

Second
Edition

PROBABILITY *and* STATISTICS with R

María Dolores Ugarte
Ana F. Militino
Alan T. Arnholt



A CHAPMAN & HALL BOOK

Second
Edition

PROBABILITY
and **STATISTICS**
with **R**

Second
Edition

PROBABILITY *and* STATISTICS

with **R**

María Dolores Ugarte

Public University of Navarre
Pamplona, Navarre, Spain

Ana F. Militino

Public University of Navarre
Pamplona, Navarre, Spain

Alan T. Arnholt

Appalachian State University
Boone, North Carolina, USA



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

© 2016 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Version Date: 20150227

International Standard Book Number-13: 978-1-4665-0440-0 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

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

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

Contents

List of Figures	xv
List of Tables	xxv
Preface to the Second Edition	xxix
Preface to the First Edition	xxxi
1 What Is R?	1
1.1 Introduction to R	1
1.2 Downloading and Installing R	1
1.2.1 Installing R under Windows	2
1.2.2 Launching R	4
1.2.3 A First Look at R (Interactive Mode)	4
1.3 Vectors	6
1.3.1 Naming Cautions	10
1.3.2 Vector Indexing	10
1.3.3 Generating Vector Sequences and Repeating Vector Constants	11
1.3.4 Filtering Vectors	12
1.4 Mode and Class of an Object	13
1.5 Getting Help	14
1.6 External Editors	15
1.7 RStudio	16
1.8 Packages	19
1.9 R Data Structures	21
1.9.1 Arrays and Matrices	21
1.9.2 Vector and Matrix Operations	27
1.9.3 Factors	28
1.9.4 Lists	29
1.9.5 Data Frames	31
1.9.5.1 Creating Data Frames	32
1.9.5.2 Accessing Data Frames	33
1.9.5.3 Accessing Data from Packages	37
1.10 Reading and Saving Data in R	39
1.10.1 Using <code>read.table()</code>	40
1.10.2 Using <code>download.file()</code>	41
1.10.3 Reading Data from Secure Websites	42
1.10.4 Using <code>scan()</code>	44
1.10.5 Reading Excel (.xlsx) Files	45
1.10.6 Saving Data Frames to External Files	47
1.11 Working with Data	47
1.11.1 Dealing with NA Values	51
1.11.2 Creating New Variables in a Data Frame	54

1.11.3	Sorting a Data Frame by One or More of Its Columns	55
1.11.4	Merging Data Frames	56
1.12	Using Logical Operators with Data Frames	58
1.13	Tables	62
1.14	Summarizing Functions	66
1.15	Probability Functions	68
1.16	Flow Control	69
1.17	Creating Functions	74
1.18	Simple Imputation	78
1.19	Using <code>plot()</code>	80
1.20	Coordinate Systems and Traditional Graphic's States	85
1.21	Problems	91
2	Exploring Data	97
2.1	What Is Statistics?	97
2.2	Data	97
2.3	Displaying Qualitative Data	98
2.3.1	Tables	98
2.3.2	Barplots	100
2.3.3	Dot Charts	100
2.3.4	Pie Charts	103
2.4	Displaying Quantitative Data	104
2.4.1	Stem-and-Leaf Plots	104
2.4.2	Strip Charts	105
2.4.3	Density Curves for Exploring Univariate Data	107
2.4.3.1	Histograms	107
2.4.3.2	Kernel Density Estimators	114
2.5	Summary Measures of Location	120
2.5.1	The Mean	120
2.5.2	The Median	121
2.5.3	Mode	123
2.5.4	Quantiles	124
2.5.5	Hinges and the Five-Number Summary	126
2.5.6	Boxplots	127
2.6	Summary Measures of Spread	129
2.6.1	Range	130
2.6.2	Interquartile Range	130
2.6.3	Variance	131
2.6.4	Sample Coefficient of Variation	132
2.6.5	The Median Absolute Deviation (<i>MAD</i>)	133
2.7	Bivariate Data	134
2.7.1	Two-Way Contingency Tables	134
2.7.2	Graphical Representations of Two-Way Contingency Tables	136
2.7.3	Comparing Samples	138
2.7.4	Relationships between Two Numeric Variables	143
2.7.5	Correlation	144
2.7.6	Fitting Lines to Bivariate Data	147
2.8	Complex Plot Arrangements	151
2.9	Multivariate Data	154
2.9.1	Graphs for Categorical Data	156
2.9.2	Lattice Graphs	162

2.9.3	Arranging Several Lattice Graphs on a Single Page	165
2.9.4	Panel Functions	167
2.9.5	Graphics with <code>ggplot2</code>	169
2.9.5.1	Shading a Region of a Density Curve	177
2.9.5.2	Violin Plots	180
2.9.5.3	Adding a Smoothed Line	184
2.9.5.4	Choropleth Maps	188
2.9.6	Arranging Several <code>ggplot</code> Graphs on a Single Page	191
2.10	Problems	193
3	General Probability and Random Variables	199
3.1	Introduction	199
3.2	Counting Techniques	199
3.2.1	Sampling with Replacement	199
3.2.2	Sampling without Replacement	200
3.2.3	Combinations	201
3.3	Axiomatic Probability	202
3.3.1	Sample Space and Events	203
3.3.2	Set Theory	203
3.3.3	Interpreting Probability	204
3.3.3.1	Relative Frequency Approach to Probability	204
3.3.3.2	Axiomatic Approach to Probability	205
3.3.4	Conditional Probability	208
3.3.5	The Law of Total Probability and Bayes' Rule	210
3.3.6	Independent Events	213
3.4	Random Variables	214
3.4.1	Discrete Random Variables	215
3.4.1.1	Mode, Median, and Percentiles	217
3.4.1.2	Expected Values	217
3.4.1.3	Moments	219
3.4.2	Continuous Random Variables	222
3.4.2.1	Numerical Integration with R	225
3.4.2.2	Mode, Median, and Percentiles	226
3.4.2.3	Expected Values	229
3.4.3	Markov's Theorem and Chebyshev's Inequality	231
3.4.4	Weak Law of Large Numbers	233
3.4.5	Skewness	234
3.5	Moment Generating Functions	235
3.6	Problems	238
4	Univariate Probability Distributions	249
4.1	Introduction	249
4.2	Discrete Univariate Distributions	249
4.2.1	Discrete Uniform Distribution	249
4.2.2	Bernoulli and Binomial Distributions	250
4.2.3	Poisson Distribution	256
4.2.4	Geometric Distribution	263
4.2.5	Negative Binomial Distribution	266
4.2.6	Hypergeometric Distribution	269
4.3	Continuous Univariate Distributions	271
4.3.1	Uniform Distribution (Continuous)	272

4.3.2	Exponential Distribution	276
4.3.3	Gamma Distribution	282
4.3.4	Hazard Function, Reliability Function, and Failure Rate	286
4.3.5	Weibull Distribution	291
4.3.6	Beta Distribution	293
4.3.7	Normal (Gaussian) Distribution	296
4.4	Problems	307
5	Multivariate Probability Distributions	315
5.1	Joint Distribution of Two Random Variables	315
5.1.1	Joint pdf for Two Discrete Random Variables	315
5.1.2	Joint pdf for Two Continuous Random Variables	317
5.2	Independent Random Variables	319
5.3	Several Random Variables	319
5.4	Conditional Distributions	322
5.5	Expected Values, Covariance, and Correlation	326
5.5.1	Expected Values	326
5.5.2	Covariance	327
5.5.3	Correlation	330
5.6	Multinomial Distribution	332
5.7	Bivariate Normal Distribution	333
5.8	Problems	342
6	Sampling and Sampling Distributions	351
6.1	Sampling	351
6.1.1	Simple Random Sampling	352
6.1.2	Stratified Sampling	354
6.1.3	Systematic Sampling	355
6.1.4	Cluster Sampling	355
6.2	Parameters	356
6.2.1	Infinite Populations' Parameters	356
6.2.2	Finite Populations' Parameters	357
6.3	Estimators	357
6.3.1	Plug-In Principle	359
6.4	Sampling Distribution of the Sample Mean	359
6.5	Sampling Distribution for a Statistic from an Infinite Population	367
6.5.1	Sampling Distribution for the Sample Mean	367
6.5.1.1	First Case: Sampling Distribution of \bar{X} When Sampling from a Normal Distribution	368
6.5.1.2	Second Case: Sampling Distribution of \bar{X} When X Is Not a Normal Random Variable	370
6.5.2	Sampling Distribution for $\bar{X} - \bar{Y}$ When Sampling from Two Independent Normal Populations	375
6.5.3	Sampling Distribution for the Sample Proportion	377
6.5.4	Expected Value and Variance of the Uncorrected Sample Variance and the Sample Variance	382
6.6	Sampling Distributions Associated with the Normal Distribution	383
6.6.1	Chi-Square Distribution (χ^2)	383
6.6.1.1	The Relationship between the χ^2 Distribution and the Normal Distribution	386

6.6.1.2	Sampling Distribution for S_u^2 and S^2 When Sampling from Normal Populations	389
6.6.2	t -Distribution	394
6.6.3	The F Distribution	397
6.7	Problems	400
7	Point Estimation	405
7.1	Introduction	405
7.2	Properties of Point Estimators	405
7.2.1	Mean Square Error	405
7.2.2	Unbiased Estimators	406
7.2.3	Efficiency	409
7.2.3.1	Relative Efficiency	410
7.2.4	Consistent Estimators	414
7.2.5	Robust Estimators	415
7.3	Point Estimation Techniques	416
7.3.1	Method of Moments Estimators	417
7.3.2	Likelihood and Maximum Likelihood Estimators	419
7.3.2.1	Fisher Information	431
7.3.2.2	Fisher Information for Several Parameters	433
7.3.2.3	Properties of Maximum Likelihood Estimators	435
7.3.2.4	Finding Maximum Likelihood Estimators for Multiple Parameters	440
7.3.2.5	Multi-Parameter Properties of MLEs	442
7.4	Problems	444
8	Confidence Intervals	453
8.1	Introduction	453
8.2	Confidence Intervals for Population Means	454
8.2.1	Confidence Interval for the Population Mean When Sampling from a Normal Distribution with Known Population Variance	454
8.2.1.1	Determining Required Sample Size	460
8.2.2	Confidence Interval for the Population Mean When Sampling from a Normal Distribution with Unknown Population Variance	464
8.2.3	Confidence Interval for the Difference in Population Means When Sampling from Independent Normal Distributions with Known Equal Variances	466
8.2.4	Confidence Interval for the Difference in Population Means When Sampling from Independent Normal Distributions with Known but Unequal Variances	470
8.2.5	Confidence Interval for the Difference in Means When Sampling from Independent Normal Distributions with Variances That Are Unknown but Assumed Equal	474
8.2.6	Confidence Interval for a Difference in Means When Sampling from Independent Normal Distributions with Variances That Are Unknown and Unequal	476
8.2.7	Confidence Interval for the Mean Difference When the Differences Have a Normal Distribution	480
8.3	Confidence Intervals for Population Variances	482
8.3.1	Confidence Interval for the Population Variance When Sampling from a Normal Population	482

8.3.2	Confidence Interval for the Ratio of Population Variances When Sampling from Independent Normal Distributions	487
8.4	Confidence Intervals Based on Large Samples	490
8.4.1	Confidence Interval for the Population Proportion	491
8.4.1.1	Score Confidence Interval	496
8.4.1.2	Agresti-Coull Confidence Interval for the Population Proportion	498
8.4.1.3	Clopper-Pearson Interval for the Population Proportion	498
8.4.1.4	So Which Confidence Interval Do I Use?	498
8.4.2	Confidence Interval for a Difference in Population Proportions . . .	506
8.4.3	Confidence Interval for the Mean of a Poisson Random Variable .	508
8.5	Problems	510
9	Hypothesis Testing	519
9.1	Introduction	519
9.2	Type I and Type II Errors	520
9.3	Power Function	524
9.4	Uniformly Most Powerful Test	527
9.5	ϕ -Value or Critical Level	529
9.6	Tests of Significance	530
9.7	Hypothesis Tests for Population Means	532
9.7.1	Test for the Population Mean When Sampling from a Normal Distribution with Known Population Variance	532
9.7.2	Test for the Population Mean When Sampling from a Normal Distribution with Unknown Population Variance	535
9.7.3	Test for the Difference in Population Means When Sampling from Independent Normal Distributions with Known Variances	542
9.7.4	Test for the Difference in Means When Sampling from Independent Normal Distributions with Variances That Are Unknown but Assumed Equal	544
9.7.5	Test for a Difference in Means When Sampling from Independent Normal Distributions with Variances That Are Unknown and Not Assumed Equal	548
9.7.6	Test for the Mean Difference When the Differences Have a Normal Distribution	551
9.8	Hypothesis Tests for Population Variances	555
9.8.1	Test for the Population Variance When Sampling from a Normal Distribution	555
9.8.2	Test for Equality of Variances When Sampling from Independent Normal Distributions	558
9.9	Hypothesis Tests for Population Proportions	562
9.9.1	Testing the Proportion of Successes in a Binomial Experiment (Exact Test)	562
9.9.2	Testing the Proportion of Successes in a Binomial Experiment (Normal Approximation)	565
9.9.3	Testing Equality of Proportions with Fisher's Exact Test	569
9.9.4	Large Sample Approximation for Testing the Difference of Two Proportions	574
9.10	Problems	579

10 Nonparametric Methods	587
10.1 Introduction	587
10.2 Sign Test	588
10.2.1 Confidence Interval Based on the Sign Test	588
10.2.2 Normal Approximation to the Sign Test	589
10.3 Wilcoxon Signed-Rank Test	594
10.3.1 Confidence Interval for ψ Based on the Wilcoxon Signed-Rank Test	599
10.3.2 Normal Approximation to the Wilcoxon Signed-Rank Test	603
10.4 The Wilcoxon Rank-Sum or the Mann-Whitney U -Test	608
10.4.1 Confidence Interval Based on the Mann-Whitney U -Test	612
10.4.2 Normal Approximation to the Wilcoxon Rank-Sum and Mann-Whitney U -Tests	615
10.5 The Kruskal-Wallis Test	622
10.6 Friedman Test for Randomized Block Designs	629
10.7 Goodness-of-Fit Tests	634
10.7.1 The Chi-Square Goodness-of-Fit Test	635
10.7.2 Kolmogorov-Smirnov Goodness-of-Fit Test	640
10.7.3 Shapiro-Wilk Normality Test	647
10.8 Categorical Data Analysis	649
10.8.1 Test of Independence	651
10.8.2 Test of Homogeneity	653
10.9 Nonparametric Bootstrapping	656
10.9.1 Bootstrap Paradigm	656
10.9.2 Confidence Intervals	665
10.9.3 Bootstrapping and Regression	677
10.10 Permutation Tests	681
10.11 Problems	688
11 Experimental Design	697
11.1 Introduction	697
11.2 Fixed Effects Model	702
11.3 Analysis of Variance (ANOVA) for the One-Way Fixed Effects Model . .	703
11.4 Power and the Non-Central F Distribution	709
11.5 Checking Assumptions	718
11.5.1 Checking for Independence of Errors	719
11.5.2 Checking for Normality of Errors	720
11.5.3 Checking for Constant Variance	722
11.6 Fixing Problems	724
11.6.1 Non-Normality	725
11.6.2 Non-Constant Variance	726
11.7 Multiple Comparisons of Means	730
11.7.1 Fisher's Least Significant Difference	731
11.7.2 The Tukey's Honestly Significant Difference	732
11.7.3 Displaying Pairwise Comparisons	733
11.8 Other Comparisons among the Means	733
11.8.1 Orthogonal Contrasts	734
11.8.2 The Scheffé Method for All Contrasts	740
11.9 Summary of Comparisons of Means	740
11.10 Random Effects Model (Variance Components Model)	745
11.11 Randomized Complete Block Design	748
11.12 Two-Factor Factorial Design	760

11.13 Problems	771
12 Regression	781
12.1 Introduction	781
12.2 Simple Linear Regression	783
12.3 Multiple Linear Regression	784
12.4 Ordinary Least Squares	785
12.5 Properties of the Fitted Regression Line	788
12.6 Using Matrix Notation with Ordinary Least Squares	789
12.7 The Method of Maximum Likelihood	796
12.8 The Sampling Distribution of $\hat{\beta}$	797
12.9 ANOVA Approach to Regression	800
12.9.1 ANOVA with Simple Linear Regression	801
12.9.2 ANOVA with Multiple Linear Regression	805
12.9.3 Coefficient of Determination	806
12.9.4 Extra Sum of Squares	807
12.9.4.1 Tests on a Single Parameter	812
12.9.4.2 Tests on Subsets of the Regression Parameters	815
12.10 General Linear Hypothesis	816
12.11 Model Building	822
12.11.1 Testing-Based Procedures	822
12.11.1.1 Backward Elimination	822
12.11.1.2 Forward Selection	822
12.11.1.3 Stepwise Regression	822
12.11.1.4 Criterion-Based Procedures	829
12.11.1.5 Summary	834
12.11.2 Diagnostics	834
12.11.2.1 Checking Error Assumptions	834
12.11.2.1.1 Assessing Normality and Constant Variance	836
12.11.2.1.2 Testing Autocorrelation	836
12.11.2.2 Identifying Unusual Observations	838
12.11.2.3 High Leverage Observations	844
12.11.3 Transformations	850
12.11.3.1 Collinearity	853
12.11.3.2 Transformations for Non-Normality and Unequal Error Variances	856
12.12 Model Validation	862
12.12.1 The Validation Set Approach	863
12.12.2 Leave-One-Out Cross-Validation	864
12.12.3 k -Fold Cross-Validation	865
12.13 Interpreting a Logarithmically Transformed Model	871
12.14 Qualitative Predictors	873
12.15 Estimation of the Mean Response for New Values \mathbf{X}_h	880
12.16 Prediction and Sampling Distribution of New Observations $Y_{h(\text{new})}$	880
12.17 Simultaneous Confidence Intervals	883
12.17.1 Simultaneous Confidence Intervals for Several Mean Responses — Confidence Band	884
12.17.2 Predictions of g New Observations	884
12.17.3 Distinguishing Pointwise Confidence Envelopes from Confidence Bands	884
12.18 Problems	891

A R Commands	903
B Quadratic Forms and Random Vectors and Matrices	917
B.1 Quadratic Forms	917
B.2 Random Vectors and Matrices	918
B.3 Variance of Random Vectors	918
Bibliography	921

List of Figures

1.1	Frequently asked questions	2
1.2	Home page for R	3
1.3	Download and install R	3
1.4	R for Windows	3
1.5	Download R for Windows link	4
1.6	R Console running in Windows	5
1.7	Screenshot of RStudio desktop	16
1.8	Environment component of RStudio	18
1.9	Create Project dialog for creating a new project	18
1.10	List from which one selects a CRAN mirror	19
1.11	List of available Packages	20
1.12	Packages component of RStudio (bottom right)	20
1.13	Available data sets	38
1.14	Help component of RStudio	39
1.15	Excel workbook faculty.xlsx worksheet 1 contents	46
1.16	Excel workbook faculty.xlsx worksheet 2 contents	47
1.17	Results of <code>file.show("FAT.txt")</code>	48
1.18	Graphical representation of the relative frequency of each of the possible means from a simulation of throwing two dice 99,999 times	73
1.19	Picture of a craps table	77
1.20	Graphs from applying <code>plot()</code> to different types of data	82
1.21	Results from using <code>plot()</code> with different types of data and arguments	83
1.22	Graphing an arbitrary function	84
1.23	Using user coordinates to label a point	85
1.24	Graph depicting how text, mathematics, and symbols are placed in the various regions of a traditional graph	88
1.25	Size, color, and choice of plotting symbol	89
1.26	Autonomous communities in Spain	92
2.1	Graphical representation of the data in Grades and Age with the function <code>barplot()</code>	101
2.2	Graphical representation of the data in Grades and Age with the function <code>dotchart()</code>	102
2.3	Dot chart of total days missed by Age and average number of days missed by Age	102
2.4	Graphical representation of the data in Grades and Age with the function <code>pie()</code>	103
2.5	Nine different graphs labeled according to their shape	104
2.6	Strip chart of the number of home runs Babe Ruth hit while playing for the New York Yankees	106
2.7	Strip chart of the number of home runs Babe Ruth hit per season according to the team for which he was playing	107

2.8	Histograms created using different bin definitions	108
2.9	Histograms with different class widths	113
2.10	Histograms created using different bin definitions for the eruption duration of Old Faithful	114
2.11	Three kernels and two bandwidths	116
2.12	Triangular kernel density estimate	117
2.13	Gaussian kernel density estimate	118
2.14	Histogram and density estimate of waiting time between Old Faithful eruptions	119
2.15	Density plot of <code>totalprice</code>	124
2.16	Graph depicting the five-number summary in relationship to original data and the boxplot	128
2.17	Boxplot of car prices with five-number summaries labeled	129
2.18	Side-by-side boxplots of bodyfat percentage	130
2.19	Stacked and side-by-side barplots for levels of palpation (<code>ease</code>) and physician (<code>doctor</code>)	137
2.20	Barplot showing percentages of treatments by obstructive contacts	139
2.21	Barplot showing percentages of obstructive contacts by treatments	139
2.22	Histograms of the BMI values of patients administered an epidural in the traditional sitting and hamstring stretch positions	141
2.23	Side-by-side boxplots of the BMI values for patients who received an epidural in the traditional sitting and hamstring stretch positions	141
2.24	Density plots of BMI for patients administered an epidural in the traditional sitting and hamstring stretch positions	142
2.25	Quantile-quantile plot of BMI in the traditional sitting and hamstring stretch positions	143
2.26	Scatterplot of <code>brain</code> versus <code>body</code> for Example 2.30 using a log base 10 scale for the x - and y -axes	144
2.27	Graph depicting residuals	148
2.28	Scatterplot of <code>log(brain)</code> versus <code>log(body)</code> with superimposed regression lines computed with (solid line) and without (dashed line) dinosaurs	150
2.29	Scatterplot of <code>log(brain)</code> versus <code>log(body)</code> with three superimposed regression lines. Solid is the OLS line; dashed is the least-trimmed squares line; and dotted is the robust line.	151
2.30	Nine equal-sized plots	152
2.31	Complex plot arrangements	153
2.32	Scatterplot and boxplots for Example 2.36	155
2.33	Mosaic plots	158
2.34	Mosaic plot where physician's assessment for ease of palpating a patient is grayscale coded with patients classified as <code>Easy</code> shaded gray80, <code>Difficult</code> shaded gray50, and <code>Impossible</code> shaded gray20	159
2.35	Mosaic plot shaded according to Pearson residuals	159
2.36	Overall admissions mosaic plots	160
2.37	Department admissions mosaic plot	162
2.38	Comparative histograms of BMI by treatment	163
2.39	Side-by-side lattice boxplots of BMI in the traditional sitting and hamstring stretch positions given <code>doctor</code>	164
2.40	Side-by-side lattice stripplots of BMI in the traditional sitting and hamstring stretch positions given <code>Doctor</code>	165
2.41	Arrangement of four different lattice graphs on the same page	167

2.42	<i>x-y</i> plot of height (cm) versus weight (kg) given physician (doctor) with superimposed least squares (solid lines) and least-trimmed squares (dashed lines)	168
2.43	Boxplots of weight by treatment	170
2.44	Plots showing the results of adding different layers to a graph	172
2.45	Left: density plot of the variable kg , center: density plots of kg split by treatment levels, right: eight density plots of kg created from faceting treatment (two levels) and doctor (four levels)	172
2.46	Scatterplots of weight versus height with ease mapped to different colors (left graph) and different shapes (right graph)	173
2.47	Scatterplots of weight versus height with ease mapped to both different colors and shapes	174
2.48	Histogram and density plot of weight	175
2.49	Barplots of ease	176
2.50	Barplots of ease with faceting	178
2.51	Density plots of BMI by ease	179
2.52	Area versus polygon plot	179
2.53	Density plots that shade BMI values greater than or equal to 40 using two different approaches	181
2.54	The left plot is a kernel density plot of the body mass index (BMI) from the data frame EPIDURALF . The right plot shows a violin plot of the body mass index (BMI) from the data frame EPIDURALF	181
2.55	Violin plots	182
2.56	The left plot shows count violin plots superimposed with boxplots. The right plot adds jittered observations to the left plot.	183
2.57	The left plot shows dotplots of the number of home runs Babe Ruth hit while playing for three different teams. The right plot shows a scatterplot of the number of home runs Babe Ruth hit versus the year for three different teams.	184
2.58	Scatterplots with loess curves	185
2.59	Scatterplots with loess and least squares lines	186
2.60	Six plots illustrating various layers used in the creation of the bottom right scatterplot of brain weight versus body weight on a log base 10 scale with superimposed and labeled least squares regression lines	188
2.61	Scatterplots of brain weight versus body weight on a log base 10 scale with superimposed least squares regression lines	189
2.62	Map of the United States of America	190
2.63	Choropleth map of North Carolina	191
2.64	Scatterplot and boxplots for Example 2.48	192
3.1	Probability of two or more students having the same birthday	207
3.2	Sample space for car batteries example	211
3.3	Circuit system diagram	214
3.4	The pdf and cdf for coin tossing	217
3.5	Fulcrum illustration of $E[X]$	218
3.6	Illustration of $\mathbb{P}(a \leq X \leq b) = \mathbb{P}(X \leq b) - \mathbb{P}(X \leq a)$	222
3.7	Illustration of pdf and cdf for Example 3.21	225
3.8	Illustration of pdf and cdf for Example 3.21 using ggplot2	225
3.9	Graph of $2\cos(2x)$ from 0 to $\frac{\pi}{4}$ with R	228
3.10	Graph of $X \sim \text{Unif}(-1, 1)$	230

3.11	Distributions with γ_1 (skewness) coefficients that are negative, zero, and positive, respectively	234
3.12	Graph of the pdf for Example 3.26	235
4.1	<i>Bin</i> (0.3, 8) pdf and cdf	252
4.2	<i>Bin</i> (10, π) pdfs for three different values of π	253
4.3	Comparison of simulated and theoretical binomial distributions	254
4.4	<i>Pois</i> (1) pdf and cdf	257
4.5	<i>Pois</i> (λ) pdfs for three different values of λ	258
4.6	<i>Geo</i> ($\pi = 0.3$) pdf and cdf	264
4.7	<i>Geo</i> (π) pdfs for three different values of π	265
4.8	<i>NB</i> ($r = 6, \pi = 0.5$) pdf and cdf	267
4.9	<i>NB</i> (r, π) pdfs for three different values of r and two different values of π	268
4.10	<i>Hyper</i> ($m = 15, n = 15, k = 10$) pdf and cdf	270
4.11	<i>Hyper</i> ($m, n = 10, k$) pdfs for three different values of m and two different values of k	271
4.12	<i>Bin</i> ($n = 5, \pi$) pdfs for three different values of π	272
4.13	The pdfs and cdfs for a <i>Unif</i> (0, 8) and a <i>Unif</i> (4, 8)	273
4.14	The pdfs and cdfs for an <i>Exp</i> ($\lambda = 1$) and an <i>Exp</i> ($\lambda = 3$)	277
4.15	Histogram of time between goals with superimposed exponential density curve	282
4.16	Graphical illustration of Γ random variables	284
4.17	Hazard functions with pdfs	288
4.18	Hazard function for printer failure	291
4.19	Graphical illustration of <i>Weib</i> random variables	292
4.20	Graphical illustration of β random variables	294
4.21	Normal distributions with increasing σ values	296
4.22	Graphical representation for computing $\mathbb{P}(a \leq X \leq b)$ given $X \sim N(\mu, \sigma)$	298
4.23	Graphical representation for computing $\mathbb{P}(a \leq X \leq b)$ given $X \sim N(100, 10)$	298
4.24	Quantile-quantile plot of the standardized test scores of 20 randomly selected college freshmen	303
4.25	Quantile-quantile plot of the standardized test scores of 20 randomly selected college freshmen created with the <i>lattice</i> package (left graph) and the <i>ggplot2</i> package (right graph)	303
4.26	Superimposed quantile-quantile plots	304
4.27	Resulting quantile-quantile plots using the function <i>ntester()</i> on standardized test scores	305
4.28	Graphical results from <i>eda(scores)</i>	306
5.1	Graphical representation of the domain of interest for Example 5.3	318
5.2	Graphical representation of $f_{X,Y}(x, y) = 8xy$, $0 \leq y \leq x \leq 1$	325
5.3	Scatterplots showing positive, negative, and zero covariance between two random variables where $p_{X,Y}(x, y) = 1/10$ for each of the ten pairs of plotted points	328
5.4	Bivariate normal density representations using <i>persp()</i>	335
5.5	Bivariate normal density representations using <i>contour()</i>	336
5.6	Bivariate normal density representations using <i>image()</i>	336
5.7	Bivariate normal density representations using <i>wireframe()</i>	337
5.8	Bivariate normal density representations using <i>contourplot()</i>	338
5.9	Bivariate normal density representations using <i>levelplot()</i>	338
5.10	Bivariate normal density representations using contours from <i>ggplot2</i> . .	339

5.11	Bivariate normal density representations using heat maps from <code>ggplot2</code>	340
6.1	Sampling distributions of \bar{X} and S^2 under random sampling (RS) and simple random sampling (SRS)	368
6.2	Comparison of uniform and normal graphs	371
6.3	Uniform and exponential simulations for samples of size $n = 2$ and $n = 16$	373
6.4	Uniform and exponential simulations for samples of size $n = 36$ and $n = 100$	373
6.5	Density histogram with superimposed normal density	377
6.6	Illustrations of the pdfs of χ_3^2 , χ_6^2 , and χ_{16}^2 random variables	384
6.7	Probability histograms for simulated distributions of $(n - 1)S^2/\sigma^2$ when sampling from normal and exponential distributions	393
6.8	Illustrations of the pdfs of t_1 (dashed line), t_3 (dotted line), and t_∞ (solid line) random variables	395
6.9	Illustrations of the pdfs of $F_{2,4}$ (solid line), $F_{4,9}$ (dotted line), and $F_{19,19}$ (dashed line) random variables	398
7.1	Visual representations of variance and bias	406
7.2	Graph representing the bias of S when it is used to estimate σ when sampling from a normal distribution	409
7.3	Graphical representations for the sampling distributions of $\hat{\mu}_1$ and $\hat{\mu}_2$	412
7.4	Illustration of the $\ln L(p \mathbf{x})$ function for a General MLE Example	421
7.5	Illustration of the $\ln L(\pi \mathbf{x})$ function for the Oriental Cockroaches Example	424
7.6	Illustration of the likelihood function in the I.I.D. Uniform Random Variables Example	428
7.7	Illustration of $\ln L(\mu \mathbf{x}, \sigma = 1)$ versus μ and $\ln L(\sigma^2 \mathbf{x}, \mu = 4)$ versus σ^2	432
8.1	Standard normal distribution with an area of $\alpha/2$ in each tail	455
8.2	Simulated confidence intervals for the population mean when sampling from a normal distribution with known variance	457
8.3	Quantile-quantile (normal distribution) plot of weekly monies spent on groceries for 30 randomly selected Watauga households	459
8.4	Quantile-quantile plot of the asking price for 14 randomly selected three-bedroom/two-bath houses in Watauga County, North Carolina	467
8.5	Normal quantile-quantile plots of the hardness values for fresh and warehoused apples	469
8.6	Normal quantile-quantile plots of mathematical assessment scores	473
8.7	Normal quantile-quantile plot of the time differences between Sun and Digital workstations to complete complex simulations	482
8.8	Quantile-quantile plot of the time differences between Sun and Digital workstations to complete complex simulations shown in the middle with normal quantile-quantile plots of random normal data depicted on the outside plots	483
8.9	Chi-square distribution with six degrees of freedom depicting the points $\chi_{\alpha/2;6}^2$ and $\chi_{1-\alpha/2;6}^2$	484
8.10	Quantile-quantile plot of 1932 barley yield in bushels/acre	485
8.11	F distribution with ten and ten degrees of freedom depicting the points $f_{\alpha/2;10,10}$ and $f_{1-\alpha/2;10,10}$	488
8.12	Coverage probability for a Wald confidence interval	497
8.13	Coverage probability for the population proportion using the Agresti-Coull, Clopper-Pearson, Wald, and Wilson 95% confidence intervals when $n = 20$	499

8.14	Expected width of the Agresti-Coull, Clopper-Pearson, Wald, and Wilson 95% confidence intervals when $n = 20$	500
9.1	Graphical representation of type I and type II errors when $H_0 : \mu = 1$ versus $H_1 : \mu = 4$	523
9.2	Graphical representation of the power function, $\text{Power}(\mu)$, for both scenarios in the Achievement Test Example	526
9.3	Graphical representation of type I and type II errors when $H_0 : \mu = 1$ versus $H_1 : \mu = 2$ with rejection region $(2.0364, \infty)$	527
9.4	Graphical representation of type I and type II errors when $H_0 : \mu = 1$ versus $H_1 : \mu = 2$ with rejection region $(1.1000, 1.3000) \cup (2.4617, \infty)$	529
9.5	Graph depicting $\beta(\mu_1 = 3)$ (dark shading) and $\text{Power}(\mu_1 = 3)$ (light shading)	535
9.6	Central t -distribution and non-central t -distribution with $\gamma = 3$	538
9.7	Central t -distribution and simulated non-central t -distribution with $\gamma = 3$	539
9.8	Exploratory data analysis of the wheat yield per plot values	540
9.9	Side-by-side boxplots and normal quantile-quantile plots of the satisfaction level for graduates from State School X and State School Y	546
9.10	Side-by-side boxplots and normal quantile-quantile plots of the sodium content for source X and source Y	550
9.11	Exploratory data analysis of the differences between 1932 barley yields from the Morris and Crookston sites	553
9.12	Graphs from using <code>eda()</code> on the washers' diameters	557
9.13	Exploratory data analysis for the blood alcohol values using the breathalyzers from company X and company Y on two volunteers after drinking four beers	560
10.1	Graphical representation of a $\text{Bin}(20, 0.5)$ distribution and a superimposed normal distribution with $\mu_S = 20(0.5) = 10$ and $\sigma_S = \sqrt{20(0.5)^2} = 2.2361$	590
10.2	Graphical representation of the data in <code>call.time</code> with the function <code>eda()</code>	592
10.3	Density plot of bus waiting times in minutes	601
10.4	Graphical representation of the Wilcoxon signed-rank distribution for $n = 15$ superimposed by a normal distribution with $\mu = n(n + 1)/4 = 60$ and $\sigma = \sqrt{n(n + 1)(2n + 1)/24} = 17.6068$	604
10.5	Density plot of differences of aggression scores	606
10.6	Density plots as well as side-by-side boxplots for piglet weight gain on diets <i>A</i> and <i>B</i>	614
10.7	Graphical representations of Wilcoxon rank-sum and Mann-Whitney <i>U</i> distributions	616
10.8	Comparative boxplot for improvements in swim times for high and low-fat diets	620
10.9	Boxplots and density plots of free-throw teaching results	624
10.10	Comparative boxplots and density plots for hydrostatic weighing (<code>hwfat</code>), skin fold measurements (<code>skfat</code>), and the Tanita body fat scale (<code>tanfat</code>)	631
10.11	Histogram of observed goals for <code>SOCCKER</code> with a superimposed Poisson distribution with $\lambda = 2.5$	638
10.12	Histogram of SAT scores in <code>GRADES</code> with superimposed expected values	641
10.13	Graphical illustration of the vertical deviations used to compute the statistic D_n	644
10.14	Graphical illustration of <code>ksdist(n = 5, sims = 10000, alpha = 0.05)</code>	645
10.15	Estimated densities for simple and composite hypotheses from running <code>ksLdist(sims = 10000, n = 10)</code>	647

10.16	Graphical representation of the bootstrap	658
10.17	Bootstrap distributions of \bar{X}	660
10.18	Bootstrap distributions of \bar{X}	665
10.19	Estimated density of interarrival times	669
10.20	Density estimate and quantile-quantile plot of $\hat{\theta}^*$	671
10.21	Density estimate of r^* with shaded 95% bootstrap percentile confidence interval	676
10.22	Density estimate for the permutation distribution of $\hat{\theta} = \bar{z} - \bar{y}$	687
11.1	Representation of a completely randomized design where treatments A, B, and C are assigned at random to six experimental units	698
11.2	Representation of a randomized complete block design where treatments A, B, and C are assigned at random to three experimental units in each block	698
11.3	Output from the function <code>oneway.plots(stopdist, tire)</code> using the data frame TIRE	701
11.4	Power for the directional alternative hypothesis $H_1 : \mu_B - \mu_A > 0$ when $\gamma = 2.5981$ at the $\alpha = 0.05$ level	712
11.5	Power for detecting treatment differences when $\lambda = 5.25$ at the $\alpha = 0.05$ level	713
11.6	Histogram of simulated $F_{3, 20}^*$ superimposed by a central $F_{3, 20}$ distribution	714
11.7	Central and non-central F distributions	718
11.8	Standardized residuals versus order for <code>mod.aov</code> using the TIRE data set	720
11.9	Quantile-quantile plot of the standardized residuals with a superimposed line with a zero intercept and a slope of one for the model <code>mod.aov</code> using the TIRE data frame	721
11.10	Plot of the standardized residuals versus the fitted values for <code>mod.aov</code> using the TIRE data set	722
11.11	Graphs to assess independence, normality, and constant variance created with <code>checking.plots(mod.aov0)</code> using the data frame TIRE0	724
11.12	Transformations in common use with the Box-Cox method	725
11.13	<code>checking.plots()</code> applied to the model <code>FCD.aov (aov(weight ~ diet))</code> with the FCD data frame	727
11.14	<code>checking.plots()</code> applied to the model <code>FCDlog.aov (aov(log(weight) ~ diet))</code> with the FCD data frame	728
11.15	Simultaneous 95% confidence intervals for the contrasts C_1 and C_2	740
11.16	Graphical representation of confidence intervals based on Tukey's HSD for the model <code>stopdist ~ tire</code> using the data frame TIRE	744
11.17	Interaction plots of block and treatments using TIREWEAR	753
11.18	The <code>ggplot2</code> strip plots for Example 11.7, which show tread wear of each car by tire (top) and tread wear of each tire by car (bottom)	754
11.19	Tire wear means due to treatments and blocks	755
11.20	<code>checking.plots()</code> applied to <code>mod.aov0</code> from Example 11.7	758
11.21	Simultaneous 95% mean pairwise confidence intervals using Tukey's HSD from Example 11.7	759
11.22	Barplot of the mean wear by tire with superimposed individual 95% confidence intervals from Example 11.7	760
11.23	Graphs from <code>twoway.plots()</code>	764
11.24	Interaction plots of Glass and Phosphor	765
11.25	Barplot of means for each level of Glass and Phosphor	766

11.26	Graphs resulting from using <code>checking.plots()</code> on the model <code>mod1.TVB</code> from Example 11.8	769
12.1	Graphical representation of simple linear regression model depicting the distribution of Y given x	783
12.2	Scatterplot of <code>gpa</code> versus <code>sat</code> using GRADES	793
12.3	Plane of best fit from regressing <code>hwfat</code> onto <code>triceps</code> and <code>abs</code>	795
12.4	Decomposition of the deviations of the observations, Y_i s, around the regression line for a simple linear regression.	802
12.5	Scatterplots to illustrate values of R^2	807
12.6	Schematic representation of extra sum of squares for Example 12.9 on page 808	810
12.7	Regression model building flow chart modified from Neter et al. (1996, Figure 8.1)	823
12.8	Enhanced scatterplot matrices	825
12.9	Residual plots for four different models with different residual patterns . .	837
12.10	Diagnostic plots for <code>mod1</code> in Figure 12.9	839
12.11	Normal quantile-quantile plot for the standardized residuals of <code>mod1</code> in Figure 12.9 on page 837	839
12.12	Standardized residuals versus fitted values for <code>mod3.hsw</code>	841
12.13	Quantile-quantile plot of studentized residuals from <code>mod3.hsw</code> with the three largest (in absolute value) studentized residuals labeled	843
12.14	Diagnostic plots of <code>mod3.hsw</code> with the three most prominent observations for each diagnostic plot labeled	843
12.15	Bubble-plot of studentized residuals versus leverage values with plotted points proportional to Cook's distance for <code>mod3.hsw</code>	849
12.16	Scatterplot, residuals versus fitted values, and quantile-quantile plot of standardized residuals for y versus x_1 and y versus $\sqrt{x_1}$ models	851
12.17	Scatterplot, residuals versus fitted values, and quantile-quantile plot of standardized residuals for y versus x_2 and y versus x_2^2 models	852
12.18	Scatterplot, residuals versus fitted values, and quantile-quantile plot of standardized residuals for y versus x_3 and Y versus x_3^{-1} models	853
12.19	Box-Cox graph of λ for Example 12.22 on page 857	857
12.20	Scatterplot and residual versus fitted plot of y_1 versus x_1 ; Box-Cox plot of λ ; scatterplot, residual versus fitted plot, and quantile-quantile plot of $\ln(y_1)$ versus x_1	859
12.21	Scatterplot and residual versus fitted plot of Y_2 versus x_2 ; Box-Cox plot of λ ; scatterplot, residual versus fitted plot, and quantile-quantile plot of Y_2^{-1} versus x_2	861
12.22	Process of model building with transformations	863
12.23	Schematic display of the validation set approach	864
12.24	Schematic display of the leave-one-out cross-validation	864
12.25	Schematic display of 5-fold cross-validation	865
12.26	Small change in x gives a similar small change in $\ln(x)$	871
12.27	Four possible results for a single dummy variable with two levels	875
12.28	Scatterplot of <code>totalprice</code> versus <code>area</code> with the fitted regression line superimposed from <code>mod.simple</code>	877
12.29	Fitted regression lines for <code>mod.inter</code>	879
12.30	Representation of 90% pointwise confidence intervals, 90% prediction intervals, and a 90% confidence band	886

12.31 Joint confidence region for β_2 and β_3 enclosed by the Bonferroni (left graph) and Scheffé (right graph) confidence limits	888
--	-----

List of Tables

1.1	Body composition (BODYFAT)	58
1.2	Missing at random values by industry example	79
2.1	Common kernels and their definitions	115
2.2	Student test scores	122
2.3	Two-way table of Doctor by Ease	134
2.4	Different values for b_0 and b_1 with various regression methods	150
2.5	Geoms and commonly used aesthetics	169
3.1	Probability of two or more students having the same birthday	207
4.1	Comparison of binomial and Poisson probabilities	263
4.2	Standardized scores (data frame SCORE)	302
5.1	B.S. graduate grades in Linear Algebra and Calculus III	316
5.2	Values used to compute covariance for scatterplots with positive, negative, and zero covariance	328
5.3	Joint probability distribution for X and Y	331
6.1	Finite populations' parameters	358
6.2	Parameters and their corresponding estimators	358
6.3	Finite population parameter estimators and the estimators of their standard deviations	359
6.4	Statistics for samples of size n	360
6.5	Possible samples of size 2 with \bar{x} and s^2 for each sample — random sampling	362
6.6	Sampling distribution of \bar{X} — random sampling	362
6.7	Sampling distribution of S^2 — random sampling	362
6.8	Possible samples of size 2 with \bar{x} and s^2 — simple random sampling	365
6.9	Sampling distribution of \bar{X} — simple random sampling	365
6.10	Sampling distribution of S^2 — simple random sampling	365
6.11	Summary results for sampling without replacement (finite population)	366
6.12	Computed values for random sampling (Case 1) and simple random sampling (Case 2)	367
6.13	Comparison of simulated uniform and exponential distributions to the normal distribution	374
6.14	Output for probability distribution of $(n - 1)S^2/\sigma^2$ example	394
8.1	Weekly spending in dollars (GROCERY)	458
8.2	House prices (in thousands of dollars) for three-bedroom/two-bath houses in Watauga County, North Carolina (HOUSE)	465
8.3	Apple hardness measurements (APPLE)	469

8.4	Mathematical assessment scores for students enrolled in a biostatistics course (CALCULUS)	472
8.5	Methods for analyzing normal data	480
8.6	Time to complete a complex simulation in minutes (SUNDIG)	481
9.1	Form of hypothesis tests	519
9.2	Possible outcomes and their consequences for a trial by jury	521
9.3	Relationship between type I and type II errors	522
9.4	Calculation of ϕ -values for continuous distributions	529
9.5	Duality of $(1 - \alpha) \cdot 100\%$ confidence intervals and α -level tests of significance	531
9.6	Summary for testing the mean when sampling from a normal distribution with known variance (one-sample z -test)	533
9.7	Summary for testing the mean when sampling from a normal distribution with unknown variance (one-sample t -test)	536
9.8	Summary for test for differences in means when taking independent samples from normal distributions with known variances (two-sample z -test)	542
9.9	Summary for test for differences in means when taking independent samples from normal distributions with unknown but assumed equal variances (two-sample pooled t -test)	545
9.10	Summary for test for differences in means when taking independent samples from normal distributions with unknown and unequal variances	549
9.11	Summary for testing the mean of the differences between two dependent samples when the differences follow a normal distribution with unknown variance (paired t -test)	552
9.12	Summary for testing the population variance when sampling from a normal distribution	556
9.13	Diameters for 20 randomly selected washers (WASHER)	556
9.14	Summary for test for equality of variances when sampling from independent normal distributions	559
9.15	Summary for testing the proportion of successes in a binomial experiment (number of successes is $Y \sim \text{Bin}(n, \pi)$)	562
9.16	Summary for testing the proportion of successes in a binomial experiment (normal approximation)	566
9.17	Correction factors when $ p - \pi_0 > \frac{1}{2n}$	566
9.18	General form of a 2×2 table	569
9.19	Summary for testing the proportion of successes with Fisher's exact test .	570
9.20	Juveniles who failed a vision test classified by delinquency and glasses-wearing (Weindling et al., 1986)	571
9.21	Seven possible 2×2 tables that can be constructed where $k = 6$, $m = 9$, and $n = 7$, with their associated probabilities	571
9.22	Observed heart attacks for those physicians taking aspirin and a placebo (Hennekens, 1988)	573
9.23	Summary for testing the differences of the proportions of successes in two binomial experiments (large sample approximation)	575
9.24	Correction factors when $ p_X - p_Y > \frac{1}{2} (\frac{1}{m} + \frac{1}{n})$	576
10.1	Summary for testing the median — sign test	589
10.2	Summary for testing the median — approximation to the sign test	591
10.3	Long-distance telephone call times in minutes (PHONE)	591
10.4	Asymptotic relative efficiency comparisons	595

10.5	Possible sign and rank combinations for the trivial T^+ distribution	596
10.6	PDF of T^+ for the trivial T^+ distribution example	596
10.7	Summary for testing the median — Wilcoxon signed-rank test	599
10.8	Waiting times in minutes (WAIT)	600
10.9	Summary for testing the median — normal approximation to the Wilcoxon signed-rank test	604
10.10	Aggression test scores (AGGRESSION)	606
10.11	Summary for testing equality of medians — Wilcoxon rank-sum test	610
10.12	Summary for testing equality of medians — Mann-Whitney U -test	610
10.13	Summary for testing the difference in two medians — normal approximation to the Wilcoxon rank-sum test	617
10.14	Summary for testing the difference in two medians — normal approximation to the Mann-Whitney U -Test	618
10.15	Sorted improvements in swim times in seconds for high (x) and low (y) fat diets, where rank refers to the rank of the data point in the combined sample of x and y data points (SWIMTIMES)	619
10.16	Number of successful free-throws	623
10.17	Actual free-throws with ranks among all free-throws	625
10.18	A representation of the ranked data from a randomized complete block design	630
10.19	The first six wrestlers' body fat as measured by the three techniques and their corresponding ranks	632
10.20	Calculating D_n	643
10.21	Twenty-six-year-olds' happiness	649
10.22	Mild dementia treatment results	650
10.23	Contingency table when sampling from a single population	650
10.24	General form and notation used for an $I \times J$ contingency table when sampling from I distinct populations	651
11.1	One-way design	701
11.2	Parameters and estimators for fixed effects, one-way CRD Model (11.1) .	703
11.3	ANOVA table for one-way completely randomized design	706
11.4	Tire ANOVA table	708
11.5	α_e values for given α_c and various numbers of comparisons	730
11.6	ANOVA table for model fecundity ~ line using DROSOPHILA data	735
11.7	ANOVA table for orthogonal contrasts with DROSOPHILA	736
11.8	Shear on frozen carrots by freezer	747
11.9	Frozen carrots ANOVA table	747
11.10	ANOVA table for the randomized complete block design	751
11.11	The tread loss from the TIREWEAR data frame	752
11.12	Sums and estimates for Example 11.7	752
11.13	Tire wear ANOVA table	756
11.14	Layout for observations in a two-factor factorial design	760
11.15	ANOVA table for two-factor factorial design	762
11.16	Data from Hicks (1956) used in Example 11.8	763
11.17	Two-factor factorial design table to complete for (c) of Example 11.8 . .	763
11.18	Two-factor factorial design table COMPLETED for (c) of Example 11.8 .	767
11.19	ANOVA table for two-factor factorial design for Example 11.8	768
12.1	ANOVA table for simple linear regression	804
12.2	ANOVA table for model.1m <- lm(gpa ~ sat)	804

12.3	ANOVA table for multiple linear regression	805
12.4	ANOVA table for <code>mod1.HSW</code>	812
12.5	ANOVA table for <code>mod2.HSW</code>	812
12.6	Summary of measures of influential observations	845
12.7	Actual change in jaguar brain weight	872
12.8	Values of \mathbf{X}_{hi} for HSWRESTLER	886
A.1	Useful Commands When Working with Numeric Vectors	903
A.1	Useful Commands When Working with Numeric Vectors (continued) . . .	904
A.2	Vector and Matrix Functions	904
A.2	Vector and Matrix Functions (continued)	905
A.3	Functions Used with Arrays, Factors, and Lists	905
A.3	Functions Used with Arrays, Factors, and Lists (continued)	906
A.4	Graphs Frequently Used with Descriptive Statistics	906
A.4	Graphs Frequently Used with Descriptive Statistics (continued)	907
A.5	Basic Plotting Functions	907
A.5	Basic Plotting Functions (continued)	908
A.6	Commonly Used Graphical Parameters	908
A.6	Commonly Used Graphical Parameters (continued)	909
A.7	Lattice Functions	909
A.8	Important Probability Distributions That Work with <code>rdist</code> , <code>pdist</code> , <code>ddist</code> , and <code>qdist</code>	910
A.9	Useful Functions for Parametric Inference	910
A.9	Useful Functions for Parametric Inference (continued)	911
A.10	Useful Functions for Nonparametric Inference	911
A.11	Useful Functions in R for Linear Regression and Analysis of Variance . . .	912
A.12	Useful Contrast Functions in R for Linear Regression and Analysis of Variance	912
A.12	Useful Contrast Functions in R for Linear Regression and Analysis of Variance (continued)	913
A.13	Useful Model-Building Functions for Linear Regression and Analysis of Variance	913
A.14	Useful Diagnostic Functions for Linear Regression and Analysis of Variance .	913
A.14	Useful Diagnostic Functions for Linear Regression and Analysis of Variance (continued)	914
A.15	Functions from PASWR2	914
A.15	Functions from PASWR2 (continued)	915

Preface to the Second Edition

Welcome to the second edition of *Probability and Statistics with R!* You are holding a text that will allow you to expand your practice of statistics into the current decade. The contributions that have been made to both packages and data since the first edition was published are truly astonishing. The number of contributed packages on CRAN has increased from around 1,000 to over 6,000, and this edition explores how some of those new packages can make analysis easier and more intuitive, as well as give more visually pleasing results in the case of graphs. The data associated with the book are available in the `PASWR2` package and have been augmented by numerous Internet sources for this edition. Furthermore, you will learn how to make use of the readily available data from numerous sites.

This text has improved over the first edition in terms of additions and clarifications of examples and problems, concepts, data, and functions. Most chapters of the book have several new examples and exercises that use the most modern functions and have problems to be solved with those functions to solidify understanding. Statistically, the concepts of the coverage probability of a confidence interval and model validation have been added to this version of the text. The text is supported at <http://alanarnholt.github.io/PASWR2E-Book/> with solutions to odd exercises, and templates for homework assignments. A complete solutions manual is available from the publisher for text adopters. The package `PASWR2` written to support this text contains data sets and functions and is available on CRAN (<http://cran.r-project.org/web/packages/PASWR2/index.html>). The most recent updates to `PASWR2` can be found on GitHub at <https://github.com/alanarnholt/PASWR2>. Throughout, based on feedback of readers of the first edition, the prose has been expanded or rewritten as needed to make comprehension more certain.

Since `R` has become the undisputed language of choice for the majority of statistical practitioners, any references to its commercial counterpart, `S-PLUS`, have been removed. Additionally, the `R` code for calculations and graph creation has been highlighted for ease of reading and referenced with numbers as examples were in the first edition. Graphs have been created primarily with `ggplot2`, which has also moved to the fore as the most comprehensive graphics package with the greatest possibility of both customization and application.

It is our fervent hope that this edition of *Probability and Statistics with R* will serve you in your quest to discover truth in our world through the analysis of data. We welcome feedback and ideas for future editions and expect to continue both expansion and modification of the text as `R` itself continues to grow and change.

Acknowledgments

We gratefully acknowledge the invaluable help provided by Susie Arnholt, senior lecturer at Appalachian State University. Her willingness to apply her expertise in L^AT_EX and knowledge of English grammar to the production of this text is appreciated beyond words. Many people were instrumental in improving the readability of this text; however, we are particularly appreciative of the contributions Tomás Goicoa, associate professor at the University of Navarre, made with respect to exercises, errata, and new ideas for the second edition.

Preface to the First Edition

The authors would like to thank their parents

Lola: Pedro and Loli

Ana: Carmelo and Juanita

Alan: Terry and Loretta

for their unflagging support and encouragement.

The Book

Probability and Statistics with R is a work born of the love of statistics and the advancements that have been made in the field as more powerful computers can be used to perform calculations and simulations that were only dreamed of by those who came before. The **S** language and its derivative, **R**, have made the practice of statistics available to anyone with the time and inclination to do so.

Teachers will enjoy the real-world examples and the thoroughly worked-out derivations. Those wanting to use this book as a reference work will appreciate the extensive treatments on data analysis using appropriate techniques, both parametric and nonparametric. Students who are visual learners will appreciate the detailed graphics and clear captions, while the hands-on learners will be pleased with the abundant problems and solutions. (A solutions manual should be available from Taylor & Francis.) It is our hope that practitioners of statistics at every level will welcome the features of this book and that it will become a valuable addition to their statistics libraries.

The Purpose

Our primary intention when we undertook this project was to introduce **R** as a teaching statistical package, rather than just a program for researchers. As much as possible, we have made a great effort to link the statistical contents with the procedures used by **R** to show consistency to undergraduate students. The reader who uses **S-PLUS** will also find this text useful, as **S-PLUS** commands are included with those for **R** in the vast majority of the examples.

This book is intended to be practical, readable, and clear. It gives the reader real-world examples of how **S** can be used to solve problems in every topic covered, including, but not limited to, general probability in both the univariate and multivariate cases, sampling distributions and point estimation, confidence intervals, hypothesis testing, experimental design, and regression. Most of the problems are taken from genuine data sets rather than created out of thin air. Next, it is unusually thorough in its treatment of virtually every topic, covering both the traditional methods to solve problems as well as many nonparametric techniques. Third, the figures used to explain difficult topics are exceptionally detailed.

Finally, the derivations of difficult equations are worked out thoroughly rather than being left as exercises. These features, and many others, will make this book beneficial to any reader interested in applying the **S** language to the world of statistics.

The Program

The **S** language includes both **R** and **S-PLUS**. “**R** can be regarded as an implementation of the **S** language which was developed at Bell Laboratories by Rick Becker, John Chambers, and Allan Wilks, and also forms the basis of the **S-PLUS** systems.” (<http://cran.r-project.org/doc/manuals/R-intro.html#Preface>)

The current **R** is the result of a collaborative effort with contributions from all over the world. **R** was initially written by Robert Gentleman and Ross Ihaka of the Statistics Department of the University of Auckland. Since mid-1997 there has been a core group with write access to the **R** source (<http://www.r-project.org/>—click “Contributors” on the sidebar).

Not only is **R** an outstanding statistical package, but it is offered free of charge and can be downloaded from <http://www.r-project.org/>. The authors are greatly indebted to the giants of statistics and programming on whose shoulders we have stood to see what we will show the readers of this text.

The Content

The core of the material covered in this text has been used in undergraduate courses at the Public University of Navarre for the last ten years. It has been used to teach engineering (agricultural, industrial, and telecommunications) and economics majors. Some of the material in this book has also been used to teach graduate students studying agriculture, biology, engineering, and medicine.

The book starts with a brief introduction to **S** that includes syntax, structures, and functions. It is designed to provide an overview of how to use both **R** and **S-PLUS** so that even a neophyte will be able to solve the problems by the end of the chapter.

Chapter 2, entitled “Exploring Data,” covers important graphical and numerical descriptive methods. This chapter could be used to teach a first course in statistics.

The next three chapters deal with probability and random variables in a generally classical presentation that includes many examples and an extensive collection of problems to practice all that has been learned.

Chapter 6 presents some important statistics and their sampling distributions. Solving the exercises will give any reader confidence that the difficult topics covered in this chapter are understood.

The next four chapters encompass point estimation, confidence intervals, hypothesis testing, and a wide range of nonparametric methods including goodness-of-fit tests, categorical data analysis, nonparametric bootstrapping, and permutation tests.

Chapter 11 provides an introduction to experimental design using fixed and random effects models as well as the randomized block design and the two-factor factorial design.

The book ends with a chapter on simple and multiple regression analysis. The procedures from this chapter are used to solve three interesting case studies based on real data.

The Fonts

Knowing several typographical conventions will help the reader in understanding the material presented in this text. **R** code is displayed in a monospaced font with the **>** symbol in front of commands that are entered at the **R** prompt.

```
> x<-0.28354
> round(x,2)
[1] 0.28
```

The same font is used for data sets and functions, though functions are followed by `()`. For example, the **PASWR** package but the `round()` function would be shown. Throughout the text, a  is found at the end of solutions to examples. In the index, page numbers in **BOLD** are where the primary occurrences of topics are found, while those in *ITALICS* indicate the pages where a problem about a topic or using a given data set can be located.

The Web

This text is supported at <http://www1.appstate.edu/~arnholta/PASWR> on the Internet. The website has up-to-date errata, chapter scripts, and a copy of the **PASWR** package (which is also on CRAN) available for download.

Acknowledgments

We gratefully acknowledge the invaluable help provided by Susie Arnholt. Her willingness to apply her expertise in **L^AT_EX** and knowledge of English grammar to the production of this book is appreciated beyond words.

Several people were instrumental in improving the overall readability of this text. The recommendations made by Phil Spector, the Applications Manager and Software Consultant for the Statistical Computing Facility in the Department of Statistics at the University of California at Berkeley, who reviewed this text for Taylor & Francis, were used in improving much of the original R code as well as decreasing the inevitable typographical error rate. Tomás Goicoa, a member of the Spatial Statistics Research Group at the Public University of Navarre, was of great help in preparing and checking exercises. Celes Alexander, an Appalachian State University graduate student, graciously read the entire text and found several typos. Any remaining typos or errors are entirely the fault of the authors.

Thanks to our editor at Taylor & Francis, David Grubbs, for embracing and encouraging our project. Many thanks the Statistics and Operations Research Department at Public University of Navarre and to the Department of Mathematical Sciences at Appalachian State University for the support they gave us in our writing of this text.

The “You choose, you decide” initiative sponsored by Caja Navarra also provided funding for in-person collaborations. Thanks to the *Universidad Nacional de Educación a Distancia*, in particular the *Centro Asociado de Pamplona*, for allowing us to present this project under their auspices.

Special thanks to José Luis Iriarte, the former Vicerector of International Relations of the Public University of Navarre, and to T. Marvin Williamsen, the former Associate Vice Chancellor for International Programs at Appalachian State University. These men were instrumental in gaining funding and support for several in-person collaborations including a year-long visit at the Public University of Navarre for the third author and two multi-week visits for the first two authors to Appalachian State University.

Finally, to the geniuses of this age who first conceived of the idea of an excellent open source software for statistics and those who reared the idea to adulthood, our gratitude is immeasurable. May the lighthouse of your brilliance guide travelers on the ocean of statistics for decades to come. Thank you, R Core Team.

Chapter 1

What Is R?

1.1 Introduction to R

R is a language and an environment for statistical computing and graphics. In this book, R is used to store, manage, and manipulate data; to create graphical displays of data using different graphical systems; to analyze data using standard statistical procedures; and to perform simulations. In short, everything a student or scientist may want or need to do with data can be done with R, and this book uses R for everything that it covers. For the reader who wants his documents to update (graphs, tables, statistics, etc.) as the data changes, R integrates seamlessly with the typesetting program L^AT_EX via the *Sweave* and *knitr* packages. The *knitr* package also integrates R with markdown, an easy-to-use text markup language. The process of creating documents and solving problems that include all code and update as data changes is called reproducible analysis. Two excellent sources to learn more about the *knitr* package and reproducible analysis are *Dynamic Documents with R and knitr* (Xie, 2014) and *Reproducible Research with R and RStudio* (Gandrud, 2014), respectively.

The goals of this chapter are to provide sufficient detail to allow the reader to install R, along with an editor, on his operating system, to have the reader use R and the editor from the start in learning some of the basics of R syntax, to practice reading, writing, and accessing data, and to produce base graphs. Since R itself is a programming language, what one can do with R is truly amazing; however, this text will only address topics the authors have found useful in helping their students understand the material in a first calculus-based probability and statistics course. For a deeper introduction to the R language, the reader should consult *An Introduction to R*, which is available as a PDF file once R is installed. This can be done by clicking **Help→Manuals (in PDF)→An Introduction to R** or from <http://cran.r-project.org/doc/manuals/R-intro.pdf>. R can be used in batch mode as well as in interactive mode. This book only uses R in the interactive mode and has the reader type commands either at the R prompt or in an R script. It is the firm belief of the authors that typing commands as opposed to using drop-down menus leads to a better understanding of exactly what the reader/user is doing. While R runs on virtually any platform, the comments provided in the text are Microsoft Windows-specific unless otherwise noted.

1.2 Downloading and Installing R

Precompiled binary distributions of the base R system and contributed packages are available for Windows, MacOS X, and Linux operating systems. Source code is also available for the reader who wants to compile R from source or to run R on an operating system other

than Windows, MacOS X, or Linux. In spite of this, no attempt is made in this material to guide the reader through building R from source. The definitive reference for installing R, regardless of whether one is installing from source or installing from a precompiled binary distribution, is the *R Installation and Administration* manual, also available once R is installed. This manual is found by clicking on **Help→Manuals (in PDF)→R Installation and Administration** as well as online at <http://cran.r-project.org/doc/manuals/R-admin.pdf>. If problems are encountered installing a binary distribution, a quick answer to the particular difficulty will often be documented in Frequently Asked Questions (FAQs). To open the FAQs web page, click FAQs on the left menu at <http://cran.r-project.org>; then, choose the respective platform one is using (see Figure 1.1).

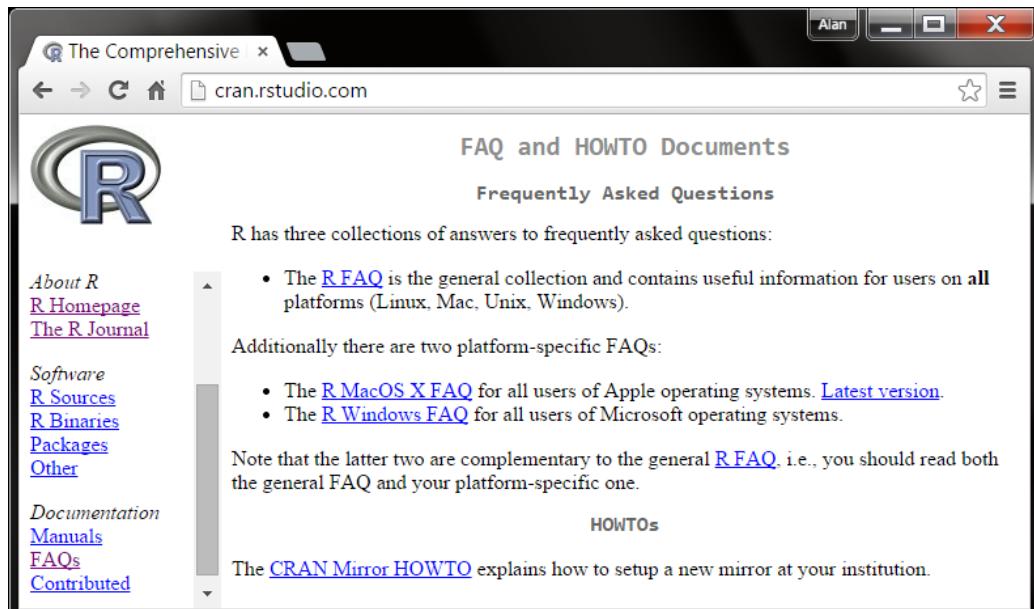


FIGURE 1.1: Frequently asked questions

1.2.1 Installing R under Windows

The home page for R is the website <http://www.r-project.org>. (See Figure 1.2 on the facing page.) By clicking on the **CRAN** link depicted on the left side of Figure 1.2 on the next page, one is presented with a selection of mirrors. If one selects “0-Cloud,” your computer will automatically be redirected to a server close to your location. A window similar to Figure 1.3 on the facing page will open once a mirror location is selected. Clicking on the **Download R for Windows** link in Figure 1.3 on the next page opens a window similar to Figure 1.4 on the facing page, and clicking on the **base** link in Figure 1.4 on the next page produces Figure 1.5 on page 4.

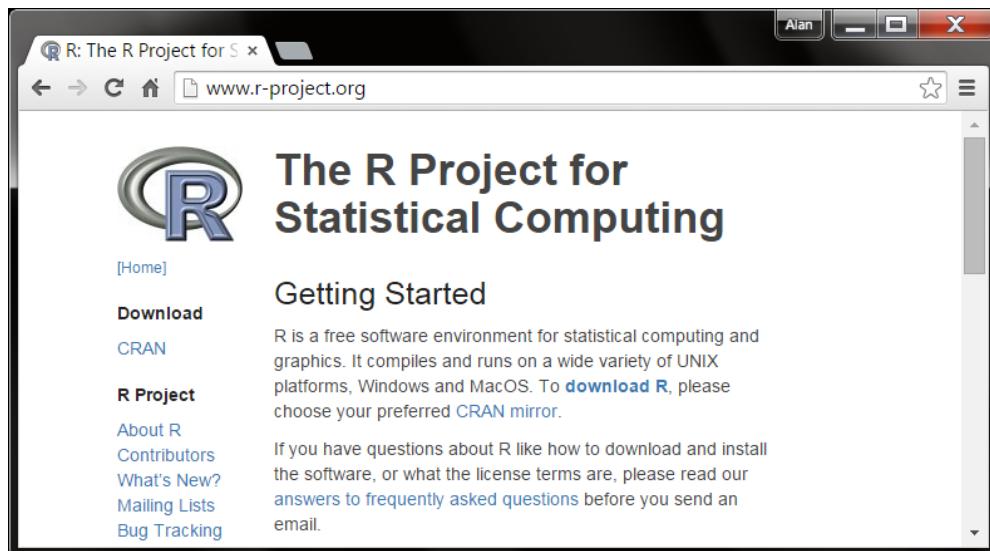


FIGURE 1.2: Home page for R

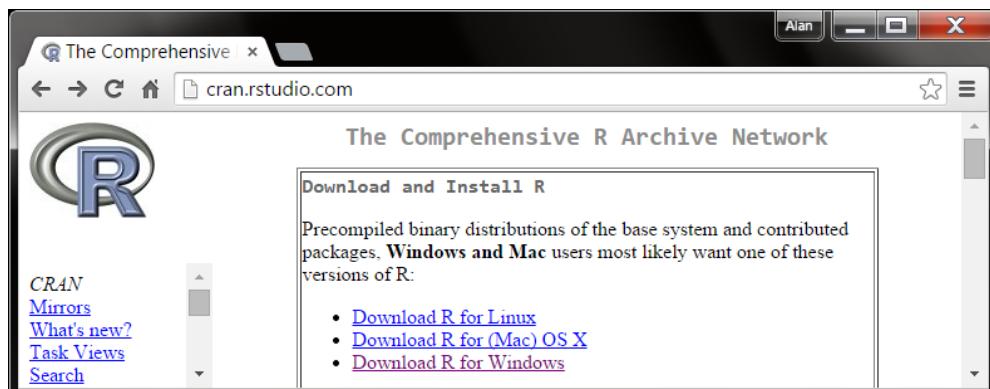


FIGURE 1.3: Download and install R

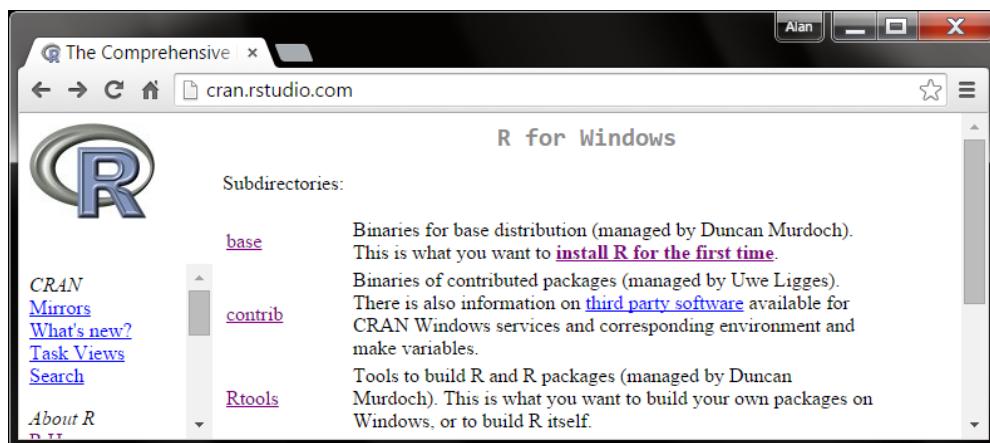


FIGURE 1.4: R for Windows

Click on the **Download R X.YY.x (R-3.2.0 as of June 2, 2015) for Windows** link, and the **R-3.2.0-win.exe** (or more current version, R-X.YY.x-win.exe) file will download to your computer. To install R, click on the downloaded **R-X.YY.x-win.exe** file. Choosing the default settings is most likely the easiest way to install R. For more information on the installation options, see the *R Installation and Administration* manual (<http://cran.r-project.org/doc/manuals/R-admin.pdf>). To install R for Mac OS X 10.6 (Snow Leopard) and higher, click on the **Download R for MacOS X** link as seen in Figure 1.3 on the previous page. Double click on the **R-3.2.0.pkg** (or more recent) file. For further information with other operating systems, refer to the *R Installation and Administration Manual*.

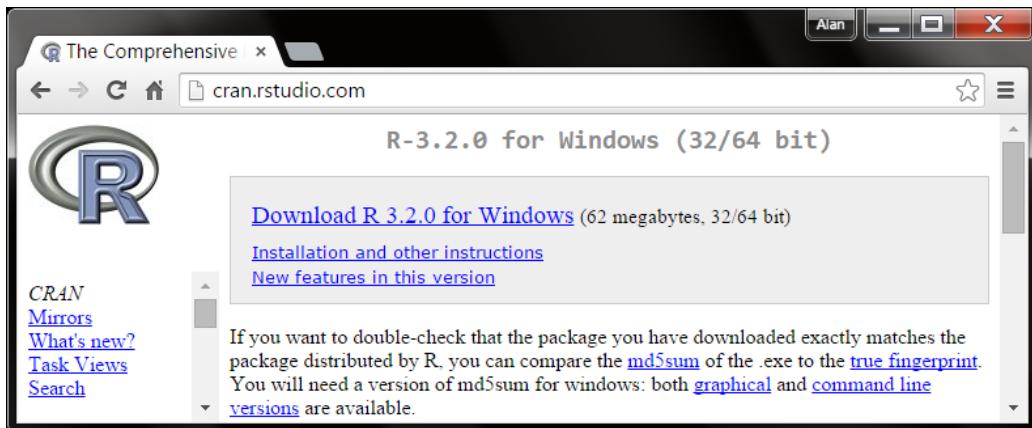


FIGURE 1.5: Download R for Windows link

1.2.2 Launching R

If R is installed on a Windows computer using the default options, the R icon will appear on the desktop as a shortcut icon. Double click the R icon, and R should open in a window resembling Figure 1.6 on the facing page. On Windows computers, R may also be launched by clicking the Windows **Start icon** → **All programs** → **R** → **R X.YY.x** (R 3.2.0 as of June 2, 2015). To launch R on a Mac, double click the R icon which, on a default installation, will be in the Applications folder. To launch R on a Linux platform, enter the command “R” at the system command line.

1.2.3 A First Look at R (Interactive Mode)

Standard mathematical operations (+ (addition), – (subtraction), / (division), * (multiplication), \wedge (raise to a power), etc.) can be used in a mathematical expression by typing in the R console at the R prompt (>). Once the user presses the Enter key, the result(s) of the requested mathematical operations appear below the code to the right of square brackets enclosing a number indicating the index of the answer for the value immediately to the right of the enclosed number.

For example, R Code 1.1 on the next page requests that twenty randomly generated values from a continuous uniform distribution with a minimum value of 0 and a maximum

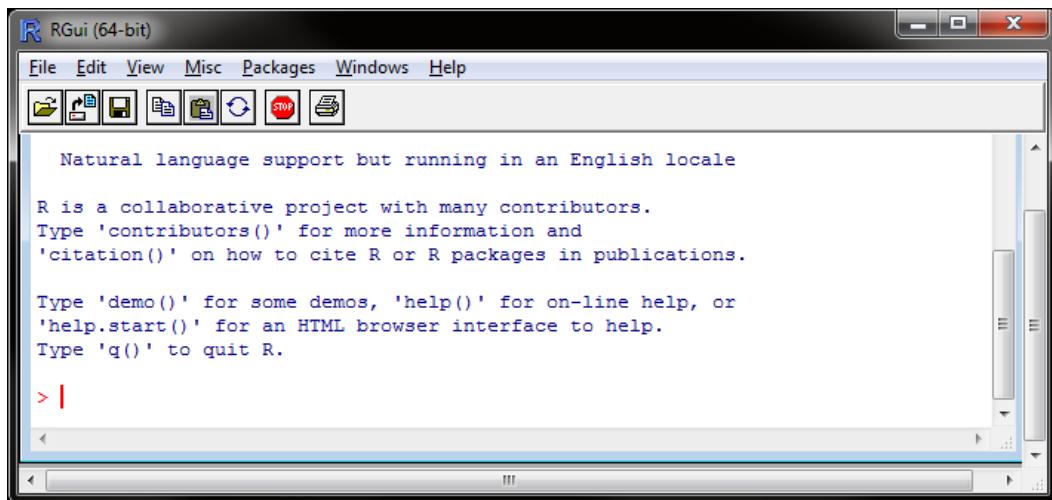


FIGURE 1.6: R Console running in Windows

value of 1 be stored in the R object `ruv` using the assignment operator (`<-`). Assignment may also be made with the equal sign (`=`). There are instances when the use of an `=` may lead to confusion, so the authors use the traditional assignment operator (`<-`) as a general rule. The value immediately to the right of `[1]`, 0.0694, is the first value; and the value immediately to the right of `[19]`, 0.6657, is the nineteenth value. When illustrating pedagogical concepts, the user will often want to generate the same set of “random” numbers at a later date. To reproduce the same set of “random” numbers, one uses the `set.seed()` function. The `set.seed()` function puts the random number generator in a reproducible state. Throughout the text, the width of printed output is set to a maximum number of 70 columns on a line using `options(width = 70)`. Text to the right of a `#` sign is not evaluated by R and is used to make comments about code.

R Code 1.1

```
> options(width = 70) # set width of console output
> set.seed(12) # set seed to make results reproducible
> ruv <- runif(n = 20, min = 0, max = 1)
> round(ruv, 4) # round answers to 4 decimal places

[1] 0.0694 0.8178 0.9426 0.2694 0.1693 0.0339 0.1788 0.6417 0.0229
[10] 0.0083 0.3927 0.8139 0.3762 0.3808 0.2649 0.4393 0.4576 0.5407
[19] 0.6657 0.1127
```

To use R as a calculator to compute $(7 \times 3) + 12 \div 2 - 7^2 + \sqrt{4}$, enter

```
> (7 * 3) + 12/2 - 7^2 + sqrt(4)

[1] -20
```

The answer to the previous computation is -20 , printed to the right of `[1]`, which indicates the answer starts at the first element of the vector. Common functions such as `log()`, `log10()`, `exp()`, `sin()`, `cos()`, `tan()`, and `sqrt()` (square root) are all recognized by R. For a quick reference to commonly used R functions, see Table A.1 on page 903. If the user omits a comma or a parenthesis, or any other type of syntax error occurs when typing at

the command line, a + sign will appear to indicate that the command is incomplete. If the user does not recognize a typographical error, an easy technique to get rid of the + sign is to press the **Esc** key for Windows and Mac users working in the console. To interrupt R in Linux or in a terminal on a Mac, type **control-C**.

1.3 Vectors

R has many types of data structures, and one of the more important is the vector. All of the elements of a vector must have the same mode, that is, data type. One might have a vector of all numeric values, all character values, or all logical values, but not a mixture of data types. If a vector is created with mixed modes, the individual elements are all forced to a single mode such as character or numeric. Single numbers, often called scalars, do not exist as scalars in R. A single number in R is simply a vector with one element.

This important paradigm will be instrumental in understanding how R recycles values to work with unequal-length vectors. For example, how do vectors with more than one element interact with vectors containing a single element (scalars)? Mathematical operations may be applied to two vectors provided the vectors are of the same length. When the two vectors are not of the same length, R recycles the values of the shorter vector until the length of the shorter vector matches that of the longer vector.

Consider R Code 1.2 where **x** is a single element vector (scalar) and **y** and **z** are vectors of lengths 3 and 4, respectively. One way to construct a vector is with the **c()** function, which combines values into either a vector or a list. When **x** (a scalar) is added to **y** (a vector), the value in the scalar is added to each element in the vector **y**. What happens is that **x** is matched in length to **y** by recycling its single value. Then, the elongated **x** is added to **y**, element-wise. When a vector of a single element (scalar) is added to a vector of more than one element, no warning appears; however, when applying mathematical operations to two vectors of unequal length (neither of which is a scalar), a warning message will appear telling the user that the lengths of the vectors are not the same. What the warning message does not tell the user is that recycling has occurred with the shorter vector to grow it to the length of the larger vector.

R Code 1.2

```
> x <- 5          # vector of length one
> y <- c(7, 3, 5)    # combining the values into y
> z <- c(2, 4, 6, 8)  # combining the values into z
> x + y          # adding x and y

[1] 12 8 10

> c(5, 5, 5) + y      # the shorter vector is recycled 3 times

[1] 12 8 10

> y + z          # adding y and z

Warning in y + z: longer object length is not a multiple of shorter object
length

[1] 9 7 11 15
```

```
> c(y, 7) + z           # values of y recycled to length of z
[1] 9 7 11 15
```

When **x** is added to **y**, the value in **x** (5) is recycled until the length of **x** is equal to the length of **y**. Then, the elements of the newly augmented **x** are added to the elements of **y**, one value per index location. When the vector **y** is added to **z**, the length of **y** must first be increased to equal the length of **z**. Since the length of **y** is three, it requires only one value, the first element in **y** (7), to be recycled to obtain a length of four for **y**.

Asking R if the values of **x** are less than the values of **z** returns a logical value (TRUE/FALSE) for each comparison. Logical values are not enclosed in quotes, and a TRUE is stored internally as a 1 and a FALSE as a 0. The logical operators are **>** for greater than, **<** for less than, **<=** for less than or equal to, **>=** for greater than or equal to, **==** for exact equality, **!=** for exact inequality, **&** for vector intersection, **|** for vector union, **&&** for scalar intersection, and **||** for scalar union. Although R does not actually work with scalar quantities, different types of Boolean operators exist for vectors and scalars. To determine the storage mode of an object, one can use the function **typeof()**. R Code 1.3 compares the elements in **x** to see if they are less than the elements in **z**, and the function **typeof()** is used to verify that the vector **LogVec** is a logical vector.

R Code 1.3

```
> LogVec <- (x < z)  # logical vector
> LogVec
[1] FALSE FALSE  TRUE  TRUE

> typeof(LogVec)      # determine how LogVec is stored internally
[1] "logical"
```

Two logical vectors **X** and **Y**, which are different from the lower case **x** and **y** created earlier, are used to illustrate vector intersection and vector union in R Code 1.4. Both ordered elements must be true for the resulting intersection to be TRUE.

R Code 1.4

```
> X <- c(FALSE, TRUE, FALSE)
> Y <- c(FALSE, TRUE, TRUE)
> X & Y  # Boolean X intersection Y
[1] FALSE  TRUE FALSE

> X | Y  # Boolean X union Y
[1] FALSE  TRUE  TRUE

> X == Y  # Boolean EQUALITY
[1]  TRUE  TRUE FALSE

> X != Y  # Boolean INEQUALITY
[1] FALSE FALSE  TRUE
```

When using scalar intersection (`&&`) or scalar union (`||`), only the first elements of each vector are compared.

```
> X && Y # only looks at first element of each vector (intersection)
[1] FALSE

> X || Y # only looks at first element of each vector (union)
[1] FALSE
```

Elements other than the first can be compared by specifying the index of the desired element in square brackets. For example, to compare the third element of Y with the second element of X with scalar intersection one would use `Y[3] && X[2]`. Examples comparing elements other than the first elements from two vectors are illustrated in R Code 1.5.

R Code 1.5

```
> Y[3] && X[2] # compares 3 element of Y to 2 element of X
[1] TRUE

> X[2] && Y[2] # compares second element of each vector (intersection)
[1] TRUE

> X[2] && Y[3] # compares element 2 of X and element 3 of Y (intersection)
[1] TRUE

> X[3] || Y[3] # compares third element of both vectors (union)
[1] TRUE
```

Character elements surrounded by quotes stored in a vector are considered to be of mode character. Objects of different modes stored together are automatically converted to the simplest mode to represent the information. The order of complexity starting with the simplest mode is usually: logical, integer, numeric/double, complex, character, and list. R can coerce objects to have different modes using one of `as.logical()`, `as.integer()`, `as.numeric()`, `as.double()`, `as.complex()`, `as.character()`, or `as.list()` functions. R has two names for its floating point vectors, double and numeric. In R Code 1.6, a logical vector is combined with an integer vector, and R subsequently coerces the resulting vector `IntVec` into an integer vector. The objects used in R Code 1.6 were created in previous R Code chunks starting with R Code 1.2 on page 6.

R Code 1.6

```
> typeof(z) # determine how z is stored internally
[1] "double"

> z1 <- as.integer(z) # coerce z from double to integer
> z1

[1] 2 4 6 8
```

```
> typeof(z1)           # determine how z is stored internally
[1] "integer"

> IntVec <- c(LogVec, z1) # integer vector
> IntVec

[1] 0 0 1 1 2 4 6 8

> typeof(IntVec)        # determine how IntVec is stored internally
[1] "integer"
```

When a logical vector (**LogVec**) is combined with a numeric vector (**z**) and stored in the object **NumVec**, **NumVec** is stored internally as a numeric vector.

```
> NumVec <- c(LogVec, z) # numeric vector
> NumVec

[1] 0 0 1 1 2 4 6 8

> typeof(NumVec) # determine how NumVec is stored internally
[1] "double"
```

When a complex number is added to a numeric vector (**NumVec**) and stored in the object **ComVec**, **ComVec** is stored internally as a complex vector.

```
> ComVec <- c(NumVec, 0+0i) # complex vector
> ComVec

[1] 0+0i 0+0i 1+0i 1+0i 2+0i 4+0i 6+0i 8+0i 0+0i

> typeof(ComVec) # determine how ComVec is stored internally
[1] "complex"
```

When a character string ("dog") is added to a complex vector (**ComVec**) and stored in the object **ChrVec**, **ChrVec** is stored internally as a character vector.

```
> ChrVec <- c(ComVec, "dog") # a character vector
> ChrVec

[1] "0+0i" "0+0i" "1+0i" "1+0i" "2+0i" "4+0i" "6+0i" "8+0i" "0+0i" "dog"

> typeof(ChrVec) # determine how ChrVec is stored internally
[1] "character"
```

When a list is combined with a character vector (**ChrVec**) and stored in the object **Lst**, **Lst** is stored internally as a list.

```
> Lst <- c(ChrVec, list(x = 4)) # a list
> typeof(Lst) # determine how Lst is stored internally

[1] "list"
```

To this point, names for variables have not appeared to have many restrictions, which is the case. Names for variables, or more appropriately for objects, can be constructed from letters, digits, and the period symbol, with the caveat that the name of the object cannot start with a digit or a period followed by a digit. It is also permissible to use the underscore between characters such as `this_one`. Finally, R is case sensitive, so the variables `ABC`, `Abc`, and `aBc` are all considered different.

1.3.1 Naming Cautions

Though R has flexible naming conventions, there are certain names one should not use. One should exercise care not to use names of functions such as `c`, `C`, `q`, `t`, `T`, `mean`, `median`, and so on, since doing so will change the meaning of said functions in the current session and may result in unexpected consequences if any code calls a function that has been overwritten. Furthermore, there are reserved words that should not be used in object names. The reserved words are `if`, `else`, `repeat`, `while`, `function`, `for`, `in`, `next`, `break`, `TRUE`, `FALSE`, `NULL`, `Inf`, `NaN`, `NA`, `NA_integer_`, `NA_real_`, `NA_complex_`, and `NA_character_`.

1.3.2 Vector Indexing

Indices for R vectors start at 1, unlike indices in C and C++, which start at 0. Individual elements of a vector are accessed with `[]` (square brackets). R uses `NA` to represent missing values, `NaN` to represent “not a number,” and `Inf` and `-Inf` to represent very large and small numbers, respectively. R has a few built-in constants such as: `LETTERS`, the 26 upper-case letters of the Roman alphabet; `letters`, the 26 lower-case letters of the Roman alphabet; and `pi`, the ratio of the circumference of a circle to its diameter. To read about other constants, type `help("Constants")` at the R prompt. R Code 1.7 returns `NaN`, `-Inf`, and `Inf` values.

R Code 1.7

```
> NV <- c(-4, 0, 2, 4, 6) # numeric vector
> NV/NV # 0/0 is not a number (NaN)

[1] 1 NaN 1 1 1

> sqrt(NV) # cannot take square root of -4
Warning in sqrt(NV): NaNs produced

[1] NaN 0.000000 1.414214 2.000000 2.449490

> NV^9999 # very large and small numbers use -Inf and Inf

[1] -Inf 0 Inf Inf Inf
```

Elements can be omitted from a vector by using a vector of negative indices inside the square brackets. Positive indices extract the indexed values of the vector. It is not possible to index values of a vector using both positive and negative indices in a single call.

```
> NV[-1] # omit first element of NV

[1] 0 2 4 6

> NV[c(1, 3)] # extract first and third element of NV
```

```
[1] -4 2

> CV <- LETTERS[c(1, 2, 3, 4)] # first four upper case letters of CV
> CV

[1] "A" "B" "C" "D"

> CV[c(2, 3)] # Extract second and third elements of CV

[1] "B" "C"

> LV <- c(TRUE, FALSE, TRUE, TRUE)
> LV

[1] TRUE FALSE TRUE TRUE

> LV[-2] # omit second element of LV

[1] TRUE TRUE TRUE

> LV[-c(1, 3)] # omit first and third elements of LV

[1] FALSE TRUE
```

1.3.3 Generating Vector Sequences and Repeating Vector Constants

The `:` operator and the `seq()` function can be used to generate useful sequences. The `:` operator generates regular sequences from a starting value to an ending value of the form `from:to`. In the previous section, the first four capital letters of the alphabet were specified by typing `LETTERS[c(1, 2, 3, 4)]`. An equivalent solution requiring less typing is `LETTERS[1:4]`. To create patterned, nonconsecutive or non-integer values, one can use the `seq()` function, which allows one to specify an increment using the `by=` argument and the desired length of the sequence using the `length.out=` argument. Consider the sequences shown in R Code 1.8.

R Code 1.8

```
> 24:20 # values 24, 23, 22, 21, and 20
[1] 24 23 22 21 20

> letters[24:20] # 24th through 20th lowercase letters
[1] "x" "w" "v" "u" "t"

> seq(from = 5, to = 25, by = 5)
[1] 5 10 15 20 25

> seq(from = 5, by = 5, length.out = 5)
[1] 5 10 15 20 25

> seq(from = 23, to = 22, length.out = 5)
[1] 23.00 22.75 22.50 22.25 22.00
```

The function `rep()` allows the user to create vectors with repeated constants stored in a vector passed to the argument `x=` by judicious use of the function's arguments `times=`, `each=`, and `length.out=`. R Code 1.9 shows examples of the function `rep()`.

R Code 1.9

```
> rep(x = 5, times = 10)
[1] 5 5 5 5 5 5 5 5 5 5

> rep(x = c(TRUE, FALSE), times = 10)
[1] TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE
[12] FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE

> rep(x = letters[5:8], each = 2)
[1] "e" "e" "f" "f" "g" "g" "h" "h"

> rep(x = 13:11, times = 1:3)
[1] 13 12 12 11 11 11

> rep(x = 1:5, length.out = 20)
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

1.3.4 Filtering Vectors

Filtering allows the user to extract elements of a vector that satisfy certain conditions. Consider R Code 1.10, which creates the vector `FEV`, returns a vector of Boolean values for the expression `FEV*FEV > 3`, and extracts the elements of `FEV` where `FEV*FEV > 3` is true.

R Code 1.10

```
> FEV <- rep(x = 3:1, times = 1:3)
> FEV
[1] 3 2 2 1 1 1

> FEV * FEV
[1] 9 4 4 1 1 1

> FEV * FEV > 3
[1] TRUE TRUE TRUE FALSE FALSE FALSE

> FEV[FEV * FEV > 3]
[1] 3 2 2

> # An equivalent approach using subset
> subset(x = FEV, subset = FEV * FEV > 3)
```

```
[1] 3 2 2

> # Instead of the actual values we may want the indices
> # where these values occur
> which(FEV * FEV > 3)

[1] 1 2 3
```

The function `subset()` returns subsets for vectors, matrices, and data frames. There are two basic arguments: `x=`, the object to be subsetted, and `subset=`, a logical expression indicating elements or rows to keep that need to be supplied regardless of the type of object (vector, matrix, data frame) one is using. When working with matrices or data frames, one may also use the argument `select=`, which requires an expression indicating the columns to select from the matrix or data frame.

1.4 Mode and Class of an Object

All objects in R have a mode and class that describe how the object is stored. The function `mode()` returns the mode of an object, and the function `class()` returns the class of an object. Many functions in R will behave differently depending on the class of their arguments. Functions that behave in this fashion are known as generic functions. Two common generic functions are `print()` and `summary()`. Although the only type of data structure discussed thus far has been the vector, which was described as a variable and subsequently called an object, everything in R is actually an object. Stored results from analyses, matrices, arrays, vectors, and so on, are all considered objects in R. Observe the mode and class of the objects created in R Code 1.11.

R Code 1.11

```
> Num <- c(1, pi, 5)
> Log <- c(TRUE, FALSE, TRUE)
> Chr <- c("a", "character", "vector")
> mode(Num)

[1] "numeric"

> class(Num)

[1] "numeric"

> mode(Log)

[1] "logical"

> class(Log)

[1] "logical"

> mode(Chr)

[1] "character"

> class(Chr)

[1] "character"
```

It is not always the case that the mode and class of an object coincide. The mode of a linear model object is stored as a list; however, the class of a linear model object is `lm`. Consider R Code 1.12, which stores the results from regressing `Y2` onto `x2` in an object named `model`. R uses the tilde (`~`) to separate the left and right sides in a model formula. A formula such as `Y ~ x1 + x2` is read `Y` is modeled by `x1` and `x2`.

R Code 1.12

```
> x2 <- 1:5
> Y2 <- x2 + rnorm(n = 5, mean = 0, sd = 0.5) # add normal errors
> model <- lm(Y2 ~ x2) # Regressing Y2 onto x2
> mode(model) # model has mode list

[1] "list"

> class(model) # model has class lm

[1] "lm"
```

1.5 Getting Help

R has extensive online help as well as HTML files one can access with a web browser. To access the HTML help, type `help.start()` at the R command prompt. The HTML version of the help system has a very useful “Search Engine & Keywords.” To open the help file of a particular function, say `median()`, one might type `?median` or `help(median)` at the R command prompt. Most help files contain useful examples of what they are describing. To run the examples in a help file without copying and pasting the code from the file, one can use the function `example()`. R Code 1.13 runs the examples given in the `median`’s help file.

R Code 1.13

```
> example(median)

median> median(1:4)          # = 2.5 [even number]
[1] 2.5

median> median(c(1:3, 100, 1000)) # = 3 [odd, robust]
[1] 3
```

Many functions will have a long list of arguments. To see the arguments of a function, use the `args()` function. To see the arguments of the `lm()` function, type `args(lm)` at the R prompt.

```
> args(lm)

function (formula, data, subset, weights, na.action, method = "qr",
  model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE,
  contrasts = NULL, offset, ...)
NULL
```

Another way to view the arguments of a function, although fleeting, is to use tab completion. After entering a function and its opening parenthesis, press the tab key, and the arguments for the function appear above the cursor. The arguments disappear as soon as a value is specified in the argument list, yet they may be recalled by pressing the tab key again, provided the closing parenthesis has not been typed. Other ways to view arguments will be discussed later in conjunction with external editors.

In addition to HTML help, the reader may consult any one of the eight (on Windows-based machines) R manuals by selecting **Help→Manuals (in PDF)**. Six of the eight manuals are available under **Help→HTML help** under the *Manuals* heading. If the help files have not provided an answer to your query, it is likely your question has been discussed in the past on the R newsgroup. To view one of the searchable archives from the newsgroup, inside of R, click **Help→R Project home page** (or manually open the URL www.r-project.org), then click on the **Mailing Lists** link along the left side of the page. Move to the R-help section, and click **web-interface**. For Windows users, inside R, one can select **Help→search.r-project.org** or manually open the website <http://search.r-project.org>. Stack Overflow ([http://stack overflow.com](http://stackoverflow.com)) is a searchable site that addresses many different programming issues. Use the “r” tag to find or post an R related programming question on Stack Overflow.

If you still do not find the answer to your question, as a last resort, you might consider posting a well-constructed question to the R help list. Before posting to the R help list, one should read and follow the posting guide at <http://www.r-project.org/posting-guide.html>. Failure to read and follow the posting guide often results in online chastisement for failure to do so when a question does not conform to the posting guidelines.

1.6 External Editors

External editors provide many advantages for command line programming, including syntax highlighting, parentheses matching, and commenting and un-commenting blocks of code. There are many editors the user can select, and the comments about the various editors are based on the authors’ experiences teaching the material in this book. Most readers of a first calculus-based probability and statistics course have not been interested in investing the time needed to master **Emacs Speaks Statistics** (<http://stat.ethz.ch/ESS/>), an extremely powerful scripting editor capable of interacting with various statistics analysis programs such as R, SAS, S-Plus, and Stata; however, Vincent Goulet has collected and distributes a modified version of **GNU Emacs** that includes a few add-ons that are very easy to install for both Windows and MAC users (<http://vgoulet.act.ulaval.ca/en/emacs/>). One of the authors has used **Eclipse** (<http://www.eclipse.org/>) with the **StatET** plug-in (<http://www.walware.de/goto/statet>) to teach the material in this book to computer science majors who have already had exposure to Eclipse in their programming courses. Unfortunately, those students without previous exposure to Eclipse struggled, even with detailed directions on how to set up and use that editor, which detracted from the primary focus of teaching basic probability and statistics concepts. All of the authors have used **Tinn-R** (<http://www.sciviews.org/Tinn-R/>) as an editor to teach the material in this book. Regrettably, students using operating systems other than Windows have been placed at a slight disadvantage, as **Tinn-R** only runs on Windows. Another Windows-only editor is the **RWinEdt** package (<http://cran.r-project.org/web/packages/RWinEdt/index.html>) that is used in conjunction with the **L^AT_EX** editor **WinEdt** (<http://www.winedt.com/>), a share-

ware editor. The **RWinEdt** package provides a nice interface to R for Windows users who are already familiar with the **L^AT_EX** editor **WinEdt** (<http://www.winedt.com/>). All of the aforementioned editors have their followers and ardent supporters. For additional editors, point your browser to http://www.sciviews.org/_rgui/projects/Editors.html.

What editor do the authors recommend you use? The answer depends largely on your personal preferences. For teaching the material in this book to students using various operating systems, the authors' students have had the best experience with **RStudio** (<http://www.rstudio.com/>) in terms of configuration, installation, and general ease of use. While an editor is not required to run R or to use the material in this book, using one is highly recommended. What follows are a few pointers for using **RStudio**.

1.7 RStudio

RStudio is an integrated development environment (IDE) for using and programming R. The desktop version of **RStudio** runs on all major platforms (Windows, Mac, and Linux). Like R, it is an open-source project, which means that it can be downloaded and installed from the Internet for free. Before installing **RStudio**, one should have a relatively recent installation of R. For Windows and Mac OS X users, one simply downloads the self-installing binary from <http://www.rstudio.org/download/desktop>. Figure 1.7 shows a screenshot of how the desktop version of **RStudio** appears in Windows.

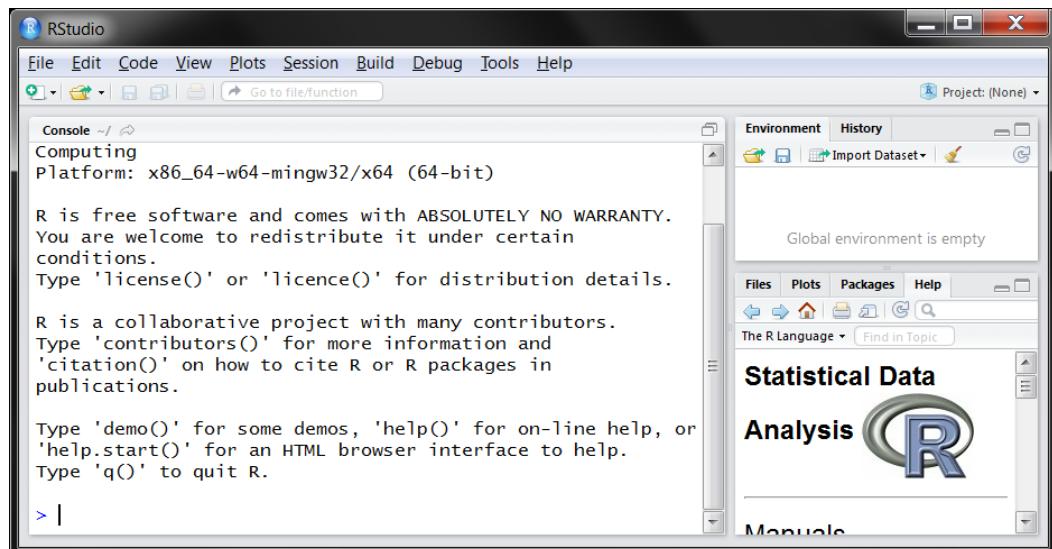


FIGURE 1.7: Screenshot of RStudio desktop

In Figure 1.7, there are three main windows: the *Console*, an *Environment* browser (currently empty), and the *Help* browser. The *Environment* browser and the *Help* browser are part of notebooks that contain other components (History, Files, Plots, and Packages). The source code editor is not open in the screenshot, as no files have been opened to view or

edit. To create a new R script, choose **File**→**New**→**R Script**. Type the code in the editor; then, select **File**→**Save** and give the script a name. The extension **.R** will automatically be appended to the file name, and the file will be saved in R’s default working directory. For Windows machines, the default working directory is the **Documents** folder. For Linux and Mac OS X users, the directory from which R is launched is the working directory. To verify the working directory, one can enter the command `getwd()` at the R prompt.

Notice that the path is specified using forward slashes even for a Windows machine. To use the backwards slash with Windows, one must enter two backward slashes. This is because the backward slash is used in R to start an escape sequence inside character constants. As you work on different projects, or possibly from different chapters of this book, it is a good idea to save your work to named folders. For example, suppose you want to save the work for this chapter to a folder named **Chap1**. First, create the folder; then, change the working directory to where the **Chap1** folder resides using the `setwd()` command. If you are using RStudio, select **Session**→**Set Working Directory**→**Choose Directory**....

Until the working directory is changed, any files created and saved will automatically be saved in the **Chap1** folder. Once the working directory has been changed, R’s workspace is also changed. The workspace in R is known as the global environment, is stored in R as **.GlobalEnv**, and is where any user-defined objects (vectors, matrices, arrays, data frames, lists, functions, etc.) are stored. At the end of an R session, one can save an image of the current workspace that will be automatically reloaded the next time R starts. To list the contents of the workspace, type `ls()` at the R prompt. The contents of the current workspace are also shown in the **Environment** component of RStudio (see Figure 1.8 on the next page). The importance of setting a working directory in R cannot be overstated. This is because all paths in R not beginning with a drive letter on Windows or a leading slash (/) on Unix-like systems are relative to the working directory. It is generally best to have a separate directory for each project you start. While one may save the workspace, the authors prefer to keep a record of the commands entered in a script (hence the emphasis on an external editor) so that the workspace can be recreated at a later date.

RStudio will automatically create a folder and set the working directory to the chosen folder when one uses the Project feature. To create a new project, select **File** →**New Project**. Enter a name for the folder where the new project will be created in the **Directory name:** box, then click the **Create Project** button (see Figure 1.9 on the following page). The current project name is listed on the far right of the toolbar in a drop-down menu that allows one to switch between projects or create new projects (see Figure 1.8 on the next page where the project name **PASWR2E** is visible in the far right of the application toolbar).

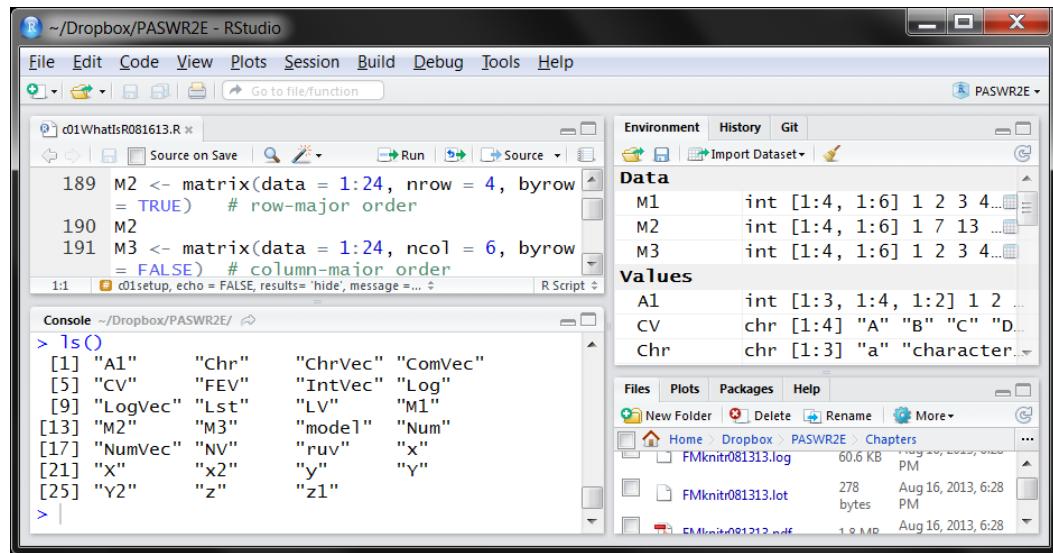


FIGURE 1.8: Contents of the current environment are shown in the top right window in the **Environment** component and in the **Console** component (lower left)

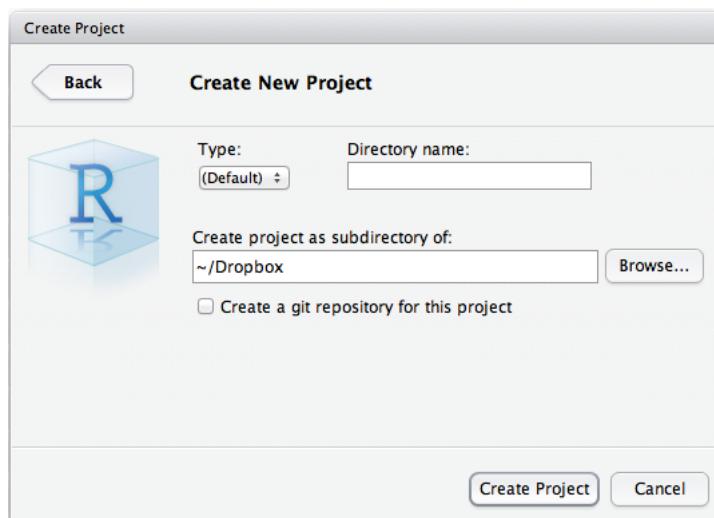


FIGURE 1.9: Create Project dialog for creating a new project

1.8 Packages

Packages are collections of R functions, data, and compiled code that have a uniform organization. All of the data sets, functions, and often the code referenced in this text are available in the package PASWR2. To work through examples and problems, the package PASWR2 should be installed once R is installed. Packages can be installed using the menu interface system on Windows and Mac platforms. On a Windows computer, click **Packages** (the first time you select Packages, you will be prompted to select a CRAN mirror—see Figure 1.10). Then, click **Install Packages(s)** and select the desired package from the menu selection as shown in Figure 1.11 on the next page. Once a package is selected, click **OK**.



FIGURE 1.10: List from which one selects a **CRAN mirror**

From the *Packages* window, use the scroll bar to find the desired package(s). In this case, select PASWR2; then, click **OK**. On a Mac, click on **Packages & Data → Package Installer**. Select the **Get List** button; then, find the package you would like to install from the list. Before clicking **OK**, select the **Install Dependencies** box so that any packages that are required to run the selected package will also be installed if they are not currently installed. The default setting for Windows machines automatically installs required packages so you do not generally have to worry about this step if you are using the menu interface. If you need to install a package (say PASWR2) at the R command line with a Unix-like operating system or just prefer typing, type `install.packages("PASWR2", dependencies = TRUE)`.

If one is using the RStudio editor, regardless of operating system, one clicks on the **Packages** component, which appears in the lower-right panel of the RStudio editor. The **Packages** component is shown in Figure 1.12 on the following page. Once the component is raised, click the **Install Packages** toolbar button in the lower-right panel. Next, select the checkbox for the packages you would like to install. A package will only need to

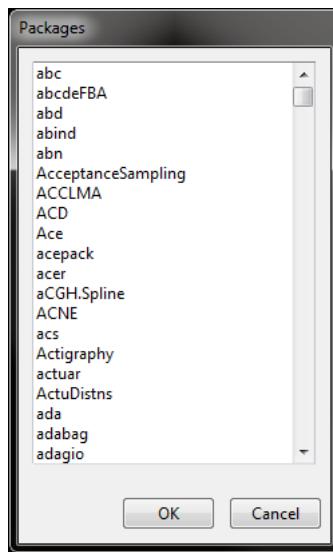


FIGURE 1.11: List of available Packages

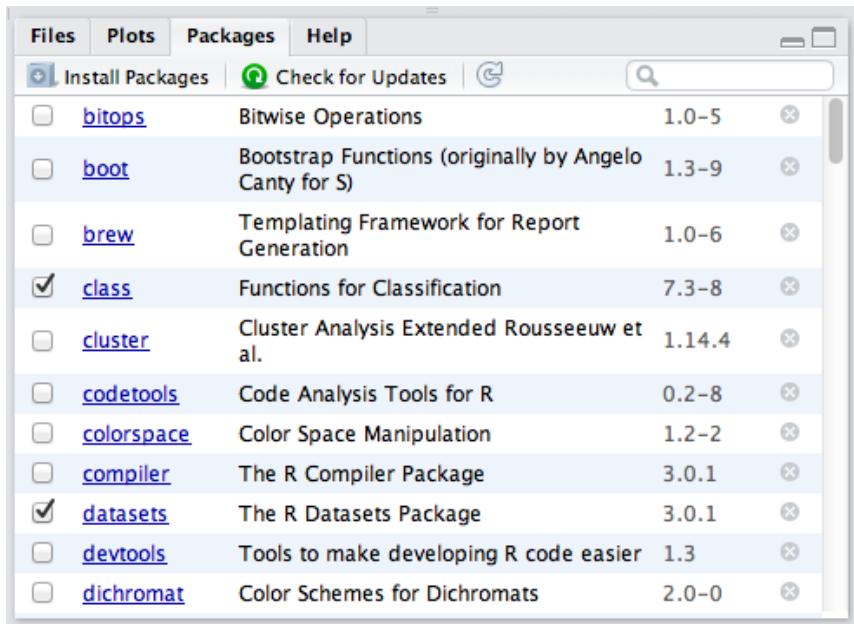


FIGURE 1.12: Packages component of RStudio (bottom right)

be installed once; however, to access the contents of a package, R must be told to search a particular path to find the contents of the package. This is done with the command `library("PackageName")`, where "PackageName" is the case-sensitive name of the package. Issuing the command `library("PackageName")` is generally referred to as "loading a package." A package can only be loaded if it is installed, and the help files for a package are only available once a package has been loaded. A directory where packages are stored on a computer is called a library. The function `.libPaths()` shows where all the libraries are

located, and the function `library()`, when used without any arguments, lists all available packages in the libraries. To summarize, a package must first be installed, and then the contents of the package are made available by issuing the command `library("PackageName")`.

1.9 R Data Structures

R data structures used in this text include vectors, arrays, matrices, factors, lists, and data frames. Construction of vectors using the `c()`, `seq()`, and `rep()` functions as well as the `:` operator was illustrated earlier. The concept of a vector is crucial, as other data structures are defined in terms of vectors. For example, an array is a vector with a dimension attribute, where the dimension attribute is a vector of non-negative integers; a matrix is also a vector with a dimension attribute (of length two that provides the number of rows and columns); a factor is an encoding of a vector into categories; most R lists are generic vectors; and data frames are data structures similar to matrices that allow the columns (vectors) to be of differing types (numeric, logical, character, etc.). Data frames are the fundamental data structure used for most of R's modeling software and are the primary structure used to archive data in the PASWR2 package.

1.9.1 Arrays and Matrices

Arrays are multidimensional arrangements of elements. A very common example of an array is a matrix, which is a two-dimensional array. The examples in R Code 1.14 show how arrays are constructed from vectors by specifying the arrays' dimensions. The reader should note that the elements of a vector are stored in an array according to the `dim=` attribute argument, which provides the maximal indices in each dimension of the array in row, column, and layer order. For array `A1` (shown in the following code), the values 1 through 24, corresponding to the vector passed to the `data=` argument, are entered down the columns from left to right. This is known as column-major order.

R Code 1.14

```
> A1 <- array(data = 1:24, dim = c(3, 4, 2))
> A1

, , 1

 [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2

 [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24

> is.array(A1)
```

```
[1] TRUE
> is.matrix(A1)
[1] FALSE
> class(A1)
[1] "array"
> dim(A1)
[1] 3 4 2
```

Next, a 4-by-6 matrix is constructed and stored in the object M1.

```
> M1 <- array(data = 1:24, dim = c(4, 6))
> M1
[,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    5    9   13   17   21
[2,]    2    6   10   14   18   22
[3,]    3    7   11   15   19   23
[4,]    4    8   12   16   20   24

> is.array(M1)
[1] TRUE
> is.matrix(M1)
[1] TRUE
> class(M1)
[1] "matrix"
> dim(M1)
[1] 4 6
```

Note that A1 is an array and not a matrix since it has more than two dimensions; however, M1 is both an array and a matrix, and it has a class designation "`matrix`." Although two-dimensional arrays are matrices and the `array()` function can be used to create a matrix, the `matrix()` function allows the user to create matrices by using row-major order as well as column-major order. When using the function `matrix()`, one may specify the number of rows using the argument `nrow=`; and the number of columns will automatically be computed based on the length of the vector provided to the `data=` argument. Likewise, if one specifies the number of columns using the `ncol=` argument, the number of rows will automatically be computed based on the length of the vector provided to the `data=` argument.

```
> M2 <- matrix(data = 1:24, nrow = 4, byrow = TRUE) # row-major order
> M2
```

```
[,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1     2     3     4     5     6
[2,]    7     8     9    10    11    12
[3,]   13    14    15    16    17    18
[4,]   19    20    21    22    23    24

> M3 <- matrix(data = 1:24, ncol = 6, byrow = FALSE) # column-major order
> M3

[,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1     5     9    13    17    21
[2,]    2     6    10    14    18    22
[3,]    3     7    11    15    19    23
[4,]    4     8    12    16    20    24
```

In the following example, different types of barley are in the columns; and different provinces in Spain are in the rows. The entries in the matrix represent the weight in thousands of metric tons for each type of barley produced in a given province. The `barley.data` matrix will be used to illustrate various functions and manipulations that can be applied to a matrix. Given the matrix

$$\begin{pmatrix} 190 & 8 & 22.0 \\ 191 & 4 & 1.7 \\ 223 & 80 & 2.0 \end{pmatrix},$$

the values are written to a matrix (reading across the rows with the command `byrow = TRUE`) with the name `barley.data` as follows:

```
> Data <- c(190, 8, 22, 191, 4, 1.7, 223, 80, 2)
> barley.data <- matrix(data = Data, nrow = 3, byrow = TRUE)
> barley.data

[,1] [,2] [,3]
[1,] 190    8 22.0
[2,] 191    4 1.7
[3,] 223   80 2.0
```

The matrix's dimensions are returned by typing `dim(barley.data)`:

```
> dim(barley.data)
[1] 3 3
```

The function `dim()` applied to `barley.data` in the previous code returns a vector of length two where the first returned value is the number of rows and the second returned value is the number of columns. Consider the following code that creates two objects where the names of the three provinces are assigned to `province` and the three types of barley to `type`:

```
> province <- c("Navarra", "Zaragoza", "Madrid")
> type <- c("typeA", "typeB", "typeC")
```

Assign the names stored in `province` to the rows of the matrix as follows:

```
> dimnames(barley.data) <- list(province, NULL)
> barley.data

[,1] [,2] [,3]
Navarra   190     8 22.0
Zaragoza  191     4 1.7
Madrid    223    80 2.0
```

Next, assign the names stored in `type` to the columns of the matrix:

```
> dimnames(barley.data) <- list(NULL, type)
> barley.data

typeA typeB typeC
[1,]   190     8 22.0
[2,]   191     4 1.7
[3,]   223    80 2.0
```

To assign row and column names simultaneously, the command that should be used is `dimnames(barley.data) <- list(province, type)`:

```
> dimnames(barley.data) <- list(province, type)
> barley.data

typeA typeB typeC
Navarra   190     8 22.0
Zaragoza  191     4 1.7
Madrid    223    80 2.0
```

One can verify the assigned names with the function `dimnames()`:

```
> dimnames(barley.data)

[[1]]
[1] "Navarra"  "Zaragoza" "Madrid"

[[2]]
[1] "typeA"   "typeB"   "typeC"
```

To delete the row and column name assignments, type

```
> dimnames(barley.data) <- NULL
> barley.data

[,1] [,2] [,3]
[1,]   190     8 22.0
[2,]   191     4 1.7
[3,]   223    80 2.0
```

If one is interested in only the second row of data, one can enter

```
> dimnames(barley.data) <- list(province, type)
> barley.data[2, ]

typeA typeB typeC
191.0  4.0   1.7
```

or

```
> barley.data["Zaragoza", ]
```

	typeA	typeB	typeC
191.0	4.0	1.7	

Note that `barley.data[2,]` and `barley.data["Zaragoza",]` are equivalent to typing `barley.data[2, 1:3]` and `barley.data["Zaragoza", 1:3]`. That is, when an index position is left empty, the full range of the index is brought into play. To see the third column, key in

```
> barley.data[, "typeC"]
```

	Navarra	Zaragoza	Madrid
	22.0	1.7	2.0

To add an additional column for a fourth type of barley (`typeD`), use the `cbind()` command. Once `typeD` is part of the the `barley.data` data frame, it is removed from the workspace to avoid confusion.

```
> typeD <- c(2, 3.5, 2.75)
> barley.data <- cbind(barley.data, typeD)
> rm("typeD") # remove typeD from workspace
> barley.data
```

	typeA	typeB	typeC	typeD
Navarra	190	8	22.0	2.00
Zaragoza	191	4	1.7	3.50
Madrid	223	80	2.0	2.75

The function `apply()` allows the user to apply a function to one or more of the dimensions of an array. To calculate the mean of the columns for the matrix `barley.data`, type `apply(X = barley.data, MARGIN = 2, FUN = mean)`, where the value passed to `X` is an array (recall that a matrix is a two-dimensional array). The value of 2 passed to `MARGIN` tells the function to work on the columns. Additionally, the value `mean` passed to `FUN` specifies the function to apply to the respective margin of the array. To read more about `apply()`, type either `?apply` or `help(apply)` at the R prompt.

```
> apply(X = barley.data, MARGIN = 2, FUN = mean)

      typeA      typeB      typeC      typeD
201.333333 30.666667  8.566667  2.750000
```

The second argument, `MARGIN = 2` in the previous example, tells the function `apply()` to work on the columns. For the function to work on rows, the second argument should be `MARGIN = 1`. For example, to find the average barley weight for each province, type

```
> apply(X = barley.data, MARGIN = 1, FUN = mean)

      Navarra      Zaragoza      Madrid
    55.5000 50.0500 76.9375
```

The function `names()` allows the assignment of names to vectors:

```
> x <- c(1, 2, 3)
> names(x) <- c("A", "B", "C")
> x
A B C
1 2 3
```

To suppress the names of a vector, type `names(x) <- NULL`:

```
> names(x) <- NULL
> x
[1] 1 2 3
```

Earlier, the text used `barley.data[2,]` to extract all columns (barley types) for the second row (Zaragoza). The object returned is a vector, not a matrix. This dimensional reduction may seem innocuous at first; however, it may cause problems in code that involves matrix operations. To prevent dimension reduction, one should use the `drop = FALSE` argument. Consider how R Code 1.15 illustrates both the potential problem of dimension reduction as well as code to prevent the dimension reduction.

R Code 1.15

```
> barley.data[2, ]
typeA typeB typeC typeD
191.0   4.0   1.7   3.5

> dim(barley.data[2, ]) # Not a matrix...a vector now
NULL

> is.vector(barley.data[2, ])
[1] TRUE

> barley.data[2, , drop = FALSE] # A 1*4 matrix not a vector
      typeA typeB typeC typeD
Zaragoza   191     4    1.7    3.5

> is.matrix(barley.data[2, , drop = FALSE])
[1] TRUE

> dim(barley.data[2, , drop = FALSE])
[1] 1 4
```

If one has a vector or a matrix that has been reduced to a vector by mistake and needs to return the object to a matrix, consider using the `as.matrix()` function. When using the `as.matrix()` function, be sure to pay particular attention to the dimension of the resulting matrix. To transpose a matrix, use the `t()` function.

```
> as.matrix(barley.data[2, , drop = FALSE]) # 1*4 matrix
  typeA typeB typeC typeD
Zaragoza    191     4    1.7   3.5

> t(as.matrix(barley.data[2, , drop = FALSE])) # 4*1 matrix
  Zaragoza
typeA     191.0
typeB      4.0
typeC      1.7
typeD      3.5
```

1.9.2 Vector and Matrix Operations

Consider the system of equations:

$$\begin{aligned} 3x + 2y + 1z &= 10 \\ 2x - 3y + 1z &= -1 \\ 1x + 1y + 1z &= 6. \end{aligned}$$

This system can be represented with matrices and vectors as

$$\mathbf{Ax} = \mathbf{b}, \text{ where } \mathbf{A} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & -3 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \text{ and } \mathbf{b} = \begin{bmatrix} 10 \\ -1 \\ 6 \end{bmatrix}.$$

To solve this system of equations, enter **A** and **b** into R and type **solve(A, b)** at the command prompt:

```
> A <- matrix(c(3, 2, 1, 2, -3, 1, 1, 1, 1), byrow = TRUE, nrow = 3)
> A
  [,1] [,2] [,3]
[1,]    3    2    1
[2,]    2   -3    1
[3,]    1    1    1

> b <- matrix(c(10, -1, 6), byrow = TRUE, nrow = 3)
> b
  [,1]
[1,]   10
[2,]   -1
[3,]    6

> x <- solve(A, b)
> x
  [,1]
[1,]    1
[2,]    2
[3,]    3
```

The operator `%*%` is used for matrix multiplication. If \mathbf{x} is an $(n \times 1)$ column vector, and \mathbf{A} is an $(m \times n)$ matrix, then the product of \mathbf{A} and \mathbf{x} is computed by typing `A %*% x`. To verify R's solution, multiply $\mathbf{A} \times \mathbf{x}$, and note that this is equal to \mathbf{b} :

```
> A %*% x
[1,] 10
[2,] -1
[3,] 6
```

Other common functions used with vectors and matrices are included in Table A.2 on page 904.

1.9.3 Factors

A factor is a vector with additional information indicating the distinct values or levels of the vector. Many statistical problems use categorical variables to classify subjects or to subdivide the data. Examples include variables such as socioeconomic status, gender, and marriage status. When using R, one should store categorical data in factors. To create a factor from a vector, one can use the `factor()` function.

Consider a variable where a physician rates how challenging it was for him to find a location for an epidural block as one of the following: easy, difficult, or impossible. For a concrete example, suppose the physician rates four patients where the first is impossible, the second is difficult, the third is easy, and the last patient is rated as impossible. To minimize typing, patients rated as easy, difficult, and impossible are assigned numerical values of 0, 1, and 2, respectively. By using the `levels=` and the `labels=` arguments for the function `factor()`, one can create a factor with three labeled levels as in R Code 1.16.

R Code 1.16

```
> v <- c(2, 1, 0, 2)
> fv <- factor(v)
> fv
[1] 2 1 0 2
Levels: 0 1 2

> fv <- factor(v, levels = 0:2, labels = c("Impossible", "Difficult",
+ "Easy"))
> fv
[1] Impossible Difficult Easy      Impossible
Levels: Easy Difficult Impossible
```

It is also possible to create a factor from a character vector. One should note that if the `levels=` argument for the `factor()` function is not used with character vectors, the levels of the factor will be sorted alphabetically, which may not be appropriate. In R Code 1.17, the levels of the factor `fv2` are ordered alphabetically as Difficult, Easy, and Impossible.

R Code 1.17

```
> v2 <- c("Impossible", "Difficult", "Easy", "Impossible")
```

```
> fv2 <- factor(v2)
> fv2
[1] Impossible Difficult Easy      Impossible
Levels: Difficult Easy Impossible
```

To establish the correct levels of ease of palpation, one can use the `levels=` argument or the `levels()` function both illustrated in R Code 1.18.

R Code 1.18

```
> fv3 <- factor(v2, levels = c("Easy", "Difficult", "Impossible"))
> fv3
[1] Impossible Difficult Easy      Impossible
Levels: Easy Difficult Impossible

> levels(fv2) <- c("Easy", "Difficult", "Impossible")
> fv2
[1] Impossible Easy      Difficult Impossible
Levels: Easy Difficult Impossible
```

1.9.4 Lists

A list is an object whose elements can be of different modes (character, numeric, logical, etc.). Lists are used to unite related data that have different modes. Since many R functions return results using lists, it is important to know how to extract information from a list. Consider two lists, the first one named `stu1` and the second one named `stu2`. In the first list, tags are used to provide the different objects with names, and the names of the different objects are displayed using the `names()` function. Since tags are not mandatory, the list `stu2` is created without tags, so the reader can better see how to index a list. An individual component, say `major`, of the `stu1` list may be accessed using `$` prefixing or with `[[]]` by specifying inside the double square brackets the name of the component in quotes or the index number of the component. To reduce the chance of accessing the wrong component, it is generally better to use the name of the component instead of its index. Creation of the first list (`stu1`) is shown in R Code 1.19. When an entire R expression does not fit on a single line, a `+` symbol appears where the expression wraps to subsequent lines to let the user know the expression is not complete. Once the R expression is complete, the R prompt `>` will reappear in the R console. If the `+` appears in the R console and the user cannot discern how to complete the R expression, a quick solution is to press the escape key, which will return the R prompt but also remove the last expression the user was typing.

R Code 1.19

```
> stu1 <- list(first.name = "Bob", last.name = "Smith",
+               major = "Statistics", semester.hours = 18,
+               grades = c("A", "B+", "A-", "C+", "B", "B-"))
> names(stu1$grades) <- c("Analysis", "Experimental Design", "English",
+                           "German", "Regression", "Programming")
> stu1
```

```
$first.name
[1] "Bob"

$last.name
[1] "Smith"

$major
[1] "Statistics"

$semester.hours
[1] 18

$grades
      Analysis Experimental Design          English
      "A"           "B+"           "A-"
German          Regression          Programming
      "C+"           "B"           "B-"

> names(stu1)
[1] "first.name"      "last.name"       "major"          "semester.hours"
[5] "grades"
```

Creation of the second list (`stu2`) is shown in R Code 1.20.

R Code 1.20

```
> stu2 <- list("Bob", "Smith", "Statistics", 18,
+                c("A", "B+", "A-", "C+", "B", "B-"))
> stu2
[[1]]
[1] "Bob"

[[2]]
[1] "Smith"

[[3]]
[1] "Statistics"

[[4]]
[1] 18

[[5]]
[1] "A"   "B+"  "A-"  "C+"  "B"   "B-"

> names(stu2) # tags not used
NULL
```

Note that `stu1$major`, `stu1[["major"]]`, and `stu1[[3]]` all return the same value ("Statistics") in R Code 1.21 on the facing page.

R Code 1.21

```
> stu1$major
[1] "Statistics"

> stu1[["major"]]
[1] "Statistics"

> stu1[[3]]
[1] "Statistics"
```

R Code 1.21 illustrates how components are extracted from lists using double square brackets ([[]]). When the n^{th} component is a vector, the i^{th} element of the n^{th} component can be extracted using single square brackets ([]). One can also extract the i^{th} element of a named component using either `Lst[["name"]][i]` or `Lst$name[i]`, where `Lst` is a list. R Code 1.22 illustrates four approaches to extract the 6th element from the `grades` vector of the `stu1` list.

R Code 1.22

```
> stu1$grades["Programming"]
Programming
  "B-"

> stu1[["grades"]][["Programming"]]
Programming
  "B-"

> stu1[[5]][["Programming"]]
Programming
  "B-"

> stu1[[5]][6]
Programming
  "B-"
```

1.9.5 Data Frames

A data frame is similar to a matrix in the sense that it is a rectangular structure used to store information. It is different in that all elements of a matrix must be of the same mode (numeric, character, etc.), but this restriction does not apply to data frames. That is, data frames have a two-dimensional structure with rows (experimental units) and columns (variables) where all columns have the same number of rows, which have unique names; yet the columns (variables) in a data frame are not required to be of the same mode. Another way to think of a data frame is as a list with the restriction that all its components are equal length vectors. The data frame is the fundamental data structure used with functions

from the `lattice` and `ggplot2` graphics packages and in most of R's modeling functions. It is also the format used to archive all of the data in the `PASWR2` package.

1.9.5.1 Creating Data Frames

The function `data.frame()` can be used to create a data frame by using equal length vectors as the first arguments of the `data.frame()` function. By default, any character variables are converted to factors. If this is undesirable, change the default value of `TRUE` to `FALSE` in the `stringsAsFactor=` argument inside the `data.frame()` function. Unique row names are assigned to the rows of the data frame based on what is passed to the `row.names=` argument. The default argument of `NULL` sequentially numbers the rows starting with a 1.

In R Code 1.23, data frame `DF1` is created. `DF1` uses the default row names. The `str()` function can be used to see the structure of an R object and is used to view the structure of `DF1`. Note that the variable `nv` has class numeric, `cv` has class Factor, and `lv` has class logical. Next, a data frame, `DF2`, is created that uses a character vector supplied to the `row.names=` argument to create unique row names. The `str()` function is also used to view the structure of `DF2`.

R Code 1.23

```
> nv <- c(1, 3, 6, 8)                                # Numeric vector
> cv <- c("a", "d", "f", "p")                         # Character vector
> lv <- c(TRUE, FALSE, FALSE, TRUE)                   # Logical vector
> DF1 <- data.frame(nv, cv, lv)
> DF1

  nv cv    lv
1  1 a  TRUE
2  3 d FALSE
3  6 f FALSE
4  8 p  TRUE

> str(DF1)

'data.frame': 4 obs. of  3 variables:
$ nv: num  1 3 6 8
$ cv: Factor w/ 4 levels "a","d","f","p": 1 2 3 4
$ lv: logi  TRUE FALSE FALSE TRUE

> DF2 <- data.frame(nv, cv, lv, row.names = c("Joe", "Bob", "Jill", "Sam"),
+                     stringsAsFactors = FALSE)
> DF2

  nv cv    lv
Joe   1 a  TRUE
Bob   3 d FALSE
Jill  6 f FALSE
Sam   8 p  TRUE

> str(DF2)

'data.frame': 4 obs. of  3 variables:
$ nv: num  1 3 6 8
$ cv: chr  "a" "d" "f" "p"
$ lv: logi  TRUE FALSE FALSE TRUE

> rm("nv", "cv", "lv")  # remove the variables from the current environment
```

From the output of R Code 1.23, note that the variable `cv` has class character (`chr`) in `DF2` but has class `Factor` in `DF1` because `stringsAsFactors` was set to `FALSE` in creating `DF2`. One final difference between `DF1` and `DF2` is that `DF2` has names for the rows, that are created by providing a vector of names to the `row.names=` argument in the `data.frame()` function. Row names can be added after the creation of a data frame with the function `row.names()`.

```
> row.names(DF1) <- c("Joe", "Bob", "Jill", "Sam")
> DF1

  nv cv    lv
Joe 1 a  TRUE
Bob 3 d FALSE
Jill 6 f FALSE
Sam 8 p  TRUE
```

1.9.5.2 Accessing Data Frames

Since a data frame is technically a list, its components can be accessed with the same techniques used to access the components of a list, namely `$` prefixing or with `[[]]` by specifying inside the double square brackets the name of the component in quotes or the index number of the component. Using the previously created data frame `DF2` and recalling that the variables `nv`, `cv`, and `lv` were removed from the current environment using the `rm()` function, different ways of accessing the information in the variable `nv` are illustrated. Since the individual vectors `nv`, `cv`, and `lv` are no longer present in the current environment, when one enters `nv` at the R prompt, R indicates it cannot find any such object. Seven different ways of accessing the information in the variable `nv` are illustrated in R Code 1.24: dollar (`$`) prefixing , component indexing, component naming, array indexing with names and indices, using the `with()` function, and using the `attach()` function. Note that `DF2[, "nv"]` is equivalent to `DF2[1:4, "nv"]`. That is, when an index position is left empty, the full range of the index is brought into play.

R Code 1.24

```
> DF2

  nv cv    lv
Joe 1 a  TRUE
Bob 3 d FALSE
Jill 6 f FALSE
Sam 8 p  TRUE

> nv          # nv not on search path it is part of DF2
Error in eval(expr, envir, enclos): object 'nv' not found

> DF2$nv        # dollar prefixing
[1] 1 3 6 8

> DF2[[1]]      # component indexing
[1] 1 3 6 8
```

```

> DF2[["nv"]]    # component naming
[1] 1 3 6 8

> DF2[, "nv"]    # all rows, column nv
[1] 1 3 6 8

> DF2[, 1]        # all rows, column 1
[1] 1 3 6 8

> with(data = DF2, expr = nv)
[1] 1 3 6 8

> attach(DF2)    # DF2 on search path
> nv
[1] 1 3 6 8

> detach(DF2)   # DF2 removed from search path
> nv            # nv no longer on search path

Error in eval(expr, envir, enclos): object 'nv' not found

```

If what the user wants is the information in `nv`, simply typing the variable name at the R prompt after attaching the data frame involves less typing than the other four solutions; however, one must exercise care when using the `attach()` function with data frames, especially if any changes are made to the data frame. To understand how R searches for objects and to illustrate the potential danger of using `attach()` after changing the values in a data frame, consider the function `search()`, which returns a list of attached packages and objects. When a data frame is attached using the `attach()` function, it moves to the second position in the search path.

```

> search()      # show attached packages
[1] ".GlobalEnv"           "package:leaps"
[3] "package:scatterplot3d" "package:multcomp"
[5] "package:TH.data"       "package:mvtnorm"
[7] "package:car"           "package:boot"
[9] "package:gridExtra"     "package:nortest"
[11] "package:coin"          "package:survival"
[13] "package:binom"         "package:cubature"
[15] "package:plyr"          "package:extrafont"
[17] "package:fontcm"        "package:mapproj"
[19] "package:maps"          "package:xtable"
[21] "package:vcd"           "package:grid"
[23] "package:tikzDevice"   "package:xlsx"
[25] "package:xlsxjars"      "package:rJava"
[27] "package:repmis"         "package:MASS"
[29] "package:PASWR2"        "package:lattice"
[31] "package:ggplot2"        "package:knitr"

```

```
[33] "package:stats"      "package:graphics"
[35] "package:grDevices"   "package:utils"
[37] "package:datasets"    "package:methods"
[39] "Autoloads"           "package:base"

> ls(1)                 # shows objects in .GlobalEnv

[1] "A"                  "A1"                 "b"                  "barley.data" "Chr"
[6] "ChrVec"              "ComVec"              "CV"                "Data"        "DF1"
[11] "DF2"                 "FEV"                "fv"                "fv2"        "fv3"
[16] "IntVec"              "Log"                "LogVec"             "Lst"        "LV"
[21] "M1"                  "M2"                 "M3"                "model"      "Num"
[26] "NumVec"              "NV"                 "province"          "ruv"        "stu1"
[31] "stu2"                "type"               "v"                 "v2"         "x"
[36] "X"                   "x2"                 "y"                 "Y"          "Y2"
[41] "z"                   "z1"

> attach(DF2)  # place DF2 in search path pos. 2
> search()      # shows DF2 in pos. 2

[1] ".GlobalEnv"          "DF2"
[3] "package:leaps"        "package:scatterplot3d"
[5] "package:multcomp"      "package:TH.data"
[7] "package:mvtnorm"       "package:car"
[9] "package:boot"          "package:gridExtra"
[11] "package:nortest"       "package:coin"
[13] "package:survival"      "package:binom"
[15] "package:cubature"      "package:plyr"
[17] "package:extrafont"     "package:fontcm"
[19] "package:mapproj"       "package:maps"
[21] "package:xtable"        "package:vcd"
[23] "package:grid"          "package:tikzDevice"
[25] "package:xlsx"          "package:xlsxjars"
[27] "package:rJava"         "package:repmis"
[29] "package:MASS"          "package:PASWR2"
[31] "package:lattice"        "package:ggplot2"
[33] "package:knitr"          "package:stats"
[35] "package:graphics"       "package:grDevices"
[37] "package:utils"          "package:datasets"
[39] "package:methods"        "Autoloads"
[41] "package:base"

> ls(2)                 # shows objects in pos. 2 (DF2)

[1] "cv" "lv" "nv"
```

Suppose the values in `nv` have been systematically recorded incorrectly such that the correct values should be the current values plus 5. One may be tempted to change the values with the code `nv <- nv + 5`. Doing this is not a good solution because it creates another `nv` object in R's search path that is not equivalent to the `nv` object in position 2. Observe how the `nv` stored in the global environment (position 1 of the search path) and the object `nv` of the `DF2` data frame (position 2 of the search path) have different values.

```

> nv <- nv + 5 # nv stored in workspace
> nv           # nv from pos. 1

[1] 6 8 11 13

> ls(2)         # list objects in position 2

[1] "cv" "lv" "nv"

> DF2$nv        # nv from pos. 2

[1] 1 3 6 8

> detach(DF2)   # remove DF2 from search path
> search()      # show attached packages

[1] ".GlobalEnv"          "package:leaps"
[3] "package:scatterplot3d" "package:multcomp"
[5] "package:TH.data"       "package:mvtnorm"
[7] "package:car"           "package:boot"
[9] "package:gridExtra"     "package:nortest"
[11] "package:coin"          "package:survival"
[13] "package:binom"         "package:cubature"
[15] "package:plyr"          "package:extrafont"
[17] "package:fontcm"        "package:mapproj"
[19] "package:maps"          "package:xtable"
[21] "package:vcd"           "package:grid"
[23] "package:tikzDevice"    "package:xlsx"
[25] "package:xlsxjars"      "package:rJava"
[27] "package:repmis"         "package:MASS"
[29] "package:PASWR2"         "package:lattice"
[31] "package:ggplot2"        "package:knitr"
[33] "package:stats"          "package:graphics"
[35] "package:grDevices"      "package:utils"
[37] "package:datasets"       "package:methods"
[39] "Autoloads"              "package:base"

```

To change the values of the variable `nv` for the data frame `DF2`, one might use something like `DF2$nv <- DF2$nv + 5`.

```

> DF2$nv <- DF2$nv + 5 # nv changed inside DF2
> DF2

  nv cv    lv
Joe  6  a  TRUE
Bob  8  d FALSE
Jill 11  f FALSE
Sam 13  p  TRUE

```

While the `attach()` function will often be used for convenience, be aware that changes to variables of attached data frames are not saved in the attached data frame. It is a good practice to remove a data frame from the search path using the `detach()` function when you no longer need the data frame. It is also possible to return the i^{th} column(s) of a data

frame as a data frame using single brackets ([]) by providing a vector specifying the desired columns. If array indexing is used, the default argument `drop = TRUE` must be changed to `drop = FALSE` as illustrated in R Code 1.25.

R Code 1.25

```
> DF2[c(1, 3)] # extract 1st and 3rd columns

      nv      lv
Joe    6  TRUE
Bob    8 FALSE
Jill  11 FALSE
Sam   13  TRUE

> DF2[, c("nv", "lv")] # extract columns named nv and lv

      nv      lv
Joe    6  TRUE
Bob    8 FALSE
Jill  11 FALSE
Sam   13  TRUE

> DF2[, c(1, 3), drop = FALSE] # extract all rows for 1st and 3rd columns

      nv      lv
Joe    6  TRUE
Bob    8 FALSE
Jill  11 FALSE
Sam   13  TRUE

> DF2[, c("nv", "lv"), drop = FALSE] # all rows for columns nv and lv

      nv      lv
Joe    6  TRUE
Bob    8 FALSE
Jill  11 FALSE
Sam   13  TRUE
```

It is also possible to use the convenience function `subset()` as follows.

```
> subset(x = DF2, select = c(nv, lv))

      nv      lv
Joe    6  TRUE
Bob    8 FALSE
Jill  11 FALSE
Sam   13  TRUE
```

1.9.5.3 Accessing Data from Packages

Package authors may store their data in many different formats, with data frames being one of the more common formats. To access data sets from a package, the package must first be in the search path. To place a package, say `PackageName`, in the search

path, one should use the `library()` function (`library(PackageName)`). Removal of a package, say `PackageName`, from the search path is done with the `detach()` function (`detach(package:PackageName)`). Once a package is in the search path, its contents can generally be viewed by typing `data()` at the R prompt. To view a short description of all available data sets installed on your machine, type `data()` at the R prompt (see Figure 1.13). The data sets of a particular package, provided it is installed, can be viewed by entering `data(package="PackageName")` where `PackageName` is the case-sensitive name of the desired package. To view the data sets for all installed packages, enter `data(package = .packages(all.available = TRUE))`.

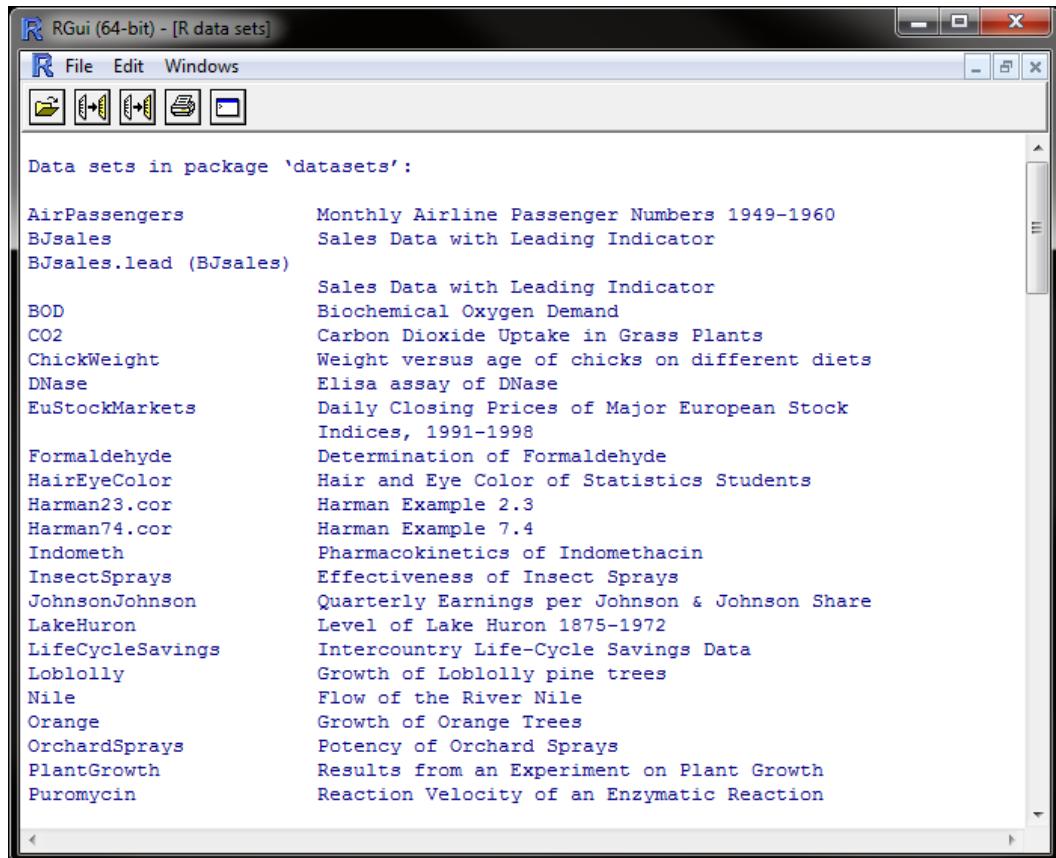


FIGURE 1.13: Available data sets

R Code 1.26 places the package MASS in the search path, opens an HTML help file (as in Figure 1.14 on the facing page) where information is provided about the data frame **Animals**, and uses the function `head(Animals)` to show the first six rows of the data frame. To view the last six rows of a data frame, use the `tail()` function. To view a different number of rows other than the default six with the functions `head()` and `tail()`, use the argument `n=` to specify the number of rows one desires to view.

R Code 1.26

```
> library(MASS) # Places MASS in search path
```

```
> help(Animals) # Opens HTML help window
> head(Animals) # shows first six rows

      body brain
Mountain beaver    1.35   8.1
Cow            465.00 423.0
Grey wolf        36.33 119.5
Goat           27.66 115.0
Guinea pig       1.04   5.5
Diplodocus     11700.00 50.0
```

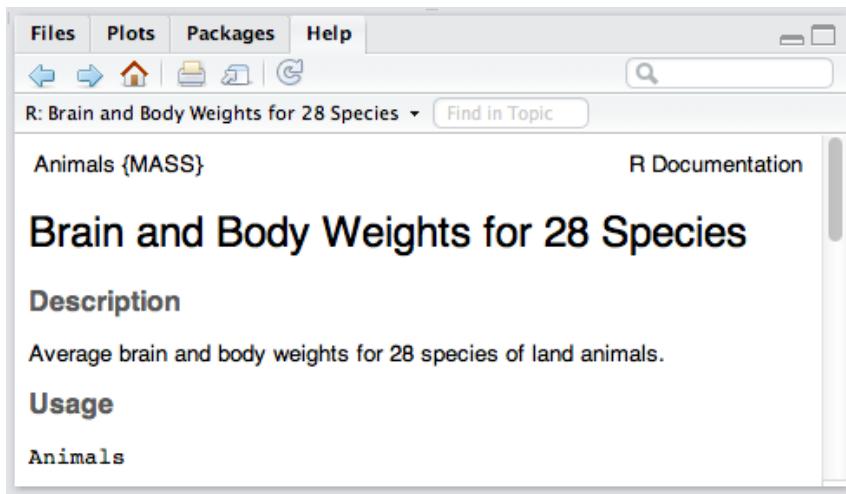


FIGURE 1.14: **Help** component showing the HTML help file for the `Animals` data frame from the `MASS` package

While most of the data in this text is stored in the `PASWR2` package, the reader may need to work with data stored in external files in a variety of different formats. For example, the reader may have data stored in a spreadsheet as an `.xls`, `.xlsx`, or `.csv` file, or one may want to read in data stored as text in a `.txt` file. R can read many different types of files and the formats that are most useful for the material in this text will be covered.

1.10 Reading and Saving Data in R

R has the ability to read data from external files stored in several formats. To read data in formats other than ASCII, the package `foreign` can be used. This package is able to read formats from other statistical programs such as `EpiInfo`, `Minitab`, `S-Plus`, `SAS`, `SPSS`, `Stata`, and `Systat`. For more information on reading data from other statistical programs, relational databases, and binary files, please reference the *R Data Import/Export Manual* available online and accessible from the **R HTML** help menu or the **Help** component of

RStudio. For all but the smallest of data sets, when working with data stored in a format not readable by R, it will almost always prove easier first to save the original data as a text file and then to read the external file using `read.table()` or `scan()`. Typically, `read.table()` is more user friendly, although `scan()` reads large data of a single mode more quickly than does `read.table()`. It also reads data from the console. To download a wide variety of files from the Internet, one can use the function `download.file()`, which allows the user to specify where they want the downloaded file saved by using the `destfile=` argument.

1.10.1 Using `read.table()`

The function `read.table()` reads a file in table format (a rectangular data set) and creates a data frame from that file. The file may be on your computer, at an unsecure (`http`) website or, for Windows users, the contents of the clipboard (`file = "clipboard"`). To create an R data frame from an external file, one can use the function `read.table()` or one of its variants. Consider the text file `FAT.txt` available online from the website: <http://www1.appstate.edu/~arnholta/PASWR/CD/data/Bodyfat.txt>.

```
> site <- "http://www1.appstate.edu/~arnholta/PASWR/CD/data/Bodyfat.txt"
> FAT <- read.table(file = site, header = TRUE, sep = "\t")
> head(FAT)      # Show first six rows

  age  fat sex
1 23  9.5   M
2 23 27.9   F
3 27  7.8   M
4 27 17.8   M
5 39 31.4   F
6 41 25.9   F
```

The website's address is stored in the R object `site`. This is done so that the `read.table()` command can appear on a single line. Inside the function `read.table()`, the only mandatory argument is the file location. Since `Bodyfat.txt` has column names and is a tab-delimited file, the arguments `header = TRUE` and `sep = "\t"` are used. There are numerous arguments for `read.table()`: the field separator character (`sep=`), which by default is set to a blank space; the character used for decimal points (`dec=`); the character vector used to represent missing values (`na.strings=`); and many others. To read about all of the arguments for `read.table()` and its variants such as `read.csv()`, which can be used to read comma-separated value (.csv) files, type `?read.table` at the R prompt. If the `Bodyfat.txt` file were stored in the current working directory of R, then one would only need to specify the file name in quotes of the `file=` argument. When files are not in the working directory, then the complete path to the desired file must be used.

Microsoft Excel spreadsheets are often stored as .csv files. For American readers, no confusion should result when viewing a .csv file, as the values are separated by commas just as the name implies. On the other hand, in certain countries, Excel will save the contents of a .csv spreadsheet using the semicolon (;) as the field separator and the comma (,) as the decimal point. Users in these locales may find the function `read.csv2()` useful, as its default arguments for field separation and decimal points are `sep = ";"` and `dec = ","`, respectively. To open the `Bodyfat.csv` file stored on the Internet, type

```
> site <- "http://www1.appstate.edu/~arnholta/PASWR/CD/data/Bodyfat.csv"
> FAT <- read.csv(file = site)
> head(FAT)  # Show first six rows
```

```

age  fat sex
1  23  9.5 M
2  23 27.9 F
3  27  7.8 M
4  27 17.8 M
5  39 31.4 F
6  41 25.9 F

```

Since the default argument for the header is `header = TRUE` in `read.csv()`, only the file was specified in the function.

1.10.2 Using `download.file()`

The function `download.file()` downloads files from the Internet and allows the user to specify where they want the downloaded file saved by providing a character string specifying the path, including the name of the downloaded file to be saved to the `destfile` argument. This function is especially useful when the file is large and/or the user's Internet connection is slow. Consider downloading the gross salaries for Baltimore city employees, which are available as one of the 110 publicly available data sets at <https://data.baltimorecity.gov>, and storing the results in a directory named `Data`. R Code 1.27 uses the `paste0()` function to concatenate two strings, each on a separate line, into a single string. The single string would extend beyond the allowable margins of the text; consequently, `paste0()` is used for cosmetic purposes.

R Code 1.27

```

> site <- paste0("http://data.baltimorecity.gov/api/",
+                  "views/7ymi-bvp3/rows.csv?accessType=DOWNLOAD")
> download.file(site, destfile = "./Data/Salaries.csv")
> list.files("./Data")                                # show files in ./Data
[1] "Salaries.csv"

> file.info("./Data/Salaries.csv")                   # file information
                                                 size isdir mode          mtime
./Data/Salaries.csv 1524254 FALSE  644 2015-06-02 09:12:27
                                         ctime          atime uid gid uname
./Data/Salaries.csv 2015-06-02 09:12:27 2015-06-02 08:09:21 501  20  alan
                                         grname
./Data/Salaries.csv  staff

> BES <- read.csv("./Data/Salaries.csv")
> head(BES, n = 2)                                  # show first two rows
                                                 name        JobTitle AgencyID
1 Aaron,Patricia G Facilities/Office Services II A03031
2   Aaron,Petra L ASSISTANT STATE'S ATTORNEY      A29005
                                         Agency   HireDate AnnualSalary GrossPay
1       OED-Employment Dev 10/24/1979      $51862.00 $52247.39
2 States Attorneys Office 09/25/2006      $64000.00 $59026.81

```

The function `file.info()` shows the size of the downloaded file as 1,524,254 bytes or 1.524254 megabytes and the date and time the file was last accessed under `atime`.

The function `download.file()` is not restricted to reading `.csv` files. Although the authors prefer to use human readable files instead of binary files, large files may need to be compressed and stored as `.zip` files. Consider using `download.file()` to download a zipped version of the Baltimore city employees salary data downloaded December 5, 2014, and stored at `http://bit.ly/12H9E01`, which is the shortened version of the original URL: `http://www1.appstate.edu/~arnholta/PASWR/CD/data/Salaries.csv.zip`.

```
> site <- "http://bit.ly/12H9E01" # URL
> download.file(site, destfile = "./DataZ/Salaries.csv.zip")
> file.info("./DataZ/Salaries.csv.zip")['size']

          size
./DataZ/Salaries.csv.zip 419937
```

Note that the zipped file is 419,937 bytes while the `.csv` file is 1,524,254 bytes. By using a compression algorithm, the zipped file is 3.6297 times the size of the `.csv` file.

1.10.3 Reading Data from Secure Websites

Secure websites start with `https`, in contrast to unsecure websites such as those used in Section 1.10.1, which begin with `http`. File-sharing services such as Dropbox and GitHub store their data on secure websites. One approach to reading a data file from a secure website is to use the function `source_data()` from the R package `repmis` (Gandrud, 2015). To read a data file into R from the *Public* folder on Dropbox, use `source_data()`; to read a data file into R from a non-*Public* folder on Dropbox, use the function `source_DropboxData()`. Dropbox has a *Public* folder on all accounts created prior to October 4, 2012. If one is using a Dropbox account created after October 4, 2012, see the website <https://www.dropbox.com/help/16/en> for directions on how to create a *Public* folder.

Consider the file `Verizon.csv` (Chihara and Hesterberg, 2011) stored in the *Public* folder of a Dropbox account, <https://db.tt/1rlTfYnk>, which is the shortened version of the original URL: <https://dl.dropboxusercontent.com/u/134274843/data/Verizon.csv>. R Code 1.28 reads the data into the data frame `Verizon1`. The same data file, `Verizon.csv`, is also stored in a non-*Public* folder on Dropbox <http://bit.ly/1mYM0jV>, which is the shortened version of the original URL: <https://www.dropbox.com/s/a9muo5wybukfs86/Verizon.csv>.

R Code 1.28

```
> library(repmis)
> URL <- "https://db.tt/1rlTfYnk"
> Verizon1 <- source_data(url = URL, sep = ",", header = TRUE)

Downloading data from: https://db.tt/1rlTfYnk
SHA-1 hash of the downloaded data file is:
6a26836a830af142c3562a6e4fe612eeb0281c30

> head(Verizon1) # show first six rows of data

  Time Group
1 17.50  ILEC
2  2.40  ILEC
3  0.00  ILEC
```

```
4 0.65 ILEC
5 22.23 ILEC
6 1.20 ILEC
```

Inside R Code 1.28, after the third line of typed code, one sees three lines of italicized text. The long string of numbers and letters on the third line of italicized text after *SHA-1 hash of the downloaded data file is:* is a unique identifier assigned by the function `source_data()` to the data file. The identifier allows the user to verify that the data file they downloaded is indeed the data file desired. That is, the unique identifier will change if the contents of the data file change. One can verify that the data downloaded in R Code 1.28 and the data downloaded in R Code 1.29 and R Code 1.30 are indeed identical. There is only one required argument for `source_data()`, `url=`. The user must provide a universal resource locator (URL, that is a website) to the argument `url=` for `source_data()` to work. Data files at unsecure websites as well as secure websites can be read into R with `source_data()`.

To download data from GitHub, one may also use `source_data()`. Caution needs to be exercised to make sure the data file one downloads is a “raw” text file. When one navigates to a directory on GitHub that contains data, the initial view may contain HTML embedded in the data file. To get a plain text file, click on the “raw” button in the upper right portion of the GitHub window. R Code 1.29 reads the file `Verizon.csv` from GitHub <http://bit.ly/1gqZCX3>, which is the shortened version of the original URL <https://raw.githubusercontent.com/alanarnholt/Data/master/Verizon.csv>, and stores the result in the data frame `Verizon2`.

R Code 1.29

```
> URL <- "http://bit.ly/1gqZCX3"
> Verizon2 <- source_data(url = URL)

Downloading data from: http://bit.ly/1gqZCX3
SHA-1 hash of the downloaded data file is:
6a26836a830af142c3562a6e4fe612eeb0281c30

> head(Verizon2) # show first six rows of data

  Time Group
1 17.50 ILEC
2  2.40 ILEC
3  0.00 ILEC
4 0.65 ILEC
5 22.23 ILEC
6 1.20 ILEC
```

Since the default arguments for `source_data()` are `sep = ","` and `header = TRUE`, which read in a `.csv` file with a header, neither argument was used as they were in R Code 1.28.

To read the `Verizon.csv` data file, which is also stored in a non-*Public* folder, one can use the function `source_DropboxData()` as illustrated in R Code 1.30. There are two required arguments for `source_DropboxData()`: the data file’s name and the data file’s Dropbox key. To find a file’s Dropbox key, right click on the data file’s name; then, click on **Share Dropbox Link** from the drop-down menu. The link to the selected file will be copied to the clipboard. The Dropbox link is a URL that can be pasted into a web browser. The URL referenced in R Code 1.30 is

<https://www.dropbox.com/s/a9muo5wybukfs86/Verizon.csv>

The last part of the URL, `Verizon.csv`, is the data file's name. The Dropbox key is the string of letters and numbers just after `https://www.dropbox.com/s/`, which in this case is `a9muo5wybukfs86`.

R Code 1.30

```
> Verizon3 <- source_DropboxData("Verizon.csv", "a9muo5wybukfs86")
> head(Verizon3)

  Time Group
1 17.50  ILEC
2  2.40  ILEC
3  0.00  ILEC
4  0.65  ILEC
5 22.23  ILEC
6  1.20  ILEC
```

1.10.4 Using `scan()`

The function `scan()` works well for entering a small amount of data by either typing in the console or by using a combination of copying and pasting procedures when the data can be highlighted and copied. To enter the ages for the subjects in the previously created `FAT` data frame whose values are also shown in Table 1.1 on page 58, one can proceed in two ways. One can enter all of the ages in one row, or one can enter one age per row. Regardless of how the data is entered, input is terminated with a blank line or an end of file signal (`Ctrl-Z` on Windows and `Ctrl-D` on a Mac). While it is possible to use `scan()` to read a file, the data sets in this text are generally tabular and `read.table()` and its variants are specifically designed to read in tabular data.

```
> age1 <- scan()
1: 23 23 27 27 39 41 45 49 50 53 53 54 56 57 58 58 60 61
19:
Read 18 items
> age1
[1] 23 23 27 27 39 41 45 49 50 53 53 54 56 57 58 58 60 61
> age2 <- scan()
1: 23
2: 23
3: 27
.
.
.
18: 61
19:
Read 18 items
> age2
[1] 23 23 27 27 39 41 45 49 50 53 53 54 56 57 58 58 60 61
```

1.10.5 Reading Excel (.xlsx) Files

The `xlsx` package, which must be installed, gives programmatic control of Excel files using R. The function `read.xlsx()` can be used to read an Excel workbook stored as a `.xlsx` file. When the workbook consists of multiple worksheets, the argument `sheetName=` is used to specify the desired worksheet. Consider an Excel workbook named `faculty.xlsx` stored in R's current working directory that has two worksheets, `Univ1` and `Univ2`. See Figures 1.15 and 1.16 on page 47 to view the contents of the Excel worksheets as they appear inside Excel. R Code 1.31 reads the worksheet named `Univ1` from the Excel file `faculty.xlsx`, stored in a directory named `Data` one level up from the working directory, into a data frame named `Faculty1`, which is then shown in the console. Next, the worksheet named `Univ2` from the same workbook is read into a data frame named `Faculty2` and subsequently displayed in the console.

R Code 1.31

```
> library(xlsx) # Loading xlsx package
> Faculty1 <- read.xlsx(file = "../Data/FACULTY.xlsx", sheetName = "Univ1")
> Faculty1

  Name Height      Rank
1  Joe     72 Assistant
2 Susie    63 Professor
3   Al     74 Associate
4  Rob     69 Professor
5 Juan    65 Professor

> Faculty2 <- read.xlsx(file = "../Data/FACULTY.xlsx", sheetName = "Univ2")
> Faculty2

  Name Height      Rank
1 Lola    62 Professor
2 Ana     61 Professor
3 Maria   65 Associate
4 Pilar   69 Assistant
5 Eva     65 Lecturer
```

It is worth reiterating the fact that one should always set R's working directory. Without the proper working directory, the previous code would not create the objects where the user wants them. Since R allows relative paths to be used in the `file=` argument, one has an additional incentive to establish a working directory so that code the user writes is portable across machines. A relative path is one that is defined in relation to the current or working directory. This means that a relative path on Windows will not start with a drive letter; and on Mac and Unix-like operating systems, a relative path will not start with a forward slash (/). To refer to `SomeFile` in the current directory, use a single dot (`./SomeFile`). Moving up the file system is done with two dots (...). To refer to `SomeFile` two levels above the working directory, type `../../SomeFile`.

The package `repmis` has the function `source_XlsxData()`, which can read Excel files stored at a URL (both http and https) into R. R Code 1.32 reads the file `FACULTY.xlsx` from GitHub <http://bit.ly/1iOWsGP>, which is the shortened version of the original URL <https://github.com/alanarnholt/Data/raw/master/FACULTY.xlsx>, and stores the results from sheet "Univ1" in the data frame `Faculty3` and sheet "Univ2" in the data frame `Faculty4`.

R Code 1.32

```
> URL <- "http://bit.ly/1iOWsGP"
> Faculty3 <- repmis::source_XlsxData(url = URL, sheet = "Univ1")

Downloading data from: http://bit.ly/1iOWsGP
SHA-1 hash of the downloaded data file is:
5beb512f1dbc421fd1bd315bc9b8b2be4bb00ec

> Faculty4 <- repmis::source_XlsxData(url = URL, sheet = "Univ2")

Downloading data from: http://bit.ly/1iOWsGP
SHA-1 hash of the downloaded data file is:
5beb512f1dbc421fd1bd315bc9b8b2be4bb00ec
```

```
> Faculty3
```

	Name	Height	Rank
1	Joe	72	Assistant
2	Susie	63	Professor
3	Al	74	Associate
4	Rob	69	Professor
5	Juan	65	Professor

```
> Faculty4
```

	Name	Height	Rank
1	Lola	62	Professor
2	Ana	61	Professor
3	Maria	65	Associate
4	Pilar	69	Assistant
5	Eva	65	Lecturer

	A	B	C	D
1	Name	Height	Rank	
2	Joe		72 Assistant	
3	Susie		63 Professor	
4	Al		74 Associate	
5	Rob		69 Professor	
6	Juan		65 Professor	

FIGURE 1.15: Excel workbook `faculty.xlsx` worksheet 1 contents

	A	B	C	D	E	F
1	Name	Height	Rank			
2	Lola		62	Professor		
3	Ana		61	Professor		
4	Maria		65	Associate		
5	Pilar		69	Assistant		
6	Eva		65	Lecturer		

FIGURE 1.16: Excel workbook `faculty.xlsx` worksheet 2 contents

1.10.6 Saving Data Frames to External Files

The function `write.table()` writes an R data frame to a file. Just as `read.table()` had variants `read.csv()` and `read.csv2()`, `write.table()` has variants `write.csv()` and `write.csv2()` to write to a `.csv` file. To write the `FAT` data frame, stored in the global environment, to the file `FAT.txt` in R's current working directory, type

```
> write.table(FAT, file = "FAT.txt")
```

The previous command, by default, stores the data frame `FAT` to the file `FAT.txt` using blank spaces as the separators. To store the file using tab separation, key in

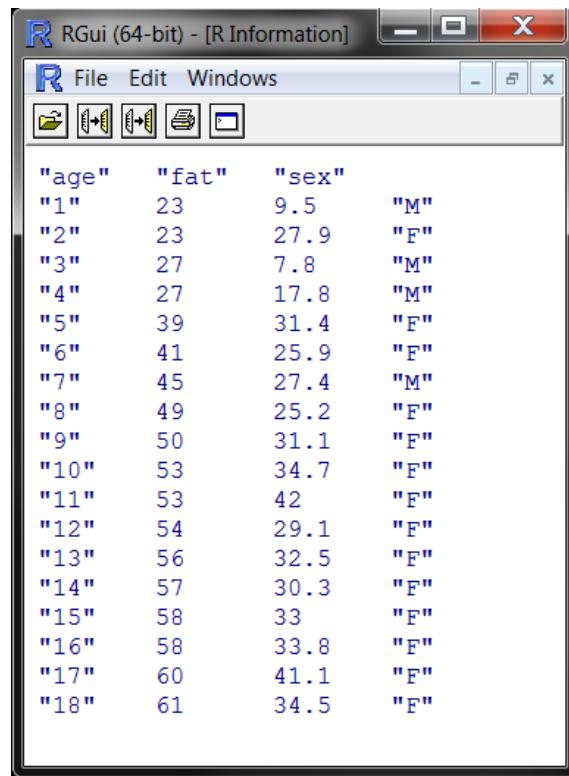
```
> write.table(FAT, file = "FAT.txt", sep = "\t")
```

To see the contents of the file `FAT.txt` in R, one can use the function `file.show()`. The result from using `file.show("FAT.txt")` on a tab delimited file can be seen in Figure 1.17 on the next page. To store a data frame to a file outside of the current working directory, the path either relative or absolute to the desired location must be given to the `file=` argument. The write analog to `read.xlsx()` is `write.xlsx()`. To write the `FAT` data frame stored in the global environment to the file `FAT.xlsx` in R's current working directory, type

```
> write.xlsx(FAT, file = "FAT.xlsx")
```

1.11 Working with Data

This section presents a data set that shows how different data types should be read into R as well as several functions that are useful for working with different types of R objects. Consider the data stored as a `.csv` file at



The screenshot shows the R GUI window titled "RGui (64-bit) - [R Information]". The menu bar includes File, Edit, and Windows. Below the menu is a toolbar with icons for file operations. The main area displays a data frame with three columns: "age", "fat", and "sex". The data consists of 18 rows, each containing a value for age, fat, and sex. The "age" column values range from 1 to 18. The "fat" column values range from 23 to 61. The "sex" column values are all "F".

"age"	"fat"	"sex"
"1"	23	9.5 "M"
"2"	23	27.9 "F"
"3"	27	7.8 "M"
"4"	27	17.8 "M"
"5"	39	31.4 "F"
"6"	41	25.9 "F"
"7"	45	27.4 "M"
"8"	49	25.2 "F"
"9"	50	31.1 "F"
"10"	53	34.7 "F"
"11"	53	42 "F"
"12"	54	29.1 "F"
"13"	56	32.5 "F"
"14"	57	30.3 "F"
"15"	58	33 "F"
"16"	58	33.8 "F"
"17"	60	41.1 "F"
"18"	61	34.5 "F"

FIGURE 1.17: Results of `file.show("FAT.txt")`

<http://www1.appstate.edu/~arnholta/PASWR/CD/data/Poplar3.CSV>.

The following description of the data is from Minitab 15:

In an effort to maximize yield, researchers designed an experiment to determine how two factors, Site and Treatment, influence the Weight of four-year-old poplar clones. They planted trees on two sites: Site 1 is a moist site with rich soil, and Site 2 is a dry, sandy site. They applied four different treatments to the trees: Treatment 1 was the control (no treatment); Treatment 2 used fertilizer; Treatment 3 used irrigation; and Treatment 4 used both fertilizer and irrigation. To account for a variety of weather conditions, the researchers replicated the data by planting half the trees in Year 1, and the other half in Year 2.

The data from Poplar3.CSV is read into the data frame `poplar` using the `read.csv()` function, and the first five rows of the data frame are shown using the function `head()` with the argument `n = 5` to show the first five rows of the data frame instead of the default `n = 6` rows in R Code 1.33.

R Code 1.33

```
> site <- "http://www1.appstate.edu/~arnholta/PASWR/CD/data/Poplar3.CSV"
> poplar <- read.csv(file = url(site))
> head(poplar, n = 3) # show first three rows
```

Site	Year	Treatment	Diameter	Height	Weight	Age
------	------	-----------	----------	--------	--------	-----

1	1	1	1	2.23	3.76	0.17	3
2	1	1	1	2.12	3.15	0.15	3
3	1	1	1	1.06	1.85	0.02	3

When dealing with imported data sets, it is always good to examine their contents using functions such as `str()` and `summary()`, which show the structure and provide appropriate summaries, respectively, for different types of objects.

```
> str(poplar)

'data.frame': 298 obs. of 7 variables:
 $ Site      : int 1 1 1 1 1 1 1 1 1 2 ...
 $ Year       : int 1 1 1 1 1 1 1 1 1 1 ...
 $ Treatment  : int 1 1 1 1 1 1 1 1 1 1 ...
 $ Diameter   : num 2.23 2.12 1.06 2.12 2.99 4.01 2.41 2.75 2.2 4.09 ...
 $ Height     : num 3.76 3.15 1.85 3.64 4.64 5.25 4.07 4.72 4.17 5.73 ...
 $ Weight     : num 0.17 0.15 0.02 0.16 0.37 0.73 0.22 0.3 0.19 0.78 ...
 $ Age        : int 3 3 3 3 3 3 3 3 3 3 ...

> summary(poplar)

  Site          Year          Treatment        Diameter      
Min.   :1.00    Min.   :1.00    Min.   :1.000   Min.   :-99.000  
1st Qu.:1.00   1st Qu.:1.00   1st Qu.:2.000   1st Qu.: 3.605  
Median :2.00    Median :2.00    Median :2.500    Median : 5.175  
Mean   :1.51    Mean   :1.51    Mean   :2.503    Mean   : 3.862  
3rd Qu.:2.00   3rd Qu.:2.00   3rd Qu.:3.750   3rd Qu.: 6.230  
Max.   :2.00    Max.   :2.00    Max.   :4.000    Max.   : 8.260  
                                             
  Height         Weight         Age          
Min.   :-99.000  Min.   :-99.000  Min.   :3.000  
1st Qu.: 5.495  1st Qu.: 0.605  1st Qu.:3.000  
Median : 6.910  Median : 1.640  Median :4.000  
Mean   : 5.902  Mean   : 1.099  Mean   :3.507  
3rd Qu.: 8.750  3rd Qu.: 3.435  3rd Qu.:4.000  
Max.   :10.900   Max.   : 6.930  Max.   :4.000
```

From typing `str(poplar)` at the R prompt, one can see that all seven variables are either integer or numeric. From the description, the variables `Site` and `Treatment` are factors. Further investigation into the experiment reveals that `year` and `Age` are factors as well. Recall that factors are an extension of vectors designed for storing categorical information. The results of `summary(poplar)` indicate the minimum values for `Diameter`, `Height`, and `Weight` are all `-99`, which does not make sense unless one is told that a value of `-99` for these variables represents a missing value. Once one understands that the variables `Site`, `Year`, `Treatment`, and `Age` are factors and that the value `-99` has been used to represent missing values for the variables `Diameter`, `Height`, and `Weight`, appropriate arguments to `read.csv()` can be entered. The data is now read into the object `poplarC` using `na.strings = "-99"` to store the `NA` values correctly. The argument `colClasses=` requires a vector that indicates the desired class of each column.

```
> poplarC <- read.csv(file = url(site), na.strings = "-99",
+                      colClasses = c(rep("factor", 3), rep("numeric", 3), "factor"))
> str(poplarC)
```

```
'data.frame': 298 obs. of 7 variables:
 $ Site      : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 2 ...
 $ Year       : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ Treatment: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
 $ Diameter   : num  2.23 2.12 1.06 2.12 2.99 4.01 2.41 2.75 2.2 4.09 ...
 $ Height     : num  3.76 3.15 1.85 3.64 4.64 5.25 4.07 4.72 4.17 5.73 ...
 $ Weight     : num  0.17 0.15 0.02 0.16 0.37 0.73 0.22 0.3 0.19 0.78 ...
 $ Age        : Factor w/ 2 levels "3","4": 1 1 1 1 1 1 1 1 1 1 ...
```

In the event different values (999, 99, 9999) for different variables (`var1`, `var2`, `var3`) are used to represent missing values in a data set, the argument `na.strings=` will no longer be able to solve the problem directly. Although one can pass a vector of the form `na.strings = c(999, 99, 9999)`, this will simply replace all values that are 999, 99, or 9999 with NAs. If the first variable has a legitimate value of 99, then it too would be replaced with an NA value. One solution for this problem in general is to read the data set into a data frame (`DF`), to assign the data frame to a different name so that the cleaned up data set is not confused with the original data, and to use filtering to assign NAs to values of `var1`, `var2`, and `var3` that have entries of 999, 99, and 999, respectively.

```
> DF <- read.table(file = url(site), header = TRUE)
> df <- DF
> df[df$var1 == 999, "var1"] = NA
> df[df$var2 == 99, "var2"] = NA
> df[df$var3 == 9999, "var3"] = NA
```

Once a variable has its class changed from `int` to `factor`, labeling the levels of the factor can be accomplished without difficulties. To facilitate analysis of the `poplarC` data, labels for the levels of the variables `Site` and `Treatment` are assigned.

```
> levels(poplarC$Site) <- c("Moist", "Dry")
> TreatmentLevels <- c("Control", "Fertilizer", "Irrigation", "FertIrriga")
> levels(poplarC$Treatment) <- TreatmentLevels
> str(poplarC$Treatment)

Factor w/ 4 levels "Control","Fertilizer",...: 1 1 1 1 1 1 1 1 1 1 ...
```

Another way to accomplish the previous labeling that makes clear the assignment of labels to levels is given in R Code 1.34. The reader should make sure that the variable being labeled is a factor before using either the `labels=` or `levels=` argument to assign labels to levels.

R Code 1.34

```
> poplarC$Site <- factor(poplarC$Site, labels = c("Moist", "Dry"))
> str(poplarC$Site)

Factor w/ 2 levels "Moist","Dry": 1 1 1 1 1 1 1 1 1 2 ...
```

If the argument `levels = c("Moist", "Dry")` is applied to a non-factor variable (as `Site` is in the original `poplar` data frame), the levels of `Site` are converted to NA values as seen in R Code 1.35 on the facing page.

R Code 1.35

```
> poplar$Site <- factor(poplar$Site, levels = c("Moist", "Dry"))
> str(poplar$Site)

Factor w/ 2 levels "Moist","Dry": NA ...
```

The previous examples illustrate assigning labels to levels of a factor. Since the default ordering of the levels of a character factor are alphabetical, one may encounter factors whose levels need to be manipulated. Consider the data frame **EPIDURALF** from the **PASWR2** package that has three levels (**Difficult**, **Easy**, and **Impossible**) for the factor **Ease**. To switch the positions of **Difficult** and **Easy** in the factor level, consider R Code 1.36.

R Code 1.36

```
> library(PASWR2)
> levels(EPIDURALF$ease) # Default levels (alphabetical)

[1] "Difficult"   "Easy"          "Impossible"

> EPIDURALF$ease <- factor(EPIDURALF$ease, levels = c("Easy",
+ "Difficult", "Impossible"))
> levels(EPIDURALF$ease) # Correct levels

[1] "Easy"         "Difficult"     "Impossible"

> rm(EPIDURALF)
```

1.11.1 Dealing with NA Values

When working with real data, values are often unavailable because the experiment failed, the subject did not show up, the value was lost, etc. R uses **NA** to denote a missing value or to denote the result of an operation performed on values that contain **NA** values. When performing a computation on an object with **NA** values, one needs to decide what to do with the **NA** values. Many functions (**mean()**, **var()**, **sd()**, **median()**, etc.) have the argument **na.rm=**, which can be set to **TRUE** to ignore the **NA** values and apply the function to the object with the **NA** values removed. Many modeling functions that accept a formula (**t.test()**, **lm()**, **glm()**, etc.) have the argument **na.action=**, which when passed the value **na.omit** will remove any row of the data frame specified in the **data=** argument that has an **NA** value. One should always ask, “Why are the values **NA**?” In certain scenarios, it may be appropriate to impute values for the **NAs**. A simple imputation example is provided in Section 1.18 on page 78. Other times, the user may want to clean up the data so that the **NA** values are removed. By applying the **summary()** function to **poplarC**, one can see that **Diameter**, **Height**, and **Weight** each have three missing values.

```
> summary(poplarC[, 4:7]) # summary of columns 4-7
```

Diameter	Height	Weight	Age
Min. :1.030	Min. : 1.150	Min. :0.010	3:147
1st Qu.:3.675	1st Qu.: 5.530	1st Qu.:0.635	4:151
Median :5.200	Median : 6.950	Median :1.680	
Mean :4.909	Mean : 6.969	Mean :2.117	
3rd Qu.:6.235	3rd Qu.: 8.785	3rd Qu.:3.470	

```
Max.    :8.260   Max.    :10.900  Max.    :6.930
NA's     :3       NA's     :3       NA's     :3
```

Two approaches are presented to remove the missing values for `Diameter`, `Height`, and `Weight`, which, by coincidence, come from the same subjects, which are trees. The first approach (R Code 1.37) uses the function `na.omit()`, while the second approach (R Code 1.38) uses the function `complete.cases()`.

R Code 1.37

```
> dim(poplarC)
[1] 298    7

> myNoMissing <- na.omit(poplarC)
> summary(myNoMissing[, 4:7]) # summary of columns 4-7

      Diameter        Height         Weight        Age
Min.    :1.030    Min.   : 1.150    Min.   :0.010    3:147
1st Qu.:3.675   1st Qu.: 5.530    1st Qu.:0.635    4:148
Median  :5.200   Median  : 6.950    Median  :1.680
Mean    :4.909   Mean    : 6.969    Mean    :2.117
3rd Qu.:6.235   3rd Qu.: 8.785    3rd Qu.:3.470
Max.    :8.260   Max.    :10.900   Max.    :6.930

> dim(myNoMissing)
[1] 295    7
```

The function `na.omit()` removed rows 179, 210, and 218 of the `poplarC` data frame, which had NA values in those rows for `Diameter`, `Height`, and `Weight`. The resulting data frame `myNoMissing` maintains the row numbers of the original data frame `poplar` with rows 179, 210, and 218 omitted, which is why the dimension of the `myNoMissing` data frame is 295 by 7, yet the last row is named 298. Compare the original data in `poplarC`, which has rows with NA values, to `myNoMissing`, which has the NA values removed and row labels maintained for all other rows.

R Code 1.38

```
> poplarC[c(178:180, 209:211, 217:219), ]

  Site Year Treatment Diameter Height Weight Age
178  Dry     1 FertIrriga    7.57   9.37   5.21   4
179  Dry     1 FertIrriga      NA     NA     NA     4
180  Dry     1 FertIrriga    7.68   9.09   5.12   4
209  Moist    1 FertIrriga    7.28   9.17   4.28   4
210  Moist    1 FertIrriga      NA     NA     NA     4
211  Moist    1 FertIrriga    5.33   8.42   2.36   4
217  Moist    1 FertIrriga    6.58   8.84   3.83   4
218  Moist    1 FertIrriga      NA     NA     NA     4
219  Moist    2 Control       7.71  10.30   5.82   4

> myNoMissing[c(178:179, 208:209, 215:216), ]
```

	Site	Year	Treatment	Diameter	Height	Weight	Age
178	Dry	1	FertIrriga	7.57	9.37	5.21	4
180	Dry	1	FertIrriga	7.68	9.09	5.12	4
209	Moist	1	FertIrriga	7.28	9.17	4.28	4
211	Moist	1	FertIrriga	5.33	8.42	2.36	4
217	Moist	1	FertIrriga	6.58	8.84	3.83	4
219	Moist	2	Control	7.71	10.30	5.82	4

The function `complete.cases()` can be applied to a vector, matrix, or data frame and returns a logical vector indicating which cases are complete. In R Code 1.39, the logical vector `complete` is used to extract the rows of `poplarC` that have no missing values.

R Code 1.39

```
> complete <- complete.cases(poplarC)
> myCompleteCases <- poplarC[complete, ]
> dim(myCompleteCases)

[1] 295    7

> summary(myCompleteCases[, 4:7]) # summary of columns 4-7

      Diameter          Height          Weight          Age      
Min.   :1.030   Min.   : 1.150   Min.   :0.010   3:147  
1st Qu.:3.675  1st Qu.: 5.530   1st Qu.:0.635   4:148  
Median :5.200  Median : 6.950   Median :1.680  
Mean   :4.909  Mean   : 6.969   Mean   :2.117  
3rd Qu.:6.235 3rd Qu.: 8.785   3rd Qu.:3.470  
Max.   :8.260  Max.   :10.900   Max.   :6.930  

> myCompleteCases[c(178:179, 208:209, 215:216), ]

      Site Year Treatment Diameter Height Weight Age
178   Dry   1 FertIrriga    7.57  9.37  5.21   4
180   Dry   1 FertIrriga    7.68  9.09  5.12   4
209   Moist  1 FertIrriga    7.28  9.17  4.28   4
211   Moist  1 FertIrriga    5.33  8.42  2.36   4
217   Moist  1 FertIrriga    6.58  8.84  3.83   4
219   Moist  2 Control       7.71 10.30  5.82   4
```

A useful function for testing for the presence of NA values in vectors is the function `is.na(x)`. The function returns a logical vector of the same size as `x` that takes on the value TRUE if and only if the corresponding element in `x` is NA. If `x` is a vector with NA values, but only the non-missing values are of interest, the function `!is.na(x)` can be used to extract them as follows.

```
> x <- c(1, 6, 9, 2, NA)
> is.na(x)

[1] FALSE FALSE FALSE FALSE  TRUE

> !is.na(x)

[1]  TRUE  TRUE  TRUE  TRUE FALSE
```

```
> x[!is.na(x)]
[1] 1 6 9 2
```

1.11.2 Creating New Variables in a Data Frame

Two different approaches for adding new variables to a data frame are the `cbind()` and the `within()` functions. The function `cbind()` takes vectors, matrices, data frames, or any combination of vectors, matrices, and data frames as arguments and combines them by columns. The function `rbind()` works in an analogous manner by combining its arguments by rows. Body mass index (BMI) is defined as weight (in kilograms) divided by height (in meters) squared. R Code 1.40 creates a new variable, BMI, using the information in the `EPIDURALF` data frame. Once the variable is created, it is bound to the `EPIDURALF` data frame using the `cbind()` function and stored in a new data frame named `EPIbmi2`.

R Code 1.40

```
> attach(EPIDURALF)
> BMI = kg/(cm/100)^2 # Creating new variable
> detach(EPIDURALF)
> EPIbmi2 <- cbind(EPIDURALF, BMI) # Column binding BMI to df
> rm(BMI) # removing BMI from .GlobalEnv
> EPIbmi2[1:3, -5] # Show first 3 rows of EPIbmi2 w/o treatment

  doctor  kg   cm      ease  oc complications      BMI
1      B 116 172 Difficult  0        None 39.21038
2      C  86 176      Easy  0        None 27.76343
3      B  72 157 Difficult  0        None 29.21011
```

The levels of the variable `Ease` of the `EPIbmi2` data frame are not in ascending order of difficulty, and are subsequently fixed using the `levels()` function.

```
> levels(EPIbmi2$ease)
[1] "Difficult"    "Easy"          "Impossible"

> EPIbmi2$ease <- factor(EPIbmi2$ease, levels = c("Easy",
+ "Difficult", "Impossible"))
> levels(EPIbmi2$ease)

[1] "Easy"          "Difficult"     "Impossible"
```

It should be noted that it is possible, and generally preferable, to create the variable `BMI` directly without using `attach()` by entering

```
> EPIDURALF$BMI <- EPIDURALF$kg/(EPIDURALF$cm/100)^2
> rm(EPIDURALF)
```

Next, the function `within()` is used both to create a new variable `BMI` and to fix the levels of the variable `ease`. The function `within()` can be used to create or modify existing data, including creating new variables with R expressions composed of current objects in the data frame specified by the `data=` argument. Note that the expressions of the `expr=` argument are enclosed in curly braces {} with one expression per line.

R Code 1.41

```
> levels(EPIDURALF$ease)
[1] "Difficult"   "Easy"        "Impossible"

> EPIbmi <- within(data = EPIDURALF, expr = {
+   BMI = kg/(cm/100)^2
+   ease = factor(ease, levels = c("Easy", "Difficult", "Impossible"))
+ })
> EPIbmi[1:6, -5]  # Show first 6 rows of EPIbmi w/o treatment

  doctor  kg  cm      ease  oc complications      BMI
1      B 116 172 Difficult  0      None 39.21038
2      C  86 176      Easy  0      None 27.76343
3      B  72 157 Difficult  0      None 29.21011
4      B  63 169      Easy  2      None 22.05805
5      B 114 163 Impossible 0      None 42.90715
6      B 121 163 Difficult  3      None 45.54180

> levels(EPIbmi$ease)
[1] "Easy"        "Difficult"    "Impossible"
```

1.11.3 Sorting a Data Frame by One or More of Its Columns

The `sort()` function can be used to sort a single variable in either increasing or decreasing order. Unfortunately, if the user wants to sort a variable in a data frame and have the other variables reflect the new ordering, `sort()` will not work. The function needed to rearrange the values in a data frame to reflect the order of a particular variable or variables in the event of ties is `order()`. Given three variables `x`, `y`, and `z` in a data frame `DF`, the command `order(x)` returns the indices of the sorted values of `x`. Consequently, the data frame `DF` can be sorted by `x` with the command `DF[order(x),]`. In the event of ties, further arguments to `order` can be used to specify how the ties should be broken. Consider how ties are broken with the following numbers in R Code 1.42. To conserve space, the transpose function `t()` is used on the data frame `DF`.

R Code 1.42

```
> x <- c(1, 1, 1, 3, 3, 3, 2, 2, 2)
> y <- c(3, 2, 3, 6, 2, 6, 10, 4, 4)
> z <- c(7, 4, 2, 9, 6, 4, 5, 3, 1)
> DF <- data.frame(x, y, z)
> rm(x, y, z)  # remove x, y, and z from workspace
> t(DF)  # transpose DF

 [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
x     1     1     1     3     3     3     2     2     2
y     3     2     3     6     2     6    10     4     4
z     7     4     2     9     6     4     5     3     1

> with(data = DF, t(DF[order(x, y, z), ]))
```

```

2 3 1 9 8 7 5 6 4
x 1 1 1 2 2 2 3 3 3
y 2 3 3 4 4 10 2 6 6
z 4 2 7 1 3 5 6 4 9

```

Note that **x** is ordered first as 1, 2, 3. Then, where **x** values are tied, **y** values determine the next ordering. Under the 1 for **x**, **y**'s values appear in 2, 3 order. Under the 2 for **x**, **y**'s values appear in 4, 10 order. Under the 3 for **x**, **y**'s values appear in 2, 6 order. Where **y**'s values are tied, the value for **z** determines the final ordering. Thus, the (**x**, **y**, **z**) triple (1, 3, 2) precedes (1, 3, 7); (2, 4, 1) is before (2, 4, 3); and (3, 6, 4) comes before (3, 6, 9).

Example 1.1 Consider the first 6 rows of the data frame **EPIbmi** (constructed in R Code 1.41 on the preceding page) for the variables **oc**, **ease**, and **BMI**. Sort the data frame **subEPI** first by **oc**, then by **ease**, and finally by **BMI**.

```

> subEPI <- EPIbmi[1:6, c("oc", "ease", "BMI")]
> subEPI

  oc      ease      BMI
1 0  Difficult 39.21038
2 0      Easy 27.76343
3 0  Difficult 29.21011
4 2      Easy 22.05805
5 0 Impossible 42.90715
6 3  Difficult 45.54180

```

Solution: To sort the data frame **subEPI** first by **oc**, then by **ease**, and finally by **BMI**, use the function **order()**. The order for the five subjects that have an **oc** of 0 is first arranged by **ease**, then by **BMI**. Consequently, the order for the two subjects with **ocs** of 0, and level of **ease** of **Easy**, is determined by the **BMI** value.

```

> my0 <- order(subEPI$oc, subEPI$ease, subEPI$BMI)
> my0

[1] 2 3 1 5 4 6

> subEPI[my0, ]

  oc      ease      BMI
2 0      Easy 27.76343
3 0  Difficult 29.21011
1 0  Difficult 39.21038
5 0 Impossible 42.90715
4 2      Easy 22.05805
6 3  Difficult 45.54180

```



1.11.4 Merging Data Frames

Related information may be stored in two or more different locations. For example, in a double blind experiment into the efficacy of a new drug claiming to boost the high-density lipoprotein (HDL) of patients, the physician may maintain one set of information, and the

supervising scientist might maintain a separate list indicating who receives the drug and who receives the placebo. In this type of experiment, neither the patient nor the physician knows what type of treatment (drug or placebo) the patient receives; however, a “secret list” is generally maintained by the scientist conducting the experiment indicating who received the drug and who received the placebo. In order to analyze the results, the two data sets will need to be joined based on some common variable. The R function `merge()` is one way to combine data frames from multiple locations. The `merge()` function has named arguments `by.x=` and `by.y=` for situations where the same information is stored in two different data frames under different names. When `merge()` is applied to two data frames without any additional arguments, `merge()` assumes the two data frames have one or more columns with names in common, merges the two data frames, and eliminates any duplicate columns. The default behavior for `merge()` can be changed using the arguments `all=`, `all.x=`, and `all.y=`. Specifying `all = TRUE` will include all rows, `all.x = TRUE` will include all rows from the first data frame, and `all.y = TRUE` will include all rows from the second data frame.

Example 1.2 Consider a fictitious example where a single physician participates in an experiment where she is provided a list indicating she should administer “Treatment One” to patients 1, 5, and 6, and “Treatment Two” to patients 2, 3, and 4. The physician maintains one data base (`DFphy`) with the patient’s ID, Gender, and HDL value after finishing some prescribed treatment protocol. The supervising scientist maintains a second data base (`DFfsci`) with information on who received the placebo and who received the drug (`secretID`) as well as patient ID. Use the function `merge()` to combine the two data frames.

Solution: Information is first stored in the data frames `DFphy` and `DFfsci`, then the two data frames are combined with `merge()`.

```
> DFphy <- data.frame(ID = 1:6, Gender = rep(c("Female", "Male"),
+                           each = 3), HDL = c(39, 42, 22, 27, 29, 45))
> DFphy
   ID Gender HDL
1  1 Female  39
2  2 Female  42
3  3 Female  22
4  4   Male  27
5  5   Male  29
6  6   Male  45

> DFfsci <- data.frame(ID = c(2, 4, 3, 5, 1, 6),
+                         secretID = rep(c("Drug", "Placebo"), each = 3))
> DFfsci
   ID secretID
1  2      Drug
2  4      Drug
3  3      Drug
4  5 Placebo
5  1 Placebo
6  6 Placebo

> merge(DFphy, DFfsci)
```

```
ID Gender HDL secretID
1 1 Female 39 Placebo
2 2 Female 42 Drug
3 3 Female 22 Drug
4 4 Male 27 Drug
5 5 Male 29 Placebo
6 6 Male 45 Placebo
```

Note that only one column for ID appears in the merged result and that the order of the secretID has been rearranged to match the variable ID.

1.12 Using Logical Operators with Data Frames

Logical operators were first introduced with vectors. Since data frames are collections of equal length vectors having possibly different modes, all of R's logical operators discussed in the context of vectors are still applicable. The data in Table 1.1 that are stored in the data frame **BODYFAT** come from a study reported in the *American Journal of Clinical Nutrition* (Mazess et al., 1984) that investigated a new method for measuring body composition. Suppose one is interested in which subjects have fat percentages less than 25%. Three common approaches that all achieve the same result are \$ prefixing, the `with()` function, and the `attach()` function in combination with a logical statement. The three approaches to answer which subjects have fat percentages less than 25% are illustrated in R Code 1.43.

Table 1.1: Body composition (**BODYFAT**)

n	age	% fat	sex	n	age	% fat	sex
1	23	9.5	M	10	53	34.7	F
2	23	27.9	F	11	53	42.0	F
3	27	7.8	M	12	54	29.1	F
4	27	17.8	M	13	56	32.5	F
5	39	31.4	F	14	57	30.3	F
6	41	25.9	F	15	58	33.0	F
7	45	27.4	M	16	58	33.8	F
8	49	25.2	F	17	60	41.1	F
9	50	31.1	F	18	61	34.5	F

R Code 1.43

```
> head(BODYFAT, n = 3) # show first 3 rows of BODYFAT
  age  fat sex
1 23  9.5 M
2 23 27.9 F
3 27  7.8 M
```

```
> BODYFAT$fat < 25

[1] TRUE FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE

> with(data = BODYFAT, fat < 25)

[1] TRUE FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE

> attach(BODYFAT)
> fat < 25

[1] TRUE FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE

> detach(BODYFAT)
```

To see the fat percentages for subjects with less than 25% fat, use one of the approaches in R Code 1.44, all of which return the same result. The first three approaches all work with the `fat` vector. The last two approaches extract all rows in the data frame `BODYFAT` where `fat < 25`, for the variable `fat`.

R Code 1.44

```
> BODYFAT$fat[BODYFAT$fat < 25]

[1] 9.5 7.8 17.8

> with(data = BODYFAT, fat[fat < 25])

[1] 9.5 7.8 17.8

> attach(BODYFAT)
> fat[fat < 25]

[1] 9.5 7.8 17.8

> detach(BODYFAT)
> BODYFAT[BODYFAT$fat < 25, "fat"]

[1] 9.5 7.8 17.8

> BODYFAT[BODYFAT$fat < 25, 2]

[1] 9.5 7.8 17.8
```

In R Code 1.44, only the values for the second column of the data frame `BODYFAT`, "fat", were returned by using `BODYFAT[BODYFAT$fat < 25, 2]`. To return a subset of the variables, say `fat` and `sex` in the data frame, one should pass appropriate values to a vector for the columns one wants to extract as illustrated in R Code 1.45.

R Code 1.45

```
> BODYFAT[BODYFAT$fat < 25, c(2, 3)] # fat < 25 for columns 2 and 3
```

```

fat sex
1 9.5   M
3 7.8   M
4 17.8  M

> BODYFAT[BODYFAT$fat < 25, c("fat", "sex")] # using names of columns

fat sex
1 9.5   M
3 7.8   M
4 17.8  M

```

Some may prefer to use the `subset()` function, a convenience function for subsetting. Everything one can do with `subset()` can be done with bracket notation; however, some may find `subset()`'s code more readable. To extract the rows of the column `fat` where `fat < 25` as a vector, use `subset()` as follows.

```

> subset(BODYFAT, select = fat, subset = fat < 25, drop = TRUE)

[1] 9.5 7.8 17.8

```

To extract the rows of the columns `fat` and `sex` where `fat < 25` into a data frame, use `subset()` as seen next.

```

> subset(x = BODYFAT, select = c(fat, sex), subset = fat < 25)

fat sex
1 9.5   M
3 7.8   M
4 17.8  M

```

Instead of returning only the values of the variable `fat` that are less than 25%, one might want to see all the values of the other variables where `fat < 25`. Consider the following solutions, which use three different ways to access the information in the data frame. Note that the second argument (`columns`) inside the square brackets after the comma is missing, which causes all variables in the data frame to be returned.

```

> BODYFAT[BODYFAT$fat < 25, ] # fat < 25 all columns

age  fat sex
1 23  9.5   M
3 27  7.8   M
4 27  17.8  M

> with(data = BODYFAT, BODYFAT[fat < 25, ]) # fat < 25 all columns

age  fat sex
1 23  9.5   M
3 27  7.8   M
4 27  17.8  M

> attach(BODYFAT)
> BODYFAT[fat < 25, ] # fat < 25 all columns

```

```

age  fat sex
1  23  9.5  M
3  27  7.8  M
4  27 17.8  M

```

```
> detach(BODYFAT)
```

The convenience function `subset()` returns the same result as shown next.

```

> subset(x = BODYFAT, subset = fat < 25)

  age  fat sex
1  23  9.5  M
3  27  7.8  M
4  27 17.8  M

```

It is also possible to extract values satisfying more complicated logical conditions. For example, to extract all fat percentages that are less than 25% and different from 7.8, one could enter

```

> with(BODYFAT, fat[fat < 25 & fat != 7.8])
[1]  9.5 17.8

```

To see all the values for the variables in the data frame where fat is less than 25% and different from 7.8, one could enter

```

> with(BODYFAT, BODYFAT[fat < 25 & fat != 7.8, ])
  age  fat sex
1  23  9.5  M
4  27 17.8  M

```

Note that there is only one vector of values for indexing a vector (`fat`) while there are two vectors for indexing a data frame (`BODYFAT`). Another solution is to use the function `subset()`.

```

> subset(x = BODYFAT, subset = fat < 25 & fat != 7.8)

  age  fat sex
1  23  9.5  M
4  27 17.8  M

```

Three additional functions that work with logical objects are `any()`, `all()`, and `which()`. The function `any()` evaluates whether at least one value from a logical statement is true. The function `all()` evaluates whether all values from a logical statement are true. The function `which()` returns the indices for values in which the logical statements are true. R Code 1.46 illustrates the use of `any()`, `all()`, and `which()` on the data frame `BODYFAT`.

R Code 1.46

```

> any(BODYFAT$fat < 10 & BODYFAT$sex == "M")      # Condition TRUE for any?
[1] TRUE

> all(BODYFAT$fat < 10 & BODYFAT$sex == "M")      # Condition TRUE for all?

```

```
[1] FALSE
> which(BODYFAT$fat < 10 & BODYFAT$sex == "M") # Indices for which TRUE
[1] 1 3
```

1.13 Tables

Tables show counts of categorical variables (factors in R). Contingency tables show counts for the intersection of two or more factors. Two functions are examined to create tables: `table()` and `xtabs()`. To create a table or contingency table of the counts for each level of a factor or at each combination of factor levels, supply the factor/factors to the function `table()`. Consider the following code to create a table of the factor `Ease` from the `EPIbmi2` data frame created in Section 1.11.2 on page 54.

```
> table(EPIbmi2$ease)
```

	Easy	Difficult	Impossible
	207	114	21

To create a contingency table using the variables `ease` and `doctor`, type:

```
> table(EPIbmi2$ease, EPIbmi2$doctor)
```

	A	B	C	D
Easy	39	49	72	47
Difficult	20	58	21	15
Impossible	2	8	0	11

Recall that the variables in a data frame can be accessed using `$` prefixing, the `with()` function, and the `attach()` function. The following illustrates the second and third approaches.

```
> with(data = EPIbmi2, table(ease, doctor))

            doctor
ease      A   B   C   D
  Easy     39  49  72  47
  Difficult 20  58  21  15
  Impossible 2   8   0  11

> attach(EPIbmi2)
> table(ease, doctor)

            doctor
ease      A   B   C   D
  Easy     39  49  72  47
  Difficult 20  58  21  15
  Impossible 2   8   0  11

> detach(EPIbmi2)
```

When working with factors stored in a data frame, the function `xtabs()` can be used to create a contingency table from cross-classifying factors using a formula interface. The `formula=` argument for `xtabs()` takes the form `~factor1 + factor2`, while a data frame is specified with the `data=` argument.

```
> xtabs(formula = ~ease + doctor, data = EPIbmi2)

      doctor
ease      A  B  C  D
  Easy     39 49 72 47
  Difficult 20 58 21 15
  Impossible 2  8  0 11
```

Consider the output created from `table()` and `xtabs()` when applied to three factors.

```
> table(EPIbmi2$ease, EPIbmi2$doctor, EPIbmi2$treatment)

, ,  = Hamstring Stretch

      A  B  C  D
  Easy     19 27 32 22
  Difficult 13 31 11  8
  Impossible 1  2  0  5

, ,  = Traditional Sitting

      A  B  C  D
  Easy     20 22 40 25
  Difficult 7 27 10  7
  Impossible 1  6  0  6

> xtabs(~ease + doctor + treatment, data = EPIbmi2)

, , treatment = Hamstring Stretch

      doctor
ease      A  B  C  D
  Easy     19 27 32 22
  Difficult 13 31 11  8
  Impossible 1  2  0  5

, , treatment = Traditional Sitting

      doctor
ease      A  B  C  D
  Easy     20 22 40 25
  Difficult 7 27 10  7
  Impossible 1  6  0  6
```

When using three-way contingency tables, `ftable()` provides more compact output than `table()` or `xtabs()`. The arguments to `ftable()` may be entered as factors in a fashion

similar to that used with `table()` or as a formula in a fashion similar to that used with `xtabs()`.

```
> ftable(EPIbmi2$treatment, EPIbmi2$ease, EPIbmi2$doctor)

          A   B   C   D

Hamstring Stretch   Easy      19 27 32 22
                  Difficult  13 31 11  8
                  Impossible 1   2   0   5
Traditional Sitting Easy      20 22 40 25
                  Difficult  7   27 10  7
                  Impossible 1   6   0   6

> # or
> ftable(doctor ~ treatment + ease, data = EPIbmi2)

                                doctor   A   B   C   D
treatment      ease
Hamstring Stretch   Easy      19 27 32 22
                  Difficult  13 31 11  8
                  Impossible 1   2   0   5
Traditional Sitting Easy      20 22 40 25
                  Difficult  7   27 10  7
                  Impossible 1   6   0   6
```

For easy computation of totals and proportions by rows or columns, R has the functions `margin.table()` (used in R Code 1.47) and `prop.table()` (used in R Code 1.48 on the facing page). The first argument (`x=`) for both functions is generally a table, although an array can be used with `margin.table()`, while the second argument (`margin=`) is a vector of indices where 1 is for rows and 2 is for columns.

R Code 1.47

```
> CT <- table(EPIbmi2$ease, EPIbmi2$doctor)
> CT

          A   B   C   D
Easy      39 49 72 47
Difficult 20 58 21 15
Impossible 2   8   0   11

> margin.table(CT)  # sum all entries in table
[1] 342

> margin.table(CT, 1)  # sum entries across rows

      Easy  Difficult Impossible
      207       114           21

> margin.table(CT, 2)  # sum entries down columns
```

```
A   B   C   D
61 115  93  73

> addmargins(CT)  # show margins
```

	A	B	C	D	Sum
Easy	39	49	72	47	207
Difficult	20	58	21	15	114
Impossible	2	8	0	11	21
Sum	61	115	93	73	342

R Code 1.48

```
> prop.table(CT)  # Equivalent to CT/margin.table(CT)

A           B           C           D
Easy      0.114035088 0.143274854 0.210526316 0.137426901
Difficult 0.058479532 0.169590643 0.061403509 0.043859649
Impossible 0.005847953 0.023391813 0.000000000 0.032163743

> prop.table(CT, 1)  # divide each entry of CT by its row total

A           B           C           D
Easy      0.1884058 0.2367150 0.3478261 0.2270531
Difficult 0.1754386 0.5087719 0.1842105 0.1315789
Impossible 0.0952381 0.3809524 0.0000000 0.5238095

> prop.table(CT, 2)  # divide each entry of CT by its column total

A           B           C           D
Easy      0.63934426 0.42608696 0.77419355 0.64383562
Difficult 0.32786885 0.50434783 0.22580645 0.20547945
Impossible 0.03278689 0.06956522 0.00000000 0.15068493
```

Using `cut()` Occasionally, the user may want to divide the range of a numeric variable into intervals returning a factor. The function `cut()` divides the range of a numeric variable into intervals and codes the values in the numeric variable according to the interval in which they fall. The two required arguments for `cut()` are `x=` (the numeric vector) and `breaks=`, which is either a numeric vector of two or more cut points or a single number greater than or equal to 2 that gives the number of intervals into which `x` is to be cut. The other arguments are optional, and the reader should consult the function's help file for further information. The R Code 1.49 on the next page first divides the range of BMI from the `EPIbmi` data frame into three equally spaced categories and codes the values of BMI into the interval where they fall, storing the result in `fBMI`. Second, it produces a table of the number of patients in the three equally spaced categories of `fBMI`. Third, it removes the variable `fBMI` from the workspace.

R Code 1.49

```
> fBMI <- cut(EPIbmi$BMI, breaks = 3) # factor BMI with 3 levels
> table(fBMI)

fBMI
(18.4,32.1] (32.1,45.8] (45.8,59.6]
    203        130         9

> rm(fBMI)
```

Instead of storing the recoded variable in the workspace, it is advisable to store the recoded factor with the data from which it was created. Consider R Code 1.50 which creates two factors `fBMI` and `f1BMI`, based on the values in `BMI` and stores the results in the `EPIbmi4` data frame.

R Code 1.50

```
> EPIbmi4 <- within(data = EPIbmi, expr = {
+   fBMI <- cut(BMI, breaks = 3, include.lowest = TRUE)
+   f1BMI <- factor(fBMI, labels = c("Low", "Med", "High"))
+ })
> head(EPIbmi4[, c("BMI", "fBMI", "f1BMI")])

      BMI      fBMI f1BMI
1 39.21038 (32.1,45.8]    Med
2 27.76343 [18.4,32.1]   Low
3 29.21011 [18.4,32.1]   Low
4 22.05805 [18.4,32.1]   Low
5 42.90715 (32.1,45.8]    Med
6 45.54180 (32.1,45.8]    Med

> levels(EPIbmi4$fBMI)
[1] "[18.4,32.1]" "(32.1,45.8]" "(45.8,59.6]"
```

Since `cut()` creates open left intervals, it is critical to use the argument `include.lowest = TRUE` with the function `cut()` to guarantee the smallest value is included in the first interval.

1.14 Summarizing Functions

In this section, the data frame `EPIbmi4` created in R Code 1.50 will be used to illustrate the functions `tapply()` and `aggregate()`. Suppose you need to compute the mean BMI for each of the twelve cells of the contingency table created from the factors `ease` and `doctor` of the `EPIbmi4` data frame. A brute force approach is to use filtering to create twelve vectors containing appropriate values and subsequently to compute the mean of each vector. R Code 1.51 on the facing page computes the mean BMI for patients doctor A classified as Easy to palpate and the mean BMI for patients doctor C classified as Difficult to palpate. This is done by first filtering the BMI values for patients doctor A classified as Easy to

palpate and storing the results in `EasyDocA` and then filtering the BMI values for patients of doctor C classified as Difficult to palpate and storing the results in `DiffDocC`. The function `mean()` is then individually applied to the two vectors, and the results are concatenated with the `c()` function to return a single vector.

R Code 1.51

```
> EasyDocA <- with(data = EPIbmi4, BMI[ease == "Easy" & doctor == "A"])
> DiffDocC <- with(data = EPIbmi4, BMI[ease == "Difficult" &
+                                         doctor == "C"])
> c(mean(EasyDocA), mean(DiffDocC))

[1] 28.89412 36.17516
```

A better approach (shown in R Code 1.52) is to use the function `tapply()`, which returns its results as an array. This R function will apply a mathematical or logical expression using the argument `FUN=` to each value of a vector in the argument `X=`. The vector corresponds to one of the categories (cells of a table or contingency table) in the argument to `INDEX=`.

R Code 1.52

```
> with(data = EPIbmi4,
+       tapply(X = BMI, INDEX = list(ease, doctor), FUN = mean)
+ )

          A         B         C         D
Easy      28.89412 28.74324 29.13575 29.28693
Difficult 33.56409 33.75993 36.17516 35.23604
Impossible 43.34582 39.95822      NA 45.09995
```

In R Code 1.52, the function supplied to the `FUN=` argument was the mean. It should be noted that the function supplied to the argument `FUN=` can be any R or user-defined function. The function `aggregate()` (used in R Code 1.53) can also be used to compute the same quantities; however, the output is returned as a data frame. Note the differences in both case and name of the arguments for `tapply()` and `aggregate()`. Note that `aggregate()` does not report a mean BMI value for patients Doctor C classified as Impossible to palpate since there are no patients in this category, while `tapply()` returns `NA` for the same category.

R Code 1.53

```
> AgDF <- with(data = EPIbmi4,
+               aggregate(x = BMI, by = list(ease, doctor), FUN = mean)
+ )
> AgDF

  Group.1 Group.2     x
1   Easy      A 28.89412
2 Difficult    A 33.56409
3 Impossible   A 43.34582
4   Easy      B 28.74324
5 Difficult    B 33.75993
6 Impossible   B 39.95822
7   Easy      C 29.13575
8 Difficult    C 36.17516
```

```
9      Easy      D 29.28693
10 Difficult    D 35.23604
11 Impossible   D 45.09995
```

Example 1.3 Assign the values (19, 14, 15, 17, 20, 23, 19, 19, 21, 18) to a vector x such that the first five values of x are in treatment A and the next five values are in treatment B. Compute the means for the two treatment groups using `tapply()`.

Solution: First, assign the values to a vector x , where the first five elements are in treatment A and the next five are in treatment B in one of two ways. The first approach uses `c()` to combine five A's and five B's.

```
> x <- c(19, 14, 15, 17, 20, 23, 19, 19, 21, 18)
> treatment <- c(rep("A", 5), rep("B", 5))
> treatment

[1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B"
```

A slightly more general approach for creating `treatment` uses the function `rep()` inside `rep()` as follows.

```
> treatment <- rep(LETTERS[1:2], rep(5, 2))
> treatment

[1] "A" "A" "A" "A" "A" "B" "B" "B" "B" "B"
```

Next, use `tapply()` to calculate the means for treatments A and B:

```
> ANSWER <- tapply(x, treatment, mean)
> ANSWER          # show results

A  B
17 20

> ANSWER['B']      # show only mean of B

B
20

> is.array(ANSWER) # confirms results are stored in an array
[1] TRUE

> rm(x, treatment) # cleanup workspace
```

Note that the results from using `tapply()` are stored as an array even though x and `treatment` are vectors. ■

1.15 Probability Functions

R has four classes of functions that perform probability calculations on all of the distributions covered in this book. These four functions generate random numbers, calculate

cumulative probabilities, compute densities, and return quantiles for the specified distributions. Each of the functions has a name beginning with a one-letter code indicating the type of function: `rdist`, `pdist`, `ddist`, and `qdist`, respectively, where `dist` is the R distribution name. Some of the more important probability distributions that work with the functions `rdist`, `pdist`, `ddist`, and `qdist` are listed in Table A.8 on page 910. For example, given vectors q and x containing quantiles (or percentiles), a vector p containing probabilities, and the sample size n for X having a $N(0, 1)$ distribution,

- `pnorm(q, mean = 0, sd = 1)` computes $\mathbb{P}(X \leq q)$
- `qnorm(p, mean = 0, sd = 1)` computes x such that $\mathbb{P}(X \leq x) = p$
- `dnorm(x, mean = 0, sd = 1)` computes $f(x)$, the probability density's value at x .
- `rnorm(n, mean = 0, sd = 1)` returns a random sample of size n from a $N(\mu = 0, \sigma = 1)$ distribution.

This class of functions will also accept a vector for the function's arguments. For example, `dpois(x = 0:10, lambda = 3)` returns a vector of values for $\mathbb{P}(X = x | X \sim Pois(\lambda = 3))$ for $x = 0, 1, 2, \dots, 10$.

```
> dpois(x = 0:10, lambda = 3)
[1] 0.0497870684 0.1493612051 0.2240418077 0.2240418077 0.1680313557
[6] 0.1008188134 0.0504094067 0.0216040315 0.0081015118 0.0027005039
[11] 0.0008101512
```

1.16 Flow Control

R, like most programming languages, has the ability to control the execution of code with programming statements such as `for`, `while`, `repeat`, and `break`. As an example, consider how `for` is used in R Code 1.54 to add the values 10, 20, and 30.

R Code 1.54

```
> sum.a <- 0
> for(i in c(10, 20, 30)){
+   sum.a <- i + sum.a
+ }
> sum.a
[1] 60
```

The `for` statement allows one to specify that a certain operation will be repeated a fixed number of times. The syntax for the `for` statement is

```
for (name in vector) {
  statements
}
```

Statements can be grouped using braces ({}). When several statements are grouped, the standard R syntax is to start with a left brace ({), then place each statement on its own line and close the group of statements with a right brace (}) on its own line.

Example 1.4 A Lucas number (Burton, 2010) is defined as

$$L_n = \begin{cases} 2 & \text{if } n = 1 \\ 1 & \text{if } n = 2 \\ L_{n-1} + L_{n-2} & \text{if } n \geq 3. \end{cases}$$

Use a **for** loop to compute the first 15 Lucas numbers.

Solution: R Code 1.55 computes the first 15 Lucas numbers after allocating space using the **numeric()** function for the answers.

R Code 1.55

```
> Number <- 15                      # Number of Lucas numbers desired
> Lucas <- numeric(Number)          # Storage for Lucas numbers
> Lucas[1] <- 2                    # First Lucas number
> Lucas[2] <- 1                    # Second Lucas number
> for(i in 3:Number){              +
+   Lucas[i] <- Lucas[i - 1] + Lucas[i - 2]
+ }
> Lucas
```

```
[1] 2 1 3 4 7 11 18 29 47 76 123 199 322 521 843
```

The **while** statement is useful for repeating a set of statements when the exact number of repeats is not known in advance. The syntax for the **while** statement is **while (condition) {statements}**. The statements are evaluated as long as the condition is TRUE. Once the condition evaluates to FALSE, nothing more is done. An alternative to the **while** statement is the **repeat** statement with a **break** statement. The syntax for using a **repeat** statement is **repeat {statements}**. The **statements** generally include a **break** statement of the form **if (condition) break**. Statements are repeated as long as the condition is TRUE. Once the condition evaluates to FALSE, nothing more is done.

The square root of a positive number, x , can be approximated iteratively using $x_{n+1} = (x_n + x/x_n)/2$ where x_n is the initial guess for the value of \sqrt{x} . Approximating the $\sqrt{113734}$ to within 0.00001 can be accomplished using a **while** statement or a **repeat** statement. Using a **while** statement to approximate $\sqrt{113734}$ to within a 0.00001 is shown in R Code 1.56.

R Code 1.56

```
> options(digits = 8)
> x <- 113734
> tolerance <- 0.00001
> oldapp <- x/2
> newapp <- (oldapp + x/oldapp)/2
> i <- 0
> while( abs(newapp - oldapp) > tolerance){
+   oldapp <- newapp
```

```
+ newapp <- (oldapp + x/oldapp)/2
+ i <- i + 1 # Iteration number
+
> c(newapp, i)
[1] 337.24472 11.00000
> options(digits = 7) # reset to default
```

Using a `repeat` statement to approximate $\sqrt{113734}$ to within a 0.00001 is shown in R Code 1.57.

R Code 1.57

```
> options(digits = 8)
> x <- 113734
> tolerance <- 0.00001
> oldapp <- x/2
> newapp <- (oldapp + x/oldapp)/2
> i <- 0
> repeat{
+   oldapp <- newapp
+   newapp <- (oldapp + x/oldapp)/2
+   i <- i + 1
+   if(abs(newapp - oldapp) < tolerance)
+     break
+ }
> c(newapp, i)
[1] 337.24472 11.00000
> options(digits = 7) # reset to default
```

The approximate $\sqrt{113734}$ (337.2447) is achieved in $i = 11$ iterations regardless of the type of statement used.

The `if()` statement allows one to control which statements are executed. The syntax for the `if()` statement has two forms:

```
if (condition) {statements when TRUE}
```

and

```
if (condition) {
  statements when TRUE
} else {
  statements when FALSE
}
```

One needs to pay close attention to how the second form is typed. In particular, entering

```
if(condition) {statements when TRUE}
else {statements when FALSE}
```

may not do what you expect. R will execute the first line before seeing the second line.

Example 1.5 Write three separate programs to simulate throwing a pair of dice 99,999 times. Compute the mean of each of the 99,999 throws, and create a table of the means using a for loop, a while statement, and a repeat statement.

Solution: The function `sample()` is used to sample 2 values with replacement from 1 to 6.

Using a for loop: Each time the loop cycles, the mean of the two dice is stored in the i^{th} position of the `means` vector. The tabled results are stored in `T1` and subsequently printed.

```
> set.seed(3) # setting seed for reproducibility
> N <- 10^5 - 1 # N = number of simulations
> means <- numeric(N) # Defining numeric vector of size N
> for (i in 1:N) {
+   means[i] <- mean(sample(x = 1:6, size = 2, replace = TRUE))
+ }
> T1 <- table(means)
> T1

means
  1   1.5    2   2.5    3   3.5    4   4.5    5   5.5    6
2802  5523  8394 11138 13938 16552 13876 11202  8287  5576 2711
```

Using a while statement: The command `matrix(0, N, 2)` creates a 99,999 by 2 matrix of zeros. While $i \leq N$, the results from simulating tossing two dice are stored in the i^{th} row of the `N2mat` matrix. The function `apply()` is used to compute the mean of each of the rows of `N2mat` and to store the result in the vector `means`. The tabled results are stored in `T2` and subsequently printed.

```
> set.seed(3) # setting seed for reproducibility
> i <- 1
> N <- 10^5 - 1 # N = number of simulations
> N2mat <- matrix(0, N, 2) # initialize N*2 matrix to all 0's
> while (i <= N) {
+   N2mat[i, ] <- sample(x = 1:6, size = 2, replace = TRUE)
+   i <- i + 1
+ }
> means <- apply(N2mat, 1, mean)
> T2 <- table(means)
> T2

means
  1   1.5    2   2.5    3   3.5    4   4.5    5   5.5    6
2802  5523  8394 11138 13938 16552 13876 11202  8287  5576 2711
```

Using a repeat statement: The line `N2mat[i,] <- sample(1:6, 2, replace = TRUE)` is repeated, filling in the i^{th} row of the matrix with values that simulate throwing two dice until $i = N$, at which point the repeat ends. The function `apply()` is used to compute the mean of each of the rows of `N2mat` and to store the result in the vector `means`. The tabled results are stored in `T3` and subsequently printed.

```
> set.seed(3) # setting seed for reproducibility
> i <- 1
> N <- 10^5 - 1 # N = number of simulations
```

```

> N2mat <- matrix(0, N, 2) # initialize N*2 matrix to all 0's
> repeat {
+   N2mat[i, ] <- sample(1:6, 2, replace = TRUE)
+   if (i == N)
+     break
+   i <- i + 1
+ }
> means <- apply(N2mat, 1, mean)
> T3 <- table(means)
> T3

means
 1   1.5    2   2.5    3   3.5    4   4.5    5   5.5    6 
2802 5523 8394 11138 13938 16552 13876 11202  8287  5576 2711

```

The function `plot()` is applied to `T3` after dividing its contents by `N`. The result, with a few embellishments, is presented in Figure 1.18. The `plot()` function will be more fully described in Section 1.19 on page 80.

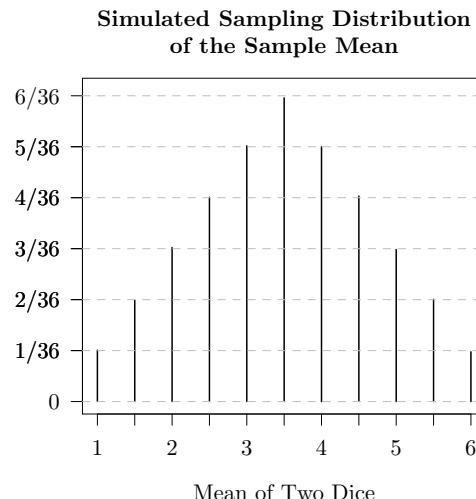


FIGURE 1.18: Graphical representation of the relative frequency of each of the possible means from a simulation of throwing two dice 99,999 times

The function `ifelse()` can be used for conditional element selection, which differs from the previous if-else constructs that were used for conditional execution. The arguments for the conditional selection are `ifelse(test, yes, no)`, where the function returns a value with the same shape as `test`, which is filled with elements selected from either `yes` or `no` depending on whether the element of `test` is `TRUE` or `FALSE`. Consider the matrix of mean BMI values for pregnant women classified by `Doctor` and `Ease` created with the commands in R Code 1.58.

R Code 1.58

```
> MBMI <- with(data = EPIbmi4, tapply(BMI, INDEX = list(ease,
```

```
+     doctor), FUN = mean))
> MBMI # show mean BMI values
```

	A	B	C	D
Easy	28.89412	28.74324	29.13575	29.28693
Difficult	33.56409	33.75993	36.17516	35.23604
Impossible	43.34582	39.95822	NA	45.09995

To label each entry of the matrix where the mean BMI is greater than 35 with "Obese" and each entry where the mean BMI is less than or equal to 35 with "Not-Obese", consider R Code 1.59.

R Code 1.59

```
> CBMI <- with(data = EPIbmi4, ifelse(test = tapply(BMI, INDEX = list(ease,
+     doctor), FUN = mean) > 35, yes = "Obese", no = "Not-Obese"))
> CBMI
```

	A	B	C	D
Easy	"Not-Obese"	"Not-Obese"	"Not-Obese"	"Not-Obese"
Difficult	"Not-Obese"	"Not-Obese"	"Obese"	"Obese"
Impossible	"Obese"	"Obese"	NA	"Obese"

Note that each entry in MBMI with a mean BMI greater than 35 is labeled "Obese" and each entry where the mean BMI is less than or equal to 35 is labeled "Not-Obese" in the array CBMI.

1.17 Creating Functions

One of the more attractive features of the R language is the flexibility the user has to modify existing functions and to create new functions. System functions in R are called by typing the name of the function and specifying the arguments being passed to the function inside parentheses. The same principle applies when constructing a new function. The basic structure of a function is

```
> fname <- function(argument1, argument2, ...) {
+   expression
+ }
```

The **expression** can be any R code. When one of the arguments takes a default value in the function definition, there is no need to type that value when the function is called. Suppose a function to sum the first n natural numbers is needed. The formula to find the sum of the first n natural numbers is $n \times (n + 1)/2$. To create the R function **SUM.N()**, type

```
> SUM.N <- function(n) {
+   n * (n + 1)/2
+ }
```

Using the function **SUM.N()**, one can see that the sum of the first 4 natural numbers is 10.

```
> SUM.N(n = 4)
[1] 10
```

The function `sum.sq()` sums the squares of the values in a vector or matrix `x`.

```
> sum.sq <- function(x) {
+   sum(x^2)
+ }
```

The result of one function can be passed as an argument to another function. Note how the function `SUM.N()` with an argument of `n = 4`, which returns an answer of 10, is passed to the function `sum.sq()`, which squares the value of 10 and returns the answer of 100.

```
> sum.sq(SUM.N(n = 4))
[1] 100
```

The astute reader will have noticed that the functions discussed are all followed by parentheses inside which the functions' arguments are specified. The arguments inside the parentheses can be positional, named, or a mixture of the two. Generally, it is a good idea to use named arguments when calling a function that has many arguments or when using lesser-known arguments. This reduces the risk of specifying the wrong argument and improves the readability of the code. When a mixture of named and unnamed arguments is used in a function, the named arguments are matched first; then the unnamed arguments are matched against the unused arguments in order.

The function `sample()` is used to illustrate named (R Code 1.60), positional (R Code 1.61 on the next page), and a mixture of named and positional arguments (R Code 1.62 on the following page). Note that the function `sample()` has four arguments: `x`, `size`, `replace = FALSE`, and `prob = NULL`. The first argument, `x`, is either a vector of one or more elements from which to choose or a positive integer. In the example, the positive integer 10 is supplied to `x`, which is a convenience feature that tells the function to sample from the numbers 1 to 10. The second argument, `size`, is given the value 20, which tells the function to pick 20 values from the values given to `x` (1 to 10). Since there are only 10 values in `x`, the logical value of TRUE is passed to the `replace` argument, which allows the generated result to be larger than the values in `x`. Since nothing is passed to the `prob` argument, sampling from the numbers 1 to 10 is done in a uniform fashion. Finally, since the names of all of the arguments are specified, the order in which the arguments in the function are assigned does not alter the value of the function.

R Code 1.60

```
> args(sample)
function (x, size, replace = FALSE, prob = NULL)
NULL

> set.seed(13)
> Named <- sample(size = 20, replace = TRUE, x = 10)
> Named

[1] 8 3 4 1 10 1 6 8 9 1 7 9 9 6 6 4 4 6 9 7
```

When names are not used for the arguments (R Code 1.61), the values provided are interpreted by the function in the order of the arguments in the function. In this case, the first value 10 will be passed to `x`, the second value, 20, will be passed to `size`, and the third value, TRUE, will be passed to `replace`.

R Code 1.61

```
> set.seed(13)
> Position <- sample(10, 20, TRUE)
> Position

[1] 8 3 4 1 10 1 6 8 9 1 7 9 9 6 6 4 4 6 9 7
```

When a mixture of named and unnamed arguments is used in a function (R Code 1.62), the named arguments are matched first; then the unnamed arguments are matched against the unused arguments in order. In the next example, since `size = 20` is matched first, the unused arguments of 10 and TRUE are matched to `x`, then to `replace`.

R Code 1.62

```
> set.seed(13)
> Mixture <- sample(10, TRUE, size = 20)
> Mixture

[1] 8 3 4 1 10 1 6 8 9 1 7 9 9 6 6 4 4 6 9 7
```

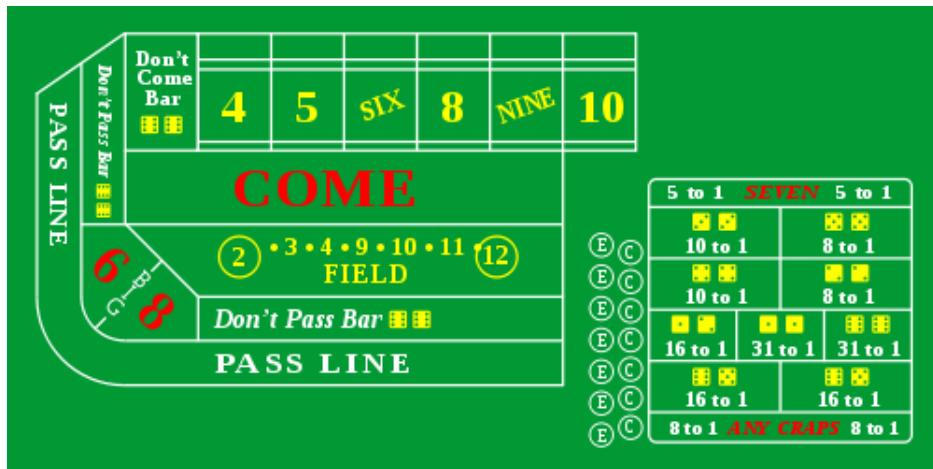
When one creates new functions, storing them in a single file can be convenient. By storing all of the functions one creates in a single file, one would be able to read all of them into the R session by typing

```
> source("C:/Rfolder/functions.txt") # For Windows
> source("/Rfolder/functions.txt") # For Mac/Unix-like
```

assuming the functions are all stored in a text file named `functions.txt` in the `Rfolder` at the machine's root.

Example 1.6 Pass line betting in bank/casino craps works as follows: A player places a bet anywhere inside the area of the table marked PASS LINE and waits for the shooter (the person rolling the dice) to roll the dice. The first roll the shooter takes is called the come out roll. If the come out roll (sum of the two dice) is either a 7 or an 11, all bettors win the amount of money they have placed in the PASS section of the table. If the come out roll is a 2, 3, or 12 (crapping out), all bettors lose the amount of money they have placed in the PASS LINE section of the table (see Figure 1.19 on the next page). If the come out roll is any other number (4, 5, 6, 8, 9, or 10), then that number is called the shooter's point, and the shooter rolls the dice repeatedly until either a 7 or the shooter's point is rolled again. If the shooter rolls a 7 before rolling point, all bets in the PASS LINE are lost. If the shooter rolls point before rolling a 7, the bank/casino pays all bets in the PASS LINE to the players. Write a function that simulates the outcomes from pass line betting in bank/casino craps.

Solution: The function `SIMcraps()` consists of two basic parts. The first part is the function `linebet()`, which simulates winning or losing a line bet in bank/casino craps. The second part is a `for` loop that runs the function `linebet()` a fixed number of times, then computes the percent of games out of the total number of games simulated that resulted in a win for the shooter.

FIGURE 1.19: Picture of a craps table taken from <http://en.wikipedia.org/wiki/Craps>

```
> SIMcraps <- function(n.games = 10000) {
+   opar <- par(no.readonly = TRUE)
+   options(scipen = 999) # Suppress scientific notation in the output
+   # linebet returns 0 or 1 based on whether 'shooter'
+   # loses or wins
+   linebet <- function() {
+     comeoutroll <- sum(sample(1:6, 2, replace = TRUE)) # first throw
+     if (comeoutroll %in% c(7, 11)) {
+       result <- 1 # win if comeoutroll is 7 or 11
+     } else if (comeoutroll %in% c(2, 3, 12)) {
+       result <- 0 # loss if comeoutroll is 2, 3, or 12
+     } else {
+       repeat {
+         substthrow <- sum(sample(1:6, 2, replace = TRUE))
+         # subsequent throw
+         if (substthrow == comeoutroll) {
+           result <- 1 # win if substthrow same as comeoutroll
+           break
+         } else if (substthrow == 7) {
+           result <- 0 # loss if substthrow is a 7
+           break
+         }
+       }
+     }
+     result
+   }
+   gameOutcome <- numeric(n.games) # vector of 0s of length n.games
+   # Play n.games simulated crap games
+   for (i in 1:n.games) {
+     gameOutcome[i] <- linebet()
+   }
+   # gameOutcomes is a vector of wins and losses
```

```

+ P.win <- mean(gameOutcome) # Percent of time shooter wins
+ Actual.answer <- 244/495
+ Error <- round(abs(Actual.answer - P.win)/Actual.answer *
+   100, 4)
+ cat("Simulated probability of winning =", P.win, "based on",
+     n.games, "simulated games.", "\n")
+ cat("Percent simulation error based on actual answer 244/495 is ",
+     Error, "%.", "\n", sep = "")
+ par(opar)
+ options(scipen = 0)
+
}

```

Running the function:

```

> set.seed(1234) # setting seed for reproducibility
> SIMcraps(n.games = 50000)

```

Simulated probability of winning = 0.49188 based on 50000 simulated games.
 Percent simulation error based on actual answer 244/495 is 0.2129%.

Computing the actual answer, which is 244/495, will be revisited in Example 3.20 on page 220, when general probability and random variables are discussed.



1.18 Simple Imputation

Section 1.11.1 on page 51 discussed the removal of NA values from data sets to form a data set without missing values. In this section, an example with imagined numbers is presented to illustrate simple imputation. The literature for missing data generally classifies missing data into one of three categories: missing completely at random (MCAR), missing at random (MAR), and not missing at random (NMAR). Missing data is considered to be MCAR if the missing data is not related to any other observable or unobservable variables. If the missing data is related to some other observed variable but not to its own unobserved values, it is considered to be MAR. If the missing data is neither MCAR or MAR, it is considered to be NMAR.

Consider the data shown in Table 1.2 on the facing page, where it will be assumed that the missing values are MAR. That is, the missing values are related to the industry in which they occur. The desire is to replace the missing values within an industry with some reasonable statistic. In this case, the median (the 50th percentile) is chosen for ease of illustration. The median is the value that splits the data into equal halves and will be addressed in greater length in Chapter 2 on page 97. From Table 1.2 on the facing page, one observes one missing value for industry A, two missing values for industry C, and one missing value for industry D.

The median for industry A is 2, computed from the non-missing values 1, 2, and 3. The median for industry C is 7.5, computed from the non-missing values of 7 and 8. The median for industry C is 10, computed from the non-missing values of 9, 10, and 11. At this point, one can simply replace the missing values with their medians or some other suitable statistic. R code 1.63 on the next page creates a data frame named **DF**, then imputes the

Table 1.2: Missing at random values by industry example

Industry:	A	B	C	D	A	A	B	B	D	D	D	A	C	C	C
Value:	1	4	7	9	NA	2	5	6	NA	10	11	3	8	NA	NA

median of the non-missing values for each industry. The imputing uses an `ifelse()` to test for `NA` values. If `NA` values are present, the `ave()` function computes group medians for each industry for each observation in `Value`. The group medians then replace the `NAs` in `Value` and store the result in a new variable `Ivalue`. The code may appear overly complex for such a trivial example; however, the same code could be used for a similar situation with a much larger data set.

R Code 1.63

```
> Industry <- c("A", "B", "C", "D", "A", "A", "B", "B", "D", "D", "D", "D",
+           "A", "C", "C", "C")
> Value <- c(1, 4, 7, 9, NA, 2, 5, 6, NA, 10, 11, 3, 8, NA, NA)
> DF <- data.frame(Industry, Value)
> rm(Industry, Value) # remove Industry and Value from work space
> DF$Ivalue <- ifelse(is.na(DF$Value),
+                       ave(DF$Value, DF$Industry,
+                           FUN = function(x){median(x, na.rm = TRUE)}),
+                       DF$Value)
> # Order DF by Industry and Value
> ODF <- DF[order(DF$Industry, DF$Value), ]
> ODF

   Industry Value Ivalue
1          A     1    1.0
6          A     2    2.0
12         A     3    3.0
5          A    NA    2.0
2          B     4    4.0
7          B     5    5.0
8          B     6    6.0
3          C     7    7.0
13         C     8    8.0
14         C    NA    7.5
15         C    NA    7.5
4          D     9    9.0
10         D    10   10.0
11         D    11   11.0
9          D    NA   10.0

> MED <- tapply(ODF$Value, ODF$Industry, median, na.rm = TRUE)
> MED

  A      B      C      D 
2.0  5.0  7.5 10.0
```

1.19 Using plot()

Data are often summarized with graphs. The R language provides a rich set of commands for creating graphs and altering the default graphic states. Tables A.5 on page 907, A.4 on page 906, and A.6 on page 908 outline some of the basic commands used to create graphs and to customize the graphic states. For further detail on any R function or parameter, the user should seek help from the extensive system help files by typing `help(function.name)`, `?function.name`, `help(par)`, or `?par`. The R function `plot()` is a high-level, generic function that produces an appropriate graph whose form depends on the type of data. The axes, labels, scales, and plotting symbols are all default values chosen automatically, any and all of which may be changed by the user. When two numeric vectors of the form `vector1 ~ vector2` are passed to `plot()`, a scatterplot is produced. If a numeric vector and a factor of the form `vector ~ factor` are used in `plot()`, boxplots are created. If the result from `table()` is passed to `plot()`, a barplot is created. Passing a single factor to `plot()` will produce a barplot, while passing a formula of the form `factor ~ numeric.vector` will create a spinogram. When a numeric vector and a factor are used with `plot()`, a stripchart is created. If two factors are passed, a spineplot is created. When a two-dimensional table is used in `plot()`, a mosaic plot is made. If a time-series object is passed, a time-series graph is produced. The user can see that `plot()` can be used to create many appropriate graphs for displaying data.

The data frame `EPIbmi`, created in R Code 1.41 on page 55 (recreated in R Code 1.64), and the time series data set `sunspots`, which is part of the R data sets, are used in R Code 1.64 and R Code 1.65 on the facing page. R Code 1.64 tests the objects `BMI`, `kg`, `ease`, `treatment`, and `sunspots` with the functions `is.numeric()`, `is.factor()`, and `is.ts()`, to verify that different objects are stored as numeric, factor, or time series, respectively.

R Code 1.64

```
> EPIbmi <- within(data = EPIDURALF, expr = {
+   BMI = kg/(cm/100)^2
+   ease = factor(ease, levels = c("Easy", "Difficult", "Impossible"))
+ })
> with(data = EPIbmi,
+       is.numeric(BMI))
[1] TRUE
> with(data = EPIbmi,
+       is.numeric(kg))
[1] TRUE
> with(data = EPIbmi,
+       is.factor(ease))
[1] TRUE
> with(data = EPIbmi,
+       is.factor(treatment))
[1] TRUE
> is.ts(sunspots) # is time series sunspots
[1] TRUE
```

Consider how `plot()` chooses an appropriate graph based on the different types of data in R Code 1.65 and displays the results in Figure 1.20 on the next page.

R Code 1.65

```
> par(mfrow= c(3, 3))      # graphics device with 3 rows and 3 columns
> with(data = EPIbmi,
+       plot(BMI ~ kg, main = "Scatterplot"))          # num ~ num
> with(data = EPIbmi,
+       plot(BMI ~ ease, main = "Boxplots"))           # num ~ factor
> with(data = EPIbmi,
+       plot(table(ease), main = "Barplot"))            # 1D table
> with(data = EPIbmi,
+       plot(ease, main = "Barplot"))                  # factor
> with(data = EPIbmi,
+       plot(ease ~ BMI, main = "Spinogram"))          # factor ~ num
> with(data = EPIbmi,
+       plot(BMI, ease, main = "Stripchart"))          # num, factor
> with(data = EPIbmi,
+       plot(ease ~ treatment, main = "Spine plot")) # factor ~ factor
> with(data = EPIbmi,
+       plot(table(ease, treatment), main = "Mosaic plot")) # 2D table
> plot(sunspots, main = "Time-series plot")         # time series
> par(mfrow=c(1, 1))      # graphics device with 1 row and 1 column
```

At this point, the reader is not expected to understand the graphs in Figure 1.20 on the following page. Rather, Figure 1.20 and the given code are meant to highlight how `plot()` produces different types of graphs based on the type of data it is given. In the remainder of this section, the function `plot()` is used with numeric vectors to illustrate some of the graphs that result from changing different arguments inside the function. To view the arguments one can change when using `plot()` with numerical vectors, enter `?plot.default` at the R prompt. The current graphic states can be seen by typing `par()` at the R prompt. R Code 1.66 illustrates the use of various arguments inside the function `plot()` and can be used to recreate Figure 1.21 on page 83. The function `text()` is also used in R Code 1.21 on page 83 to place a character string inside the plot given *x* and *y* coordinates.

R Code 1.66

```
> par(mfrow=c(3, 3), pty = "m")  #3 by 3 layout
> x <- -4:4
> y <- x^2
> plot(x, y, xlim=c(-8, 8), ylim = c(0, 20), main = "")
> title(main = "Default values with limits \n for x and y axes altered")
> plot(x, y, pch = "x", xlim=c(-8, 8), ylim = c(0, 20), main="")
> title(main = "Default plotting character \n changed to x")
> plot(x, y, type = "l", xlim = c(-8, 8), ylim = c(0, 20), main="")
> title(main = "Lines connecting the data")
> plot(x, y, type = "b", xlim = c(-8, 8), ylim = c(0, 20), main="")
> title(main = "Both point and lines \n between data")
> plot(x, y, type = "h", xlim = c(-8, 8), ylim = c(0, 20), main="")
> title(main = "Vertical lines")
> plot(x, y, type = "o", xlim = c(-8, 8), ylim = c(0, 20), main="")
```

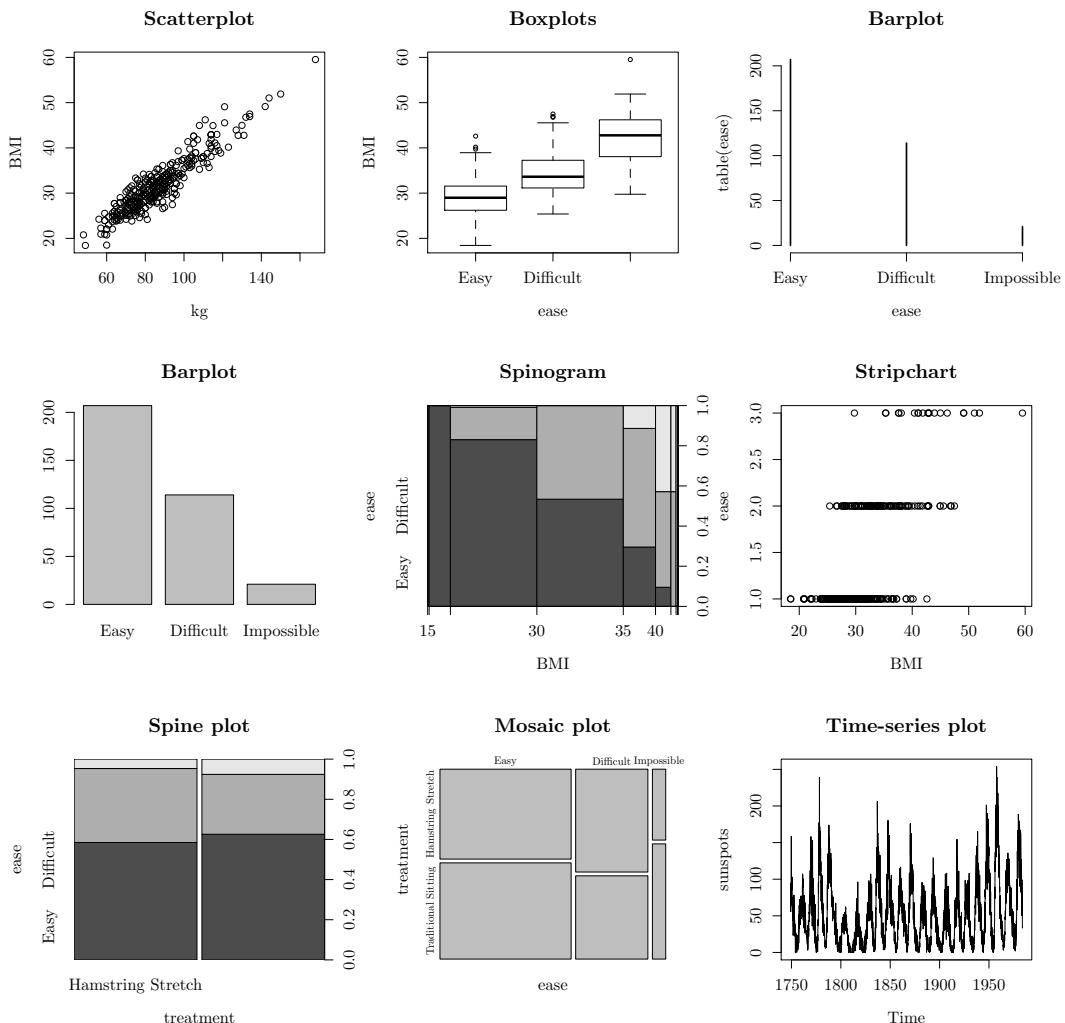
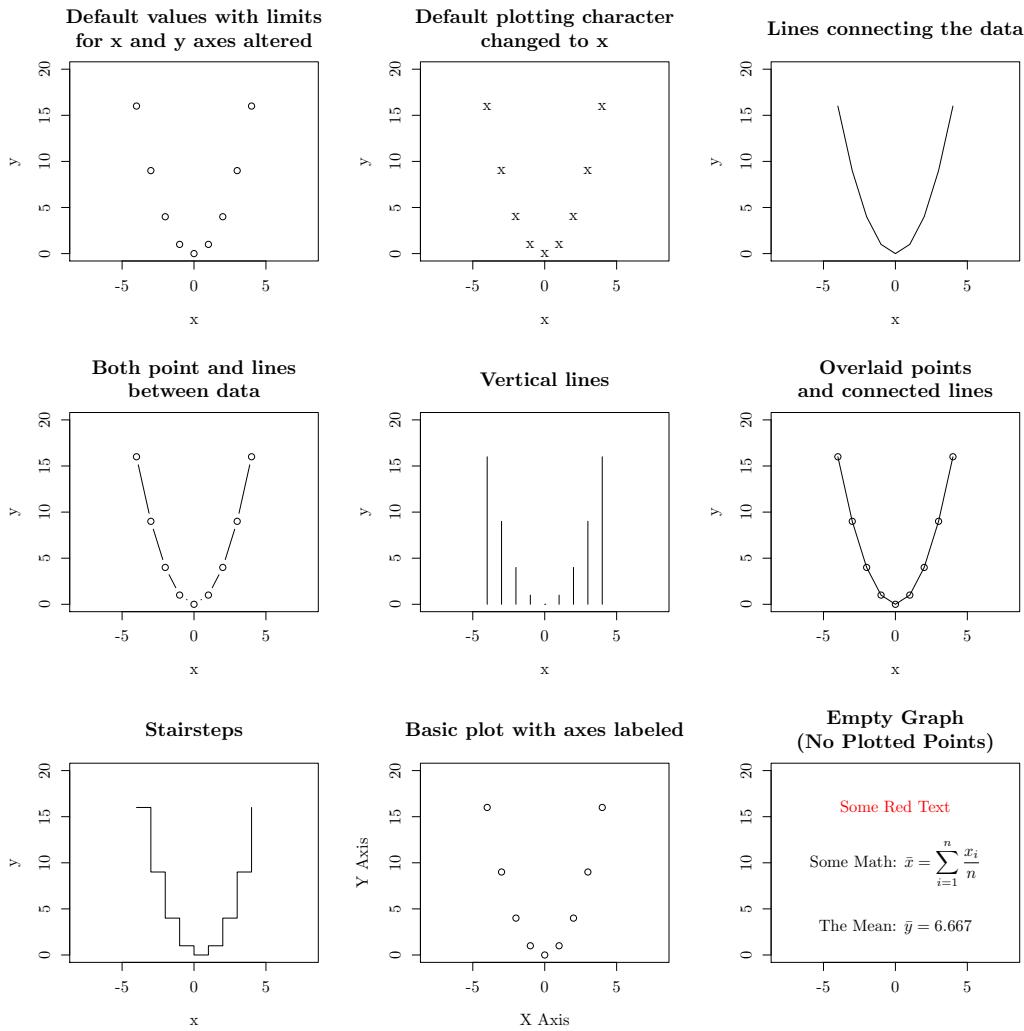


FIGURE 1.20: Graphs from applying `plot()` to different types of data

```
> title(main = "Overlaid points \n and connected lines")
> plot(x, y, type = "s", xlim = c(-8, 8), ylim = c(0, 20), main="")
> title(main = "Stairsteps")
> plot(x, y, xlim = c(-8, 8), ylim = c(0, 20), main = "", xlab = "X Axis",
+       ylab = "Y Axis")
> title(main = "Basic plot with axes labeled")
> plot(x, y, type = "n", xlim = c(-8, 8), ylim = c(0, 20), xlab = "",
+       ylab = "", main = "")
> title(main = "Empty Graph \n(No Plotted Points)")
> text(0, 16, "Some Red Text", col = "red")
> text(0, 10, expression(paste("Some Math: ", bar(x)==sum(frac(x[i],
+           n), i==1, n))))
> Alpha <- round(mean(y), 3)
> text(0, 3, bquote(paste("The Mean: ", bar(y)==.(Alpha))))
> par(mfrow=c(1, 1))
```

FIGURE 1.21: Results from using `plot()` with different types of data and arguments

The bottom right graph of Figure 1.21 illustrates that one can create a graph without any points and subsequently add points, lines, text, mathematical notation, and so on. To learn more about adding mathematical notation to a plot in R, enter `demo(plotmath)` at the R prompt, which runs the R `plotmath` demo, or type `?plotmath` at the R prompt to read the `plotmath` help file. Two of the more frequently used arguments with `par()` are `mfrow` and `mfcol`, which subdivide the plotting region into an array of figures. For example, `par(mfrow = c(3, 3))` divides the screen into nine figure regions (3 rows by 3 columns). The graphical parameter `mfrow` stands for *multiframe rowwise layout*. The commands `\n`, `\b`, `\t` tell R to make a new line, to make a backspace, and to make a tab, respectively. More information of various uses of quoting in R can be found by entering `?Quotes` at the R prompt. Running the code while sitting at a computer and experimenting with changing other arguments and graphical parameters is a great way to jump start your knowledge of the `plot()` function.

The bottom right graph of Figure 1.21 on the previous page requires further explanation. First, the placement of lines, text, points, and so on, in a plot is based on the coordinates of the plotting region. This coordinate system is referred to as the user coordinates. Figure 1.23 on the facing page illustrates the user coordinate system for traditional graphics (Murrell, 2011). To be complete, every graph in the traditional graphics system has three regions: the outer margins, the figure region, and the plot region. Figure 1.24 on page 88 fully illustrates the three regions when there is only one figure on the page.

Another use of the `plot()` function is to graph functions instead of explicitly plotting data points. R Code 1.67 illustrates two separate approaches to graphing an arbitrary function. The first approach uses the function `plot()` and the second approach uses the function `curve()`. Both approaches produce the same result shown in Figure 1.22.

R Code 1.67

```
> f <- function(x){sin(1+x^2)/(1+x^2) + x^2/100}
> plot(f, from = -10, to = 10, n = 1000, xlab = "", ylab = "")
> curve(f, from = -10, to = 10, n = 1000, xlab = "", ylab = "")
```

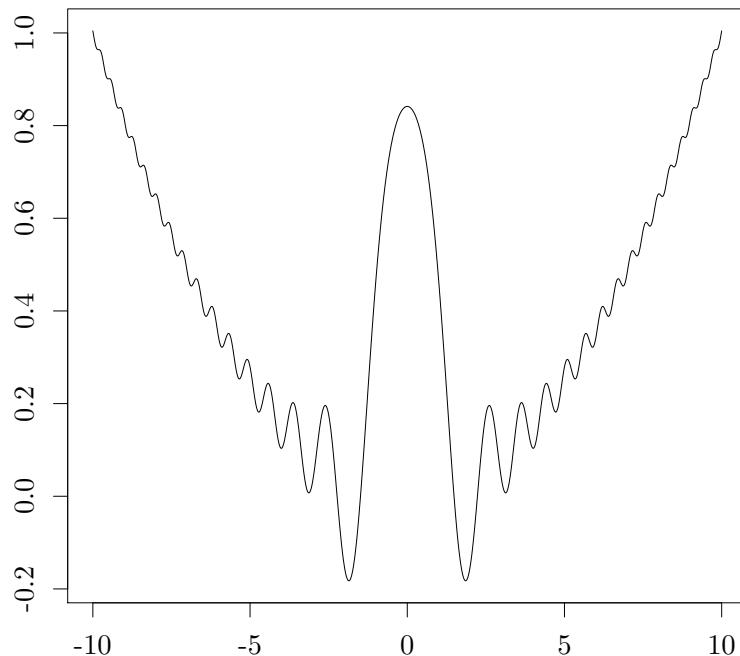


FIGURE 1.22: Graphing an arbitrary function

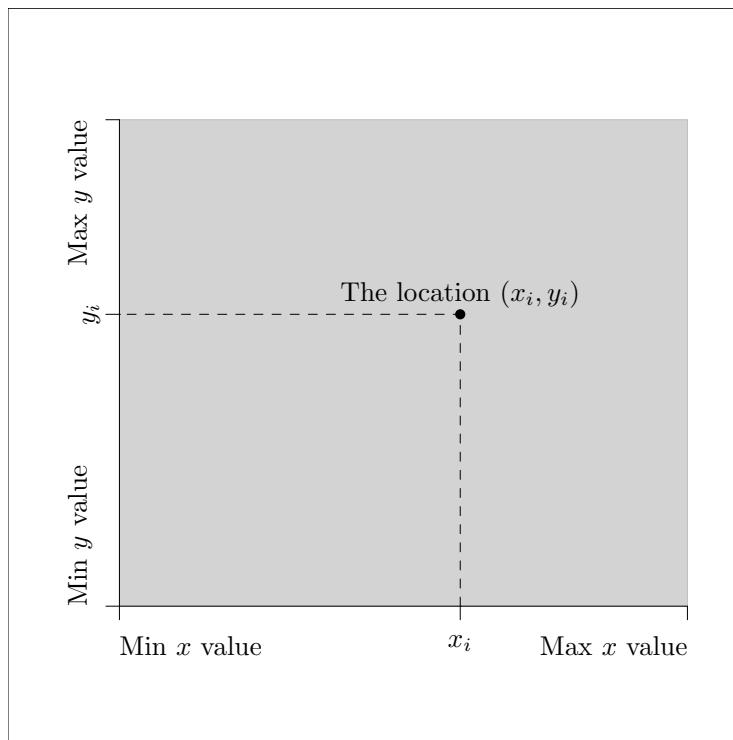


FIGURE 1.23: Using user coordinates to label a point

1.20 Coordinate Systems and Traditional Graphic's States

Each plotting region has at least one coordinate system, and drawing or annotating text in each region is done with respect to the relevant coordinate system. The coordinate system for the plot region, referred to as user coordinates, uses the range of values on the axes of the plot. Any drawing or annotation done in the plot region is accomplished with respect to the user coordinates. For most graphs, text and labels are plotted in the figure margins. One of the coordinate systems used for the figure margins is a combination of the user coordinates and the number of lines of text away from the boundary of the plot region. The coordinate system used for the outer margins uses text away from the boundary of the figure region and the fraction from the minimum toward the maximum of the boundary of the figure region for the respective side. Each graphics device has a number of graphic states that can be queried by typing `par()` at the R prompt. For example, to query R about the current figure margins and outer margins enter

```
> par(c("mar", "oma"))

$mar
[1] 5.1 4.1 4.1 2.1

$oma
[1] 0 0 0 0
```

By default, R produces graphs without outer margins. Outer margins are controlled with the `oma` graphics state setting by a vector that has four values for the four margins specified in the order bottom, left, top, right. From the previous output, one notices that the current values for `oma` are all zero. The default figure margins are 5.1, 4.1, 4.1, and 2.1 lines of text away from the plot region for the bottom, left, top, and right side, respectively. Figure 1.23 on the preceding page shows the default placement of the plot region and figure margins.

R Code 1.68 is used to produce Figure 1.24 on page 88, which illustrates the use of different coordinate systems, as well as changing the default graphics state settings for the figure margins and for the outer margins. The sixth line of code establishes outer margins on all sides of 2.1 lines of text away from the boundary of the figure region and figure margins on all sides of 5.1 lines of text away from the plot region. The code uses the function `paste()`, which converts its arguments to character strings and concatenates them (separating them by the string specified in `sep=`).

R Code 1.68

```
> x <- -4:4; y <- x^2                                # Create two vectors x and y
> # Create figure margins of 5.1 lines for all four sides, create outer
> # margins of 2.1 lines for all sides, set background color to grey90
> par(mar = rep(5, 4) + 0.1, oma = rep(2, 4) + 0.1, bg = "grey90")
> plot(x, y, type = "n")                               # Creates empty plot
> box("outer")                                       # draws outer box
> par(xpd = TRUE)                                     # clipping set to figure region
> rect(-7, -6, 7, 23, col = "grey95")                # large rectangle (figure region)
> box("figure")                                      # draws figure box
> par(xpd = FALSE)                                    # clipping set back to plot region
> rect(-7, -6, 7, 23, col = "grey90")                # large rectangle (plot region)
> box("plot")                                         # draws box around plot region
> axis(side = 1)                                       # add axis to bottom
> axis(side = 2)                                       # add axis to left side
> # label sides
> mtext(text = paste("Side", 1:4), side = 1:4, line = -1.5, cex = 1.5)
> # label outer margins
> mtext(text = paste("Outer Margin", 1:4), side = 1:4, line = 0.5,
+       cex=1.5, outer = TRUE, col="blue")
> # label Figure Region
> mtext(text = "Figure Region", side = 3, at = 4, line = 2, cex = 2,
+       col = "gray40")
> # show Lines -1 to 4 at x = -3 for sides 1 and 3 (bottom & top)
> for(side in c(1, 3))
+   mtext(text = paste("Line", -1:4), side = side, line = -1:4, at = -3,
+         col = "red")
> # show Lines -1 to 4 at y = 2.5 for sides 2 and 4 (left & right)
> for(side in c(2, 4))
+   mtext(text = paste("Line", -1:4), side = side, line = -1:4, at = 2.5,
+         col = "red")
> # show Lines -1 to 1 at 0.8 of y axis in outer margins for all sides
> for(side in 1:4)
+   mtext(text = paste("Line", -1:1), side = side, at = 0.8,
+         line = -1:2, col = "blue", outer = TRUE)
> # "Plot Region" placed at 0, 12
> text(x = 0, y = 12, "Plot Region", cex = 3, col = "gray40")
```

```

> # Red solid circle placed at 0, 4
> points(x = 0, y = 4, pch = 19, col = "red", cex = 2)
> # "(0, 4)" placed centered (0.5) and 2 lines below 0, 4
> text(x = 0, y = 4, "(0, 4)", adj = c(0.5, 2), cex = 1.5)
> # text with math symbols created with expression placed at (0, 7.5)
> text(x = 0, y = 7.5, expression(paste("Some Math: ", bar(x)==
+     sum(frac(x[i], n), i==1, n))), cex = 1.5, col = "blue")
> # "x-label" placed at side 1 (bottom) line 3 of figure margin
> mtext(text = "x-label", side = 1, line = 3)
> # "subtitle" placed at side 1 (bottom) line 4 of figure margin
> mtext(text = "subtitle", side = 1, line = 4)
> # "y-label" placed at side 3 (top) line 3 of figure margin
> mtext(text = "y-label", side = 2, line = 3)
> # "Title" placed at side 3 (top) line 2 of figure margin
> mtext(text = "Title", side = 3, line = 2, cex = 1.25)
> # "0%" placed at side 3 (top) line 0.5 at x = 0 of outer margin
> mtext(text = "0%", side = 3, line = 0.5, outer = TRUE, at = 0)
> # "100%" placed at side 3 (top) line 0.5 at x = 1 of outer margin
> mtext(text = "100%", side = 3, line = 0.5, outer = TRUE, at = 1)
> # "(line = 2, at = 5)" placed at side 1 (bottom) line 2 of figure margin
> mtext(text = "(line = 2, at = 4.5)", side = 1, line = 2, at = 4.5)
> # "(line = 0.5, at = 0.2)" placed at x = 0.2 on side 1 (bottom) line 0.5
> # of the outer margin
> mtext(text = "(line = 0.5, at = 0.2)", side = 1, line = 0.5, at = 0.2,
+       outer = TRUE)

```

When writing text inside the plot region, one uses the function `text()`, and when writing text in a margin, one uses the function `mtext()`. The default value for the `outer=` argument of `mtext()` is `FALSE`, which means that text will only be placed in the figure margins. By changing the `outer = TRUE` argument to `TRUE`, one is able to place text in the outer margins. When using the `outer = TRUE` argument to `mtext()`, text placement is partially determined by the value passed to the argument `at=`, which is a number between 0 and 1 that indicates the percent from the minimum toward the maximum of the boundary of the figure region for the respective side. When using the default argument of `outer = FALSE` with `mtext()`, text placement is still partially controlled with the `at=` argument, but the coordinate system used in this case is the user coordinate system, that is the axes values of the plot region. The other controlling argument for text placement is the `line=` argument, which determines the number of lines away from the plot region when `outer = FALSE` or the number of lines away from the boundary of the figure region when `outer = TRUE`.

R Code 1.69 illustrates the use of different plotting symbols, different colors, and different character expansion (`cex`) values and can be used to create a graph similar to Figure 1.25 on page 89. Color names can be used with a `col=` specification in graphics functions. Numbers or names of colors can be assigned to `col=` as vectors. To see a list of R's 657 named colors enter `colors()` at the R prompt. To learn more about R colors, including how to specify a color with systems such as `rgb`, `hcl`, `hsv`, and so on, enter `?colors` at the R prompt.

R Code 1.69

```

> # figure margins of 2.2, 2.2, 0.2, and 0.2 lines
> par(mar=c(2, 2, 0, 0) + 0.2)
> plot(x = 1, y = 1, xlim = c(1, 16), ylim = c(-1.5, 5), type = "n",
+       xlab = "", ylab = "") # create empty plot with x and y axes

```

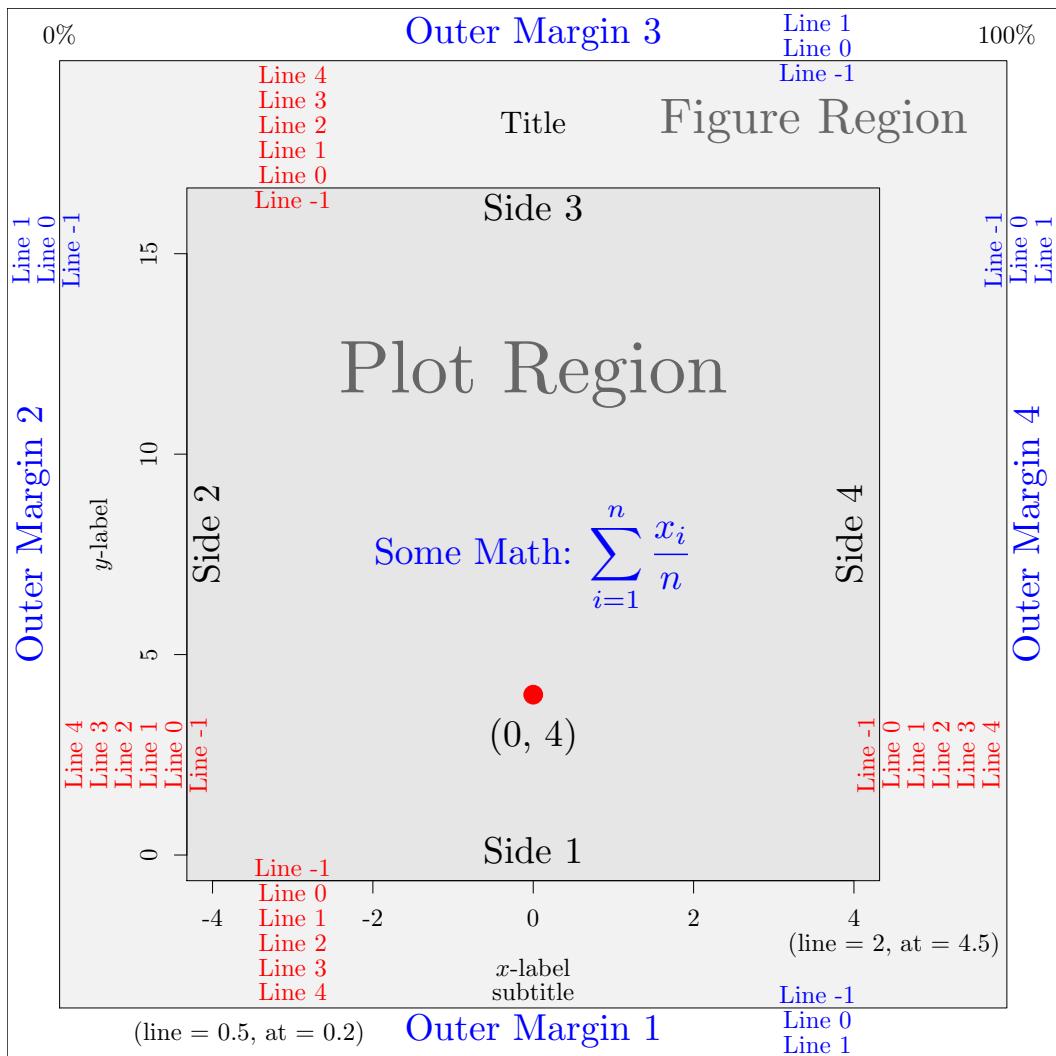


FIGURE 1.24: Graph depicting how text, mathematics, and symbols are placed in the various regions of a traditional graph

```
> COLORS <- c("black", "red", "green", "darkblue", "darkgreen",
+           "magenta", "orange", "cyan") # vector of colors
> # symbols (pch = 0:7) placed at (1, 4), (3, 4), ... (15, 4) with
> # character expansion 1:8 with color specified in COLORS
> points(x = seq(1, 15, 2), y = rep(4, 8), cex = 1:8, col = COLORS,
+          pch = 0:7, lwd = 2)
> # labels 0:7 placed at (1, 2), (3, 2), ..., (15, 2) with
> # character expansion 1:8 with color specified in COLORS
> text(x = seq(1, 15, 2), y = rep(2, 8), labels = paste(0:7), cex = 1:8,
+        col = COLORS)
> # symbols (pch = 8:15) placed at (1, 0), (3, 0), ..., (15, 0)
> # with character expansion of 2
> points(x = seq(1, 15, 2), y = rep(0, 8), pch = 8:15, cex = 2)
```

```
> # labels 8:15 placed 0.7 to the right of (1, 0), (3, 0), ..., (15, 0)
> # with character expansion of 2
> text(x = seq(1, 15, 2) + 0.7, y = rep(0, 8), labels = paste(8:15),
+       cex = 2)
> # symbols (pch = 16:23) placed at (1, -1), (3, -1), ..., (15, -1)
> # with character expansion of 2
> points(x = seq(1, 15, 2), y = rep(-1, 8), pch = 16:23, cex = 2)
> # labels 16:23 placed 0.7 to the right of (1, -1), (3, -1), ..., (15, -1)
> # with character expansion of 2
> text(x = seq(1, 15, 2) + 0.7, y = rep(-1, 8), labels = paste(16:23),
+       cex = 2)
```

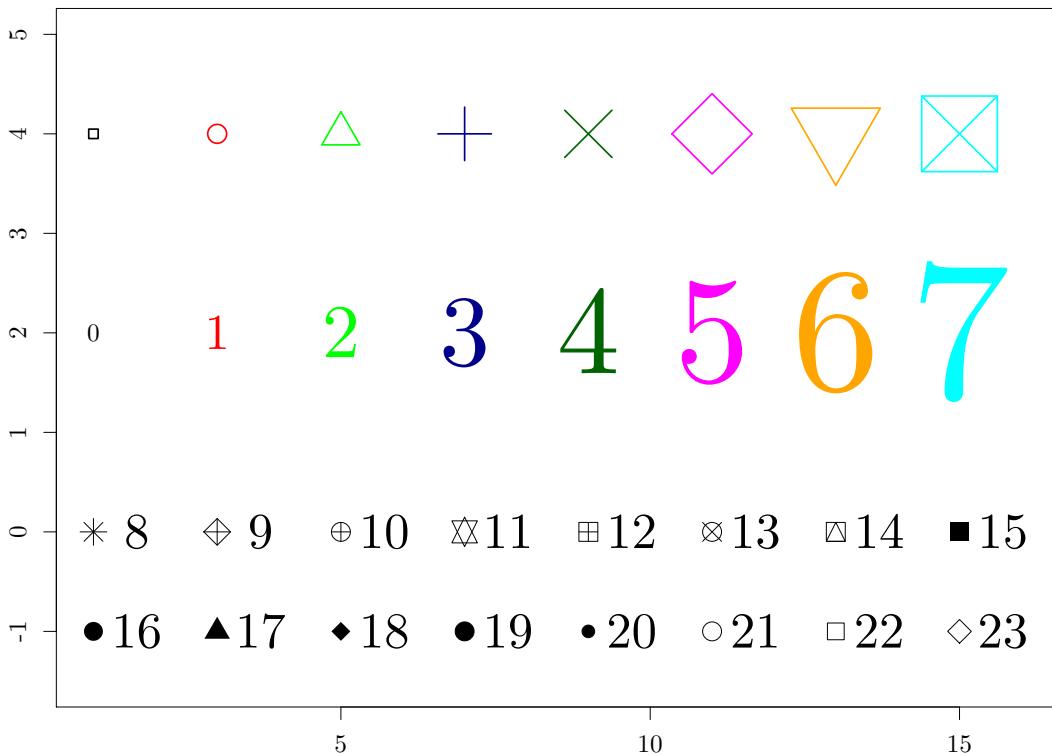


FIGURE 1.25: The numbers in the second row correspond to the plotting symbol directly above them in the first row. The different plotting symbols in the first row and their corresponding numbers in the second row also reflect a character expansion of 1 through 8. The plotting symbols in the third and fourth rows have their corresponding numbers printed to the right.

R typically opens a screen device (window) using default parameters. If no device is open, using a high-level function will cause a device to be opened. The plotting device that is opened is platform specific. That is, `windows()` opens a screen device for Microsoft-Windows, while `quartz()` opens a screen device for Mac OS X. Some devices can be opened on all operating systems such as `postscript()` and `pdf()`. For a complete list of devices, enter `?Devices` at the R prompt.

Some characteristics, such as the width, height, and background color of the graphic, can be set when the user explicitly opens a plotting device. Once a device is open, graphical parameters can be changed with a call to the `par()` function. When changing values in `par()`, a standard approach is to make a copy of the current graphic state settings, make the desired changes to the state settings, to create graphics on the current device, and finally to restore the original settings. R Code 1.70 demonstrates this standard approach.

R Code 1.70

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mar=c(2, 4, 2, 1), lty = 3, bg = "gray", las = 3, ...) # par changes
> plot(x) # plot x with changed parameters
> par(opar) # restore original settings
```

The parameter changes made to an open device remain in effect until the device is closed or the user issues commands to restore the original default parameters. Closing a window with graphics or using `dev.off()` shuts down the specified (by default current) device. To close multiple devices, use `graphics.off()`.

1.21 Problems

1. Calculate the following numerical results to three decimal places with R:
 - (a) $(7 - 8) + 5^3 - 5 \div 6 + \sqrt{62}$
 - (b) $\ln 3 + \sqrt{2} \sin(\pi) - e^3$
 - (c) $2 \times (5 + 3) - \sqrt{6} + 9^2$
 - (d) $\ln(5) - \exp(2) + 2^3$
 - (e) $(9 \div 2) \times 4 - \sqrt{10} + \ln(6) - \exp(1)$
2. Create a vector named `countby5` that is a sequence of 5 to 100 in steps of 5.
3. Create a vector named `Treatment` with the entries “Treatment One” appearing 20 times, “Treatment Two” appearing 18 times, and “Treatment Three” appearing 22 times.
4. Provide the missing values in `rep(seq(__, __, __), __)` to create the sequence 20, 15, 15, 10, 10, 5, 5, 5, 5.
5. Vectors, sequences, and logical operators
 - (a) Assign the names `x` and `y` to the values 5 and 7, respectively. Find x^y and assign the result to `z`. What is the value stored in `z`?
 - (b) Create the vectors `u = (1, 2, 5, 4)` and `v = (2, 2, 1, 1)` using the `c()` function.
 - (c) Provide R code to find which component of `u` is equal to 5.
 - (d) Provide R code to give the components of `v` greater than or equal to 2.
 - (e) Find the product `u × v`. How does R perform the operation?
 - (f) Explain what R does when two vectors of unequal length are multiplied together. Specifically, what is `u × c(u, v)`?
 - (g) Provide R code to define a sequence from 1 to 10 called `G` and subsequently to select the first three components of `G`.
 - (h) Use R to define a sequence from 1 to 30 named `J` with an increment of 2 and subsequently to choose the first, third, and eighth values of `J`.
 - (i) Calculate the scalar product (dot product) of $q = (3, 0, 1, 6)$ by $r = (1, 0, 2, 4)$.
 - (j) Define the matrix `X` whose rows are the `u` and `v` vectors from part (b).
 - (k) Define the matrix `Y` whose columns are the `u` and `v` vectors from part (b).
 - (l) Find the matrix product of `X` by `Y` and name it `W`.
 - (m) Provide R code that computes the inverse matrix of `W` and the transpose of that inverse.

6. How many of the apartments in the **VIT2005** data frame, part of the PASWR2 package, have a **totalprice** greater than €400,000 and also have a garage? Use a single line of R code to determine the answer.

7. Wheat harvested surface in Spain in 2004: Figure 1.26, made with R, depicts the autonomous communities in Spain. The Wheat Table that follows gives the wheat harvested surfaces in 2004 by autonomous communities in Spain measured in hectares. Provide R code to answer all the questions.



FIGURE 1.26: Autonomous communities in Spain

Wheat Table			
community	wheat.surface	community	wheat.surface
Galicia	18817	Castilla y León	619858
Asturias	65	Madrid	13118
Cantabria	440	Castilla-La Mancha	263424
País Vasco	25143	C. Valenciana	6111
Navarra	66326	Región de Murcia	9500
La Rioja	34214	Extremadura	143250
Aragón	311479	Andalucía	558292
Cataluña	74206	Islas Canarias	100
Islas Baleares	7203		

- (a) Create the variables **community** and **wheat.surface** from the Wheat Table in this problem. Store both variables in a **data.frame** named **wheatspain**.
- (b) Find the maximum, the minimum, and the range for the variable **wheat.surface**.
- (c) Which community has the largest harvested wheat surface?

- (d) Sort the autonomous communities by harvested surface in ascending order.
- (e) Sort the autonomous communities by harvested surfaces in descending order.
- (f) Create a new file called `wheat.c` where `Asturias` has been removed.
- (g) Add `Asturias` back to the file `wheat.c`.
- (h) Create in `wheat.c` a new variable called `acre` indicating the harvested surface in acres (1 acre = 0.40468564224 hectares).
- (i) What is the total harvested surface in hectares and in acres in Spain in 2004?
- (j) Define in `wheat.c` the `row.names()` using the names of the communities. Remove the `community` variable from `wheat.c`.
- (k) What percent of the autonomous communities have a harvested wheat surface greater than the mean wheat surface area?
- (l) Sort `wheat.c` by autonomous communities' names (`row.names()`).
- (m) Determine the communities with less than 40,000 acres of harvested surface and find their total harvested surface in hectares and acres.
- (n) Create a new file called `wheat.sum` where the autonomous communities that have less than 40,000 acres of harvested surface are consolidated into a single category named "less than 40,000" with the results from (m).
- (o) Use the function `dump()` on `wheat.c`, storing the results in a new file named `wheat.txt`. Remove `wheat.c` from your path and check that you can recover it from `wheat.txt`.
- (p) Create a text file called `wheat.dat` from the `wheat.sum` file using the command `write.table()`. Explain the differences between `wheat.txt` and `wheat.dat`.
- (q) Use the command `read.table()` to read the file `wheat.dat`.

8. Access the data from url

<http://www.stat.berkeley.edu/users/statlabs/data/babies.data>

and store the information in an object named `BABIES` using the function `read.table()`. A description of the variables can be found at

<http://www.stat.berkeley.edu/users/statlabs/labs.html>.

These data are a subset from a much larger study dealing with child health and development.

- (a) The variables `bwt`, `gestation`, `parity`, `age`, `height`, `weight`, and `smoke` use values of 999, 999, 9, 99, 99, 999, and 9, respectively, to denote "unknown." R uses `NA` to denote a missing or unavailable value. Recode the missing values in `BABIES`. Hint: use something similar to `BABIES$bwt[BABIES$bwt == 999] = NA`.
- (b) Use the function `na.omit()` to create a "clean" data set that removes subjects if any observations on the subject are "unknown." Store the modified data frame in a data frame named `CLEAN`.
- (c) How many missing values are there for `gestation`, `age`, `height`, `weight`, and `smoke`, respectively? How many rows of `BABIES` have no missing values, one missing value, two missing values, and three missing values, respectively? Note: the number of rows in `CLEAN` should agree with your answer for the number of rows in `BABIES` that have no missing values.

(d) Use the function `complete.cases()` to create a “clean” data set that removes subjects if any observations on the subject are “unknown.” Store the modified data frame in a data frame named `CLEAN2`. Write a line of code that shows all of the values in `CLEAN` are the same as those in `CLEAN2`.

(e) Sort the values in `CLEAN` by `bwt`, `gestation`, and `age`. Store the sorted values in a data frame named `BGA` and show the last six rows.

(f) Store the data frame `CLEAN` in your working directory as a `*.csv` file.

(g) What percent of the women in `CLEAN` are pregnant with their first child (`parity = 0`) and do not smoke?

9. The data frame `WHEATUSA2004` from the `PASWR2` package has the USA wheat harvested crop surfaces in 2004 by states. It has two variables, `states` for the state and `acres` for thousands of acres.

(a) Use the function `row.names()` to define the states as the row names for the data frame `WHEATUSA2004`.

(b) Define a new variable called `ha` for the surface area given in hectares where 1 acre = 0.40468564224 hectares.

(c) Sort the file according to the harvested surface area in acres.

(d) Which states fall in the top 10% of states for harvested surface area?

(e) Save the contents of `WHEATUSA2004` in a new file called `WHEATUSA.txt` in your favorite directory. Then, remove `WHEATUSA2004` from your workspace, and check that the contents of `WHEATUSA2004` can be recovered from `WHEATUSA.txt`.

(f) Use the command `write.table()` to store the contents of `WHEATUSA2004` in a file with the name `WHEATUSA.dat`. Explain the differences between storing `WHEATUSA2004` using `dump()` and using `write.table()`.

(g) Find the total harvested surface area in acres for the bottom 10% of the states.

10. Use the data frame `VIT2005` in the `PASWR2` package, which contains data on the 218 used apartments sold in Vitoria (Spain) in 2005 to answer the following questions. A description of the variables can be obtained from the help file for this data frame.

(a) Create a table of the number of apartments according to the number of garages.

(b) Find the mean of `totalprice` according to the number of garages.

(c) Create a frequency table of apartments using the categories: number of garages and number of elevators.

(d) Find the mean flat price (total price) for each of the cells of the table created in part (c).

(e) What command will select only the apartments having at least one garage?

(f) Define a new file called `data.c` with the apartments that have `category = "3B"` and have an elevator.

- (g) Find the mean of `totalprice` and the mean of `area` using the information in `data.c`.
11. Use the data frame `EPIDURALF` to answer the following questions:
- How many patients have been treated with the `Hamstring Stretch`?
 - What percent of the patients treated with `Hamstring Stretch` were classified as each of `Easy`, `Difficult`, and `Impossible`?
 - What percent of the patients classified as `Easy` to palpate were assigned to the `Traditional Sitting` position?
 - What is the mean weight for each cell in a contingency table created with the variables `Ease` and `Treatment`?
 - What percent of the patients have a body mass index ($BMI = \text{kg}/(\text{cm}/100)^2$) less than 25 and are classified as `Easy` to palpate?
12. The millions of tourists visiting Spain in 2003, 2004, and 2005 according their nationalities are given in the following table:
- | Nationality | 2003 | 2004 | 2005 |
|-------------------|--------|--------|--------|
| German | 9.303 | 9.536 | 9.918 |
| French | 7.959 | 7.736 | 8.875 |
| British | 15.224 | 15.629 | 16.090 |
| American | 0.905 | 0.894 | 0.883 |
| Rest of the world | 17.463 | 18.635 | 20.148 |
- Store the values in this table in a matrix with the name `tourists`.
 - Calculate the totals of the rows.
 - Calculate the totals of the columns.
13. Use a `for` loop to convert a sequence of temperatures (18 to 28 by 2) from degrees Celsius to degrees Fahrenheit.
14. If 1 km = 0.6214 miles, 1 hectare = 2.471 acres, and 1 L = 0.26 gallons, write a function that converts kilometers, hectares, and liters into miles, acres, and gallons, respectively. Use the function to convert 10.2 km, 22.4 hectares, and 13.5 L.
15. Write a function that randomly selects m students to present problems at the board given a total number of n students in the classroom. Assume the students are numbered from one to n . Suppose the class has six students whose first names are Joe, Bill, Mark, Karen, Anne, and Mary. Modify the previous function so that the new function returns the first names of the students randomly selected to present board work. Make sure the code returns a message if the user attempts to select more than 6 students to present problems at the chalk board.
16. Compound interest allows money to grow as it stays with an organization that uses the funds productively. The amount of money, A , one has in a bank after t years at n compoundings annually and at an interest rate i with a single initial deposit P is given by $A = P \left(1 + \frac{i}{n}\right)^{n \cdot t}$. Write a generalized R function to solve the problems.

- (a) How much money will an investor have in the bank after 7 years with an initial deposit of \$10,000 if he can invest at 8% annual interest compounded monthly?
- (b) How much money will an investor have in the bank after 10 years with an initial deposit of \$5,000 if he can invest at 6% annual interest compounded monthly?
17. Compound interest allows money to grow as it stays with an organization that uses the funds productively. The amount of money, A , one has in a bank after t years at n compoundings annually and at an interest rate i with a single initial deposit P is given by $A = P \left(1 + \frac{i}{n}\right)^{n \cdot t}$. Write a generalized R function to solve the problems.
- (a) If John needs to have \$3000 to pay for a trip in 3 years, how much should he deposit today if he can invest at 4% interest compounded daily?
- (b) Fred received an inheritance of \$9000 today. If he can invest it at 10% compounded semi-annually for 15 years, how much will he have at the end of that time?
18. A common way to save money is to deposit a certain amount at regular intervals for a period of time. The amount of money available at the end of the time is given by $A = \frac{R \left[(1 + \frac{i}{n})^{n \cdot t} - 1 \right]}{\frac{i}{n}}$ where R is the regular payment, i is the annual interest rate, and n is the number of compoundings per year. Write a generalized R function to solve the problems.
- (a) If George starts a job in his twenties and has 40 years to save a certain amount from each paycheck at 5% compounded monthly to save \$1,000,000 before he retires, how much should he save each month?
- (b) If George waits until he only has 20 years to save, how much must he deposit each month?
19. A common way to save money is to deposit a certain amount at regular intervals for a period of time. The amount of money available at the end of the time is given by $A = \frac{R \left[(1 + \frac{i}{n})^{n \cdot t} - 1 \right]}{\frac{i}{n}}$ where R is the regular payment, i is the annual interest rate, and n is the number of compoundings per year. Write a generalized R function to solve the problems.
- (a) If Mary invests \$200 at the end of each month for 30 years at 3% compounded monthly, how much will she have at the end of those 30 years?
- (b) By how much should Mary increase her monthly savings if she needs to have \$200,000 in her savings account at the end of 30 years if the interest is 3% compounded monthly?

Chapter 2

Exploring Data

2.1 What Is Statistics?

You may be wondering “What is statistics?”, “Who uses it?”, and “Why do I need to study this material?” Statistics is the process of discovering more about a topic by collecting information and subsequently analyzing that information. In essence, statistics is concerned with methods for collecting, organizing, summarizing, analyzing, and presenting data. Data-laden information is present in virtually every sector of society, and the need to understand our surroundings is a basic human need. More to the point of why you, the reader, might want to study this material can be answered in one of two ways. First, if you are a student, you are required to study this material as part of your major because there are certain topics that are deemed important by your teachers. Second, no matter what your stage in life, you desire to have some modicum of control in decision making and want to learn more about how probability and statistics help people, corporations, and governmental agencies make decisions/policies. Even if your reason for reading this material is because it is required, it is the authors’ fervent hope that your ability to make sound decisions will be strengthened through the material in this book.

2.2 Data

Data, according to *The American Heritage Dictionary*, are “information, especially information organized for analysis or used as the basis for a decision.” A characteristic that is being studied in a statistical problem is called a **variable**. A variable will be either **qualitative** or **quantitative**. When a variable is qualitative, it is essentially defining groups or categories. When the categories have no ordering the variable is called **nominal**. For example, the variable **gender** can take on the values **male** and **female** or the variable **music preference** could have values such as **classical**, **jazz**, **rock**, or **other**. When the categories have a distinct ordering, the variable is called **ordinal**. Such a variable might be **educational level** with values **elementary school**, **high school**, **college graduate**, **graduate**, or **professional school**. Values on a scale can be either interval or ratio. Interval data have interpretable distances, while ratio data have a true zero. A variable that is quantitative (numeric) may be either **discrete** or **continuous**. A discrete variable is a numerical variable that can assume a finite, or at most a countably infinite, number of values. Such variables include the **number** of people arriving at a bank on Thursday, **students** in a class, or **dogs** in the pound. A continuous variable is a numerical variable that can assume an infinite number of values associated with the numbers on an interval of the real number line; for example, the **height** of a tree, the **life** of a light bulb, and the

`weight` of an apple are all continuous random variables. An important distinction between discrete and continuous variables is that discrete variables can take on the same value repeatedly, while continuous variables have few or no repeated values. It is important to be able to distinguish between different types of variables because methods for viewing and summarizing data depend on variable type. More to the point, it will be imperative to distinguish between qualitative (categorical) variables and quantitative (numerical) variables. The **distribution** of a variable specifies the values that the variable assumes and how often these values occur. For a discrete variable, the distribution is often presented as a table that provides distinct categories and how often they occur. For a continuous variable, the distribution is typically represented with a graph such as a histogram or a density plot.

When a data set consists of a single variable, it is called a **univariate** data set. When there are two variables in a data set, it is called a **bivariate** data set; and when there are two or more variables, the data set is called a **multivariate** data set. The next four sections discuss univariate variables before switching to bivariate data in Section 2.7 on page 134.

2.3 Displaying Qualitative Data

Recall that a qualitative variable defines categories or groups. The membership in these categories is summarized with tables and is graphically illustrated with bar plots, dot charts, and pie charts. The construction of tables as well as the construction of bar plots, dot charts, and pie charts are illustrated with qualitative variables in what follows.

2.3.1 Tables

A table that lists the different groups of categorical data and the corresponding frequencies with which they occur is called a **frequency table**. Qualitative information is typically presented in the form of a frequency table. The functions `table()` and `xtabs()` can be used to create various types of tables.

Example 2.1 Suppose the letter grades of an English essay in a small class are A, D, C, D, C, C, C, F, and B. Create both a frequency table showing the counts and a relative frequency table showing the proportions of the various grades.

Solution: First, the character data are read into a vector named `Grades`. Then, the functions `table()` and `xtabs()` are used with `Grades`. Recall that the `formula=` argument for `xtabs()` takes the form `factor1 + factor2`.

```
> Grades <- c("A", "D", "C", "D", "C", "C", "C", "C", "F", "B")
> Grades
[1] "A" "D" "C" "D" "C" "C" "C" "C" "F" "B"
> table(Grades)
Grades
A B C D F
1 1 5 2 1
> xtabs(~Grades)
```

```

Grades
A B C D F
1 1 5 2 1

> table(Grades)/length(Grades)      # Relative frequency table

Grades
  A   B   C   D   F
0.1 0.1 0.5 0.2 0.1

> # or
> prop.table(table(Grades))

Grades
  A   B   C   D   F
0.1 0.1 0.5 0.2 0.1

```

Of course, there is no need to use a computer to create a table for such a small data set; however, tables can be created for much larger data sets with no more work than that required for this small data set.

Example 2.2 The `quine` data frame in the `MASS` package has information on children from Walgett, New South Wales, Australia, who were classified by `Culture`, `Age`, `Sex`, and `Learner` status, as well as the number of `Days` absent from school in a particular school year. Use the functions `table()` and `xtabs()` to create a frequency table for the variable `Age`.

Solution: To gain access to information stored in `MASS`, the package is placed on the search path with the function `library()`:

```

> library(MASS)
> table(quine$Age)  # accessing Age using dollar prefixing

F0 F1 F2 F3
27 46 40 33

> with(data = quine, table(Age))

Age
F0 F1 F2 F3
27 46 40 33

> xtabs(~Age, data = quine)

Age
F0 F1 F2 F3
27 46 40 33

```

The variable `Age` is a type of grade classification with four levels `F0`, `F1`, `F2`, and `F3`. Based on the summarized information from the tables created, one notes that the number of children in ages `F0`, `F1`, `F2`, and `F3` are 27, 46, 40, and 33, respectively.

2.3.2 Barplots

One of the better graphical methods for summarizing categorical data is a **barplot**. Barplots are also known as bar charts or bar graphs. The R function **barplot()** is used to create barplots using a summarized version of the data, often the result of using either **table()** or **xtable()** on a data set. This summarized form of the data can be either frequencies or proportions. Regardless of whether one uses frequencies or proportions, the resulting shape is identical. The scales on the y -axes are different, however. Since the argument **height=** for the **barchart()** function accepts either a vector or a matrix, which it subsequently displays, one will often use a function such as **table()** or **xtabs()** that summarizes the original data in the form of a vector or matrix, which can subsequently be passed to the **height=** argument.

Example 2.3 Construct barplots for the variables **Grades**, used in Example 2.1, and **Age** in the **quine** data frame from the **MASS** package, seen in Example 2.2 on the preceding page, using both frequencies and relative frequencies.

Solution: First, R Code 2.1 reads the current parameters and stores them in the variable **opar**. Next, the device region is split into four smaller regions with the command **par(mfrow = c(2, 2))**. Then, the requested barplots are created with four **barplot()** calls. The results are in Figure 2.1.

R Code 2.1

```
> opar <- par(no.readonly = TRUE) # read in current parameters
> par(mfrow=c(2, 2)) # change parameters
> barplot(xtabs(~ Grades), col = "gray40", xlab = "Grades",
+           ylab = "Frequency")
> barplot(prop.table(xtabs(~ Grades)), col = "gray40", xlab = "Grades",
+           ylab = "Relative Frequency")
> barplot(xtabs(~ Age, data = quine), col = "gray90", xlab = "Age",
+           ylab = "Frequency")
> barplot(prop.table(xtabs(~ Age, data = quine)), col = "gray90",
+           xlab = "Age", ylab = "Relative Frequency")
> par(opar) # reset to original parameters
```

2.3.3 Dot Charts

Dot charts are just as effective as barplots in displaying qualitative data. Dot charts are also called Cleveland dotplots. A dot chart shows the values of the variables of interest (levels of the qualitative variable) as dots in a horizontal display over the range of the data. The function used to create a dot chart is **dotchart(data)**, where **data** is a vector containing frequencies for all the different levels of a variable. The function **dotchart()** requires summarized data. When working with un-summarized data, **table()** or **xtabs()** can be used to create summarized data that can be passed to the function **dotchart()**.

Example 2.4 Construct dot charts to show the total number of days missed by **Age** and the average number of days missed by **Age** using the variables **Days** and **Age** from the **quine** data frame of the **MASS** package, first seen in Example 2.2.

Solution: Before creating any dot charts, the device region is split into two smaller regions with the command **par(mfrow=c(1, 2))**. From the left dot chart in Figure 2.2, one quickly

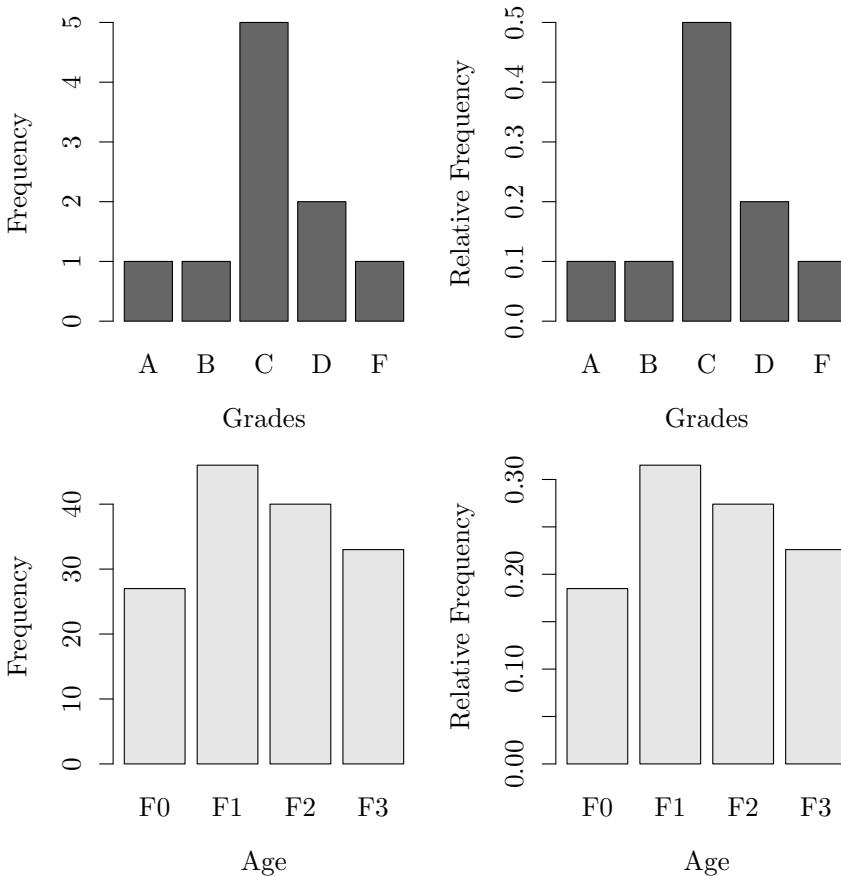


FIGURE 2.1: Graphical representation of the data in `Grades` and `Age` with the function `barplot()`

sees that more Cs were awarded as grades than any other grade. In a similar fashion, the right dotchart of Figure 2.2 shows there are more children in `Age` level F1 than any other level.

```
> opar <- par(no.readonly = TRUE) # read in current parameters
> par(mfrow=c(1, 2))
> dotchart(xtabs(~ Grades), main = "Grades", bg = "gray40",
+           xlim = c(0, 6))
> dotchart(xtabs(~ Age, data = quine), main = "Age", bg = "gray60",
+           xlim = c(25, 50))
> par(opar) # reset to original parameters
```

Example 2.5 Construct dot charts for the variables `Days` and `Age` used in the `quine` data frame from the `MASS` package in Example 2.2 on page 99. Specifically, show the total number of days missed by `Age` and the average number of days missed by `Age`.

Solution: Before creating any dot charts, the device region is split into two smaller regions with the command `par(mfrow=c(1, 2))`. The total days missed for each level of `Age` is

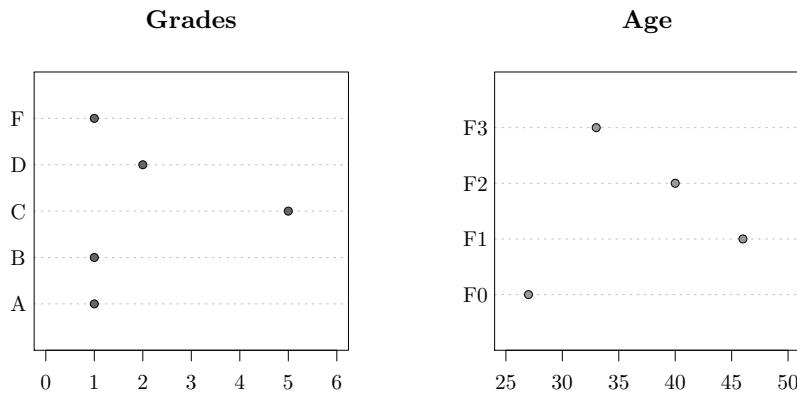


FIGURE 2.2: Graphical representation of the data in `Grades` and `Age` with the function `dotchart()`

stored in the object `TDM`. The function `tapply()` is used to find the average number of days missed for each level of `Age` with the result stored in `ADM` (average days missed).

```
> opar <- par(no.readonly = TRUE)      # read in current parameters
> par(mfrow=c(1, 2))                  # one row two columns
> TDM <- xtabs(Days ~ Age, data = quine)
> dotchart(TDM, bg = "gray40", xlab = "Total Days Missed",
+           xlim = c(400, 900))
> ADM <- with(data = quine, tapply(Days, list(Age), mean))
> dotchart(ADM, xlab = "Average Days Missed", bg = "gray60",
+           xlim = c(10, 22))
> par(opar)                          # reset to original parameters
```

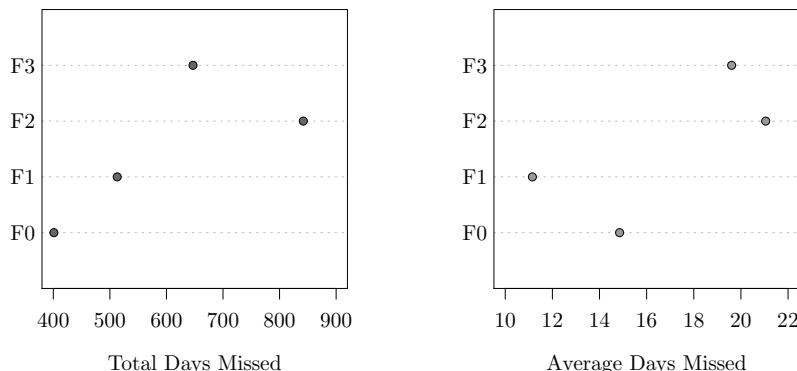


FIGURE 2.3: Dot chart of total days missed by `Age` and average number of days missed by `Age`

2.3.4 Pie Charts

Pie charts represent the relative frequencies or percentages of the levels of a categorical variable with wedges of a pie (circle). While the media often use **pie charts** to display qualitative data, the pie chart has fallen out of favor with most statisticians. Pie charts are most useful when the emphasis is on each category in relation to the total. When such an emphasis is not the primary point of the graph, a bar chart or a dot chart should be used.

Example 2.6 Construct pie charts for the variables `Grades` in Example 2.1 and `Age` from the `quine` data frame in the MASS package used in Example 2.2 on page 99.

Solution: Before creating any pie charts, the device region is split into two regions with the command `par(mfrow=c(1, 2))`. The default title placement for the pie charts is too high, so the function `mtext()` is used to place the titles manually.

```
> opar <- par(no.readonly = TRUE)           # read in current parameters
> par(mfrow = c(1, 2))                      # one row two columns
> GS <- gray(c(0.1, 0.4, 0.7, 0.8, 0.95)) # different grays
> pie(xtabs(~ Grades), radius = 1, col = GS)
> mtext("Grades", side = 3, cex = 1.25, line = 1)
> pie(xtabs(~ Age, data = quine), radius = 1, col = GS)
> mtext("Age", side = 3, cex = 1.25, line = 1)
> par(opar)                                # reset to original parameters
```

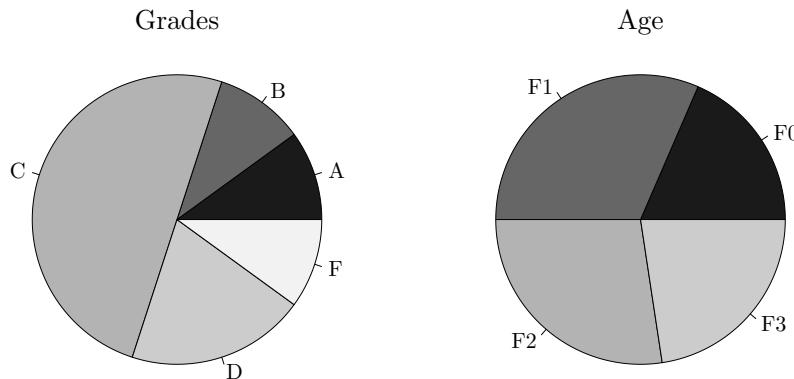


FIGURE 2.4: Graphical representation of the data in `Grades` and `Age` with the function `pie()`

The pie chart for the `Age` does not show the differences between `F1` and `F2` nearly as obviously as either the bar chart (Figure 2.1 on page 101) or the dot chart (Figure 2.3 on the facing page). In fact, with the different intensities on the grey scale, the larger `F1` ceases to be obviously larger than the smaller `F2`.

2.4 Displaying Quantitative Data

When presented with quantitative data, understanding begins with knowing about the data's shape, center, and spread. Some of the more common distribution shapes are shown in Figure 2.5. Of the nine different shapes in Figure 2.5, all are symmetric, with the exception of the second and the eighth graphs, which are characterized as skewed to the right and skewed to the left, respectively. Of the nine different shapes in Figure 2.5, all are unimodal with the exception of the first, the fourth, and the ninth graphs, which are characterized as bimodal, uniform, and multi-modal, respectively. One final highlight: When presented with a symmetric unimodal data set, it will be important to classify the distribution as either short-tailed, long-tailed, or normal. The fourth and the sixth graphs, in addition to being symmetric, are also short-tailed. What follows are graphical tools that can help in assessing the shape, center, and spread of a data set. As a general rule, the shape of the data dictates the most appropriate measures of center and spread for that data set.

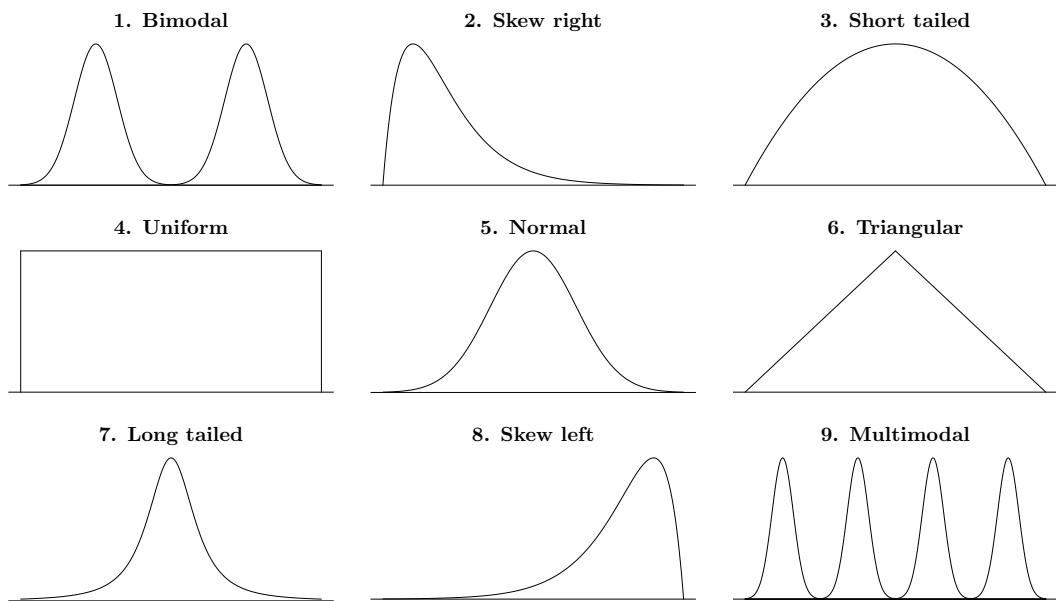


FIGURE 2.5: Nine different graphs labeled according to their shape

2.4.1 Stem-and-Leaf Plots

One way to get a quick impression of the data is to use a **stem-and-leaf plot**. The stem-and-leaf plot is useful for arranging the observations from smallest to largest so that specific positions within a data set can be located. When a stem-and-leaf plot is constructed, each observation is split into a stem and a leaf. Regardless of where the observation is split, the leaf in a stem-and-leaf plot is represented with a single digit. Although it is possible to use a stem-and-leaf plot with a moderately sized data set (more than 100 values), the plot becomes increasingly difficult to read as the number of values plotted increases.

Consequently, it is recommended that stem-and-leaf plots be used graphically to illustrate smallish data sets (less than 100 values). The R command to create a stem-and-leaf plot is `stem(x)`, where `x` is a numeric vector.

Example 2.7 Use the data frame `BABERUTH` to construct a stem-and-leaf plot for the number of home runs (`hr`) Babe Ruth hit while he played for the New York Yankees.

Solution: A quick glance at the data frame `BABERUTH` shows that Babe Ruth played for the New York Yankees for his seventh through twenty-first seasons. The information in `hr` is for Babe Ruth's entire (22 seasons) professional career. To extract the home runs he hit while he was a New York Yankee, use `BABERUTH$hr[BABERUTH$team == "NY-A"]` or `BABERUTH$hr[7:21]` (seventh through twenty-first season home runs) as shown in R Code 2.2.

R Code 2.2

```
> NYYHR <- BABERUTH$hr [BABERUTH$team == "NY-A"]
> NYYHR

[1] 54 59 35 41 46 25 47 60 54 46 49 46 41 34 22

> stem(NYYHR)
```

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

```
2 | 25
3 | 45
4 | 1166679
5 | 449
6 | 0
```

```
> rm(NYYHR) # clean up
```

In this example, see how the stems 2–6 represent the values twenty through sixty and the leaves represent the second digit of the numbers in `hr`. Reading the first row of the stem-and-leaf plot, notice the values 22 and 25. The stem-and-leaf plot reveals a fairly symmetric distribution. 

2.4.2 Strip Charts

An alternative to the stem-and-leaf plot is a **strip chart** (also referred to as a dotplot by many authors). A strip chart plots values along a line. The R function `stripchart()` will stack the tied observations in a column at each value observed along a line that covers the range of the data when given the argument `method = "stack"`. The function requires the data to be a vector, a list of vectors, or a formula of the form `x ~ g`, where values are in a vector `x` and groups are in a vector `g`. Strip charts are often useful for comparing the distribution of a quantitative variable at different qualitative levels (groups).

Example 2.8 Use the data frame `BABERUTH` to

- Construct a strip chart of the number of home runs Babe Ruth hit each season while playing for the New York Yankees.

- (b) Create a strip chart of the number of home runs Babe Ruth hit per season according to the team for which he was playing. Based on the strip chart, when Babe Ruth played, for which team did he generally hit more home runs per season?

Solution: (a) Figure 2.6 is a strip chart of the number of home runs Babe Ruth hit each season while playing for the New York Yankees. The code to construct this graph is shown in R Code 2.3.

R Code 2.3

```
> head(BABERUTH)
  year team g ab r h X2b X3b hr rbi sb bb ba slg
1 1914 Bos-A 5 10 1 2 1 0 0 0 0 0 0 0.200 0.300
2 1915 Bos-A 42 92 16 29 10 1 4 21 0 9 0.315 0.576
3 1916 Bos-A 67 136 18 37 5 3 3 16 0 10 0.272 0.419
4 1917 Bos-A 52 123 14 40 6 3 2 12 0 12 0.325 0.472
5 1918 Bos-A 95 317 50 95 26 11 11 66 6 58 0.300 0.555
6 1919 Bos-A 130 432 103 139 34 12 29 114 7 101 0.322 0.657

> NYYHR <- with(data = BABERUTH, hr[7:21])
> stripchart(NYYHR, xlab = "Home runs per season", method = "stack",
+   main = "Strip chart of home runs while a New York Yankee",
+   pch = 1)
> rm(NYYHR) # clean up
```

Strip chart of home runs while a New York Yankee

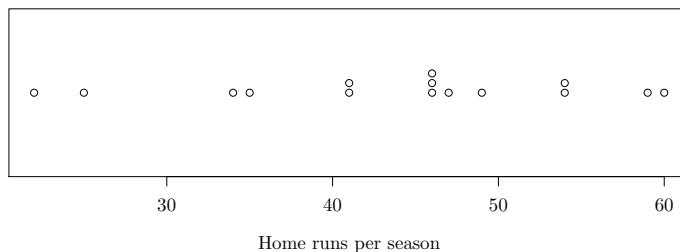


FIGURE 2.6: Strip chart of the number of home runs Babe Ruth hit while playing for the New York Yankees

- (b) Figure 2.7 on the next page, created from R Code 2.4, is a strip chart of the number of home runs Babe Ruth hit per season according to the team for which he was playing. Based on Figure 2.7 on the facing page, Babe Ruth generally hit more home runs per season while playing for NY-A (New York Yankees).

R Code 2.4

```
> opar <- par(no.readonly = TRUE)      # read in current parameters
> par(mfrow = c(1, 2))
> stripchart(hr ~ team, data = BABERUTH, xlab = "Home runs per season",
+   pch = 1, method = "stack")
```

```
> title("Strip chart of home runs \n by team")
> par(las = 1) # Makes labels horizontal
> stripchart(hr ~ team, data = BABERUTH, pch = 19, method = "stack",
+   col = c("gray30","gray50","gray70"), xlab = "Home runs per season",
+   main = "Grayscale strip chart of \n home runs by team")
> par(opar) # reset to original parameters
```

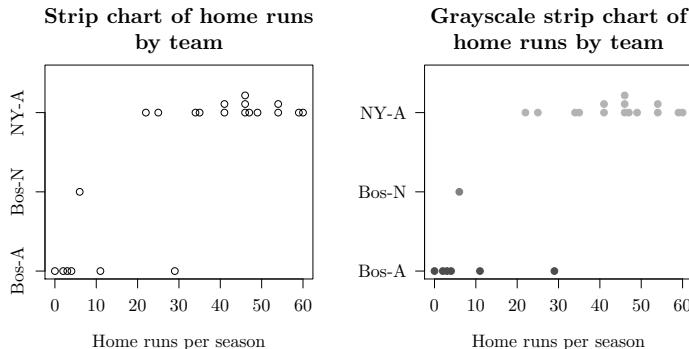


FIGURE 2.7: Strip chart of the number of home runs Babe Ruth hit per season according to the team for which he was playing

2.4.3 Density Curves for Exploring Univariate Data

Knowing the shape of a set of values is crucial for determining the best statistical procedure to apply to the data at hand. One way to estimate the overall shape of a set of values is to use a density curve. A density curve is a function drawn in such a way as to have an area of 1 between the function and the x -axis. When one explores data, generally only the data at hand is available with no additional knowledge about the target distribution. In such cases, the density curve is often approximated using only the data. One well-recognized density estimator is the histogram. All modern statistical software programs, including R, have the ability to produce histograms. Histograms are ubiquitous, seen even in grade school; however, there are a number of challenging questions one must address while constructing histograms, such as:

- How many bins should there be?
- How wide should the class intervals be?

2.4.3.1 Histograms

The **histogram** is a graph that illustrates quantitative (numerical) data. Although the barplot and the histogram look similar, the barplot is used for qualitative data while the histogram is used for numerical data, yet the bins that either the user specifies or those that R uses by default are, in essence, categories. Histograms created in R with the function `hist(x)`, where `x` is a numeric vector, are, by default, frequency histograms. To create density histograms, use the optional argument `freq = FALSE`. A density histogram has a total area of one. For class intervals of equal width h , the histogram density estimate based on a sample of size n is written

$$\hat{f}(x) = \frac{\nu_k}{nh}, \quad t_k < x \leq t_{k+1}, \quad (2.1)$$

where ν_k is the number of sample points in the class interval $(t_k, t_{k+1}]$. Many books define the class interval to be $[t_k, t_{k+1})$, that is closed on the left and open on the right, just the opposite of (2.1). The rationale for using open intervals on the left and closed intervals on the right is to match the defaults R uses when creating histograms. It should be noted that the `hist()` function uses the argument `include.lowest = TRUE` so that the minimum value of a data set is included in the first class interval.

Example 2.9 Construct a histogram that resembles the stem-and leaf plot from Example 2.7 using the `BABERUTH` data.

Solution: The first histogram uses the default arguments for `hist()`. Since the bins R uses are of the form `[]`, the default histogram does not resemble the stem-and-leaf plot. To change the bins to the form `[]`, use the argument `right = FALSE` illustrated in R Code 2.5.

R Code 2.5

```
> opar <- par(no.readonly = TRUE) # read in current parameters
> par(mfrow=c(1, 2)) # one row two columns
> bin <- seq(20, 70, 10) # creating bins 20-70 by 10
> hist(BABERUTH$hr[7:21], breaks = bin, xlab = "Home Runs", col = "pink",
+       main = "Bins of form []")
> hist(BABERUTH$hr[7:21], breaks = bin, right = FALSE, xlab = "Home Runs",
+       col = "pink", main = "Bins of form ()")
> par(opar) # reset to original parameters
```

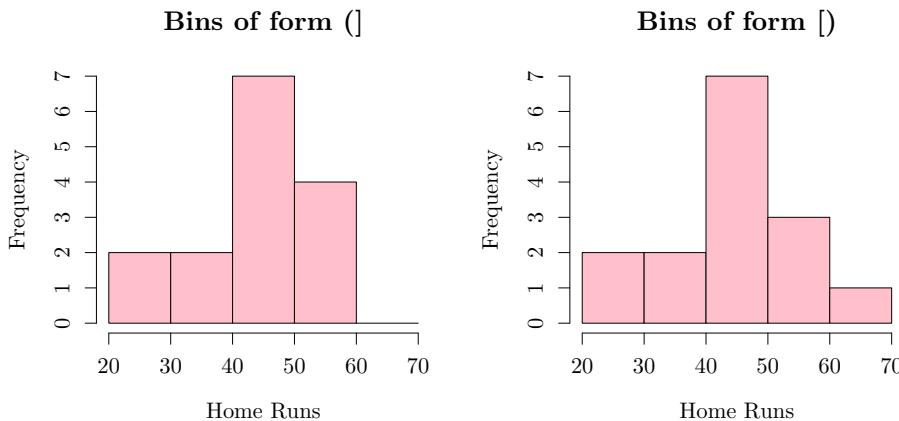


FIGURE 2.8: Histograms created using different bin definitions for the number of home runs hit by Babe Ruth while playing for the New York Yankees

One of the problems with using histograms to describe the shape of the data is the arbitrary nature of the bin width. In Example 2.9, it was seen how simply including or excluding an end point changed the shape of the histogram. If the user does not provide

either a vector of breakpoints, a function to compute the breakpoints, a single number specifying the number of breakpoints, or a character string naming the algorithm to compute the number of class intervals to the argument `breaks`, R will use the default argument `breaks = "Sturges"`. The argument `breaks` will accept three character strings, "Sturges", "FD" / "Freedman-Diaconis", and "Scott". The optimal width for class intervals according to the Sturges algorithm is

$$h_{\text{Sturges}} = \frac{R}{1 + \log_2 n}, \quad (2.2)$$

where R is the range of the sample. The width of the class intervals with Equation (2.2) depends only on sample size. The formula in Equation (2.2) was derived based on an underlying normal population. Consequently, if one uses Equation (2.2) to compute the class interval width of multimodal or skewed data, the resulting histogram will not be a good estimate of the true underlying density function. R does not use the result from Equation (2.2) to compute the class width exactly, but rather adjusts the value of h returned from Equation (2.2) with the `pretty()` function to return 'nice' breakpoints. The optimal width for class intervals according to the Freedman-Diaconis algorithm is

$$h_{\text{FD}} = \frac{2(IQR)}{n^{1/3}}. \quad (2.3)$$

IQR in (2.3) is the interquartile range as defined in Section 2.6.1 on page 130. The optimal width for class intervals according to the Scott algorithm is

$$h_{\text{Scott}} = \frac{2 \cdot 3^{1/3} \cdot \pi^{1/6} \cdot \hat{\sigma}}{n^{1/3}}. \quad (2.4)$$

The value used to estimate the population standard deviation σ , $\hat{\sigma}$, in (2.4) is the sample standard deviation as defined in Section 2.6.3 on page 131. R does not use the result from Equation (2.3) to compute the class width exactly, but rather adjusts the value of h returned from Equation (2.3) with the `pretty()` function to return 'nice' breakpoints. Instead of using $2 \cdot 3^{1/3} \cdot \pi^{1/6} \cdot \hat{\sigma}$ for the numerator of Equation (2.4), R uses the approximation $3.5 \cdot \hat{\sigma}$ in the numerator (see `nclass.scott` for further details):

$$h_{\text{ScottR}} = \frac{3.5 \cdot \hat{\sigma}}{n^{1/3}}. \quad (2.5)$$

As with `breaks = "Sturges"` and `breaks = "FD"`, the argument `breaks = "Scott"` does not use the value of h_{ScottR} directly, rather the function `pretty()` uses the number of desired intervals, determined from h_{ScottR} , to compute 'nice' breakpoints.

Example 2.10 Use the data frame **BABERUTH** to construct histograms for the number of home runs Babe Ruth hit while a New York Yankee using:

1. breakpoints defined by the class width from Equation (2.2),
2. the argument `breaks = "Sturges"`,
3. breakpoints defined by the class width from Equation (2.3),
4. the argument `breaks = "FD"`,
5. breakpoints defined by the class width from Equation (2.4), and
6. the argument `breaks = "Scott"`.

Solution: R Code 2.6 creates breakpoints using the definition of class width from Equation (2.2) and uses those breakpoints to create a histogram shown in the top left of Figure 2.9 on page 113. In the same code chunk, a histogram is created using the argument `breaks = "Sturges"`, which can be seen in the top right of Figure 2.9 on page 113. R Code 2.7 creates breakpoints using the definition of class width from Equation (2.3) and uses those breakpoints to create a histogram shown in the center left of Figure 2.9 on page 113. In the same code chunk, a histogram is created using the argument `breaks = "FD"`, which can be seen in the center right of Figure 2.9 on page 113. R Code 2.8 creates breakpoints using the definition of class width from Equation (2.4) and uses those breakpoints to create a histogram shown in the bottom left of Figure 2.9 on page 113. In the same code chunk, a histogram is created using the argument `breaks = "Scott"`, which can be seen in the bottom right of Figure 2.9 on page 113.

R Code 2.6

```
> xs <- BABERUTH$hr[7:21]           # xs = home runs while NYY
> R <- diff(range(xs))            # R = range of data
> n <- length(xs)                # number of observations in xs
> hs <- R/(1 + log2(n))          # class interval width Sturges
> hs

[1] 7.744212

> nclassS <- ceiling(R/hs)        # number of classes
> nclassS

[1] 5

> bpS <- min(xs) + hs*0:nclassS   # breakpoints using Definition
> bpS

[1] 22.00000 29.74421 37.48842 45.23264 52.97685 60.72106

> sturgesD <- hist(xs, breaks = bpS, main = "Sturges Definition",
+                     xlab = "", col = "pink") # Histogram using Def
> sturgesD$breaks                 # show breakpoints

[1] 22.00000 29.74421 37.48842 45.23264 52.97685 60.72106

> sturgesD$counts                # count in each bin

[1] 2 2 2 5 4

> pretty(xs, n = nclassS)         # breakpoints using pretty
[1] 20 30 40 50 60

> sturgesA <- hist(xs, breaks = "Sturges", main = "Sturges Adjusted",
+                     xlab = "", col = "blue") # Histogram Adjusted
> sturgesA$breaks                 # show adjusted breakpoints

[1] 20 30 40 50 60

> sturgesA$counts                # count in each bin

[1] 2 2 7 4
```

Using the function `range()` on the values in `xs` returns a vector with the minimum and maximum values of `xs`, respectively. By applying the function `diff()`, which returns a one-lag difference (maximum – minimum for this problem), the range is computed and stored in the object `R`. The class width computed with the Sturges algorithm is 7.7442. The required number of classes (5) is determined by taking the largest integer after dividing the range of the data (38) by the class interval width (7.7442). Breakpoints are determined by finding the minimum value in `xs` and then adding multiples, up to the total number of class intervals (5), of width 7.7442 to the minimum value in `xs`. The breakpoints are stored in the object `pbS`. The histogram created from using the breakpoints in `bpS` is stored in the object `sturgesA`. A check of the breakpoints used in the histogram is performed with `sturgesA$breaks`, which shows the actual breakpoints used to create the histogram. The command `sturgesA$counts` shows the number of observations falling in each class interval. When using the argument `breaks = "Sturges"`, the function `pretty()` is used to determine ‘nice’ breakpoints for the resulting histogram based on the desired number of class intervals. This can be seen indirectly by comparing the output from `pretty(xs, n = nclassS)` and the output from `sturgesA$breaks`.

R Code 2.7

The class width computed with the Friedman-Diaconis algorithm is 10.948. The required number of classes (4) is determined by taking the largest integer after dividing the range of the data (38) by the class interval width (10.948). Breakpoints are determined by finding the minimum value in `xs` and then adding multiples, up to the total number of class intervals (4), of width 10.948 to the minimum value in `xs`. The breakpoints are stored in the object `pbFD`. The histogram created from using the breakpoints in `bpFD` is stored in the object `FDdef`. A check of the breakpoints used in the histogram is performed with `FDdef$breaks`, which shows the actual breakpoints used to create the histogram. The command `FDdef$counts` shows the number of observations falling in each class interval. When using the argument `breaks = "FD"`, the function `pretty()` is used to determine ‘nice’ breakpoints for the resulting histogram based on the desired number of class intervals. This can be seen indirectly by comparing the output from `pretty(xs, n = nclassFD)` and the output from `FDadj$breaks`.

R Code 2.8

```
> hsc <- 2*3^(1/3)*pi^(1/6)*sd(xs)/n^(1/3) # class interval width Scott
> hsc
[1] 15.91972

> nclassSC <- ceiling(R/hsc) # number of classes
> nclassSC
[1] 3

> bpSC <- min(xs) + hsc*0:nclassSC # breakpoints using Definition
> scottD <- hist(xs, breaks = bpSC, main = "Scott Definition", xlab = "", 
+                   col = "pink") # Histogram using Definition
> scottD$breaks # show breakpoints
[1] 22.00000 37.91972 53.83944 69.75916

> scottD$counts # count in each bin
[1] 4 7 4

> pretty(xs, n = nclassSC) # breakpoints using pretty
[1] 20 30 40 50 60

> scottA <- hist(xs, breaks = "Scott", main = "Scott Adjusted",
+                  xlab = "", col = "blue") # Histogram Adjusted
> scottA$breaks # show adjusted breakpoints
[1] 20 30 40 50 60

> scottA$counts # count in each bin
[1] 2 2 7 4
```

The class width computed with the Scott algorithm is 15.9197. The required number of classes (3) is determined by taking the largest integer after dividing the range of the data (38) by the class interval width (15.9197). Breakpoints are determined by finding the mini-

mum value in `xs` and then adding multiples, up to the total number of class intervals (3), of width 15.9197 to the minimum value in `xs`. The breakpoints are stored in the object `pbSC`. The histogram created from using the breakpoints in `bpSC` is stored in the object `scottD`. A check of the breakpoints used in the histogram is performed with `scottD$breaks`, which shows the actual breakpoints used to create the histogram. The command `scottD$counts` shows the number of observations falling in each class interval. When using the argument `breaks = "Scott"`, the function `pretty()` is used to determine ‘nice’ breakpoints for the resulting histogram based on the desired number of class intervals. This can be seen indirectly by comparing the output from `pretty(xs, n = nclassSC)` and the output from `scottA$breaks`. In this particular problem, `hist()` uses the same breakpoints, second column of histograms in Figure 2.9, for `breaks = "Sturges"`, `breaks = "FD"`, and `breaks = "Scott"`.

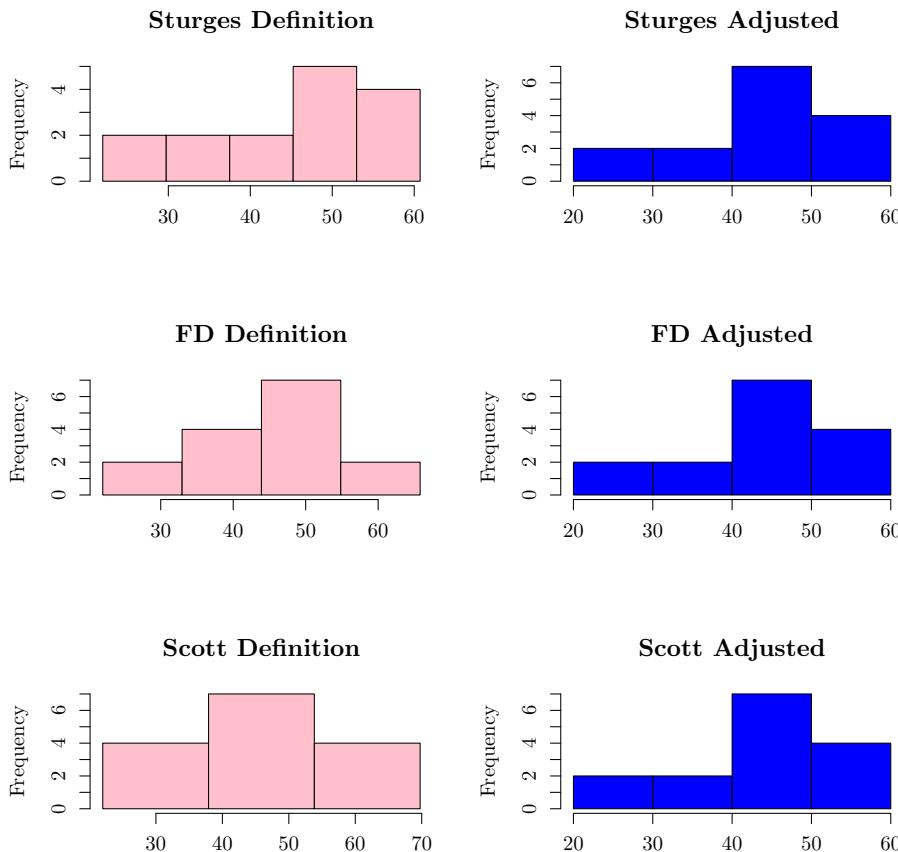


FIGURE 2.9: Histograms in the first column use the Sturges, Freedman-Diaconis, and Scott definitions for computing class width. Histograms in the second column are constructed using the arguments `breaks = "Sturges"`, `breaks = "FD"`, and `breaks = "Scott"` inside the `hist()` function, respectively.



2.4.3.2 Kernel Density Estimators

Selecting different bin widths may alter the perceived shape of the underlying distribution. Consider the differences among the shapes of the histograms in Figure 2.10 produced by altering the bin width in R Code 2.9. The data set used to produce Figure 2.10 is `geyser`, available in the `MASS` package.

R Code 2.9

```
> opar <- par(no.readonly = TRUE)      # read in current parameters
> par(mfrow=c(2, 2))                  # two rows two columns
> attach(geyser)                      # place geyser on search path
> hist(duration, breaks = 3, freq = FALSE, ylim = c(0, 1), col = "pink")
> hist(duration, breaks = 6, freq = FALSE, ylim = c(0, 1), col = "pink")
> hist(duration, breaks = 12, freq = FALSE, ylim = c(0, 1), col = "pink")
> hist(duration, breaks = 24, freq = FALSE, ylim = c(0, 1), col = "pink")
> detach(geyser)                      # remove geyser from search path
> par(opar)                           # reset to original parameters
```

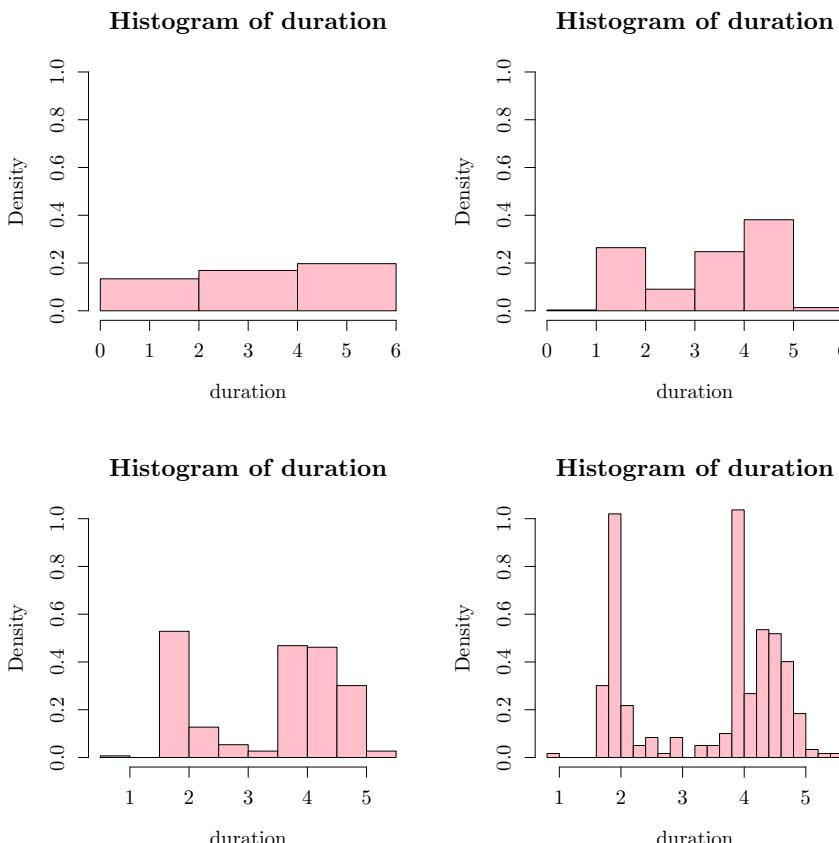


FIGURE 2.10: Histograms created using different bin definitions for the eruption duration of Old Faithful

Table 2.1: Common kernels and their definitions

Name	Definition
Rectangular	$K(x) = \frac{1}{2}, \quad x < 1$
Triangular	$K(x) = 1 - x , \quad x < 1$
Gaussian	$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}, \quad -\infty < x < \infty$

A much better choice to get an idea of what the shape of a distribution looks like is to use a **kernel density estimator**. The R function `density(x)`, where `x` is a numeric vector, can be used to create a kernel density estimate. Basically, a kernel density estimate uses shapes with $\frac{1}{n}$ area added up at each point in the data set to create a graph with area 1. The resulting shape is a kernel density estimate. The result of the kernel density estimate can be viewed with either the `plot()` or `lines()` function. Recall that `plot()` is a high-level function while `lines()` is a low-level function. That is, `plot()` will create a graph while `lines()` will add to an existing graph. A kernel density estimate is a generalization of a density histogram defined as

$$\hat{f}(x) = \frac{1}{hn} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right), \quad (2.6)$$

where $K(\cdot)$ is known as the kernel function and h is the bandwidth or smoothing parameter. Kernel functions are generally symmetric density functions that must satisfy the condition

$$\int_{-\infty}^{\infty} K(x) dx = 1.$$

Three commonly used kernels are the rectangular kernel, the triangular kernel, and the normal or Gaussian kernel shown in Table 2.1 with the functions that define each kernel. Figure 2.11 on the next page displays these kernels for $h = 1$ and $h = 2$. Note that when the bandwidth (h) increases from $h = 1$ to $h = 2$, the corresponding kernels are twice as wide.

R Code 2.10 on the following page is used to construct a triangular kernel density estimate of the two values 0.5 and 1 using a bandwidth of $h = 0.3$, with the result shown Figure 2.12 on page 117. The values 0.5 and 1 are assigned to the variable `xi`. A sequence of values starting at 0, incrementing by 0.1, and ending at 1.5 are stored in the variable `x`. The incrementing value, 0.1, of the sequence is kept relatively large in order to show all of the values. A function named `tri` is created to compute the triangular kernel. The condition $|x| < 1$ is evaluated and returns either a `TRUE` or `FALSE`. When the condition is `TRUE`, the result from applying `tri()` is $1 - |x|$; when the condition is `FALSE`, the result from applying `tri()` is 0. The values under the column labeled `f` in the `RES` object are the estimated heights of the kernel density, $\hat{f}(x)$ obtained by adding across rows of the columns labeled 0.5 and 1. The function `sapply()` is used to obtain Equation (2.6) minus the summation by passing the values in `xi` to the function defined as Equation (2.6) minus the summation, inside the `sapply()` function, and storing the results in the matrix `shapes`. The matrix `shapes` will be of dimension $n_x \times n_{xi}$ in general, where n_x represents the number of values in `x`, and n_{xi} represents the number of values in `xi`, 16 × 2 for this example. The estimated kernel density, $\hat{f}(x)$, is obtained by adding the values for each `x` at each `xi` with the function `apply()`. That is, the rows of the matrix `shapes` are added and the results are stored in

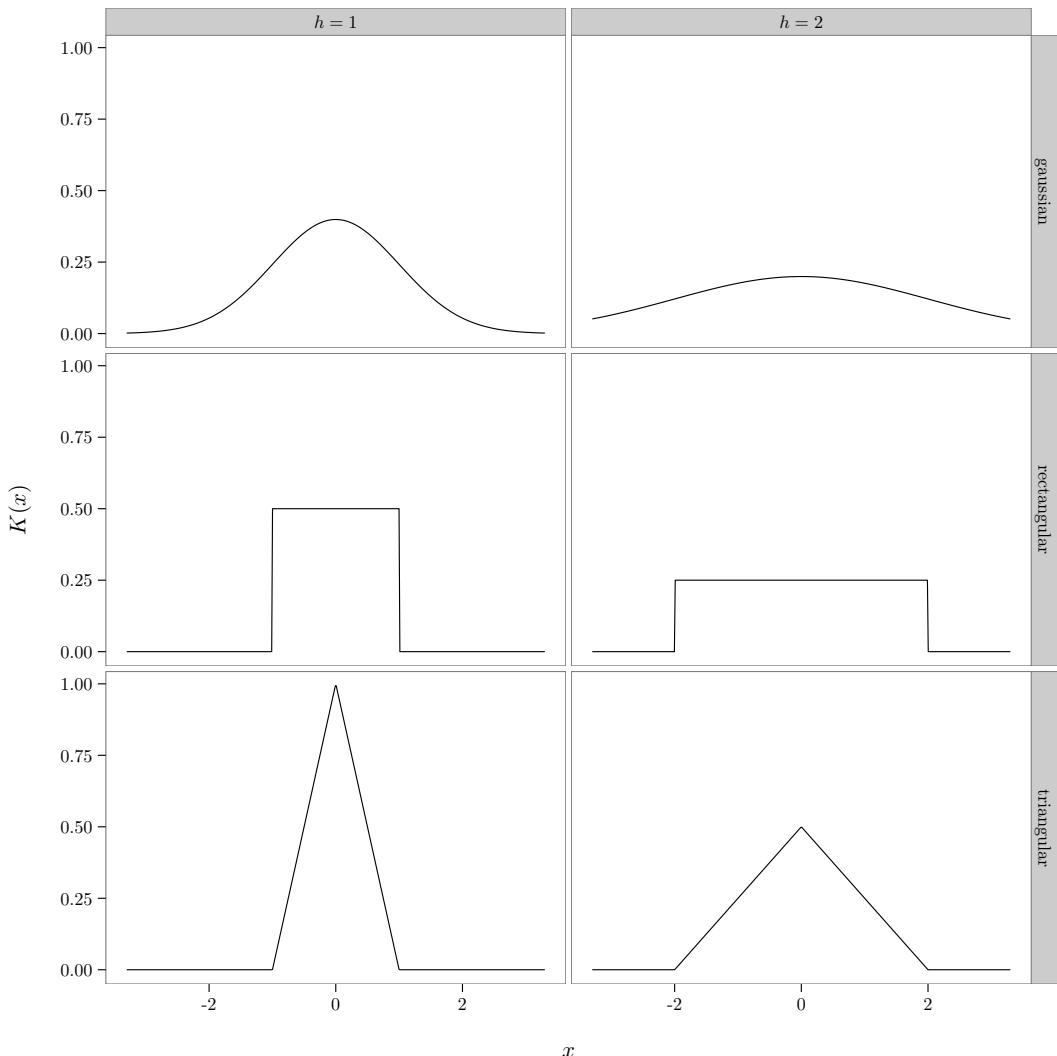


FIGURE 2.11: Three kernels and two bandwidths

the vector `fx`. The final result, visible in Figure 2.12 on the facing page, is created by connecting the points in `x` and `fx` with lines by changing the default argument in `plot()` from points to lines with `type = "l"`.

R Code 2.10

```
> xi <- c(0.5, 1.0)    # assign values to xi
> n <- length(xi)      # number of values
> x <- seq(from = min(xi) - .5, to = max(xi) + .5, by = .1)
> h <- 0.3
> tri <- function(x){(abs(x) < 1)*(1 - abs(x))}
> shapes <- sapply(xi, function(xi){(1/(h*n))*tri((x - xi)/h)})
> fx <- apply(shapes, 1, sum)
> plot(x, fx, type = "l", xlab = "x", ylab = expression(hat(f)(x)))
```

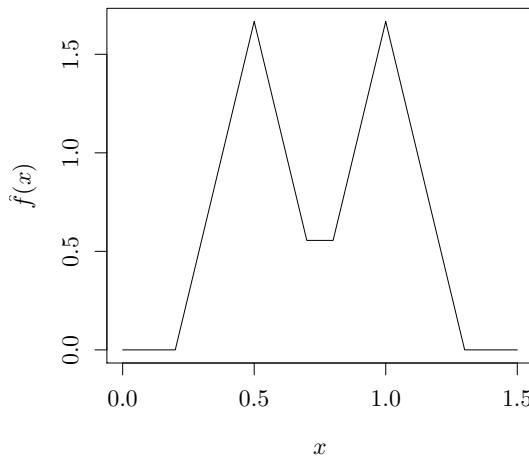


FIGURE 2.12: Triangular kernel density estimate

R Code 2.11

```
> dimnames(shapes) <- list(x, c(xi[1], xi[2])) # list
> RES <- cbind(shapes, f = apply(shapes, 1, sum))
> RES

      0.5      1      f
0  0.0000000 0.000000e+00 0.0000000
0.1 0.0000000 0.000000e+00 0.0000000
0.2 0.0000000 0.000000e+00 0.0000000
0.3 0.5555556 0.000000e+00 0.5555556
0.4 1.1111111 0.000000e+00 1.1111111
0.5 1.6666667 0.000000e+00 1.6666667
0.6 1.1111111 0.000000e+00 1.1111111
0.7 0.5555556 3.700743e-16 0.5555556
0.8 0.0000000 5.555556e-01 0.5555556
0.9 0.0000000 1.111111e+00 1.1111111
1  0.0000000 1.666667e+00 1.6666667
1.1 0.0000000 1.111111e+00 1.1111111
1.2 0.0000000 5.555556e-01 0.5555556
1.3 0.0000000 0.000000e+00 0.0000000
1.4 0.0000000 0.000000e+00 0.0000000
1.5 0.0000000 0.000000e+00 0.0000000
```

The matrix `shapes` is given row and column names in R Code 2.11. The row names are the values of the values in x and the column names are the values in x_i . The estimated kernel density, labeled `f`, is column bound to `shapes` and stored in the object `RES`. Plotting the values in x versus x_i for each i results in individual triangles (`shapes`). The next example shows both the final estimated kernel density and the individual shapes used to create the final estimate.

Example 2.11 Construct a Gaussian kernel density estimate for the values 2.1, 2.2, 2.3, 2.4, 2.6, 2.7, 3.2, 3.3, 3.6, and 3.7. Show both the final Gaussian kernel density estimate and the individual shapes used to obtain the kernel density estimate in the same graph.

Use a bandwidth of $h = 0.3$ and 800 evenly spaced points from 1.1 to 4.7 for x .

Solution: In first line of R Code 2.12, the given values are read into `xi`. The 800 evenly spaced points are generated with the function `seq()`. A function `gauss` is created and used to create the values in `shapes`. The resulting `shapes` matrix has dimensions 800×10 . After plotting the estimated kernel density estimate with the function `plot()`, the function `rug()` is used to make vertical tick marks in the plot for each value in `xi`. Finally, the individual shapes (normals) are added to the plot with the function `lines()`, which joins the points in `x` and `shapes` for each of the values in `xi`. The resulting graph is shown in Figure 2.13.

R Code 2.12

```
> xi <- c(2.1, 2.2, 2.3, 2.4, 2.6, 2.7, 3.2, 3.3, 3.6, 3.7)
> n <- length(xi)
> x <- seq(from = min(xi) - 1, to = max(xi) + 1, length.out = 800)
> h <- 0.3
> gauss <- function(x){1/sqrt(2*pi)*exp(-(x^2)/2)}
> shapes <- sapply(xi, function(xi){(1/(h*n))*gauss((x - xi)/h)})
> plot(x, apply(shapes, 1, sum), type = "l", ylab = "", lwd = 3)
> rug(xi, lwd = 2)
> apply(shapes, 2, function(b){lines(x, b)})
```

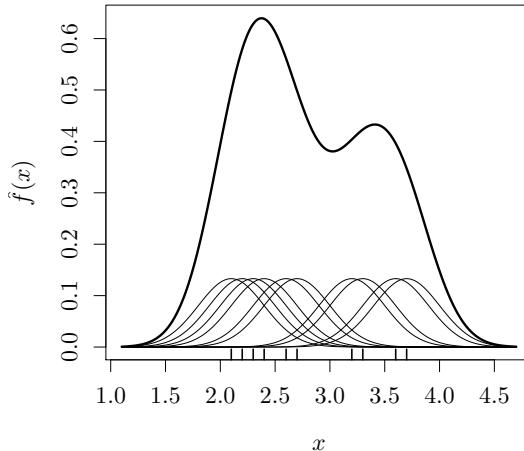


FIGURE 2.13: Gaussian kernel density estimate



Example 2.11 on the preceding page was given to illustrate how Equation (2.6) works by adding shapes to obtain $\hat{f}(x)$. The function `density()` is much more sophisticated than the code given in Example 2.11 on the previous page; however, the resulting kernel density estimator is essentially the same. Bandwidth is automatically estimated with `density()` unless the argument `bw=` is specified. The function `density()` will accept seven definitions for the `kernel=` argument: `gaussian`, `epanechnikov`, `rectangular`, `triangular`, `biweight`, `cosine`, and `optcosine`. The default kernel for `density()` is `gaussian`. For further details of the `density()` function, one should consult the `density` help file.

Example 2.12 Construct a density histogram of the waiting time until the next eruption using the data frame `geyser` available in the MASS package. Superimpose a Gaussian kernel density estimate over the density histogram. In the same graph, show the kernel density estimate without showing the density histogram.

Solution: Note that to superimpose a kernel density estimate over a histogram, the histogram must be a density histogram. Recall that density histograms are produced with the optional argument `freq = FALSE`. R Code 2.13 is used to construct Figure 2.14.

R Code 2.13

```
> opar <- par(no.readonly = TRUE) # read in current parameters
> par(mfrow=c(1, 2)) # make device region 1 by 2
> with(data = geyser,
+       hist(waiting, freq = FALSE, col="grey95")
+       ) # density histogram
> with(data = geyser,
+       lines(density(waiting), col="red", lwd=2)
+       ) # superimpose kernel density estimate over histogram
> with(data = geyser,
+       plot(density(waiting), col="red", lwd=2,
+             main="Density of waiting")
+       ) # create kernel density estimate by itself
> par(opar) # reset to original parameters
```

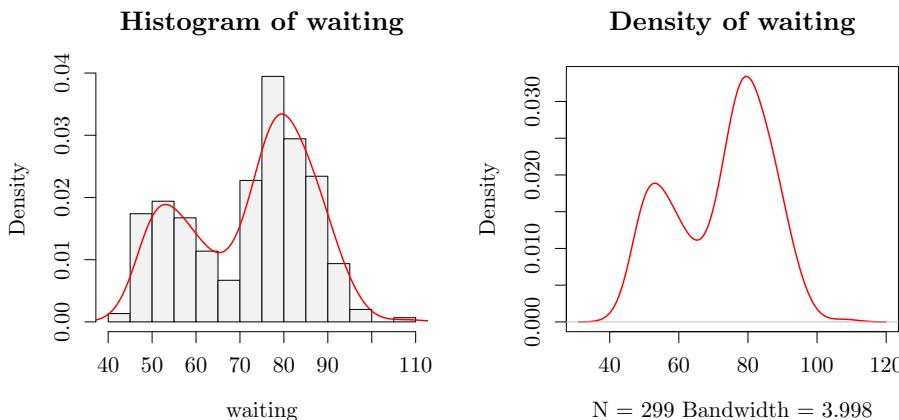


FIGURE 2.14: Histogram of waiting time between Old Faithful eruptions with superimposed Gaussian kernel density estimate as well as the kernel density estimate by itself

Based on the kernel density estimate, it appears there are two modes for waiting time until the next eruption. It seems one will usually have to wait close to either 50 or 80 minutes until the next eruption. ■

2.5 Summary Measures of Location

One of the main objectives of statistics is to make inference to a population based on information obtained from a sample. Since it can be overwhelming to work with the entire population and/or sample, summary measures are introduced to help characterize the data at hand. These summary measures may apply to either the population or to the sample. Numerical summaries of the population are called **parameters**, while numerical summaries of the sample are called **statistics**. More formal definitions of both parameters and statistics will be given later. Measures of central location are introduced first. The measures covered are generally familiar to the reader from everyday usage. Specifically, the **mean**, the **trimmed mean**, and the **median** are introduced. Other measures of location addressed include quartiles, hinges, and quantiles.

2.5.1 The Mean

The most common measure of center is the average, which locates the balance point of the distribution or data. The mean is an appropriate measure of center for symmetric distributions; however, it is not appropriate for skewed distributions. In statistics, the average of a sample is called the **sample mean** and is denoted by \bar{x} . Given some numeric data x_1, x_2, \dots, x_n , the sample mean is defined as

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \sum_{i=1}^n \frac{x_i}{n} \quad (2.7)$$

The R function `mean(x)` will compute the mean of a data vector `x`. Additional arguments of `mean(x)` include `na.rm = TRUE`, for removal of missing values, and `trim=`, to compute a trimmed mean. The trimmed mean is generally used to estimate the center when working with long-tailed distributions. When a $p\%$ trimmed mean is computed, $p\%$ of the sorted data is deleted from each end of the distribution, and a mean is computed from the remaining values. When $p \times n$ is not an integer, the integer portion, ($\lfloor p \times n \rfloor$), should be deleted from each end of the sorted values and the mean computed from the remaining values.

Example 2.13 Compute the mean number of home runs per season Babe Ruth hit while playing for the New York Yankees. Compute a 5%, a 10%, a 15%, and a 50% trimmed mean for the number of home runs per season Babe Ruth hit while playing for the New York Yankees using the information stored in the data frame **BABERUTH**.

Solution: In Example 2.7 on page 105, the variable `NYYHR` was created that contained the number of home runs Babe Ruth made while playing for the New York Yankees. If `NYYHR` is no longer available, recreate it with the command `NYYHR <- hr[7:21]` once the data frame **BABERUTH** is on the search path. Since there are 15 values in `NYYHR`, to compute 5%, 10%, 15%, and 50% trimmed means, $\lfloor 0.05 \times 15 \rfloor = \lfloor 0.75 \rfloor = 0$, $\lfloor 0.10 \times 15 \rfloor = \lfloor 1.5 \rfloor = 1$, $\lfloor 0.15 \times 15 \rfloor = \lfloor 2.25 \rfloor = 2$, and $\lfloor 0.50 \times 15 \rfloor = \lfloor 7.5 \rfloor = 7$ values, respectively, will need to be deleted from the sorted values of `NYYHR` before computing means on the remaining values. A second solution is also presented using the function `mean()` using the `trim=` argument:

```
> NYYHR <- with(data = BABERUTH, hr[7:21])
> NYYHR
[1] 54 59 35 41 46 25 47 60 54 46 49 46 41 34 22
```

```

> SNYYHR <- sort(NYYHR)
> SNYYHR

[1] 22 25 34 35 41 41 46 46 46 47 49 54 54 59 60

> p.05 <- floor(0.05 * 15)
> p.10 <- floor(0.1 * 15)
> p.15 <- floor(0.15 * 15)
> p.50 <- floor(0.5 * 15)
> num.to.delete <- c(p.05, p.10, p.15, p.50)
> num.to.delete

[1] 0 1 2 7

> m.05 <- mean(SNYYHR[(1 + p.05):(15 - p.05)])
> m.10 <- mean(SNYYHR[(1 + p.10):(15 - p.10)])
> m.15 <- mean(SNYYHR[(1 + p.15):(15 - p.15)])
> m.50 <- mean(SNYYHR[(1 + p.50):(15 - p.50)])
> t.m <- c(m.05, m.10, m.15, m.50)
> names(t.m) <- c("5%tmean", "10%tmean", "15%tmean", "50%tmean")
> t.m

5%tmean 10%tmean 15%tmean 50%tmean
43.93333 44.38462 44.81818 46.00000

> tm.05 <- mean(NYYHR, trim = 0.05)
> tm.10 <- mean(NYYHR, trim = 0.1)
> tm.15 <- mean(NYYHR, trim = 0.15)
> tm.50 <- mean(NYYHR, trim = 0.5)
> tms <- c(tm.05, tm.10, tm.15, tm.50)
> names(tms) <- c("5%tmean", "10%tmean", "15%tmean", "50%tmean")
> tms

5%tmean 10%tmean 15%tmean 50%tmean
43.93333 44.38462 44.81818 46.00000

```

The trimmed means are all fairly similar, confirming a rather symmetric distribution. Note that the 50% trimmed mean is the value in the middle of the sorted observations. This value is also known as the median. ■

2.5.2 The Median

While the mean is the most commonly encountered measure of center, it is not always the best measure of center. The **sample median** is the middle value of a distribution of numbers, denoted by the letter m . Since the median ignores the information in surrounding values, it is more resistant to extreme fluctuations in the data than is the mean. When working with skewed distributions, the median is the most appropriate measure of center. The sample median, m , of x_1, x_2, \dots, x_n is the $(\frac{n+1}{2})^{\text{st}}$ observation of the sorted values. When n is odd, $\frac{n+1}{2}$ is an integer, and finding the observation is straightforward. When n is even, an average of the two middle observations is taken to find the median. When the values x_1, x_2, \dots, x_n are sorted, they are called **order statistics** and denoted as $x_{(1)}, x_{(2)}, \dots, x_{(n)}$.

A more concise definition of the sample median is then

$$m = \begin{cases} x_{(k+1)} & n = 2k + 1 \text{ (odd)}, \\ \frac{1}{2}(x_{(k)} + x_{(k+1)}) & n = 2k \text{ (even)}. \end{cases} \quad (2.8)$$

To find the sample median with R, use the function `median(x)`, where `x` is a numeric vector.

Example 2.14 \triangleright **Means and Medians** \triangleleft The numerical grades achieved by three students on four exams during the course of a semester are recorded in Table 2.2. Compute means and medians for the students. Which student would you characterize as overconfident? Which was a procrastinator? Which was consistent?

Table 2.2: Student test scores

	Test1	Test2	Test3	Test4
Student1	73	75	74	74
Student2	95	94	12	95
Student3	66	67	63	100

Solution: First, the students' exam scores are read into individual vectors denoted `Student1`, `Student2`, and `Student3`. The function `median()` is used first to find the median test score for each student. It is possible to compute the mean test score for each student in a similar fashion to that used to find the median test score for each student; however, another solution is provided by using the functions `rbind()`, `cbind()`, and `apply()` in R Code 2.14.

R Code 2.14

```
> Student1 <- c(73, 75, 74, 74)
> Student2 <- c(95, 94, 12, 95)
> Student3 <- c(66, 67, 63, 100)
> median(Student1)

[1] 74

> median(Student2)

[1] 94.5

> median(Student3)

[1] 66.5

> SM <- rbind(Student1, Student2, Student3) # combine rows
> colnames(SM) <- c("Test1", "Test2", "Test3", "Test4")
> SM

      Test1 Test2 Test3 Test4
Student1    73    75    74    74
Student2    95    94    12    95
Student3    66    67    63   100
```

```

> means <- apply(SM, 1, mean)                      # mean of rows
> medians <- apply(SM, 1, median)                  # median of rows
> TOT <- cbind(SM, means, medians)                 # combine columns
> TOT

      Test1 Test2 Test3 Test4 means medians
Student1    73    75    74    74    74    74.0
Student2    95    94    12    95    74    94.5
Student3    66    67    63   100    74    66.5

```

As seen in the output, the mean test score for the three students is 74. One possible characterization of the three students might be: Student 1: is consistent; Student 2: is overconfident; and Student 3: is a procrastinator. Would the mean or the median be the better representative in assigning their final grades? One may want to consider using the median instead of the mean because it is a more robust estimator of center as seen in this example.

2.5.3 Mode

The mode of a sample is the most frequently occurring value in the sample. In the event there are ties in the sample for the most frequently occurring values, there are as many modes as there are ties. The mode is not sensitive to extreme data and is the only measure of location discussed that makes sense to use with categorical data. The mode of a continuous distribution is the peak of the density curve. When distributions have a single mode, they are said to be unimodal. Likewise, if a distribution has two modes or more than two modes, it is referred to as bimodal or multimodal, respectively.

Example 2.15 In Example 2.1, the letter grades of an English essay in a small class were given as A, D, C, D, C, C, C, F, and B. What is the mode of the letter grades?

Solution: Since the most frequently occurring letter grade is a C, the mode of the letter grades is C.

```

> Grades <- c("A", "D", "C", "D", "C", "C", "C", "C", "F", "B")
> table(Grades)

Grades
A B C D F
1 1 5 2 1

> names(which.max(table(Grades)))

[1] "C"

```

Example 2.16 Use the data framed **VIT2005** and find the mode of the variable **totalprice**.

Solution: A density plot is created with R Code 2.15 and shown in Figure 2.15 on the next page. R Code 2.16 finds the mode of the density curve given in Figure 2.15.

R Code 2.15

```
> plot(density(VIT2005$totalprice), main = "")
```

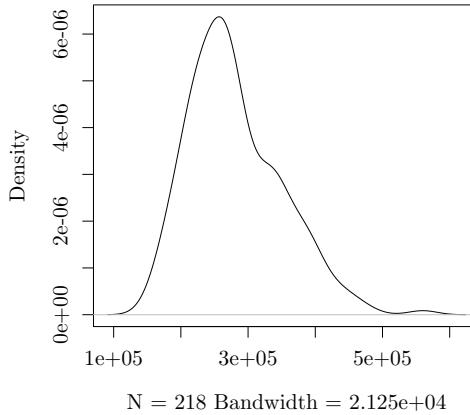


FIGURE 2.15: Density plot of `totalprice`

R Code 2.16

```
> DV <- density(VIT2005$totalprice)
> yval <- max(DV$y)
> ID <- which(DV$y == yval)
> MODE <- DV$x[ID]
> MODE
[1] 256944.5
```

The density plot is unimodal with a mode of €256944.4885.



2.5.4 Quantiles

The p^{th} quantile, $0 \leq p \leq 1$, of a distribution is the value x_p such that $\mathbb{P}(X \leq x_p) \geq p$ and $\mathbb{P}(X \geq x_p) \geq 1 - p$. For discrete data, there are often many values of x_p that satisfy the definition of the p^{th} quantile. In this book, the definition used by R to compute quantiles will be used. R defines the p^{th} quantile of a distribution to be the $(p(n - 1) + 1)^{\text{st}}$ order statistic. When $p(n - 1) + 1$ is not an integer, linear interpolation is used between order statistics to arrive at the p^{th} quantile. Given values $x_{(1)}, x_{(2)}, \dots, x_{(n)}$, the p^{th} quantile for the k^{th} order statistic, $p(k)$, is

$$p(k) = \frac{(k - 1)}{(n - 1)}, \quad k \leq n. \quad (2.9)$$

By this definition, it is seen that the 50% quantile (50th percentile) is the median since

$$0.50 = \frac{k - 1}{n - 1} \Rightarrow k = \frac{n + 1}{2},$$

which, by definition, is the location of the order statistic that is the median. Other definitions for quantiles exist and are used in other texts and other statistical software packages; however, the definition used here is consistent with the default algorithm used in R for computing quantiles. To read about alternative algorithms for computing quantiles with R, type `?quantile` at the R prompt. To compute the quantiles of a data set stored in a vector \mathbf{x} , use the function `quantile(x)`. By default, the function `quantile(x)` returns the 0%, 25%, 50%, 75%, and 100% quantiles of the data vector \mathbf{x} . The p^{th} quantile is the same thing as the $(p \times 100)^{\text{th}}$ percentile. That is, percentiles and quantiles measure the same thing; however, percentiles use a scale from 0 to 100 instead of the 0 to 1 scale used by quantiles.

Just as the sample median is the value that divides the sample into equal halves, the **sample quartiles** can be thought of as the values that divide the sample into quarters. The first, second, and third sample quartiles are denoted as Q_1 , Q_2 , and Q_3 , respectively, and are (by default) computed with the R function `quantile(x)`. To compute other quantiles, use the argument `probs=` to specify either a single value or to pass a vector of values to the `quantile()` function.

Example 2.17 Compute Q_1 , Q_2 , and Q_3 for the values $x_{(1)} = 1$, $x_{(2)} = 4$, $x_{(3)} = 7$, $x_{(4)} = 9$, $x_{(5)} = 10$, $x_{(6)} = 14$, $x_{(7)} = 15$, $x_{(8)} = 16$, $x_{(9)} = 20$, and $x_{(10)} = 21$.

Solution: First, the order statistics for the 0.25, 0.50, and 0.75 quantiles are computed using (2.9):

$$\begin{array}{l} .25 = \frac{k-1}{10-1} \\ k = 3.25, \end{array} \quad \begin{array}{l} .50 = \frac{k-1}{10-1} \\ k = 5.50, \text{ and} \end{array} \quad \begin{array}{l} .75 = \frac{k-1}{10-1} \\ k = 7.75. \end{array}$$

Linear interpolation is then used on the order statistics to find the requested quantiles/quartiles. Specifically, since Q_1 , Q_2 , and Q_3 occur at the 3.25, 5.50, and 7.75 order statistics, 0.25 of the distance between the third and fourth order statistics is added to the third order statistic to arrive at Q_1 . Likewise, 0.50 of the distance between the fifth and sixth order statistics is added to the fifth order statistic to compute Q_2 . Finally, 0.75 of the distance between the seventh and eighth order statistics is added to the seventh order statistic to compute Q_3 :

$$\begin{aligned}
 Q_1 &= x_{(3)} + .25(x_{(4)} - x_{(3)}) & Q_2 &= x_{(5)} + .50(x_{(6)} - x_{(5)}) \\
 &= 7 + .25(9 - 7) & &= 10 + .5(14 - 10) \\
 &= 7.50, & &= 12.00, \text{ and}
 \end{aligned}$$

$$\begin{aligned}
 Q_3 &= x_{(7)} + .75(x_{(8)} - x_{(7)}) \\
 &= 15 + .75(16 - 15) \\
 &= 15.75.
 \end{aligned}$$

Code to compute the requested quartiles according to the quantile definition is provided in R Code 2.17. Subsequently, the function `quantile()` is used to compute the same quartiles/quartiles.

R Code 2.17

```
[1] 3.25 5.50 7.75

> Q1 <- x[3] + 0.25*(x[4] - x[3])      # linear interpolation
> Q2 <- x[5] + 0.50*(x[6] - x[5])      # linear interpolation
> Q3 <- x[7] + 0.75*(x[8] - x[7])      # linear interpolation
> QU <- c(Q1, Q2, Q3)
> names(QU) <- c("Q1", "Q2", "Q3")
> QU                                         # quartiles

  Q1     Q2     Q3
7.50 12.00 15.75

> quantile(x, probs=c(0.25, 0.5, 0.75))  # the easy way!
25%    50%    75%
7.50 12.00 15.75
```



2.5.5 Hinges and the Five-Number Summary

An alternative method to calculating quartiles is to compute **hinges**. The idea behind both quartiles and hinges is to split the data into fourths. When a computer is not available, hinges are somewhat easier to compute by hand than are quartiles. The lower and upper hinges are the $x_{(j)}$ and $x_{(n-j+1)}$ order statistics, where

$$j = \frac{\lfloor \frac{n+1}{2} \rfloor + 1}{2}. \quad (2.10)$$

In short, the lower hinge is the median of the lower half of the data and the upper hinge is the median of the upper half of the data. Lower and upper hinges can be different from quartiles. For example, consider Example 2.17 on the preceding page where the locations of the first, and third quartiles were found to be at the 3.25th and 7.75th order statistics. However, since

$$\frac{\lfloor \frac{n+1}{2} \rfloor + 1}{2} = \frac{\lfloor \frac{10+1}{2} \rfloor + 1}{2} = 3,$$

the locations for the lower and upper hinges are at the 3rd, $x_{(3)}$, and 8th, $x_{(n-3+1)} = x_{(10-3+1)} = x_{(8)}$, order statistics.

Hinges are typically returned as part of the **five-number summary**. A five-number summary for a data set consists of the smallest value, the lower hinge, the median, the upper hinge, and the largest value, all of which are computed with R's function **fivenum()**.

Example 2.18 Compute the 0.25, 0.50, and 0.75 quantiles as well as a five-number summary for the number of runs batted in (RBIs) by Babe Ruth while he played for the New York Yankees. The variable **rbi** in the data frame **BABERUTH** contains the RBIs per season for Babe Ruth over his professional baseball career.

Solution: The quartiles and hinges are first computed by their definitions. Subsequently, the function **quantile()** and the function **fivenum()** are used to obtain the same results:

```

> NYYRBI <- with(data=BABERUTH, rbi[7:21]) # Extract RBIs while a NYY
> SNYYRBI <- sort(NYYRBI)
> p <- c(0.25, 0.50, 0.75)
> n <- length(NYYRBI)
> order.stat <- p*(n - 1) + 1
> order.stat

[1] 4.5 8.0 11.5

> Q1 <- SNYYRBI[4] + 0.5*(SNYYRBI[5] - SNYYRBI[4])
> Q2 <- SNYYRBI[8]
> Q3 <- SNYYRBI[11] + 0.5*(SNYYRBI[12] - SNYYRBI[11])
> QU <- c(Q1, Q2, Q3)
> names(QU) <- c("Q1", "Q2", "Q3")
> QU

    Q1      Q2      Q3
112.0 137.0 153.5

> quantile(NYYRBI, probs=c(0.25, 0.50, 0.75))

 25%    50%    75%
112.0 137.0 153.5

> j <- (floor((n + 1)/2) + 1)/2                      # Number to count in
> j

[1] 4.5

> lower.hinge <- SNYYRBI[4] + 0.5*(SNYYRBI[5] - SNYYRBI[4])
> upper.hinge <- SNYYRBI[11] + 0.5*(SNYYRBI[12] - SNYYRBI[11])
> small <- min(NYYRBI)
> large <- max(NYYRBI)
> five.numbers <- c(small, lower.hinge, Q2, upper.hinge, large)
> five.numbers

[1] 66.0 112.0 137.0 153.5 171.0

> fivenum(NYYRBI)

[1] 66.0 112.0 137.0 153.5 171.0

```

In this particular example, the first and third quartile are equal to the lower and upper hinge, respectively.

2.5.6 Boxplots

A popular method of representing the information in a five-number summary is the **boxplot**. To show spread, a box is drawn from the lower hinge (H_L) to the upper hinge (H_U) with a vertical line drawn through the box to indicate the median or second quartile (Q_2). A “whisker” is drawn from H_U to the largest data value that does not exceed the upper fence. This value is called the **adjacent value**. The upper fence is defined as

$Fence_U = H_U + 1.5 \times H_{spread}$, where $H_{spread} = H_U - H_L$. A whisker is also drawn from H_L to the smallest value that is larger than the lower fence, where the lower fence is defined as $Fence_L = H_L - 1.5 \times H_{spread}$. Any value smaller than the lower fence or larger than the upper fence is considered an **outlier** and is generally depicted with a hollow circle. Figure 2.16 illustrates a boxplot for the variable `fat` from the data frame `BODYFAT`. Boxplots are useful for detecting skewness, finding outliers, and comparing two or more variables that are all measured on the same scale; however, a boxplot will not detect multi-modality.

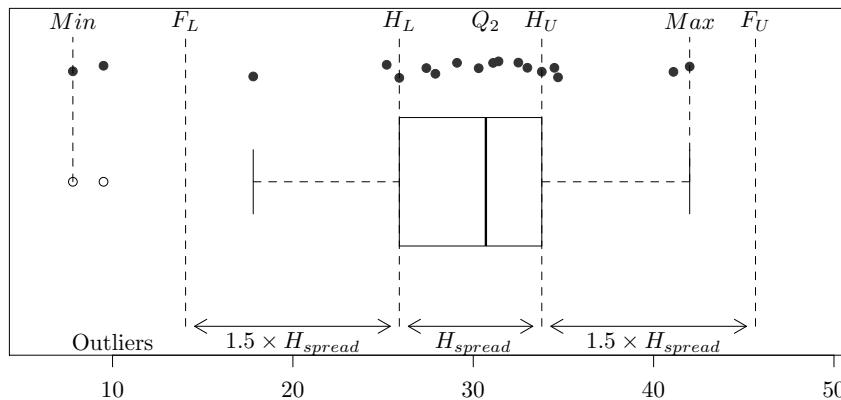


FIGURE 2.16: Graph depicting the five-number summary in relationship to original data and the boxplot

To create a boxplot, use the command `boxplot()`. By default, boxplots have a vertical orientation. To create a horizontal boxplot, use the optional argument `horizontal = TRUE`. Common arguments for `boxplot()` include `col=` to set the box color and `notch = TRUE` to add a notch to the box to highlight the median.

Example 2.19 Use the data frame `Cars93` in the `MASS` package to create a boxplot of the variable `Min.Price`. Use the `text()` function to label the five-number summary values in the boxplot.

Solution: The final boxplot created from R Code 2.18 is shown in Figure 2.17 on the facing page. Additionally, the labels in R contain mathematical notation. To learn more about R's ability to plot mathematical expressions, type `?mathplot` at the R prompt. The function `text()` uses the argument `labels =`, which takes a character vector and draws the items specified in the character vector at x and y coordinates specified by the `x =` and `y =` arguments, which accept numeric vectors of coordinates indicating where the labels should be written. In this particular problem, the function `locator()` was used to determine x coordinates in the boxplot since the boxplot created with `boxplot()` does not show a scale for the x -axis.

R Code 2.18

```
> library(MASS)                                     # load MASS package
> opar <- par(no.readonly = TRUE)                  # read in current parameters
> par(mar=c(0.1, 7.1, 0.1, 0.1))                # more space needed on side 2
> with(data = Cars93, boxplot(Min.Price, ylim = c(0, 50),
+     ylab = "Minimum Price (in \\$1000)\n for basic version",
```

```

+     col = "springgreen3"))
> f <- with(data=Cars93, fivenum(Min.Price)) # store fivenum values in f
> text(x = rep(1.25, 5), y = f, labels=c("Min", expression(H[L]),
+   expression(Q[2]), expression(H[U]), "Max"), pos=4)
> par(opar)                                     # reset to original parameters

```

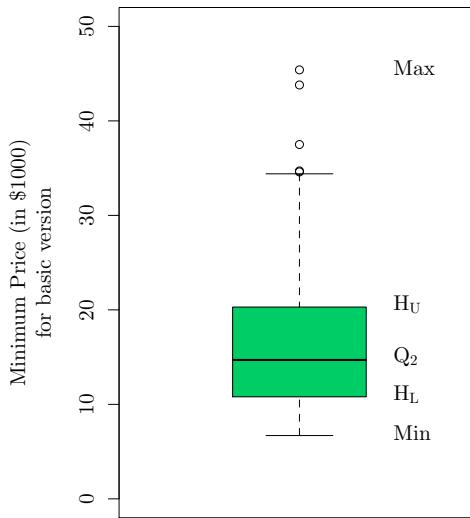


FIGURE 2.17: Boxplot of car prices with five-number summaries labeled



Example 2.20 Use the data frame `BODYFAT` to create side-by-side boxplots of the variable `fat` according to `sex`.

Solution: Side-by-side boxplots created from R Code 2.19 are shown in Figure 2.18 on the next page. The labels for the levels of `sex` were changed from F and M to Female and Male, respectively.

R Code 2.19

```

> BODYFAT$sex <- factor(BODYFAT$sex, labels = c("Female", "Male"))
> boxplot(fat ~ sex, data = BODYFAT, col = c("gray80", "gray20"),
+   ylab = "Percent Bodyfat")

```



2.6 Summary Measures of Spread

Summary measures of center such as the mean and median are important because they describe “typical” values in a data set; however, it is possible to have two data sets with the

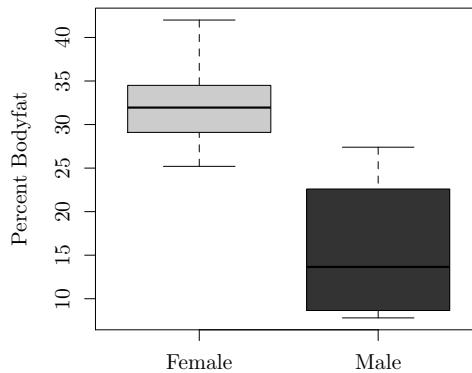


FIGURE 2.18: Side-by-side boxplots of bodyfat percentage

same means (Example 2.14 on page 122) and/or medians while still having different spreads. For this reason, it is important to measure not only typical values but also the spread of the values in a distribution in order to describe the distribution fully. There are many ways to measure spread, some of which include range, interquartile range, and variance.

2.6.1 Range

The easiest measure of spread to compute is the range. At times, the range refers to the difference between the smallest value in a data set and the largest value in the data set. Other times, the range refers to the smallest and largest values of a data set as a pair. The function `range(x)` returns the smallest and largest values in `x`. If the distance between the largest and smallest value is desired, one can use `diff(range(x))`:

```
> range(1:10)
[1] 1 10
> diff(range(1:10))
[1] 9
```

2.6.2 Interquartile Range

Instead of looking at the entire range of the data, looking at the middle 50% will often prove to be a useful measure of spread, especially when the data are skewed. The interquartile range (*IQR*) is defined as $IQR = Q_3 - Q_1$ and can be found with the function `IQR()`:

```
> quantile(1:10)
 0%   25%   50%   75% 100%
1.00  3.25  5.50  7.75 10.00
> IQR(1:10)
[1] 4.5
```

2.6.3 Variance

The **sample variance**, s^2 , can be thought of as the average squared distance of the sample values from the sample mean. It is not quite the average because the quantity is divided by $n - 1$ instead of n in the formula

$$s^2 = \sum_{i=1}^n \frac{(x_i - \bar{x})^2}{n-1}. \quad (2.11)$$

When the positive square root of the sample variance is taken, the **sample standard deviation**, s , results. It is often preferable to report the sample standard deviation instead of the variance since the units of measurement for the sample standard deviation are the same as those of the individual data points in the sample. To compute the variance, use the function `var(x)`. One could compute the standard deviation by taking the square root of the variance `sqrt(var(x))` or use the built-in function to do so (`sd()`). The standard deviation is an appropriate measure of spread for unimodal symmetric distributions. Consider how the variance and standard deviation are computed with the values 1 through 5 (stored in `x`) in R Code 2.20.

R Code 2.20

```
> x <- 1:5
> n <- length(x)
> mean.x <- mean(x)
> mean.x
[1] 3
> x-mean.x
[1] -2 -1  0  1  2
> (x-mean.x)^2
[1] 4 1 0 1 4
> NUM <- sum((x-mean.x)^2)    # numerator of s^2 hard way
> NUM
[1] 10
> DEN <- n-1                  # denominator of s^2
> DEN
[1] 4
> VAR <- NUM/DEN             # variance hard way
> VAR
[1] 2.5
> var(x)                      # variance easy way
[1] 2.5
> SD <- sqrt(VAR)            # standard deviation hard way
> SD
[1] 1.581139
> sd(x)                      # standard deviation with R
[1] 1.581139
```

An interesting function that will return different results depending on the class of the object to which it is applied is the function `summary()`. When the object is a numeric vector, as is the case with `x`, six summary statistics are returned: the minimum, the first quartile, the median, the mean, the third quartile, and the maximum:

```
> summary(x)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1	2	3	3	4	5

To view what `summary()` returns when applied to objects of class `aov`, `lm`, and `glht` see R Code 11.2 on page 708, 12.2 on page 794, and 12.14 on page 819, respectively.

2.6.4 Sample Coefficient of Variation

The sample coefficient of variation is a unitless measure that can be used to compare data sets with different units or very different means. It is computed as the ratio of the sample standard deviation to the sample mean and is denoted as

$$\widehat{CV} = \frac{S}{\bar{X}}. \quad (2.12)$$

This ratio lets the user gauge the variability of a sample in terms of the mean. Samples where the mean is larger than the sample standard deviation will result in \widehat{CV} values less than 1. The coefficient of variation is known in many settings and can be used to obtain a better estimate of the mean. Since the definition of the \widehat{CV} allows negative and undefined values, a useful \widehat{CV} can generally be obtained only with non-negative observations.

Some fields use a measure very similar to \widehat{CV} known as the relative standard deviation. The relative standard deviation is the absolute value of the sample coefficient of variation multiplied by 100 to render a percentage.

$$\widehat{RSD} = \left| \widehat{CV} \right| \times 100 = \left| \frac{S}{\bar{X}} \right| \times 100. \quad (2.13)$$

The interpretation of the \widehat{RSD} is similar to \widehat{CV} , as both measures form a ratio of the standard deviation to the mean.

Example 2.21 Forty interarrival times for vehicles traveling on the M1 motorway in England are reported in the data frame `SDS4`. Compute the mean interarrival time in seconds and in minutes. Show that the sample coefficient of variation is the same regardless of whether one uses seconds or minutes for the interarrival times.

Solution: The interarrival times are converted to minutes by dividing the values in `times` by 60.

```
> meanSEC <- mean(SDS4$times)
> meanMIN <- mean(SDS4$times/60)
> c(meanSEC, meanMIN)
```

[1] 7.80 0.13

The mean interarrival time is 7.8 seconds or 0.13 minutes. It is not appropriate to compare values that use different units of measurement; however, by using the sample coefficient of variation, the result is unitless.

```
> sdSEC <- sd(SDS4$times)
> sdMIN <- sd(SDS4$times/60)
> c(sdSEC, sdMIN)

[1] 7.871402 0.131190

> CVsec <- meanSEC/sdSEC
> CVmin <- meanMIN/sdMIN
> c(CVsec, CVmin)

[1] 0.9909289 0.9909289
```

In this particular problem, the \widehat{CV} , regardless of the unit of measurement, is approximately 1, which is what one expects when working with interarrival times that follow an approximate exponential distribution. ■

Example 2.22 Compute the relative standard deviation for the drying times of half gallon and whole gallon milk cartons in the data frame **MILKCARTON**.

Solution: The function **rsd()** is created to compute the relative standard deviation.

```
> rsd <- function(x){
+   abs(sd(x)/mean(x))*100
+ }
> ANS <- tapply(MILKCARTON$seconds, MILKCARTON$size, rsd)
> ANS

halfgallon wholegallon
 6.228337 16.832572
```

The relative standard deviation for half gallon containers is 6.2283%, while the relative standard deviation for whole gallon containers is 16.8326%. The standard deviation is relatively small compared to the mean for both half gallon and whole gallon containers. ■

2.6.5 The Median Absolute Deviation (*MAD*)

The median absolute deviation is a robust measure of spread that can be used to describe the spread of the data. The *MAD* is often used when the median is reported to describe the center of a skewed data set. The sample median absolute deviation is defined as

$$MAD = \text{median} |x_i - m|. \quad (2.14)$$

R has the function **mad()**, which will compute the *MAD* as defined in (2.14) when the argument **constant = 1** is specified.

Example 2.23 Use the data frame **SDS4** to compute the *MAD* for the interarrival times of cars traveling along the M1 motorway.

Solution: The *MAD* is computed according to the definition in (2.14), and the answer is verified using the **mad()** function.

```
> median(abs(SDS4$times - median(SDS4$times)))
[1] 3

> mad(SDS4$times, constant = 1)
[1] 3
```

The *MAD* interarrival time for cars traveling along the M1 motorway is 3 seconds.

2.7 Bivariate Data

Methods to summarize and display relationships between two variables (bivariate data) will be the focus of the next few pages. In Section 2.3, two of the methods used to gain a deeper understanding of categorical variables were tables and barplots. When two variables are categorical, tables, called contingency tables, and barcharts will still prove useful. When presented with quantitative bivariate data, relevant questions will deal with the relationships between the two variables. For example, is there a relationship between a person's height and his weight? Is there a relationship between the amount of time a student spends studying and her grades? Graphical techniques such as scatterplots can be used to explore bivariate relationships. When relationships exist between variables, different correlation coefficients are used to characterize the strengths of the relationships. Finally, a brief introduction to the simple linear regression model is given before multivariate data is covered.

2.7.1 Two-Way Contingency Tables

The commands `table(x)` and `xtabs(~x, data = DF)` were used for creating frequency tables with univariate, categorical variables. For bivariate, categorical data, the commands `table(x, y)` and `xtabs(~x + y, data = DF)` are used to create two-way contingency tables where `x` and `y` represent the two categorical variables and `DF` is a data frame containing `x` and `y`.

Example 2.24 Consider the data frame `EPIDURAL`, which contains information from a study to determine whether the traditional sitting position or the hamstring stretch position is superior for administering epidural anesthesia to pregnant women in labor as measured by the number of obstructive (needle to bone) contacts `oc`. The variable `doctor` specifies which of the four physicians in the study administered the procedure. The physician's assessment prior to administering the epidural of how well bony landmarks for epidural placement can be felt is stored in the variable `ease`. Produce a two-way contingency table for the variables `doctor` and `ease`.

Solution: The goal is to produce a two-way table such as the one in Table 2.3 with R. The levels of categorical variables by default are alphabetical. Consequently, the levels of `ease` are Difficult, Easy, and Impossible. Pay particular attention to how the levels of a variable are rearranged in R Code 2.21.

Table 2.3: Two-way table of Doctor by Ease

	Easy	Difficult	Impossible
Dr. A	19	3	1
Dr. B	7	10	4
Dr. C	18	3	0
Dr. D	13	4	3

R Code 2.21

```
> head(EPIDURAL) # First six rows of EPIDURAL
  doctor kg cm      ease      treatment oc complications
1 Dr. B 116 172 Difficult Traditional Sitting 0      None
2 Dr. C 86 176      Easy   Hamstring Stretch 0      None
3 Dr. B 72 157 Difficult Traditional Sitting 0      None
4 Dr. B 63 169      Easy   Hamstring Stretch 2      None
5 Dr. B 114 163 Impossible Traditional Sitting 0      None
6 Dr. B 121 163 Difficult   Hamstring Stretch 3      None

> xtabs(~doctor + ease, data = EPIDURAL) # levels listed alphabetically
      ease
doctor Difficult Easy Impossible
Dr. A        3   19       1
Dr. B       10    7       4
Dr. C        3   18       0
Dr. D        4   13       3

> EPIDURAL$ease <- factor(EPIDURAL$ease, levels = c("Easy", "Difficult",
+                               "Impossible")) # levels in order of difficulty
> xtabs(~doctor + ease, data = EPIDURAL) # levels in proper order
      ease
doctor Easy Difficult Impossible
Dr. A   19       3       1
Dr. B    7      10       4
Dr. C   18       3       0
Dr. D   13       4       3
```

Extensions to multi-way contingency tables can be accomplished by specifying additional factors to the functions `table()` and `xtabs()` or by using the flattened table function `ftable()`. More options for `table`, `xtabs()`, and `ftable()` can be found in their respective help files. An example of a flattened three-way contingency table using the factors `doctor`, `treatment`, and `ease` is shown in R Code 2.22.

R Code 2.22

```
> with(data = EPIDURAL, ftable(doctor, treatment, ease))
      ease Easy Difficult Impossible
doctor treatment
Dr. A Hamstring Stretch      7       1       0
          Traditional Sitting  12       2       1
Dr. B Hamstring Stretch      3       3       0
          Traditional Sitting  4       7       4
Dr. C Hamstring Stretch      8       3       0
          Traditional Sitting 10       0       0
Dr. D Hamstring Stretch      7       1       2
          Traditional Sitting  6       3       1
```

2.7.2 Graphical Representations of Two-Way Contingency Tables

Barplots can be used to depict graphically the information from two-way contingency tables. This is accomplished by picking one of the variables to form the categories of the barplot. The second variable's levels are graphed either in a single bar (stacked) or as several bars (side-by-side).

Example 2.25 Produce stacked and side-by-side barplots of the information contained in Table 2.3 on page 134.

Solution: Barplots where the variable of interest is `ease` then `doctor` are created first. The top left graph of Figure 2.19 on the next page is created by passing the values in `X` from R Code 2.23 to the function `barplot()`. The top right graph of Figure 2.19 on the next page is created by passing the values in `t(X)` from R Code 2.23 to the function `barplot()` in R Code 2.24. The bottom left and right side-by-side barplots in Figure 2.19 on the facing page are constructed by using the additional argument `beside = TRUE` with the function `barplot()`.

R Code 2.23

```
> X <- xtabs(~doctor + ease, data = EPIDURAL)
> X

      ease
doctor   Easy Difficult Impossible
Dr. A     19       3        1
Dr. B      7      10        4
Dr. C     18       3        0
Dr. D     13       4        3

> t(X)  # Transpose X

      doctor
ease       Dr. A Dr. B Dr. C Dr. D
Easy        19    7    18    13
Difficult     3   10     3     4
Impossible    1    4     0     3
```

R Code 2.24

```
> opar <- par(no.readonly = TRUE) # read in current parameters
> par(mfrow=c(2, 2))           # 2 rows and 2 columns
> barplot(X)                  # top left graph
> title("Doctor Stacked within \n Levels of Palpation")
> barplot(t(X))               # top right graph
> title("Levels of Palpation \n Stacked within Doctor")
> barplot(X, beside = TRUE)    # bottom left graph
> title("Doctor Grouped within \n Levels of Palpation")
> barplot(t(X), beside = TRUE) # bottom right graph
> title("Levels of Palpation \n Grouped within Doctor")
> par(opar)                   # rest original parameters
```

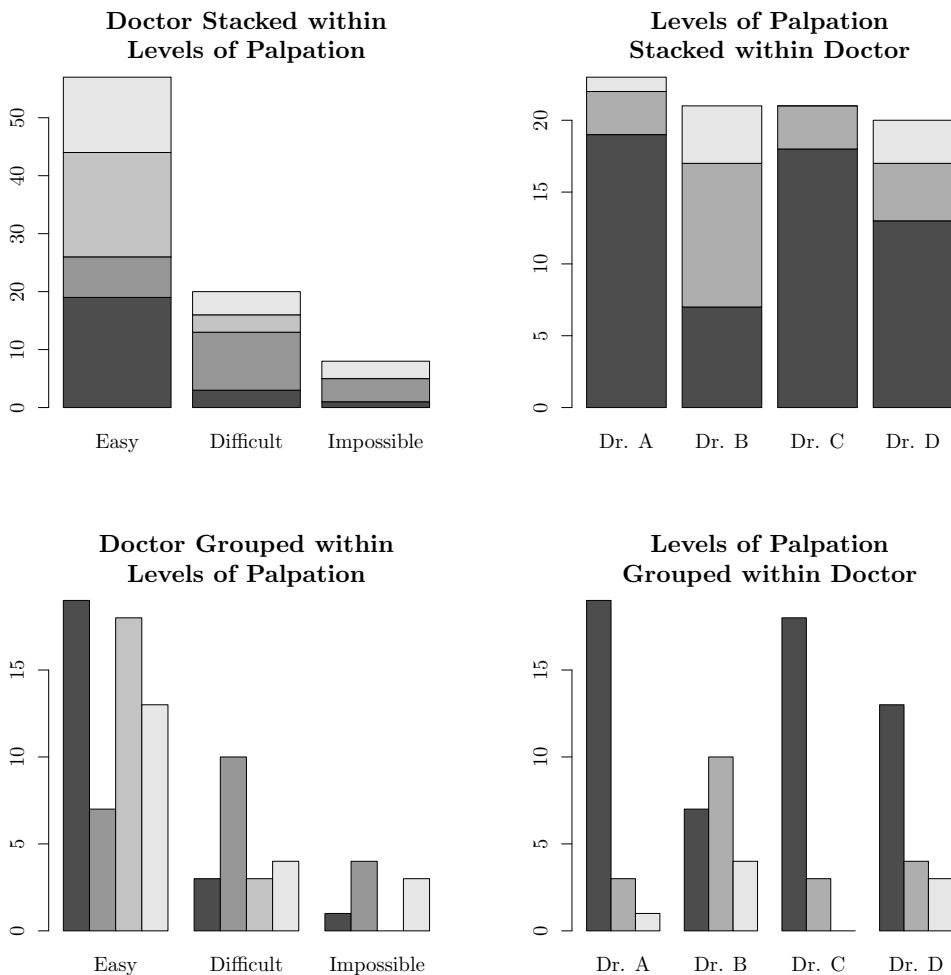


FIGURE 2.19: Stacked and side-by-side barplots for levels of palpation (`ease`) and physician (`doctor`)



Relationships are often better represented with proportions than with counts. R has the function `prop.table(x)`, which can be used to compute proportions based on the number of entries in either the entire table, `x`, which is the default, or by entering `prop.table(x, margin=1)` for row totals or `prop.table(x, margin=2)` for column totals.

Example 2.26 Using the data frame `EPIDURAL`, create a side-by-side barplot of `treatment` versus `oc`.

Solution: Since 35 patients have been treated with the hamstring stretch position and 50 patients have been treated with the traditional sitting position in data frame `EPIDURAL`, it would not be rational to compare the frequencies for the two treatment methods. Instead, one should compare the percentages within the categories of `oc` by `treatment`. Close examination of the data frame `EPIDURAL` reveals there is a missing value reported for the number of obstructive contacts for one patient assigned to the traditional sitting position. The missing value is why the table created with `xtabs()` shows only 49 patients being treated with the traditional sitting position.

R Code 2.25

```
> xtabs(~treatment + oc, data = EPIDURAL)

          oc
treatment      0  1  2  3  4  5  6 10
  Hamstring Stretch 17  6  6  2  1  1  0  2
  Traditional Sitting 23 16  3  1  2  2  2  0

> addmargins(xtabs(~treatment + oc, data = EPIDURAL)) # addmargins

          oc
treatment      0  1  2  3  4  5  6 10 Sum
  Hamstring Stretch 17  6  6  2  1  1  0  2 35
  Traditional Sitting 23 16  3  1  2  2  2  0 49
  Sum            40 22  9  3  3  3  2  2 84

> X <- prop.table(xtabs(~treatment + oc, data = EPIDURAL), 1)
> X # Percents by rows

          oc
treatment      0           1           2           3
  Hamstring Stretch 0.48571429 0.17142857 0.17142857 0.05714286
  Traditional Sitting 0.46938776 0.32653061 0.06122449 0.02040816

          oc
treatment      4           5           6           10
  Hamstring Stretch 0.02857143 0.02857143 0.00000000 0.05714286
  Traditional Sitting 0.04081633 0.04081633 0.04081633 0.00000000
```

Note that the categories for the barplot in Figure 2.20 on the next page created from R Code 2.26 are the `oc` categories in the two-way contingency table named `X` in R Code 2.25. Within each `oc` category, comparisons are shown side-by-side based on the treatment.

R Code 2.26

```
> barplot(X, beside = TRUE, legend = TRUE, ylim = c(0, 0.5))
```

If the user wants the categories to be reversed, transpose the table of percents using the transpose command `t(X)`, where `X` is the two-way contingency table of percents. R Code 2.27 transposes the values in `X` and creates Figure 2.21 on the next page.

R Code 2.27

```
> barplot(t(X), beside=TRUE, legend=TRUE, ylim = c(0, 0.5))
```

2.7.3 Comparing Samples

The need to compare two samples is quite common. Simple experiments will often compare a control group to a treatment group in an effort to see if the treatment provides some added benefit. For example, the data in the `EPIDURAL` data frame are from an ongoing experiment to see which of two positions results in fewer obstructive bone contacts (the times the needle hits a bone). When comparing two samples, typically some type of inference to the samples' populations is desired. That is, are the centers the same? Are the spreads

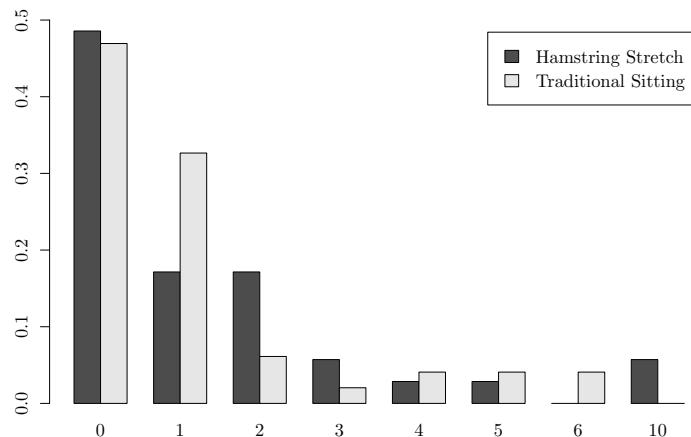


FIGURE 2.20: Barplot showing percentages of treatments by obstructive contacts

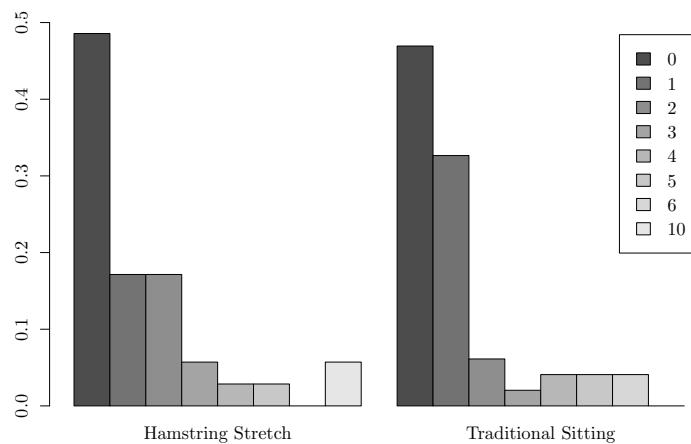


FIGURE 2.21: Barplot showing percentages of obstructive contacts by treatments

similar? Are the shapes of the two distributions similar? Graphs such as histograms, density plots, boxplots, and quantile-quantile plots can help answer these questions. Histograms and density plots were introduced in Section 2.4, and boxplots were introduced in Section 2.5. A **quantile-quantile (Q-Q) plot** plots the quantiles of one distribution against the quantiles of another distribution as (x, y) points. When the two distributions have similar shapes, the points will fall along a straight line. The function to make a quantile-quantile plot is `qqplot(x, y)`. Histograms can be used to compare two distributions; however, it is rather challenging to put both histograms on the same graph. Example 2.27 shows the user how histograms can be used to compare distributions; better approaches, such as multi-panel Trellis/lattice graphics and `ggplot2` graphics that are faceted, exist and are explained in Section 2.9.

Example 2.27 Use histograms to compare the body mass index (BMI) for patients who received an epidural using the traditional sitting position and for those who received an epidural using the hamstring stretch position. The two treatments (traditional sitting and hamstring stretch are stored in the variable `treatment`) of the `EPIDURAL` data frame.

Solution: R Code 2.28 starts by reading the current graphical parameters into the object `opar`. Next, the graphics device is split into two rows and two columns. Since the data frame `EPIDURAL` does not contain a BMI variable, one is created. For this study, BMI is defined as kg/m^2 . The default options for the BMI histograms of the traditional sitting and hamstring stretch groups are shown in the first column of Figure 2.22 on the next page, while the BMI histograms of the traditional sitting and hamstring stretch groups are shown in the second column of Figure 2.22 after the axes limits for both the x - and y -axes have been set to the same values for both histograms.

R Code 2.28

```
> opar <- par(no.readonly = TRUE)           # read in current parameters
> par(mfrow=c(2, 2))                      # 2*2 plotting region
> EPIDURAL$BMI <- EPIDURAL$kg/(EPIDURAL$cm/100)^2    # Create BMI variable
> with(data = EPIDURAL, hist(BMI[treatment == "Traditional Sitting"],
+     xlab = "BMI", main = "Sitting", col = 'gray70'))   # top left
> with(data = EPIDURAL, hist(BMI[treatment == "Traditional Sitting"],
+     xlim = c(20, 60), ylim = c(0, 17), xlab = "BMI", main = "Sitting",
+     col = 'gray30'))                                     # top right
> with(data = EPIDURAL, hist(BMI[treatment == "Hamstring Stretch"],
+     xlab = "BMI", main = "Hamstring", col = 'gray70')) # bottom left
> with(data = EPIDURAL, hist(BMI[treatment == "Hamstring Stretch"],
+     xlim = c(20, 60), ylim = c(0, 17), xlab = "BMI", main = "Hamstring",
+     col = 'gray30'))                                     # bottom right
> par(opar)                                         # reset to original parameters
```

Note that it is misleading to compare histograms where the bin widths and/or units on the axes of the two histograms are different. Both axes are different in the first column of Figure 2.22 on the facing page, and this pair should not be used for comparisons. The bins of the two histograms are set with the argument `breaks=`, and the x - and y -axes are set with the arguments `xlim=` and `ylim=`, respectively. Comparing the second column graphs (dark gray), the general shape of the BMI for the patients administered epidurals in the hamstring stretch position is unimodal skewed to the right. While the distribution of BMI for patients administered epidurals in the traditional sitting position is also unimodal skewed to the right, it is not quite as skewed as the distribution where patients are administered epidurals from the hamstring stretch position. ■

Example 2.28 Use side-by-side boxplots and superimposed density plots to compare the BMI for the two treatments (traditional sitting and hamstring stretch stored in `treatment`) using the data frame `EPIDURAL`.

Solution: The argument `horizontal = TRUE` is used with the `boxplot()` function in R Code 2.29 to create the horizontal boxplots shown in Figure 2.23 on the facing page. Using boxplots, as seen in Figure 2.23 on the next page, one sees that the median for both treatments is around $30 \text{ kg}/\text{m}^2$, and both distributions appear to be skewed to the right.

R Code 2.29

```
> with(data = EPIDURAL, boxplot(BMI ~ treatment, horizontal = TRUE))
```

R Code 2.30 on page 142 is used to create the density plots of BMI shown in Figure 2.24 on page 142. In the first line of code, the BMI values for patients given an epidural in the traditional sitting position are extracted and stored in the variable `BMITS`. The second

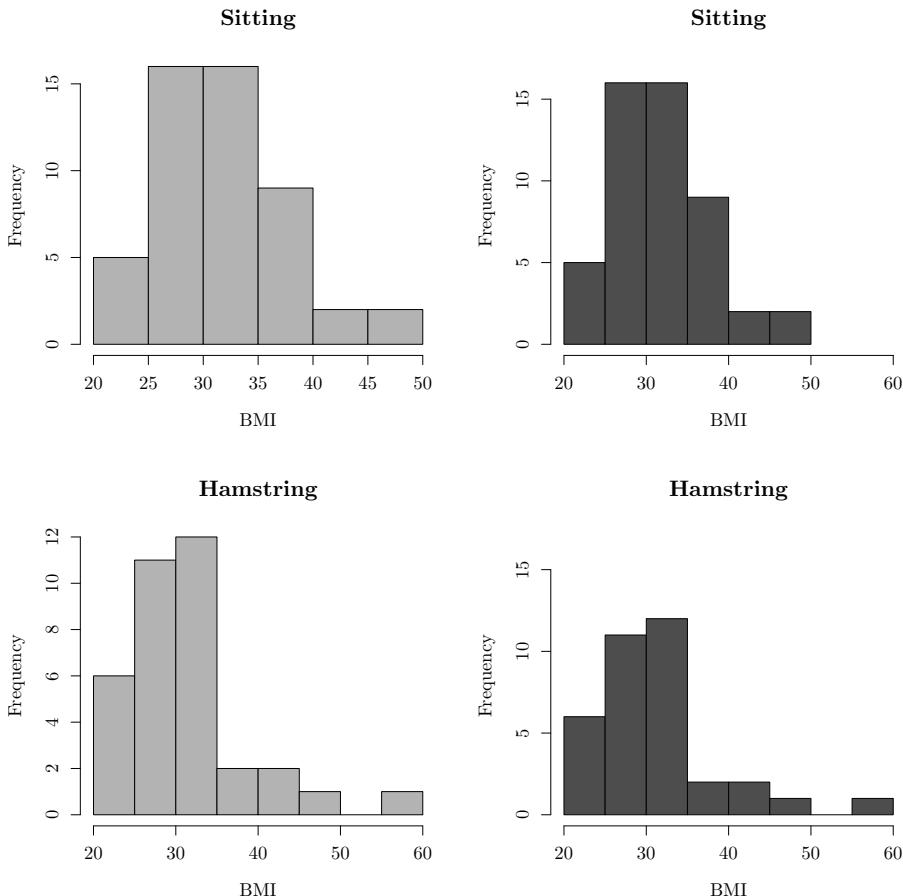


FIGURE 2.22: Histograms of the BMI values of patients administered an epidural in the traditional sitting and hamstring stretch positions with different `xlim` and `ylim` values in the first column (light gray) and the same `xlim` and `ylim` values in the second column (dark gray)

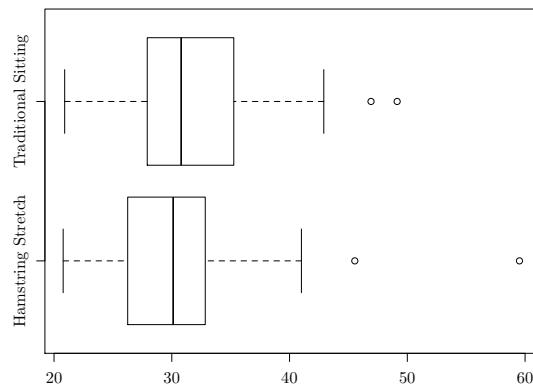


FIGURE 2.23: Side-by-side boxplots of the BMI values for patients who received an epidural in the traditional sitting and hamstring stretch positions

line of code extracts the BMI values for patients given an epidural in the hamstring stretch position and stores the results in `BMIHS`.

R Code 2.30

```
> BMITS <- with(data = EPIDURAL, BMI[treatment == "Traditional Sitting"])
> BMIHS <- with(data = EPIDURAL, BMI[treatment == "Hamstring Stretch"])
> plot(density(BMITS), xlim = c(20, 60), lwd = 2, main = "", xlab = "BMI")
> lines(density(BMIHS), lty = 2, lwd = 2)
> rm(BMITS, BMIHS) # clean up
```

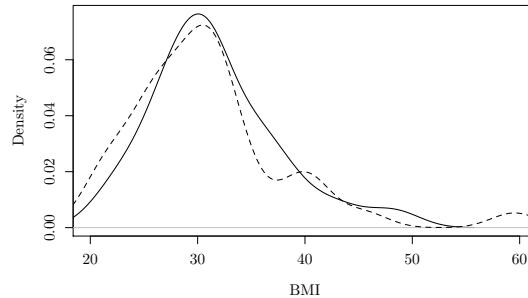


FIGURE 2.24: Density plots of the BMI for patients administered an epidural in the traditional sitting (solid line) and hamstring stretch positions (dashed line)

The density plots in Figure 2.24 further indicate that the distribution of BMI for patients who received an epidural using the traditional sitting position and patients who received an epidural using the hamstring stretch position are skewed to the right. ■

Example 2.29 Use a quantile-quantile plot to compare the BMI values of patients who received an epidural using the traditional sitting position and those who received an epidural using the hamstring stretch position. Use the information in the `EPIDURAL` data frame to create the quantile-quantile plot.

Solution: Commands to recreate the quantile-quantile plot shown in Figure 2.25 on the next page are given in R Code 2.31. Note that both the x - and y -axes have the same limits.

R Code 2.31

```
> BMITS <- with(data = EPIDURAL, BMI[treatment == "Traditional Sitting"])
> BMIHS <- with(data = EPIDURAL, BMI[treatment == "Hamstring Stretch"])
> qqplot(x = BMITS, y = BMIHS, xlim = c(20, 60), ylim = c(20, 60),
+         xlab = "Traditional Sitting", ylab = "Hamstring Stretch")
> abline(a = 0, b = 1) # Line  $y = 0 + 1*x$ 
> rm(BMITS, BMIHS) # clean up
```

The quantile-quantile plot in Figure 2.25 suggests the distributions are fairly similar since the points roughly follow the $y = x$ line. ■

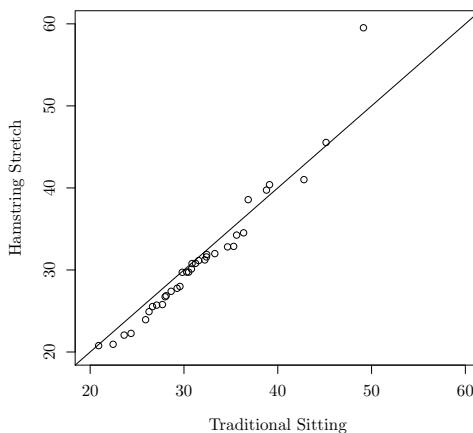


FIGURE 2.25: Quantile-quantile plot of BMI in the traditional sitting and hamstring stretch positions

2.7.4 Relationships between Two Numeric Variables

Relationships between two numeric variables can be viewed with **scatterplots**. A scatterplot plots the values of one variable against the values of a second variable as points (x_i, y_i) in the Cartesian plane. Typical questions researchers seek to answer with numeric variables include “Is there a relationship between the two variables?”, “How strong is the relationship between the two variables?”, and “Is the relationship linear?” Questions such as “Is there a relationship between a person’s height and his weight?” or “Is there a relationship between a student’s grades and the time spent studying?” are typical. Given two numeric variables, say x and y , entering the function `plot(x, y)` produces a scatterplot.

Example 2.30 Use the data frame `Animals` from the `MASS` package to investigate whether the brain weights of animals are related to their body weights. In other words, is a bigger brain required to govern a bigger body?

Solution: Because of the large range in body and brain weights, (0.023 kg to 87,000 kg) and (0.4 g to 5,712 g), respectively, a scatterplot of the values in `body` and `brain` is too distorted to reveal any clear pattern. Consequently, the data is graphed on a log base 10 scale for both axes as shown in Figure 2.26 on the next page.

```
> library(MASS)
> range(Animals$body)
[1] 2.3e-02 8.7e+04

> range(Animals$brain)
[1] 0.4 5712.0

> range(log(Animals$body))
[1] -3.772261 11.373663

> range(log(Animals$brain))
[1] -0.9162907 8.6503245

> with(data = Animals, plot(body, brain, log = "xy"))
```

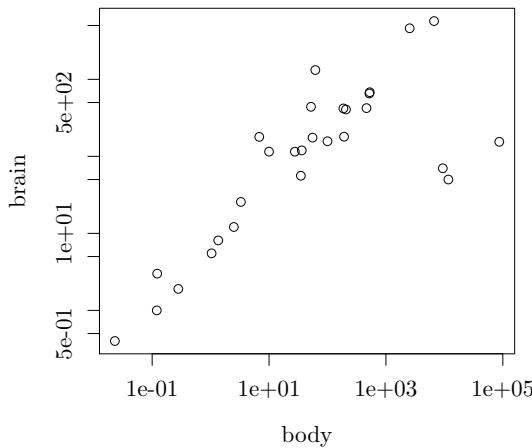


FIGURE 2.26: Scatterplot of `brain` versus `body` for Example 2.30 using a log base 10 scale for the x - and y -axes

■

The function `identify()` can be used to label points in a scatterplot. R Code 2.32, which is not run, gives an idea of how to identify points in a scatterplot. The function `identify()` labels the closest point in the scatterplot with each mouse click (left click with Windows) until instructed to stop. How the function is instructed to stop varies by operating system. Right clicking with Windows, middle clicking with Linux, and using the escape key in Mac OS X will generally stop the identification process. Based on Figure 2.26, there appears to be linear relationship between the logarithm of the body weights and the logarithm of the brain weights. The dinosaurs, the three open circles in the middle right of Figure 2.26, can be classified as bivariate outliers as they do not fit the overall pattern seen in the rest of the data. Use R Code 2.32 to label the dinosaurs as an exercise.

R Code 2.32

```
> with(data = Animals,
+       plot(body, brain, log = 'xy') )
> with(data = Animals,
+       identify(body, brain, log = "xy", labels = row.names(Animals)) )
```

2.7.5 Correlation

The **correlation coefficient**, denoted by r , measures the strength and direction of the linear relationship between two numeric variables X and Y and is defined by

$$r = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_X} \right) \left(\frac{y_i - \bar{y}}{s_Y} \right). \quad (2.15)$$

The value for r will always be between -1 and $+1$. When r is close to $+1$, it indicates a strong positive linear relationship. That is, when x increases so does y , and vice versa. When the value of r is close to -1 , it indicates a strong negative linear relationship. Values of r close to zero indicate weak linear relationships. To compute the correlation between two numeric vectors, one may use the function `cor(x, y)`.

Example 2.31 Find the correlation coefficient, r , between the logarithms of the body and brain weights in the data frame `Animals` from the MASS package using (2.15). Verify the calculated answer using the function `cor()`.

Solution: First, the variables `logbody`, `logbrain`, `Zbody`, and `Zbrain` are created. The new variables are subsequently column bound to the `Animals` data frame and stored in a new data frame named `Anim`.

```
> library(MASS)
> logbody <- log(Animals$body)
> logbrain <- log(Animals$brain)
> Zbody <- (logbody - mean(logbody))/sd(logbody)
> Zbrain <- (logbrain - mean(logbrain))/sd(logbrain)
> Anim <- cbind(Animals, logbody, logbrain, Zbody, Zbrain)
> n <- sum(!is.na(logbody))
> r <- (1/(n - 1)) * sum(Zbody * Zbrain) # Definition of r
> r
[1] 0.7794935

> cor(logbody, logbrain)
[1] 0.7794935

> head(Anim) # Show first 6 rows of Anim
      body  brain    logbody logbrain      Zbody      Zbrain
Mountain beaver     1.35   8.1 0.30010459 2.091864 -0.92058294 -0.9726165
Cow                 465.00 423.0 6.14203741 6.047372  0.62873205  0.6760048
Grey wolf            36.33 119.5 3.59264385 4.783316 -0.04738213  0.1491574
Goat                27.66 115.0 3.31998733 4.744932 -0.11969225  0.1331592
Guinea pig           1.04   5.5 0.03922071 1.704748 -0.98977088 -1.1339630
Diplodocus          11700.00 50.0 9.36734412 3.912023  1.48410239 -0.2139901
```

The correlation between `logbrain` and `logbody` is 0.7795, which indicates a positive linear relationship between the two variables. An alternative to computing the z-scores directly is to use the function `scale()`, which stores objects as a matrix with some added attributes the user can easily ignore if desired. R Code 2.33 stores the Z score of `logbody` in the matrix `ZB0`, then names the only column of the matrix `ZB0`, `ZLBO` using the function `dimnames()`. A similar process is used to name the column of the matrix `ZBR`, `ZLBR`.

R Code 2.33

```
> ZB0 <- scale(logbody) # Z score of logbody
> dimnames(ZB0) <- list(NULL, "ZLBO") # name the column ZLBO
> ZBR <- scale(logbrain) # Z score of logbrain
> dimnames(ZBR) <- list(NULL, "ZLBR") # name the column ZLBR
> SAME <- cbind(Zbody, ZB0, Zbrain, ZBR)
> SAME[1:5,] # Show first five rows of data frame
      Zbody      ZLBO      Zbrain      ZLBR
[1,] -0.92058294 -0.92058294 -0.9726165 -0.9726165
[2,]  0.62873205  0.62873205  0.6760048  0.6760048
```

```
[3,] -0.04738213 -0.04738213  0.1491574  0.1491574
[4,] -0.11969225 -0.11969225  0.1331592  0.1331592
[5,] -0.98977088 -0.98977088 -1.1339630 -1.1339630
```



Example 2.32 Find the correlation coefficient, r , between the logarithms of the body and brain weights in the data frame `Animals` from the `MASS` package with and without dinosaurs.

Solution: To save space, only four rows of the data frames `SA` and `NoDINO` are shown in the output of R Code 2.34. Note that there are a total of 28 animals in the data frame `Animals`.

R Code 2.34

```
> library(MASS)
> CWD <- with(data = Animals,
+               cor(log(body), log(brain))) # Correlation with dinosaurs
> CWD

[1] 0.7794935

> SA <- Animals[order(Animals$body), ]      # Sorted by body weight
> tail(SA, n = 4) # Equivalently SA[25:28, ], shows four heaviest animals

          body   brain
African elephant 6654 5712.0
Triceratops       9400    70.0
Diplodocus        11700   50.0
Brachiosaurus     87000  154.5

> NoDINO <- SA[-(28:26), ]                  # Remove rows 26-28 of SA
> NoDINO[22:25, ]                            # Show four heaviest animals

          body   brain
Horse           521    655
Giraffe          529    680
Asian elephant   2547   4603
African elephant 6654 5712

> CND <- with(data = NoDINO,
+               cor(log(body), log(brain))) # Correlation without dinosaurs
> CND

[1] 0.9600516
```

The correlation between `log(brain)` and `log(body)` when dinosaurs are included is 0.7795 and the correlation between `log(brain)` and `log(body)` is 0.9601 when the dinosaurs are removed from the computation.



2.7.6 Fitting Lines to Bivariate Data

When a linear pattern is evident from a scatterplot, the relationship between the two variables is often modeled with a straight line. When modeling a bivariate relationship, Y is called the **response** or **dependent** variable, and x is called the **predictor** or **independent** variable. There are relationships that are of interest that are not linear; however, before addressing more complicated models, this material attempts to provide a foundation for the simpler models (simple linear regression) from which more complicated models can later be built. Chapter 12 is devoted to standard regression techniques for both the simple and multiple linear regression model. The simple linear regression model is written

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i. \quad (2.16)$$

Model (2.16) is said to be simple, linear in the parameters (β_0 and β_1), and linear in the predictor variable (x_i). It is simple because there is only one predictor; linear in the parameters because no parameter appears as an exponent nor is multiplied or divided by another parameter; and linear in the predictor variable since the predictor variable is raised only to the first power. When the predictor variable is raised to a power, this power is called the **order** of the model. For now, only the simple linear model will be discussed. The goal is to estimate the coefficients β_0 and β_1 in (2.16). The most well-known method of estimating the coefficients β_0 and β_1 is to use ordinary least squares (OLS). OLS provides estimates of β_0 and β_1 by minimizing the sum of the squared deviations of the Y_i s for all possible lines. Specifically, the sum of the squared residuals ($\hat{\varepsilon}_i = Y_i - \hat{Y}_i$) is minimized when the OLS estimators of β_0 and β_1 are

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x} \quad (2.17)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad (2.18)$$

respectively. Note that the estimated regression function is written as

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i.$$

A graphical representation of the residuals and a line fit to some data using OLS can be seen in Figure 2.27 on the following page.

The OLS estimators of β_0 and β_1 are affected by outliers just as the mean and standard deviation are subject to outliers. Recall that the median and IQR were suggested as measures of center and spread, respectively, when working with skewed distributions. This recommendation was made because the median and IQR provide more robust measures of center and spread in the presence of outliers. In the presence of bivariate outliers, several robust alternatives exist for computing estimates of β_0 and β_1 . Two alternatives to OLS implemented in the MASS package will be considered. Specifically, least-trimmed squares using the function `lqs()` and robust regression using an M estimator with the function `r1m()` are discussed. Just as OLS sought to minimize the squared vertical distance between all of the Y_i s over all possible lines, least-trimmed squares minimizes the q smallest residuals over all possible lines where $q = \lfloor (n + p + 1)/2 \rfloor$. Fitting for the function `r1m()` is done by iterated re-weighted least squares. Although `lqs()` and `r1m()` are computationally intensive, the interfaces for `lm()`, `lqs()`, and `r1m()` are essentially identical. All three functions require a model formula of the form $y \sim x$. The \sim in this notation is read “is modeled by.”

Example 2.33 In Exercise 2.32 on the preceding page, the correlation between the logarithms of the body and brain weights in the data frame `Animals` from the MASS package with

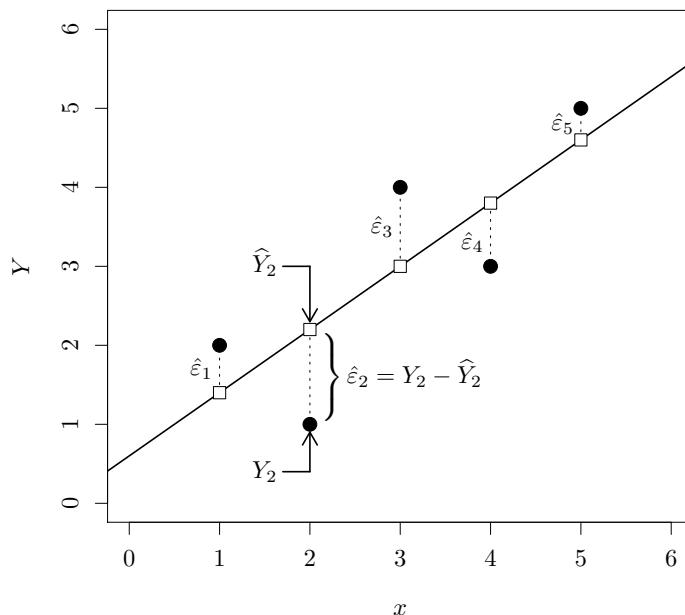


FIGURE 2.27: Graph depicting residuals. The vertical distances shown with a dotted line between the Y_i s, depicted with a solid circle, and the \hat{Y}_i s, depicted with a clear square, are the residuals.

and without dinosaurs was computed. Find the estimates for the least squares regression lines with and without dinosaurs where the logarithm of brain is modeled by the logarithm of body using Equations (2.17) and (2.18) as well as the R function `lm()`. Superimpose both lines on the scatterplot using the function `abline()` (see Table A.5 on page 907).

Solution: Recall that there are a total of 28 animals in the data frame `Animals` and 25 animals in the `NoDINO` data frame. R Code 2.35 created the scatterplot shown in Figure 2.28 on page 150 with superimposed regression lines including the dinosaurs and omitting the dinosaurs.

R Code 2.35

```
> Y <- log(Animals$brain)
> X <- log(Animals$body)
> plot(X, Y, xlab = "log(body)", ylab = "log(brain)")
> b1 <- sum((X - mean(X))*(Y - mean(Y)))/sum((X - mean(X))^2)
> b0 <- mean(Y) - b1*mean(X)
> estimates <- c(b0, b1)
> estimates

[1] 2.5548981 0.4959947
```

```
> modDINO <- lm(Y ~ X)
> modDINO
```

Call:

```
lm(formula = Y ~ X)
```

```

Coefficients:
(Intercept)          X
2.555            0.496

> abline(modDINO, col = "pink", lwd = 2)
> SA <- Animals[order(Animals$body),]      # Sorted by body weight
> NoDINO <- SA[-(28:26), ]                # Remove rows 26-28 (dinosaurs)
> Y <- log(NoDINO$brain)
> X <- log(NoDINO$body)
> b1 <- sum((X - mean(X))*(Y - mean(Y)))/sum((X - mean(X))^2)
> b0 <- mean(Y) - b1*mean(X)
> estimates <- c(b0, b1)
> estimates

[1] 2.1504121 0.7522607

> modNODINO <- lm(Y ~ X)
> modNODINO

Call:
lm(formula = Y ~ X)

Coefficients:
(Intercept)          X
2.1504            0.7523

> abline(modNODINO, col = "blue", lwd = 2, lty = 2)
> leglab <- c("OLS with Dinosaurs", "OLS without Dinosaurs")
> leglty <- c(1, 2)
> legcol=c("pink","blue")
> legend("bottomright", legend=leglab, lty=leglty, col=legcol, lwd=2)

```

The intercept and slope of the regression line with dinosaurs are 2.5549 and 0.496, respectively. Without the dinosaurs, the intercept and slope of the regression line are 2.1504 and 0.7523, respectively. ■

Example 2.34 From Figure 2.28 in Exercise 2.33, one notices three bivariate outliers (dinosaurs). Fit regression lines to the same data used in Exercise 2.28 using ordinary least squares, least-trimmed squares, and robust regression with an M estimator. Superimpose the resulting regression lines on a scatterplot and label the lines accordingly.

Solution: R Code 2.36 created the scatterplot shown in Figure 2.29 on page 151 with the three superimposed regression lines.

R Code 2.36

```

> plot(log(Animals$body), log(Animals$brain), col="blue",
+       xlab = "log(body)", ylab = "log(brain)")
> f <- log(Animals$brain) ~ log(Animals$body)
> modellM <- lm(f)
> abline(modellM, col = "pink", lwd = 2)
> modellQS <- lqs(f)

```

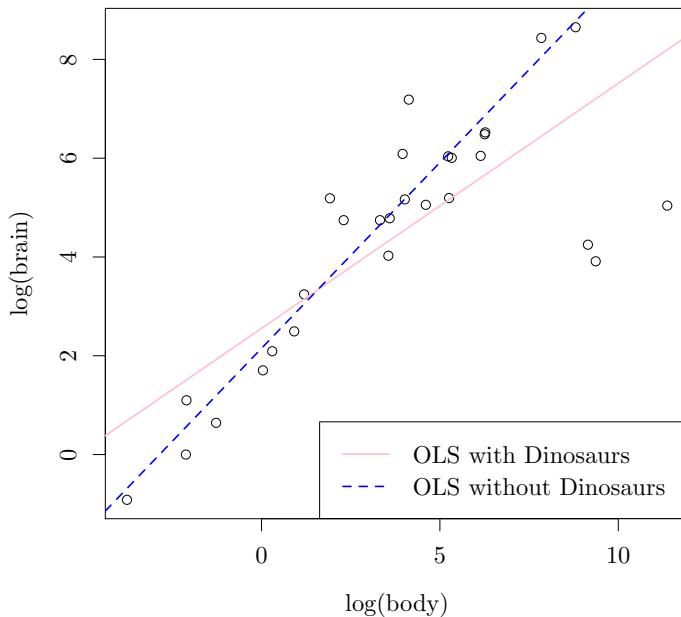


FIGURE 2.28: Scatterplot of $\log(\text{brain})$ versus $\log(\text{body})$ with superimposed regression lines computed with (solid line) and without (dashed line) dinosaurs

```
> abline(modellQS, lty = 2, col = "red", lwd = 2)
> modelRLM <- rlm(f, method = "MM")
> abline(modelRLM, lty = 3, col = "black", lwd = 2)
> leglabels <- c("Least Squares Line", "Least-Trimmed Squares",
+                 "Robust Line: M-estimator")
> leglty <- c(1, 2, 3)
> legend("bottomright", legend = leglabels, lty = leglty,
+         col = c("pink", "red", "black"), lwd = 2, cex = 0.85)
```

The least-trimmed squares (`lqs()`) procedure and the robust line with M estimator (`rlm()`) method produce lines that put relatively little importance on outliers (dinosaurs). This is further highlighted when one considers the estimates β_0 and β_1 for the OLS estimates without dinosaurs compared to the estimates of β_0 and β_1 for the least-trimmed squares and robust procedures given in Table 2.4.

Table 2.4: Different values for b_0 and b_1 with various regression methods

Method	b_0	b_1
OLS with dinosaurs	2.5549	0.496
OLS without dinosaurs	2.1504	0.7523
least-trimmed squares	1.8163	0.7761
robust line with M estimator	2.0487	0.7513

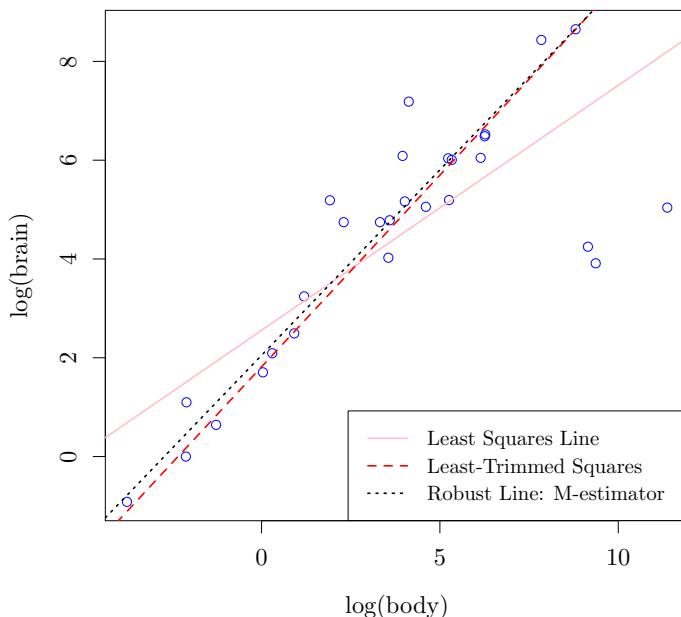


FIGURE 2.29: Scatterplot of $\log(\text{brain})$ versus $\log(\text{body})$ with three superimposed regression lines. Solid is the OLS line; dashed is the least-trimmed squares line; and dotted is the robust line.

2.8 Complex Plot Arrangements

R Code 1.68 on page 86 introduced the idea of splitting the graphics device into several plots of equal size using `par(mfrow = c(nr, nc))` where `nr` and `nc` are integer values for the number of rows and the number of columns by which the graphics device is split, respectively. In this section, the function `layout()` is introduced to handle more complex layouts. There is one required argument for `layout()`, `mat`, which is a matrix specifying the location of the next N figures on the graphics device. For more options, see the `layout` help file.

Example 2.35 Use the function `layout()` to split the graphics device into nine equal sized plots.

Solution: R Code 2.37 creates a 3×3 matrix for the values 1 through 9 and stores the result in the object `mat33`. The graphics device is split according to `mat33` with the function `layout()`, and the resulting split of the graphics device shown in Figure 2.30 on the following page is accomplished with the function `layout.show()`.

R Code 2.37

```
> mat33 <- matrix(1:9, byrow = TRUE, nrow = 3)
> mat33
```

```
[,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

> layout(mat33)  # split the device according to mat33
> layout.show(9) # show the nine plots
```

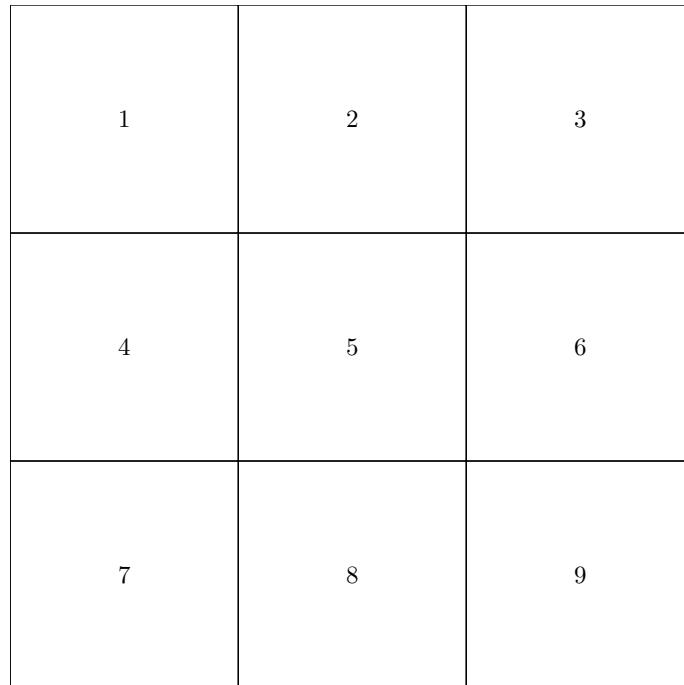


FIGURE 2.30: Nine equal-sized plots

R Code 2.38 creates a 3×2 matrix and splits the graphics device according to the values in `mat32`. The resulting partition of the graphics device is shown on the left of Figure 2.31 on the facing page.

R Code 2.38

```
> mat32 <- matrix(c(1, 1, 2, 2, 3, 3), byrow = TRUE, nrow = 3)
> mat32

 [,1] [,2]
[1,]    1    1
[2,]    2    2
[3,]    3    3

> layout(mat32)
> layout.show(3)
```

R Code 2.39 creates a 3×5 matrix and splits the graphics device according to the values in `mat35`. The resulting partition of the graphics device is shown on the right of Figure 2.31.

R Code 2.39

```
> mat35 <- matrix(c(1, 1, 1, 2, 3, 1, 1, 1, 4, 5, 6, 6, 7, 7, 7),
+                      byrow = TRUE, nrow = 3)
> mat35

[,1] [,2] [,3] [,4] [,5]
[1,]    1    1    1    2    3
[2,]    1    1    1    4    5
[3,]    6    6    7    7    7

> layout(mat35)
> layout.show(7)
```

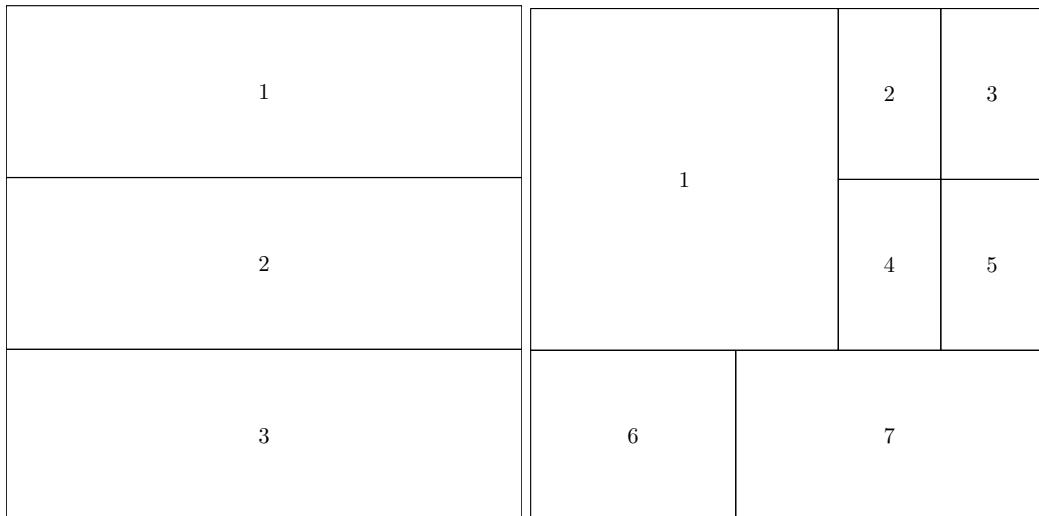


FIGURE 2.31: The left figure shows a graphics device split according to the matrix `mat32`. The right figure shows a graphics device split according to the matrix `mat35`.

Example 2.36 Use the function `layout()` to split the graphics device into four plotting regions. Use the data frame `HWWRESTLER` and place a scatterplot of `hwfat` versus `tanfat` in the top left of the graphics device. Directly below the x -axis of the scatterplot, place a horizontal boxplot of the variable `tanfat`. To the right of the scatterplot, place a vertical boxplot of the variable `hwfat`. Leave the bottom right plotting region empty.

Solution: R Code 2.40 splits the graphics device according to the values in `mat44`. The requested graphs are shown in Figure 2.32 on page 155.

R Code 2.40

```
> opar <- par(no.readonly = TRUE) # copy of current settings
```

```

> mat44 <- matrix(c(1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 3, 3, 3, 0),
+                     byrow = TRUE, nrow = 4)
> mat44

      [,1] [,2] [,3] [,4]
[1,]    1    1    1    2
[2,]    1    1    1    2
[3,]    1    1    1    2
[4,]    3    3    3    0

> layout(mat44)                      # split graphics device
> par(mar = c(4, 4, 4, 4))
> plot(hwfat ~ tanfat, col = "red", pch = 19, main = "", data = HSWRESTLER)
> boxplot(HSWRESTLER$hwfat, col = "blue", ylab = "hwfat")
> boxplot(HSWRESTLER$tanfat, col = "purple", horizontal = TRUE,
+           xlab = "tanfat")
> par(opar)                          # restore original settings

```

2.9 Multivariate Data

Graphs created up to this point have been limited to base graphics. In this section, three additional graphical systems, `vcd` (Meyer et al., 2014b), `lattice` (Sarkar, 2015), and `ggplot2` (Wickham and Chang, 2015b), are introduced that are helpful when exploring statistical relationships among multiple variables. The graphics functions from the `vcd`, `lattice`, and `ggplot2` packages are all built using the `grid` (R Core Team, 2015b) graphics system. Traditional base graphics functions such as `hist()` and `boxplot()` are created with the `graphics` package, which is automatically loaded in a standard installation of R. To learn more about the `grid` graphics system, the user should consult the text *R Graphics* (Murrell, 2011). The functions in the `vcd`, `lattice`, and `ggplot2` packages will all take a `data = argument`. A further requirement in `ggplot2` is that the object provided to the `data = argument` must be a data frame.

The major difference between `ggplot2` and the other graphics systems is that it uses a defined structure (grammar) to create its graphs. That is, a large number of graphs can be created from working with a relatively small number of elements (aesthetics, geoms, scales, coordinate system, and facets). Other systems have different function calls for each individual graph. The implementation of `ggplot2` is based on the ideas presented in Leland Wilkinson's book *The Grammar of Graphics* (Wilkinson, 2005). Mastering the grammar used in `ggplot2` is worth the time investment for anyone working with graphs on a routine basis.

For R users comfortable with the base graphics function `plot()`, the package `ggplot2` provides the wrapper function `qplot()` (quick plot) that allows users familiar with `plot()` to produce `ggplot2` graphs with commands very similar to those one would provide to `plot()`. Although there are similarities between the arguments given to `plot()` and `qplot()` for simple graphs, graphs produced with the package `ggplot2()` are much more sophisticated than those produced with base graphics. This text will not discuss the `qplot()` function further; but, will focus on the function `ggplot()` which is part of the package `ggplot2`.

Specifically, re-arrangement of graphics components that account for whether the final

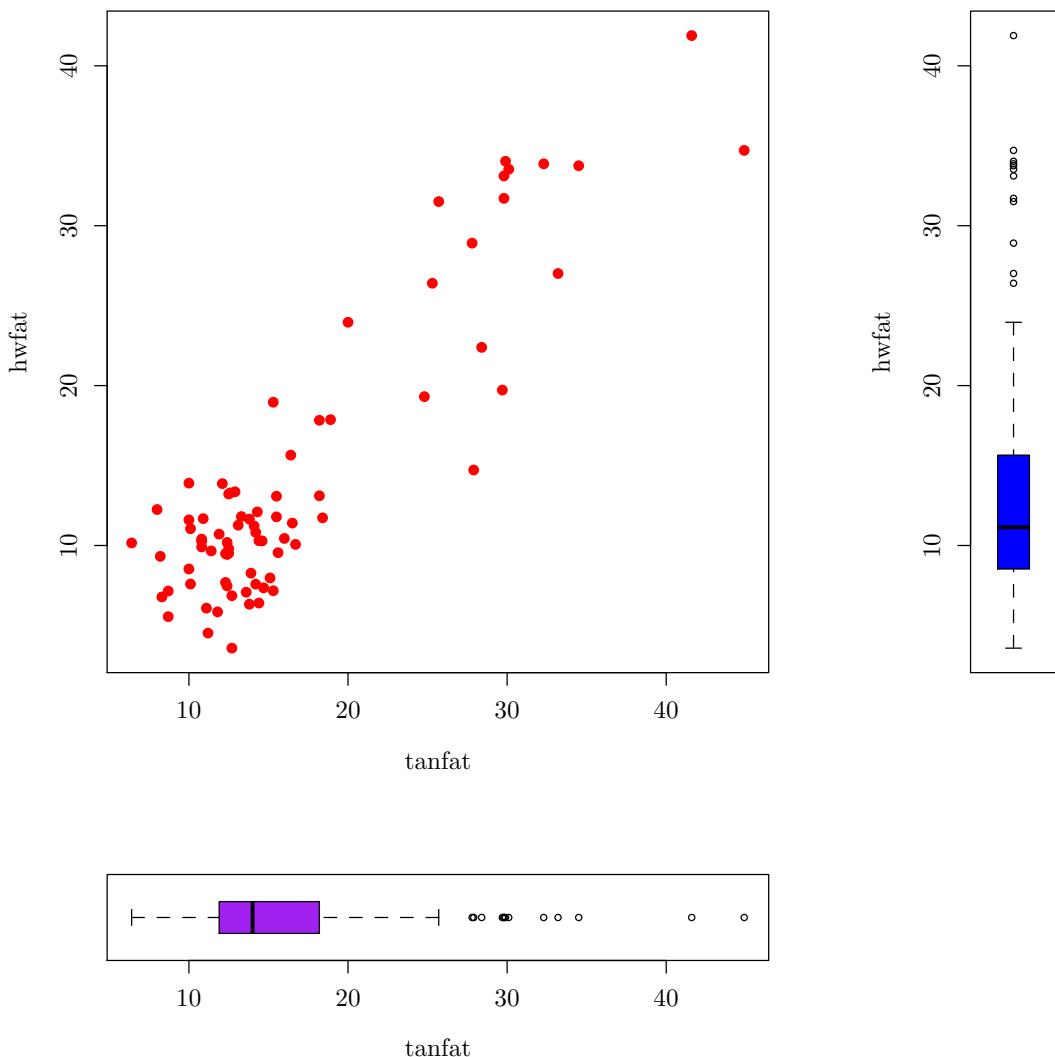


FIGURE 2.32: Scatterplot and boxplots for Example 2.36

graphic has labels on the axes, a title, and or a legend are all done automatically. While the user still has control of the individual components, just as in base graphs, the initial graph produced with `ggplot2` is more likely to be of publication quality without tinkering with additional arguments than a graph produced with a call to a base graphics function such as `boxplot()`. There is a slight penalty for the additional convenience: More knowledge is required to change the default appearance of a `ggplot2` graph than to change the default appearance of a graph created with base graphics.

Graphs produced with the packages `vcd` and `lattice` are also more sophisticated than base graphs, but they lack the formal structure of `ggplot2` graphs. A `ggplot2` graph is created by mapping the data to a geometric shape (`geom`) via an aesthetics (`aes`) argument. Other elements of `ggplot2` graphs include scales, coordinate system, and facets. By mastering the grammar of `ggplot2`, one can create a wide range of graphs without having to use different arguments for many specialized functions. The material in this section is brief with respect to the `vcd`, `lattice`, and `ggplot2` packages and is limited to talking

about graphs that will be helpful for a reader engaged in the material covered in this text. More detail about `lattice` and `ggplot2` can be found in the books *Lattice: Multivariate Data Visualization with R* (Sarkar, 2008) and *ggplot2: Elegant Graphics for Data Analysis* (Wickham, 2009), written by Deepayan Sarkar and Hadley Wickham, authors of the `lattice` and `ggplot2` packages, respectively. While the reader can always refer to the help pages for individual functions of any package, the online documentation for `ggplot2`, <http://docs.ggplot2.org/current/>, is extremely useful in showing code and graphical output for different geoms, statistics, scales, facets, and position adjustments. A general online reference for R is the *Cookbook for R* found at <http://wiki.stdout.org/rcookbook/>, which includes a very nice section on graphs with particular attention paid to `ggplot2` graphs.

2.9.1 Graphs for Categorical Data

In Section 2.3 on page 98, graphs for displaying qualitative data such as barplots, dot charts, and pie charts were discussed. These often work well for one or two variables. To effectively display categorical multivariate data, new graphs are needed. One such graph is a mosaic plot. A mosaic plot is a graphical display that allows one to explore relationships among several categorical variables. The mosaic plot starts as a unit square, a square with sides of length one. The function `mosaic()` from the `vcd` package (Meyer et al., 2014b) starts by dividing the square into horizontal bars (this differs from the mosaic plot created with the base function `mosaicplot()`, which starts by dividing the unit square into vertical bars) whose area is proportional to the number of observations in each category of the first categorical variable passed to the argument `formula=`. Next, each horizontal row is split by the second categorical variable passed to the argument `formula=` of the `mosaic()` function. The vertical splits are proportional to the probability of being in a particular category of the second categorical variable given the first categorical variable (conditional probability). Further splits with additional categorical variables follow the same horizontal then vertical division process. The end result shows rectangles corresponding to cells of a table of one or more dimensions. The area of each rectangle is proportional to the corresponding cell entry of a table.

Consider the data frame `EPIDURALF` to illustrate the construction of a mosaic plot. Recall that the levels of factors are stored alphabetically by default. The variable `ease` in the `EPIDURALF` data frame has levels "Difficult", "Easy", and "Impossible". Before proceeding, the levels of `ease` are reordered in R Code 2.41.

R Code 2.41

```
> levels(EPIDURALF$ease)
[1] "Difficult"    "Easy"          "Impossible"

> EPIDURALF$ease <- factor(EPIDURALF$ease, levels = c("Easy", "Difficult",
+                                         "Impossible"))
> levels(EPIDURALF$ease)
[1] "Easy"          "Difficult"      "Impossible"
```

To find out how many patients each physician treated, a table of the variable `doctor` is created with the function `xtabs()`.

```
> T1 <- xtabs(~doctor, data = EPIDURALF)
> T1
```

```

doctor
A   B   C   D
61 115 93 73

> addmargins(T1) # add margins to T1

doctor
A   B   C   D Sum
61 115 93 73 342

> prop.table(T1) # Fraction of patients each physician treated

doctor
A           B           C           D
0.1783626 0.3362573 0.2719298 0.2134503

```

Note that Doctors A, B, C, and D administered epidurals to 17.84, 33.63, 27.19, and 21.35 percent of the patients, respectively. Consequently, the unit square is split into horizontal bars whose height is proportional to the number of patients each physician treated. The initial horizontal splits are shown in the left plot of Figure 2.33. Next, each horizontal row is split by the second categorical variable passed to the argument `formula=` of the `mosaic()` function. That is, for all of the patients treated by Doctor A (61), vertical splits are made whose width is proportional to the number of patients who were easy, difficult, and impossible to palpate, respectively. This process is repeated across each of the horizontal splits. The proportional widths of each vertical split are computed in R Code 2.42.

R Code 2.42

```

> T2 <- xtabs(~doctor + ease, data = EPIDURALF)
> addmargins(T2) # add margins

      ease
doctor Easy Difficult Impossible Sum
    A     39       20        2   61
    B     49       58        8 115
    C     72       21        0  93
    D     47       15       11  73
    Sum  207      114      21 342

> prop.table(T2, 1) # compute fraction going across rows

      ease
doctor      Easy Difficult Impossible
    A 0.63934426 0.32786885 0.03278689
    B 0.42608696 0.50434783 0.06956522
    C 0.77419355 0.22580645 0.00000000
    D 0.64383562 0.20547945 0.15068493

```

The final mosaic plot with both horizontal and vertical splits is shown in the right plot of Figure 2.33. The code used to create both plots in Figure 2.33 is shown in R Code 2.43. To use the functions inside the `vcd` package, make sure the `vcd` package is loaded.

R Code 2.43

```
> library(vcd) # load vcd package
> mosaic(~doctor, data = EPIDURALF) # left plot
> mosaic(~doctor + ease, data = EPIDURALF) # right plot
```

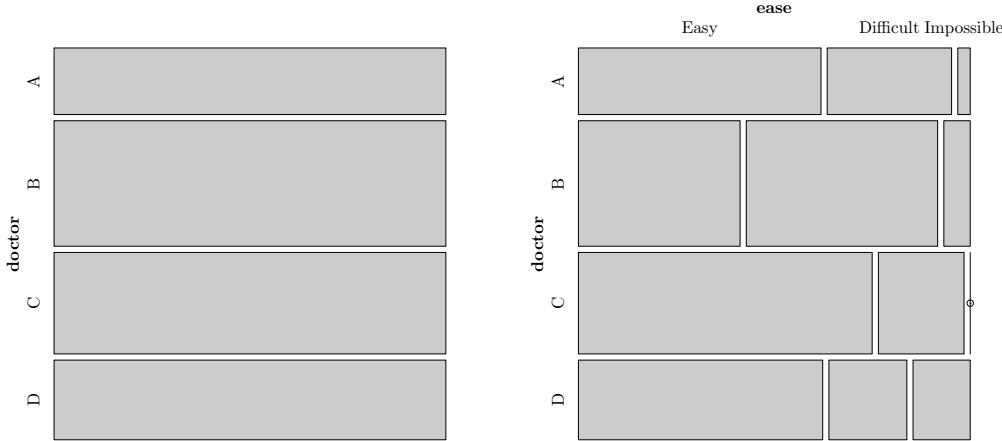


FIGURE 2.33: The left mosaic plot shows the initial horizontal splits that correspond to the proportion of patients treated by each physician. The right mosaic plot shows the relative proportion of patients each physician classified as easy, difficult, and impossible to palpate.

Doctor C did not classify any patients as impossible to palpate, and this is illustrated in the right plot of Figure 2.33 by the superimposed circle on a vertical line. In this case, a line is used to represent the area of the rectangle with no observations.

Rectangles can be colored based on a particular categorical variable by providing values to the arguments `highlighting_fill=` and `highlighting=`. Figure 2.34, created from R Code 2.44, shows a mosaic plot where the variable `ease` is shaded `gray80` for easy, `gray50` for difficult, and `gray20` for impossible.

R Code 2.44

```
> mosaic(~doctor + ease, data = EPIDURALF, highlighting_fill = c("gray80",
+ "gray50", "gray20"), highlighting = "ease")
```

One can also use the option `shade = TRUE` to help in assessing independence among categories. R Code 2.45 is used to create Figure 2.35 on the next page

R Code 2.45

```
> mosaic(~doctor + ease, data = EPIDURALF, shade = TRUE)
```

The light gray shading in Figure 2.35 indicates Doctor B classified fewer patients as easy to palpate than did his colleagues, while the dark gray rectangle for Doctor B indicates he

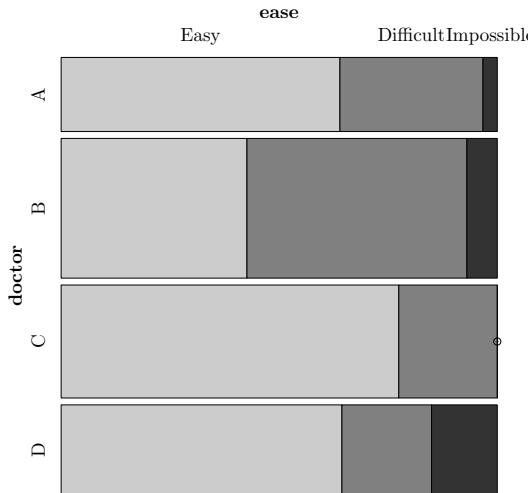


FIGURE 2.34: Mosaic plot where physician's assessment for ease of palpating a patient is grayscale coded with patients classified as **Easy** shaded gray80, **Difficult** shaded gray50, and **Impossible** shaded gray20

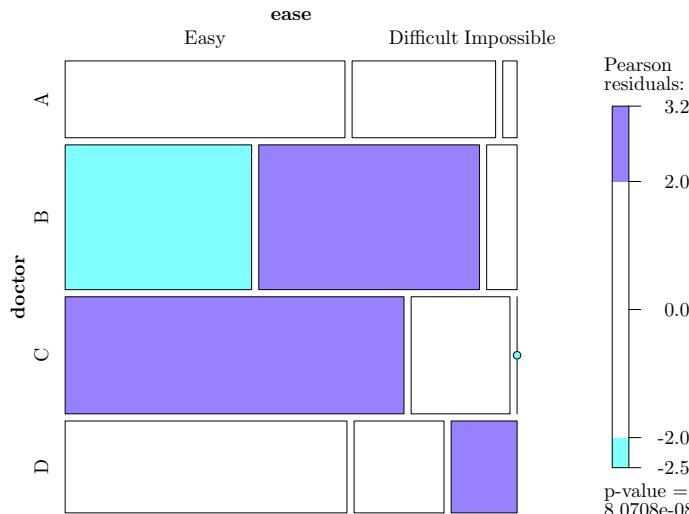


FIGURE 2.35: Mosaic plot shaded according to Pearson residuals

classified more patients as difficult to palpate than did his colleagues. When the shading color is white, the numbers in that cell are relatively close to what one might expect under an assumption of independence among the categorical variables.

Example 2.37 Create both tables and mosaic plots to show why the University of California appeared to have gender bias in graduate admissions when a group considered only gender and admission status of the applicants. Aggregate data on applicants who applied to graduate school at the University of California, Berkeley in 1973 is stored in **UCBAdmissions** as a three-dimensional array. Next, include departments in the analysis and explain why the gender bias disappears.

Solution: The **UCBAdmissions** array is coerced into a data frame and a two-way table

of gender versus admission status is created in R Code 2.46 showing that 44.52% of male graduate applicants were admitted while only 30.35% of female graduate applicants were admitted. A shaded mosaic plot is created and shown in Figure 2.36. If the analysis were to stop at this point, one might erroneously claim gender bias.

R Code 2.46

```
> UCB <- as.data.frame(UCBAdmissions)
> prop.table(xtabs(Freq ~ Gender + Admit, data = UCB), 1)

      Admit
Gender   Admitted Rejected
Male    0.4451877 0.5548123
Female  0.3035422 0.6964578
```

```
> mosaic(~Gender + Admit, data = UCB, shade = TRUE)
```

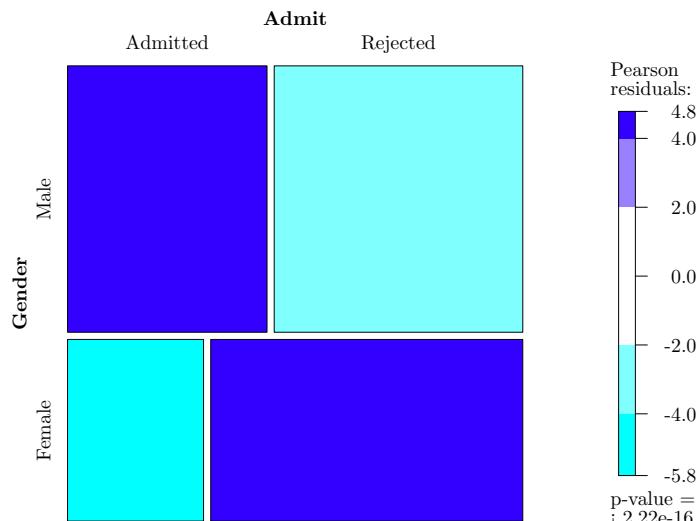


FIGURE 2.36: Shaded mosaic plot suggesting too few female applicants were admitted (bottom left light gray rectangle) as well as too few male applicants were rejected (upper right light gray rectangle). The dark gray rectangles suggest too many male applicants were admitted and too many female applicants were rejected.

Admission percentages based on both gender and departments are constructed with R Code 2.47. An examination of the results shows females were admitted at percentages similar to or higher than males. The University of California at Berkeley graduate admission data illustrates how collapsing categories can change or reverse the relationship among variables. This particular phenomenon is known as Simpson's paradox.

R Code 2.47

```
> prop.table(xtabs(Freq ~ Admit + Gender + Dept, data = UCB), c(2, 3))
```

```
, , Dept = A

      Gender
Admit      Male      Female
  Admitted 0.62060606 0.82407407
  Rejected 0.37939394 0.17592593

, , Dept = B

      Gender
Admit      Male      Female
  Admitted 0.63035714 0.68000000
  Rejected 0.36964286 0.32000000

, , Dept = C

      Gender
Admit      Male      Female
  Admitted 0.36923077 0.34064081
  Rejected 0.63076923 0.65935919

, , Dept = D

      Gender
Admit      Male      Female
  Admitted 0.33093525 0.34933333
  Rejected 0.66906475 0.65066667

, , Dept = E

      Gender
Admit      Male      Female
  Admitted 0.27748691 0.23918575
  Rejected 0.72251309 0.76081425

, , Dept = F

      Gender
Admit      Male      Female
  Admitted 0.05898123 0.07038123
  Rejected 0.94101877 0.92961877
```

Because of overlapping text on the *x*-axis, R Code 2.48 changes the values of **Admit** from **Admitted** to **Yes** and **Rejected** to **No**. Students admitted to the graduate program are shaded light gray and those rejected are shaded dark gray in Figure 2.37 on the next page.

R Code 2.48

```
> UCB$Admit <- factor(UCB$Admit, labels = c("Yes", "No"))
> mosaic(~ Dept + Gender + Admit, data = UCB,
+         direction = c("v", "h", "v"),
+         highlighting = "Admit",
```

```
+     highlighting_fill = c("gray80", "gray20"))
```

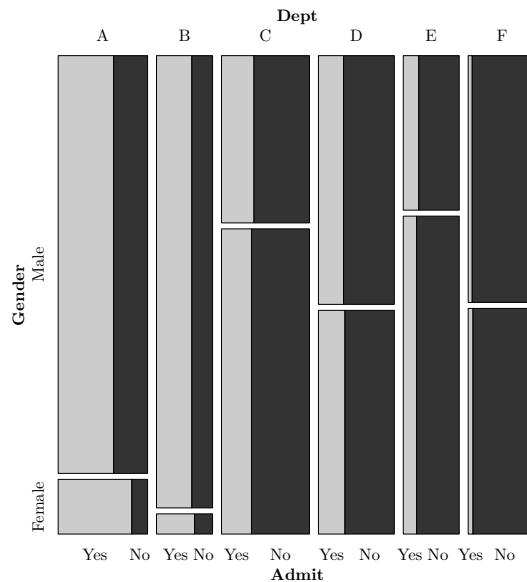


FIGURE 2.37: Mosaic plot showing the percent of females admitted to individual departments are similar to the percent of males admitted to individual departments with the exception of department A where the percent of females admitted was 82% versus 62% for males.

Departments A and B had primarily male applicants. These two departments also admitted a larger percentage of their applicants than the other four departments, thus driving up the overall male admissions. In general, female applicants applied to departments with tougher admission criteria while males applied to departments with easier admission requirements. When looking only at admission by gender, it seems females have a case for discrimination; however, by evaluating admission by department, the percent of females admitted to a particular department is similar to or higher than the percent of males who were admitted.

2.9.2 Lattice Graphs

The R implementation of Trellis displays, developed by Cleveland (1993), is implemented with the package `lattice` (Sarkar, 2015). Lattice displays are graphs that examine higher dimensional structure in data by conditioning on one or more variables. Lattice graphs are implemented in a slightly different fashion from traditional R graphs; however, some readers may find the layout, rendering, and default coloring of lattice graphs more appealing than base R graphs. Lattice graphs are created with a formula syntax. The formula expresses the dependencies between the variables as follows:

```
response ~ predictor | conditioning.variable
```

The expression $y \sim x | z$ is read “ y is modeled as x given z .” Depending on the type of graph, all three components may not need to be specified. Table A.7 on page 909 lists

the arguments for some of the more popular lattice functions. A conditioning variable, generally a factor, defines the subsets plotted in different panels of a lattice graph.

If there is more than one conditioning variable, they are all listed, separated by the multiplication symbol (*).

Example 2.38 Use lattice histograms to compare the body mass index (BMI) for the two treatments (traditional sitting and hamstring stretch stored in Treatment) using the data frame **EPIDURAL**.

Solution: Recall that BMI is typically defined as kg/m^2 . Since the data frame **EPIDURAL** does not contain a BMI variable, one is created:

```
> EPIDURAL$BMI <- EPIDURAL$kg/(EPIDURAL$cm/100)^2 # create BMI variable
> library(lattice) # load lattice package
> histogram(~BMI|treatment, data = EPIDURAL, layout = c(1, 2))
```

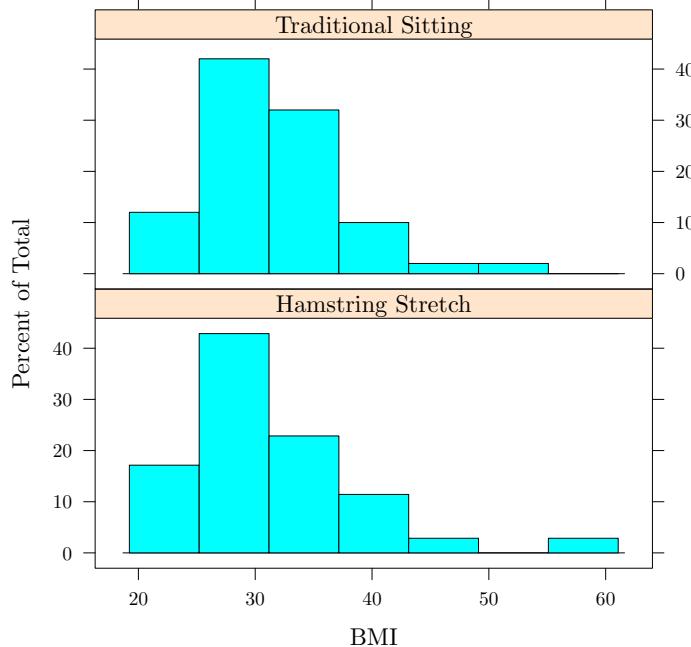


FIGURE 2.38: Comparative histograms of BMI by treatment

The `histogram()` function used the additional argument of `layout = c(1, 2)`. The first value of `layout` determines the number of columns (1) in the lattice graph and the second value determines the number of rows (2) in the lattice graph. This is in contrast to how dimensions are specified in a matrix, which is number of rows by number of columns. The basic shapes of the two histograms shown in Figure 2.38 are quite similar, just as was observed in Example 2.27 on page 139 when the histograms were created using base R graphs. ■

Example 2.39 In Example 2.28 on page 140, side-by-side boxplots were used to compare the BMI for the two treatments. An additional concern is that not only should the distri-

bution of BMI be similar for treatments, but it should also be similar for each physician. Use lattice graphs to create side-by-side boxplots of BMI by treatments given doctor using the data frame **EPIDURAL**.

Solution: R Code 2.49 uses the argument `as.table = TRUE` in the `bwplot()` function to arrange the graphs the way one reads a book. The default arrangement of graphs is to start in the lower left and move to the upper right. This is done so that the graphs appear with the smallest values in the lower left, analogous to a scatterplot.

R Code 2.49

```
> bwplot(treatment ~ BMI | doctor, data = EPIDURAL, as.table = TRUE)
```

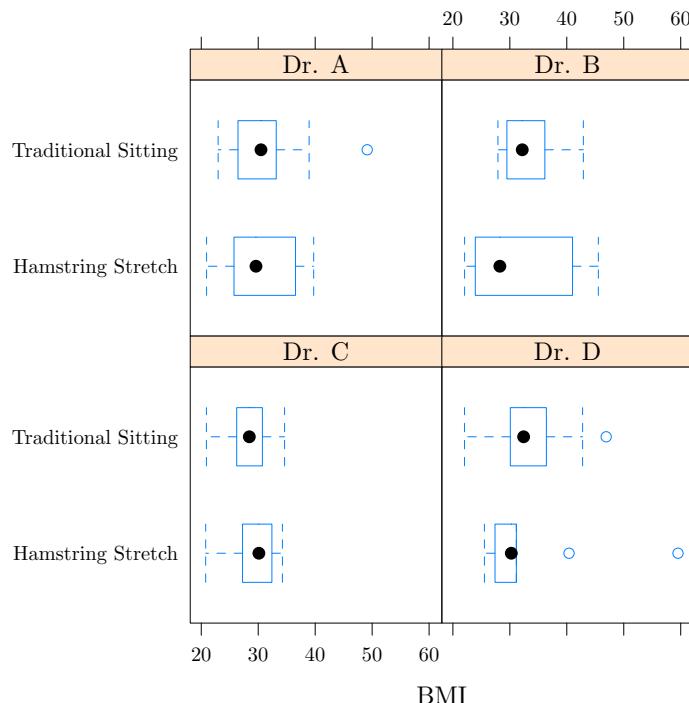


FIGURE 2.39: Side-by-side lattice boxplots of BMI in the traditional sitting and hamstring stretch positions given `doctor`

Since the number of observations for each of the treatments is relatively small (the range is from 6 to 15, see R Code 2.50), it might be a better to look at the data with a **stripplot**.

R Code 2.50

```
> xtabs(~treatment + doctor, data = EPIDURAL)
```

treatment	doctor			
	Dr. A	Dr. B	Dr. C	Dr. D
Hamstring Stretch	8	6	11	10
Traditional Sitting	15	15	10	10

R Code 2.51 is used to create the stripplot of the treatments conditioning on physician illustrated in Figure 2.40 using the function `stripplot()`. The optional argument `jitter = TRUE` adds a small amount of noise to the values in the stripplot so that overlapping values are easier to distinguish. Based on the stripplots shown in Figure 2.40, it seems that Dr. C's patients have a consistently smaller BMI for both treatment positions. Further investigation is needed to see why Dr. C's patients have consistently smaller BMI measurements versus those of the other physicians.

R Code 2.51

```
> stripplot(treatment ~ BMI | doctor, jitter = TRUE, data = EPIDURAL,
+           as.table = TRUE)
```

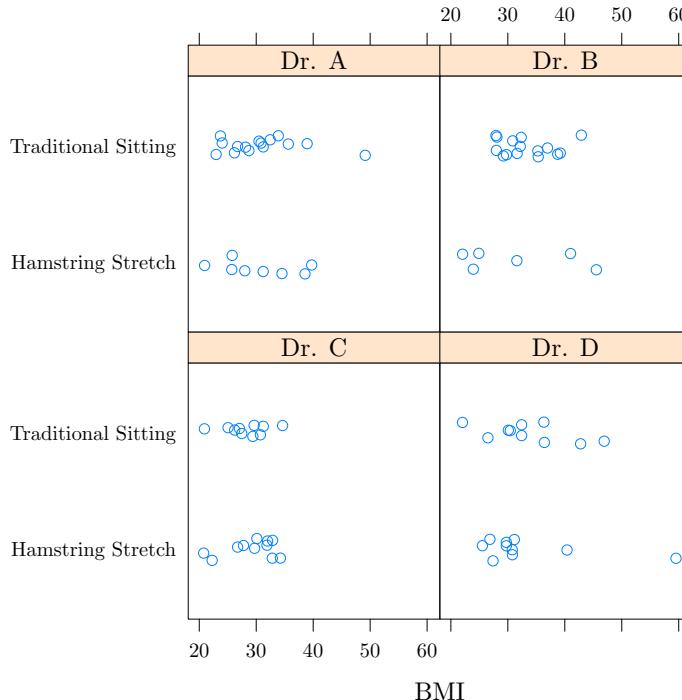


FIGURE 2.40: Side-by-side lattice stripplots of BMI in the traditional sitting and hamstring stretch positions given Doctor

2.9.3 Arranging Several Lattice Graphs on a Single Page

The arrangement of lattice graphs on a single page is different from the arrangement of traditional graphs on a single page. Specifically, approaches involving `par(mfrow=c())`, `par(mfcol=c())`, `split.screen()`, or `layout()` will not work with lattice graphs. Two different approaches can be taken when arranging several lattice graphs on a single page. The first approach discussed is to arrange the graphs in equally sized rectangles based on the dimensions of a matrix. In other words, if one wants to plot six graphs on a single page, it might be accomplished with a 3 by 2 or a 2 by 3 matrix where each position of the matrix

represents a graph. To print each graph, the following structure is used:

```
print(latticegraph, split=c(column, row, number_of_columns,
                           number_of_rows), more=TRUE/FALSE)
```

The value one passes to the argument `more=` is always TRUE until one reaches the last graph, at which point the value passed to `more=` is FALSE since no more graphs need to be printed.

A second approach to producing multiple graphs on a single page is literally to specify the lower left and upper right coordinates for each graph. The lower left of the graph is denoted by the coordinates (0, 0), and the upper right corner is denoted by the coordinates (1, 1). The form for specifying each graph is $(x_{LL}, y_{LL}, x_{UR}, y_{UR})$, where LL and UR denote the lower left and upper right, respectively. To print each graph, the following structure is used:

```
print(latticegraph, position=c(x_LL, y_LL, x_UR, y_UR), more=TRUE/FALSE)
```

As with the first approach, the value one passes to the argument `more=` is always TRUE until one reaches the last graph, at which point the value passed to `more=` is FALSE since no more graphs need to be printed.

Example 2.40 Use lattice graphs to create boxplots of BMI given `doctor`, a scatterplot of `cm` versus `kg` given `doctor`, a histogram of BMI, and a density plot of BMI given `treatment` using the data frame `EPIDURAL`. Show all four graphs on the same page.

Solution: The argument `as.table = TRUE` is used with both the `bwplot()` and the `xyplot()` functions since most people like to read from left to right and top to bottom. The four graphs are created and stored in variables named `graph1`, `graph2`, `graph3`, and `graph4`, respectively. By splitting the graph into a 2 by 2 matrix with the argument `split = c(show_row_i, show_col_j, 2, 2)`, see R Code 2.52 for complete details, or by specifying the literal position for each of the four graphs, see R Code 2.53 for complete details, one can reproduce Figure 2.41 on the facing page.

R Code 2.52

```
> graph1 <- histogram(~BMI, data = EPIDURAL)
> graph2 <- xyplot(cm ~ kg|doctor, data = EPIDURAL, as.table = TRUE)
> graph3 <- densityplot(~BMI|treatment, data = EPIDURAL, as.table = TRUE)
> graph4 <- bwplot(~BMI|doctor, data = EPIDURAL, as.table = TRUE)
> print(graph1, split=c(1, 2, 2, 2), more = TRUE)    # Lower left
> print(graph2, split=c(2, 2, 2, 2), more = TRUE)    # Lower right
> print(graph3, split=c(1, 1, 2, 2), more = TRUE)    # Upper left
> print(graph4, split=c(2, 1, 2, 2), more = FALSE)   # Upper right
```

R Code 2.53

```
> print(graph1, position=c(0, 0, 0.5, 0.5), more = TRUE)    # Lower left
> print(graph2, position=c(0.5, 0, 1, 0.5), more = TRUE)    # Lower right
> print(graph3, position=c(0, 0.5, 0.5, 1), more = TRUE)    # Upper left
> print(graph4, position=c(0.5, 0.5, 1, 1), more = FALSE)   # Upper right
```

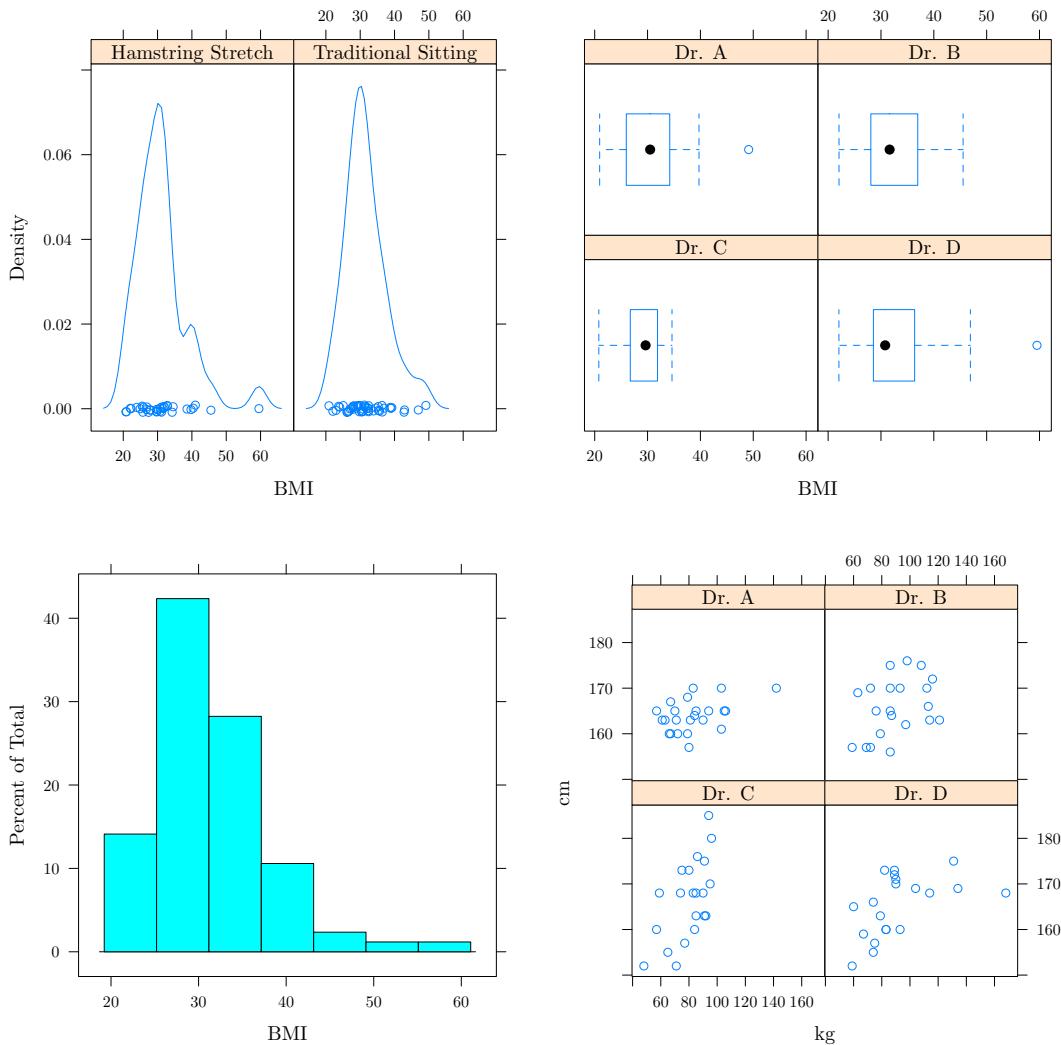


FIGURE 2.41: Arrangement of four different lattice graphs on the same page

2.9.4 Panel Functions

Panel functions can be used to add additional features to a lattice graph. For example, given a lattice x - y plot, one can add a line using the panel function `panel.abline()`. For a list of available panel functions in R, type `?panel.functions` at the R prompt.

Example 2.41 Create a lattice x - y plot of `cm` versus `kg` given `doctor` using the data frame `EPIDURAL`. Use panel functions to superimpose the ordinary least squares line and a least-trimmed squares line over the x - y plot.

Solution: R Code 2.54 creates Figure 2.42 on the next page. The package `MASS` is loaded for the `lqs()` function.

R Code 2.54

```
> library(MASS) # Needed for lqs
```

```
> xyplot(cm ~ kg | doctor, data = EPIDURAL, as.table = TRUE,
+         panel=function(x, y) {
+             panel.xyplot(x, y)                                     # x-y plot
+             panel.abline(lm(y ~ x))                            # lm line
+             panel.abline(lqs(y ~ x), col = 3, lty = 2, lwd = 2) # lqs line
+         })
})
```

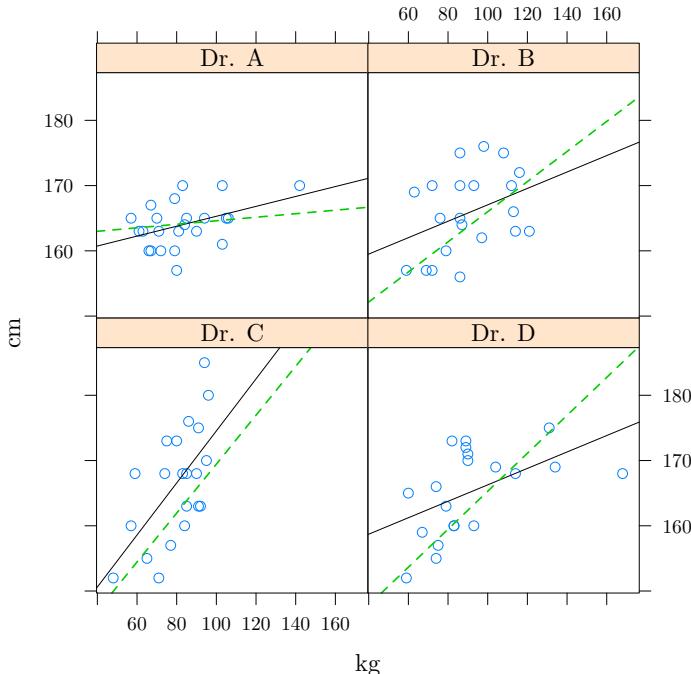


FIGURE 2.42: x - y plot of height (cm) versus weight (kg) given physician (doctor) with superimposed least squares (solid lines) and least-trimmed squares (dashed lines)

Another approach, illustrated in R Code 2.55, is to create a panel function that will superimpose the least squares and least-trimmed squares lines on an x - y plot and then to call that function within the `xyplot()`.

R Code 2.55

```
> panel.scatreg <- function(x, y)                                # name function
+ {
+     panel.xyplot(x, y)                                         # make x-y plot
+     panel.abline(lm(y ~ x), lwd = 2)                           # lm line
+     panel.abline(lqs(y ~ x), col = 3, lty = 2, lwd = 2) # lqs line
+ }
> xyplot(cm~kg|doctor, data = EPIDURAL, as.table = TRUE,
+         panel = panel.scatreg)
```

Either R Code 2.54 or R Code 2.55 can be used to create Figure 2.42. The dashed lines (`lty = 2`) in Figure 2.42 are the least-trimmed squares lines. ■

2.9.5 Graphics with ggplot2

Graphs in `ggplot2` are created in layers and follow a particular “grammar.” Specifically, a `ggplot2` graph is created by mapping the data, which must be a data frame, to a geometric object (`geom`) via an aesthetic (`aes`) function. The `aes()` function describes how variables in the data frame are mapped to visual properties (aesthetics) that one perceives on a plot. The geometric object determines how the data is viewed (i.e., as points or as a histogram) and adds these elements to the plot with one of the `geom` functions such as `geom_point()` or `geom_histogram()`. Several of the `geoms` and their aesthetics used in the text are provided in Table 2.5. For a complete list of `geoms`, see the online documentation for `ggplot2` at <http://docs.ggplot2.org/current/>.

Table 2.5: Geoms and commonly used aesthetics

Geom	Description	Example	Aesthetics
<code>geom_abline()</code>	Line	R Code 2.72, Figure 2.60	<code>color, linetype</code>
<code>geom_area()</code>	Area plot	R Code 2.66, Figure 2.53	<code>x</code>
<code>geom_bar()</code>	Barplot	R Code 2.62, Figure 2.49	<code>x, color, fill</code>
<code>geom_boxplot()</code>	Boxplot	R Code 2.57, Figure 2.44	<code>x, y, fill</code>
<code>geom_density()</code>	Density	R Code 2.64, Figure 2.51	<code>x, y, fill, linetype</code>
<code>geom_dotplot()</code>	Dotplot	R Code 2.69, Figure 2.57	<code>x, y, color, fill</code>
<code>geom_histogram()</code>	Histogram	R Code 2.61, Figure 2.48	<code>x, color, fill</code>
<code>geom_point()</code>	Symbols	R Code 2.59, Figure 2.46	<code>x, y, color, shape</code>
<code>geom_polygon()</code>	Polygon	R Code 2.66, Figure 2.53	<code>x, y, color fill</code>
<code>geom_smooth()</code>	Smoothed mean	R Code 2.70, Figure 2.58	<code>x, y, color, fill</code>
<code>geom_text()</code>	Text	R Code 2.73, Figure 2.61	<code>x, y, label, angle</code>
<code>geom_violin()</code>	Violin plot	R Code 2.67, Figure 2.55	<code>x, y, color, fill</code>

The default theme for `ggplot2` is a gray background, which works well with the predefined colors `ggplot2` uses. Since this text is published in black and white, an attempt is made to use a black-and-white theme along with gray shades, which produce easier-to-read graphs. Consider how the left graphs in Figure 2.43 on the following page show the default gray theme and use predefined colors that are mapped to treatment levels (although the book only shows grayscale) versus the right graphs that use a black-and-white theme with an intentional grayscale used to map the levels of treatment to `color`. When creating graphs that are to be viewed or printed in color, use the default gray theme without the `scale_fill_grey()` or `scale_color_grey()` functions for a more colorful result. Using the black-and-white theme with a grayscale for different `treatment` values makes it easier to see the differences between boxplots used with the `hamstring` stretch and those used with traditional sitting than does the default gray theme and predefined colors when printed in black and white, as seen in Figure 2.43 on the next page. The default gray theme is used with `ggplot2` unless a different theme such as `theme_bw()` is issued with a call similar to that shown in R Code 2.56. The default arguments for `scale_fill_grey()` are `scale_fill_grey(start = 0.2, end = 0.8)`, which produces both a dark gray and a light gray. The boxes in the right plot of Figure 2.43 on the following page were created using the arguments `scale_fill_grey(start = 0.4, end = 0.6)`.

R Code 2.56

```

> previous_theme <- theme_set(theme_bw())           # set black-and-white theme
> p <- ggplot(data = EPIDURALF)                  # Empty plot
> p1 <- p + geom_boxplot(aes(x = treatment, y = kg, fill = treatment)) +
+   guides(fill = FALSE) + facet_grid(doctor ~ .)
> p1                                              # Left plot
> p2 <- p1 + scale_fill_grey(start = .4, end = .6) # Right plot
> p2                                              # Restore original theme

```

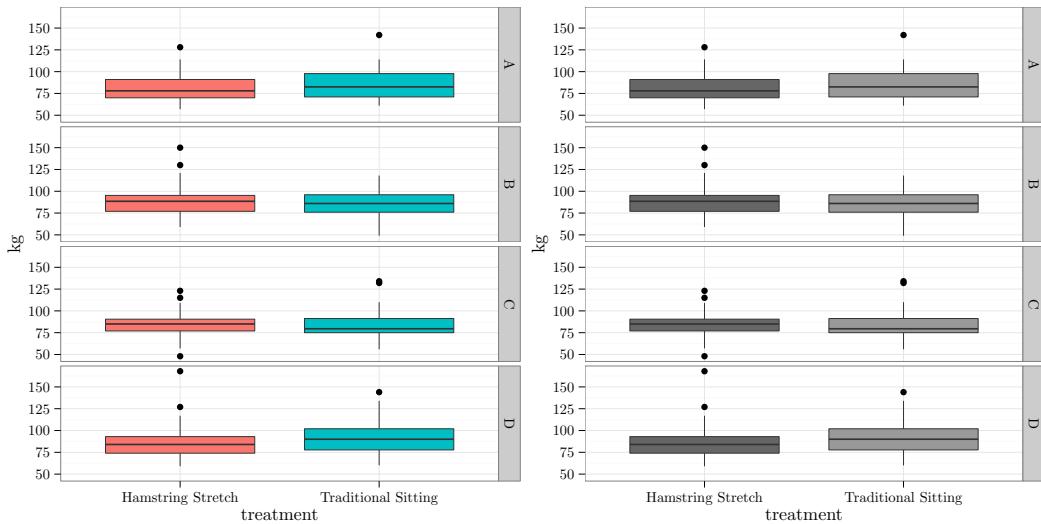


FIGURE 2.43: The left plot shows boxplots of the weight of patients for each of the two treatments for each of the physicians using the default gray theme with the variable `treatment` mapped to predefined colors (red and cyan in e-book version of this text). The right plot shows the same boxplots with `treatment` mapped to a grayscale with a black-and-white theme.

A `ggplot2` graph is a layered plot built up with the `+` operator. Consider how the graphs in Figure 2.44 on page 172 are created with three layers (`geom_boxplot()`, `guides()`, and `facet_grid()`) from R Code 2.57. The second line in R Code 2.57 creates a new plot for the data set `EPIDURALF`. At this point, no information has been provided to indicate how the data should be displayed, so nothing is drawn; however, the result, a `ggplot` object, is assigned to the symbol `p` so that different layers can be added to the plot using the `+` operator. The first layer added to `p` creates the side-by-side boxplots shown in the left plot of Figure 2.44 on page 172. The `geom_boxplot()` call is given aesthetic arguments of `x = treatment`, which maps the levels of the `treatment` variable to the *x*-axis; `y = kg`, which maps the weight of the patients to the *y*-axis; and `fill = treatment`, which fills the boxplots with different colors (visible in the e-book version of this text) based on the values in `treatment`. Since the *x* axis has labels indicating the left boxplot plot is of the weights for patients administered an epidural while using the hamstring stretch position and the right boxplot is of the weights for patients administered an epidural using the traditional

sitting position, the legend (automatically generated) shown on the right side of the first plot is not needed.

Legends and scales are referred to as guides in `ggplot2`. The legends and axes of a `ggplot2` graph are automatically generated based on the scales and geoms used in the plot. Scales control the mapping from data to aesthetics. That is, scales take data and create something the user perceives visually such as color, position, shape, or size. Scales also generate the means to interpret the plot in the form of axes and legends. Every aesthetic has a default scale that is added to the plot whenever the aesthetic is used. When the aesthetic `fill = treatment` is used in R Code 2.57, a legend is automatically generated for the variable `treatment`. The automatic generation of axes and legends is different from most other graphing paradigms where the user is responsible for adding the appropriate axes and legends. To override the default guides, one can use the function `guides()`. The second layer, shown in the middle plot of Figure 2.44 on the following page, removes the legend with the call `guides(fill = FALSE)`.

Finally, the third layer, shown in the right plot of Figure 2.44 on the next page, breaks the data into four subsets (one for each physician) and creates side-by-side boxplots for the weights of the patients according to the physician who administered the epidural with a call to `facet_grid()`.

Creating subsets of data to be plotted with `ggplot2` is called facetting. Faceting in `ggplot2` is performed with a call to the function `facet_grid()`. The result from facetting is similar to that of multipanels used with the `lattice` package. The default argument for `facet_grid()` is `facet_grid(. ~ .)`, which returns a single panel where neither the rows nor the columns are faceted. To create a single column with multiple rows, one would use the call `facet_grid(row ~ .)`, where `row` is the name of a categorical variable included in the data frame one is using. In a similar fashion, a single row with multiple columns is created with the call `facet_grid(. ~ col)` where `col` is the name of a categorical variable included in the data frame one is using. The call `facet_grid(row ~ col)` creates the number of levels in `row` \times the number of levels in `col` subplots.

R Code 2.57

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = treatment, y = kg,
+                                         fill = treatment)) # Empty plot
> p1 <- p + geom_boxplot() + scale_fill_grey(start = .3, end = .7)
> p1                                         # Left plot
> p2 <- p1 + guides(fill = FALSE)           # removes legend
> p2                                         # Center plot
> p3 <- p2 + facet_grid(doctor ~ .)          # splits boxplot by physician
> p3                                         # Right plot
> theme_set(previous_theme)                  # Restore original theme
```

Consider R Code 2.58 used to create Figure 2.45 on the next page, which shows a density plot of the variable `kg`, two density plots of `kg` split by treatment levels, and finally eight density plots of `kg` created from the eight subsets of data created from facetting `treatment` (two levels) and `doctor` (four levels).

R Code 2.58

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = kg)) # Empty plot
> p1 <- p + geom_density()
```

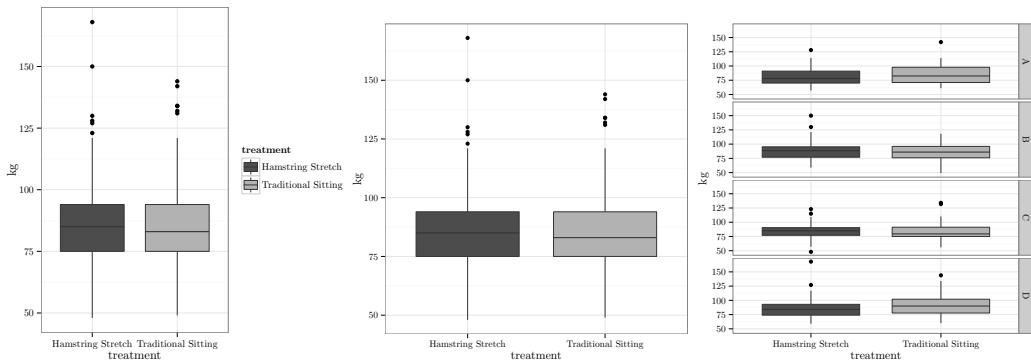


FIGURE 2.44: Plots showing the results of adding different layers to p from R Code 2.57. The left plot shows the addition of the first layer; the center plot shows the addition of the second layer; and the right plot shows the addition of the third layer.

```
> p1 # density plot of kg
> p2 <- p1 + facet_grid(treatment ~ .)
> p2 # density plot of kg split by treatment
> p3 <- p1 + facet_grid(treatment ~ doctor)
> p3 # density plot of kg split by treatment and physician
> theme_set(previous_theme) # Restore original theme
```

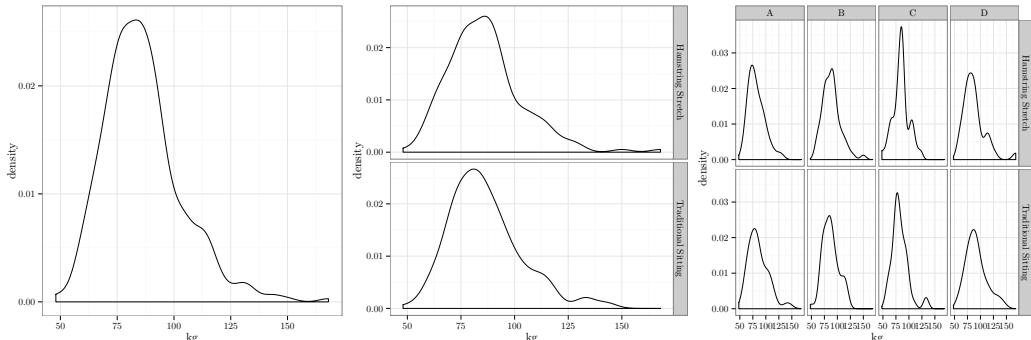


FIGURE 2.45: Left: density plot of the variable kg , center: density plots of kg split by treatment levels, right: eight density plots of kg created from facetting `treatment` (two levels) and `doctor` (four levels)

Example 2.42 Use the data frame `EPIDURALF` to create a scatterplot of weight (kg) versus height (cm). Use the `color=` and `shape=` arguments to map the different levels of `ease` onto the scatterplot. Change the default labels on the x and y axes to read `Height (cm)` and `Weight (kg)`, respectively, using the `labs()` function. Finally, change the default legend guide from `ease` to `Ease of Palpatating Patient`.

Solution: The scatterplots in Figure 2.46 on the facing page and the left scatterplot in Figure 2.47 on page 174 are built by mapping the aesthetics `cm`, `kg`, `color`, and `shape` to

the geometric object `point`. The right scatterplot in Figure 2.47 on the following page adds additional layers to the scatterplot with calls to `labs`, which allows one to change the default labels along the axes as well as add a title, and `guides`, to set guides for different scales. The first line in R Code 2.59 switches the default `ggplot2` gray theme to a black-and-white theme. Since the book is published in black and white, the `scale_color_grey()` command is also used to display points mapped to `color` and `shape` with gray shades.

R Code 2.59

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = cm, y = kg))  # Empty plot
> p1 <- p + geom_point(aes(color = ease)) + scale_color_grey()
> p1                                         # Left scatterplot
> p2 <- p + geom_point(aes(shape = ease))
> p2                                         # Right scatterplot
> theme_set(previous_theme)                  # Restore original theme
```

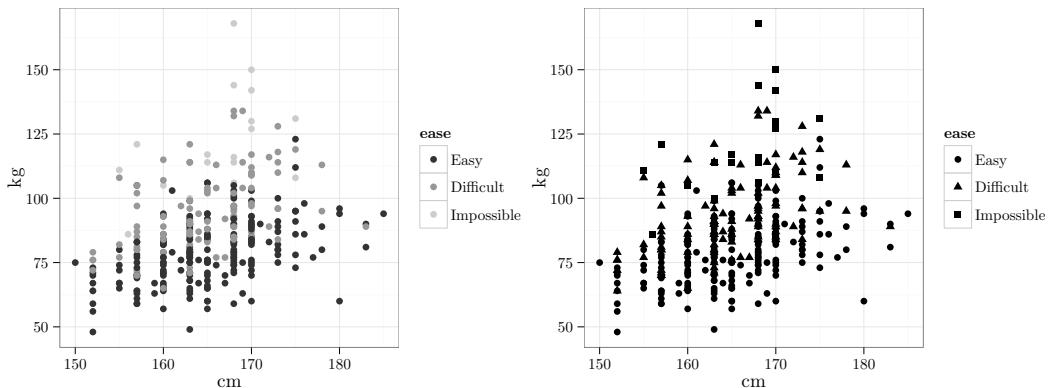


FIGURE 2.46: The left plot maps the `cm` and `kg` values in the data frame `EPIDURALF` to points in the plot that are mapped to different colors (visible in the e-book version of this text) based on the values in `ease`. The right plot also maps the `cm` and `kg` values to points but this time the shapes of the points are determined by the mapping of the levels of `ease`.

In R Code 2.60, the text above the legend is changed in the right plot of Figure 2.47 on the following page by using `guide_legend(TEXT)`. A text string is assigned to `TEXT`, and `guide_legend(TEXT)` is assigned to both `color` and `shape`. Note that the variable `ease` is passed to the aesthetic `color=` as well as `shape=`. Consequently, specifying a `guide_legend()` for only one of `color` or `shape` when both are used as aesthetic arguments results in two legends with the desired text appearing over only one of either `color` or `shape` depending on which one was used with the `guide_legend()` argument.

R Code 2.60

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = cm, y = kg, color = ease,
+                                         shape = ease))  # Empty plot
> TEXT <- "Ease of\nPalpating\nPatient"        # \n returns a new line
```

```

> p1 <- p + geom_point() + scale_color_grey()
> p1
# Left plot
> p2 <- p1 + guides(color = guide_legend(TEXT),
+ shape = guide_legend(TEXT)) + labs(x = "Height (cm)",
+ y = "Weight (kg)")
# Right plot
> p2
# Restore original theme

```

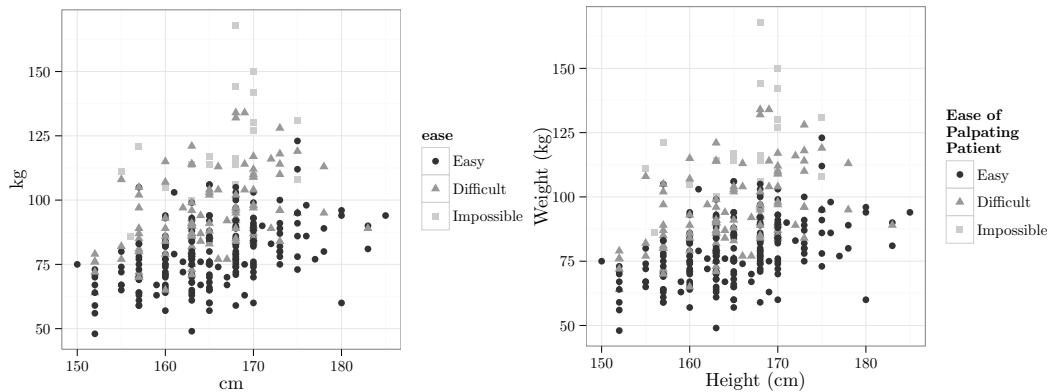


FIGURE 2.47: The left graph maps the `cm` and `kg` values to points where the levels of `ease` are further mapped to both different colors (visible in the e-book version of this text) and shapes. The right graph is identical to the left graph with additional layers added to change the appearance of the axes and legend labels.

Example 2.42 created a scatterplot by mapping values to the x - y plane. Some univariate graphs, such as histograms and barplots, do not require a `y=` aesthetic mapping. R Code 2.61 creates three slightly different histograms. The first plot in Figure 2.48 on the next page shows a histogram with bins of width 10. The default for histograms in `ggplot2` is to show the frequency/counts of each bin along the y -axis. The middle plot of Figure 2.48 shows a gray histogram created by assigning "gray65" to `fill=` and setting the bin width to 5 with the argument `binwidth = 5`. The third plot in Figure 2.48 on the facing page shows a density histogram with a superimposed density curve that also has a bin width of 5. Although the scale of the y -axis changes in the third plot, the shape of the histogram does not change from the second plot to the third plot of Figure 2.48 on the next page. The y -axis scale for the third plot is changed from the default `count` (`y = ..count..`) to `density` by using the aesthetic `y = ..density..` inside `geom_histogram()`.

R Code 2.61

```

> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = kg)) # Empty plot
> p1 <- p + geom_histogram(binwidth = 10)      # Left histogram
> p2 <- p + geom_histogram(binwidth = 5, fill = "gray65") # Center histogram
> p2
# Restore original theme

```

```
> p3 <- p + geom_histogram(aes(y = ..density..), binwidth = 5,
+                           fill = "gray65") + geom_density() +
+   labs(x = "Weight (kg)", y = "density")
> p3                                         # Right histogram
> theme_set(previous_theme)                  # Restore original theme
```

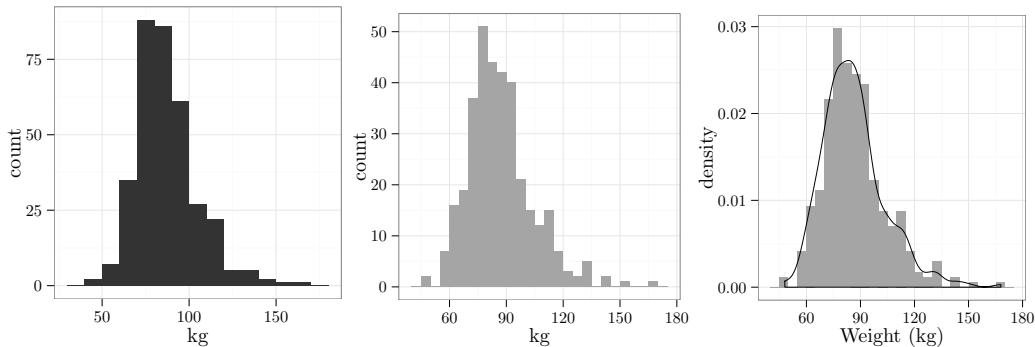


FIGURE 2.48: First plot shows a histogram of the variable `kg` with bins of width ten. The second plot shows a gray histogram with bins of width five. The third plot shows a density histogram with bins of width five with a superimposed density curve.

Barplots are often confused with histograms, possibly because both barplots and histograms report frequency/counts along the *y*-axis. The difference is that histograms are used with quantitative variables and barplots are used with categorical variables. Consider R Code 2.62, which creates barplots for the variable `ease` from the **EPIDURALF** data frame. The default `position`= argument for `geom_bar()` is `position = "stack"`, which is shown in the middle graph of Figure 2.49 on the next page. Side-by-side bars can be created with `position = "dodge"`. To create a barplot where each bar represents 100% of some variable, use the argument `position = "fill"`.

R Code 2.62

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = ease)) # Empty plot
> p1 <- p + geom_bar(position = "stack") +
+   theme(axis.text.x = element_text(angle = 75, vjust = 0.5))
> p1                                         # Left barplot
> p2 <- p + geom_bar(aes(fill = doctor), position = "stack") +
+   scale_fill_grey() +
+   theme(axis.text.x = element_text(angle = 75, vjust = 0.5))
> p2                                         # Center stacked barplot
> p3 <- p + geom_bar(aes(fill = doctor), position = "dodge") +
+   scale_fill_grey() +
+   theme(axis.text.x = element_text(angle = 75, vjust = 0.5))
> p3                                         # Right dodged barplot
> theme_set(previous_theme)                  # Restore original theme
```

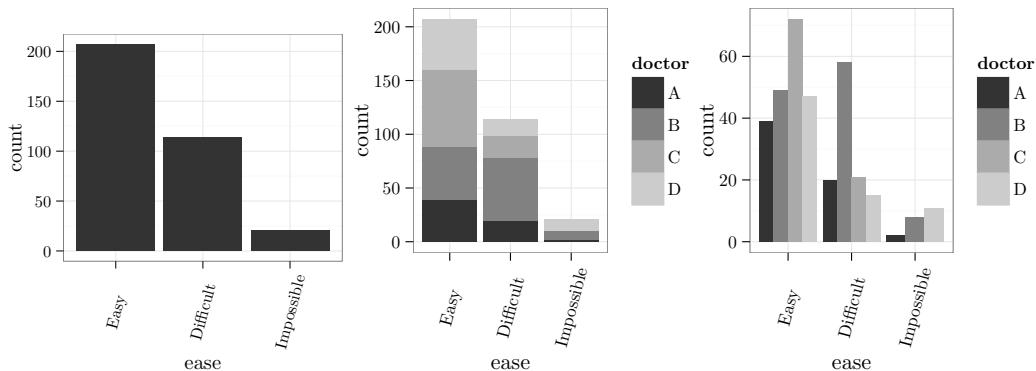


FIGURE 2.49: First plot shows a barplot where each bar represents the number of patients in each of the levels of `ease`. The second barplot color codes the patients classified by each physician within each bar (visible in the e-book version of this text). The last plot shows side-by-side bars for the number of patients each physician classified as easy, difficult, and impossible to palpate.

Example 2.43 Use the data frame `EPIDURALF` to create a `ggplot2` barplot where the fraction of patients classified in the variable `ease` are shown for each physician. Create a second barplot using `facet_grid()` to see if the fraction of patients each physician classified as easy, difficult, and impossible are similar for the treatments hamstring stretch and traditional sitting.

Solution: The fraction of patients each physician classified as easy, difficult, and impossible are not the same. Doctor “B” classified a smaller fraction of his patients as easy to palpate than did the other three physicians. Doctor “C” did not classify any of his patients as impossible to palpate. Further differences in the fraction of patients each physician classified as easy, difficult, and impossible to palpate can be seen in the left plot of Figure 2.50 on page 178. The remaining three plots in Figure 2.50 on page 178 suggest that each physician classifies a similar fraction of patients as easy, difficult, and impossible to palpate regardless of treatment position. R Code 2.63 maps `doctor` to the *x*-axis for the top two plots and `treatment` to the *x*-axis for the bottom two plots of Figure 2.50 on page 178. The labels for the levels of `treatment` are shortened so that they do not overlap in the bottom right plot of Figure 2.50 on page 178.

R Code 2.63

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = doctor, fill = ease))  # empty plot
> p1 <- p + geom_bar(position = "fill") +
+     labs(y = "") +
+     scale_fill_grey()
> p1                                         # top left barplot
> p2 <- p1 + guides(fill = guide_legend("Ease of\nPalpating\nPatient")) +
+     labs(y = "") +
+     facet_grid(treatment ~ .)
> p2                                         # top right barplot
> pn <- ggplot(data = EPIDURALF, aes(x = treatment, fill = ease))
>
> p3 <- pn + geom_bar(position = 'fill') + facet_grid(doctor ~ .) +
+     
```

```

+     labs(y = "", x = "") +
+     guides(fill = guide_legend("Ease of\nPalpating\nPatient")) +
+     scale_fill_grey()
> p3                                         # bottom left barplot
> EPIDURALF$treatment <- factor(EPIDURALF$treatment,
+                                   labels = c("HS", "TS")) # shorten labels
> p4 <- ggplot(data = EPIDURALF, aes(x = treatment, fill = ease)) +
+     geom_bar(position = 'fill') + facet_grid(. ~ doctor) +
+     labs(y = "", x = "") +
+     guides(fill = guide_legend("Ease of\nPalpating\nPatient")) +
+     scale_fill_grey()
> p4                                         # bottom right barplot
> EPIDURALF$treatment <- factor(EPIDURALF$treatment,
+                                   labels = c("Hamstring Stretch", "Traditional Sitting")) # reset
> theme_set(previous_theme)                  # restore original theme

```

R Code 2.58 on page 171 created density plots of the variable `kg` based on various subsets of the data frame `EPIDURALF` by faceting first on `treatment`, then faceting on both `treatment` and `doctor`. It is also possible to put more than a single density estimate in a single plot. R Code 2.64 creates density estimates for the BMI values of the patients administered an epidural based on the physicians' assessment of the level of difficulty of palpating the patients. The variable `BMI` is mapped to the *x*-axis, and the variable `ease` is mapped to the aesthetic `fill=`. The density plots are filled based on the variable `ease`. In this case, the first density estimate to be filled corresponds to the level `Easy`, followed by `Difficult` and then `Impossible`. Since the fill colors are by default opaque, the outlines of the density estimates for the BMI values of `Easy` and `Difficult` are partially obscured. To change the transparency of the fill colors, one can use the argument `alpha=`. An `alpha` value of 1 corresponds to completely opaque coloring while an `alpha` value of 0 corresponds to completely transparent coloring. Figure 2.51 on page 179 was created from R Code 2.64 where the `alpha` value for the fill aesthetic was 0.2.

R Code 2.64

```

> previous_theme <- theme_set(theme_bw()) # set black-and-white theme
> EPIDURALF$BMI <- EPIDURALF$kg/(EPIDURALF$cm/100)^2 # Create BMI
> p <- ggplot(data = EPIDURALF, aes(x = BMI, fill = ease)) # Empty plot
> p1 <- p + geom_density(alpha = 0.2) + labs(x = "Body Mass Index") +
+     guides(fill = guide_legend("Ease of\nPalpating\nPatient")) +
+     scale_fill_grey()
> p1 # Left density plots
> p2 <- p1 + facet_grid(treatment ~ .)
> p2 # Right density plots
> theme_set(previous_theme) # Restore original theme

```

2.9.5.1 Shading a Region of a Density Curve

To shade a region (not necessarily the entire area beneath the density curve) of a density plot, one may use either `geom_area()` or `geom_polygon()`. The distinction between `geom_area()` and `geom_polygon()` is that `geom_area()` draws an area plot, which is a line filled to the *y*-axis, while `geom_polygon()` draws polygons that are filled paths. For shading subregions under a density curve, `geom_area()` involves less thought since one just

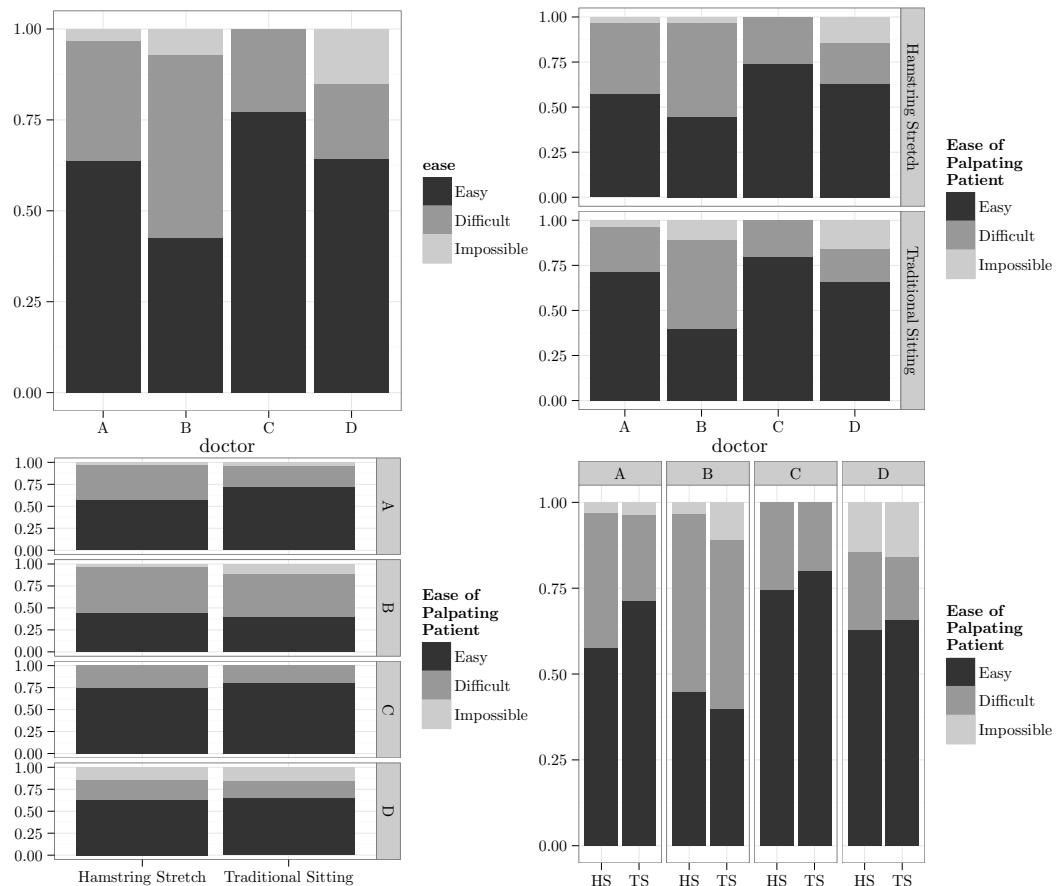


FIGURE 2.50: The top left plot shows the fraction of patients each physician classified as easy, difficult, and impossible to palpate. The top right plot shown the fraction of patients each physician classified as easy, difficult, and impossible to palpate conditioning on the treatment (hamstring stretch or traditional sitting). The bottom left plot shows the fraction of patients each physician classified as easy, difficult, and impossible to palpate according to treatment while row faceting on `doctor`. The bottom right plot shows the fraction of patients each physician classified as easy, difficult, and impossible to palpate according to treatment while column faceting on `doctor`.

specifies the x region one wants to shade in contrast to using `geom_polygon()` where one needs to make sure the path starts and finishes where desired.

Figure 2.52 on the facing page is created from R Code 2.65, which shows the results of using `geom_area()` and `geom_polygon()` on the same data set. Then, it shows how `geom_polygon()` can be used to create an area plot. That is, the third plot in Figure 2.52 on the next page created with a call to `geom_polygon()` appears identical to the first plot in Figure 2.52 on the facing page created with a call to `geom_area()`. The third plot in Figure 2.52 on the next page is filled in by tracing the points $(0, 0)$, $(0, 3)$, $(1, 0)$, $(2, 4)$, $(2, 0)$, and $(0, 0)$.

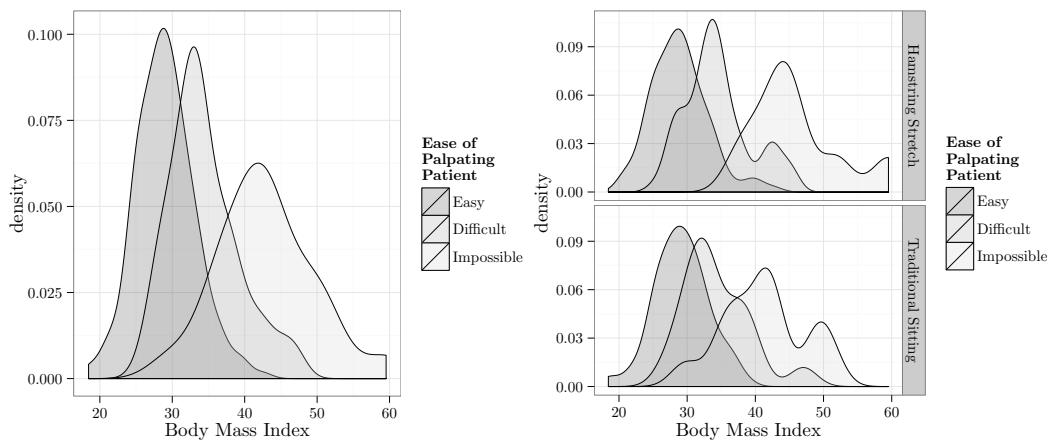


FIGURE 2.51: Left plot shows three density estimates of body mass index based on the physicians' assessment of ease of palpating the patient. The right plot shows three density estimates of body mass index also based on the physicians' assessment of ease of palpating the patient after subsetting the data into two groups by facetting on the variable `treatment`.

R Code 2.65

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> DF <- data.frame(x = c(0, 1, 2), y = c(3, 0, 4))
> p <- ggplot(data = DF, aes(x = x, y = y))
> p + geom_area()
> p + geom_polygon()
> p + geom_polygon(data = rbind(c(0, 0), DF, c(2, 0))) # Restore original theme
```

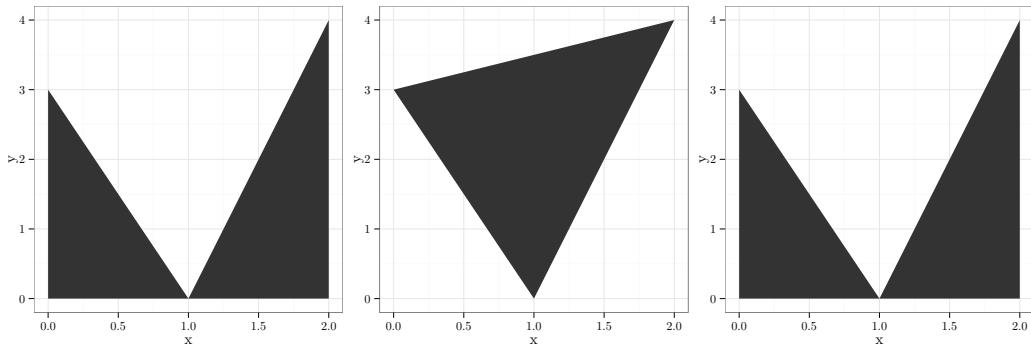


FIGURE 2.52: The left and center plot show the distinction between an area plot and a polygon with a filled path, respectively. The far right plot shows how an area plot can be created with a polygon-filled path by augmenting the start and end points of the polygon.

Example 2.44 Create a density plot of the body mass index values based on the infor-

mation in the data frame **EPIDURALF**. Use an area plot as well as a polygon to shade body mass values that are greater than or equal to 40.

Solution: Since the data frame **EPIDURALF** does not have a variable for body mass index, one is created and stored as **BMI**. The base function **density()** is applied to the values in **BMI**, and the results are stored in the object **dens**, which records the **x** and **y** coordinates for the estimated density curve. Since **ggplot2** requires a data frame, the information in **dens** is stored in a data frame named **df.dens** for later use when a subset of the values in **df.dens** (those greater than or equal to 40) are used to create an area plot with the function **geom_area()**. The area under the density plot for BMI values greater than or equal to 40 is also shaded with **geom_polygon**; however, care must be taken to ensure the same area is shaded by making sure the points given to the polygon enclose the same area as those provided to the area plot.

R Code 2.66

```
> previous_theme <- theme_set(theme_bw())    # set black-and-white theme
> EPIDURALF$BMI <- EPIDURALF$kg/(EPIDURALF$cm/100)^2  # Create BMI
> dens <- density(EPIDURALF$BMI)
> df.dens <- data.frame(x = dens$x, y = dens$y)
> p <- ggplot(data = EPIDURALF, aes(x = BMI)) +
+     geom_density(fill = "gray", alpha = 0.4)
> p1 <- p + geom_area(data = subset(df.dens, x >= 40 &
+     x <= max(EPIDURALF$BMI)), aes(x = x, y = y)) +
+     labs(x = "Body Mass Index", y = "", title = "geom\\_area()")
> p1 # Left density plot
> p2 <- p + geom_polygon(data = rbind(c(min(df.dens$x[df.dens$x >= 40]), 0),
+     subset(df.dens, x >= 40 & x <= max(EPIDURALF$BMI)),
+     c(max(EPIDURALF$BMI), 0)), aes(x = x, y = y)) +
+     labs(y = "", x = "Body Mass Index", title = "geom\\_polygon()")
> p2 # Right density plot
> theme_set(previous_theme) # Restore original theme
```

2.9.5.2 Violin Plots

A violin plot is a standard kernel density plot that has been rotated around the **x**-axis. Like boxplots, violin plots can be used to compare the distribution of a quantitative variable for several levels of a qualitative variable; however, unlike boxplots, violin plots do not hide multi-modality. Consider how the right plot of Figure 2.54 on the facing page (a violin plot) is created from reflecting the left plot (a kernel density) around the **x**-axis. By default, violin plots in **ggplot2** have a vertical orientation with factors appearing on the **x**-axis.

The default argument for **scale**= when using **geom_violin()** is "area", which ensures the area for each side-by-side violin plot is the same. If the number of observations in the side-by-side violin plots are different, use **scale = "count"** so that the areas of the side-by-side violin plots are proportionally scaled to the number of observations in each violin plot. R Code 2.67 on the next page is used to create Figure 2.55 on page 182. The left side of Figure 2.55 on page 182 creates side-by side violin plots of body mass index values according to the physicians' assessments of ease of palpating a patient using **scale = "area"**. Since the numbers of patients classified as easy, difficult, and impossible to palpate are 207, 114, and 21, respectively, the right plot of Figure 2.55 on page 182 is created using **scale = "count"** so that the violin plots are scaled proportionally according to the total number of observations.

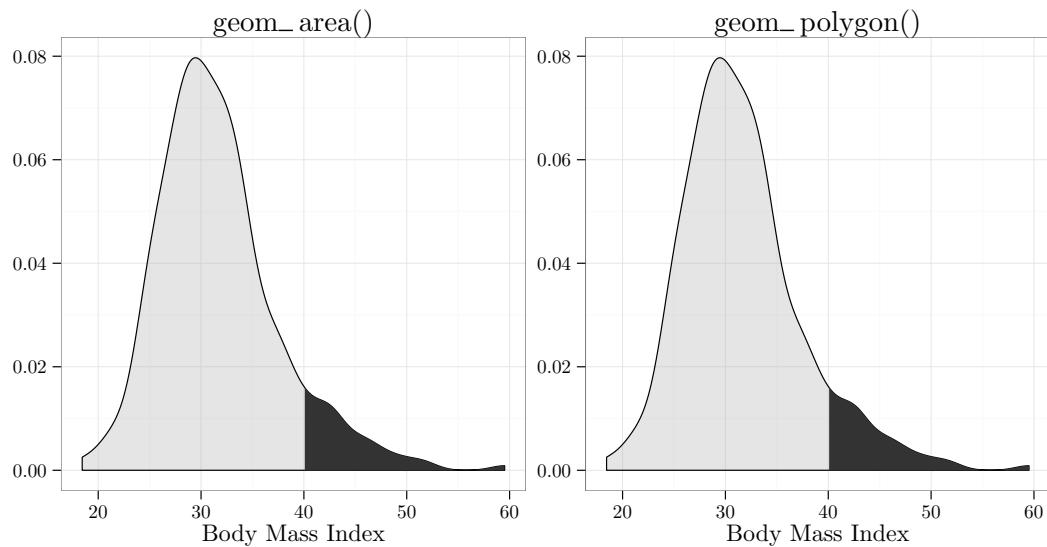


FIGURE 2.53: Density plots that shade BMI values greater than or equal to 40 using two different approaches

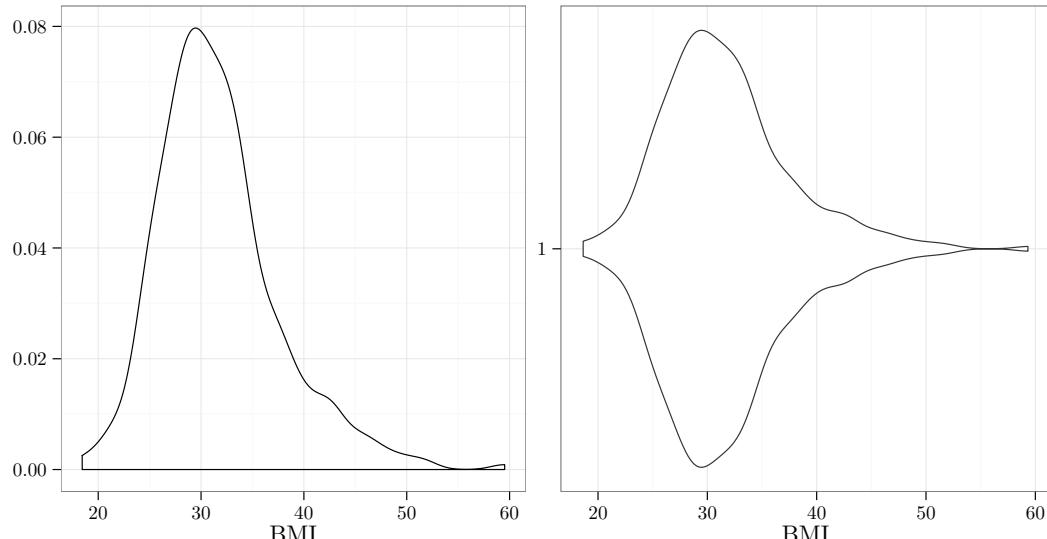


FIGURE 2.54: The left plot is a kernel density plot of the body mass index (BMI) from the data frame `EPIDURALF`. The right plot shows a violin plot of the body mass index (BMI) from the data frame `EPIDURALF`.

R Code 2.67

```
> previous_theme <- theme_set(theme_bw())    # set black-and-white theme
> EPIDURALF$BMI <- EPIDURALF$kg/(EPIDURALF$cm/100)^2  # Create BMI
> p <- ggplot(data = EPIDURALF, aes(x = ease, y = BMI, fill = ease)) +
+     guides(fill = FALSE) + scale_fill_grey()
> p1 <- p + geom_violin(scale = "area") +
```

```
+     labs(title = "Area", x="", y = "Body Mass Index (BMI)")
> p1    # Left area violin plots
> p2 <- p + geom_violin(scale = "count") +
+     labs(title = "Count", x="", y = "Body Mass Index (BMI)")
> p2    # Right count violin plots
> theme_set(previous_theme) # Restore original theme
```

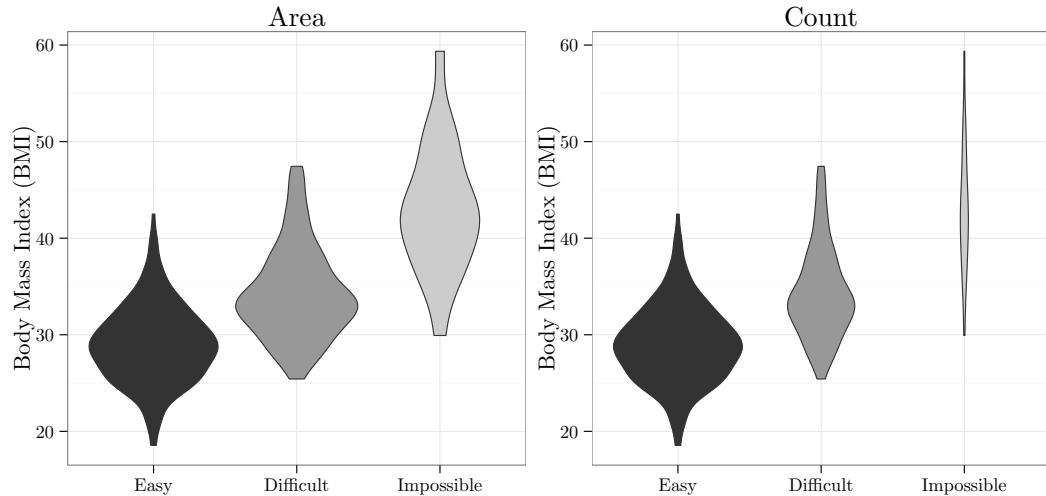


FIGURE 2.55: The left plot shows side-by-side violin plots of body mass index for patients according to the physicians' assessment of ease of palpating their spine using the default `scale = "area"` argument. The right plot shows side-by-side violin plots of body mass index for patients according to the physicians' assessment of ease of palpating their spine using the argument `scale = "count"` to create plots scaled proportionally to their number of observations.

R Code 2.68 is used to create Figure 2.56 on the facing page, which shows the relationship between boxplots and count violin plots. The left plot in Figure 2.56 on the next page superimposes the count violin plots with boxplots. The right plot of Figure 2.56 on the facing page adds an additional layer of jittered observations to each count violin plot so the reader can see the relationship between the actual observations and the scaling of the count violin plots.

R Code 2.68

```
> previous_theme <- theme_set(theme_bw())    # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = ease, y = BMI))  # Empty plot
> p1 <- p + geom_violin(scale = "count") +
+     geom_boxplot(aes(fill = ease), width = 0.25, outlier.size = 1.25) +
+     scale_fill_grey() +
+     guides(fill = FALSE) +
+     labs(x="", y = "Body Mass Index (BMI)", title = "Count")
> p1    # Left violin plots/boxplots
> p2 <- p1 + geom_jitter(aes(color = ease), size = 1.25) +
```

```
+     scale_color_grey(start = 0.8, end = 0.2) +
+     guides(color = FALSE)
> p2 # Right violin plots/boxplots
> theme_set(previous_theme)           # Restore original theme
```

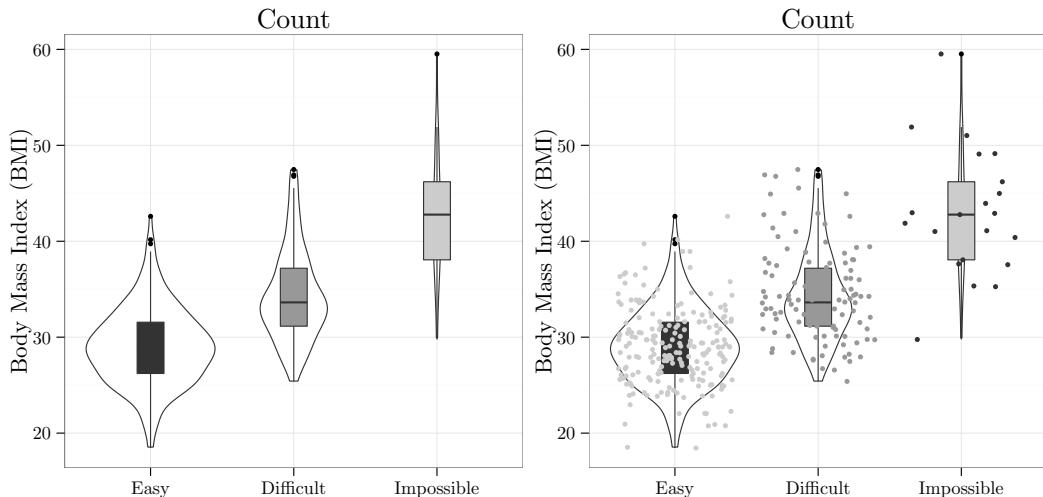


FIGURE 2.56: The left plot shows count violin plots superimposed with boxplots. The right plot adds jittered observations to the left plot.

Strip charts using the base R function `stripchart()`, often referred to as dot plots, were discussed in Section 2.4.2 on page 105. To create a dotplot with `ggplot2`, one uses the geom `geom_dotplot()`. A dotplot similar to Figure 2.7 on page 107, which shows the number of home runs Babe Ruth hit while playing for three different teams, is created with `ggplot2` using R Code 2.69 and is shown in the left plot of Figure 2.57 on the following page. Although the dotplot shows the distribution for the number of home runs Babe Ruth hit while playing for three different baseball teams, it does not make use of the `year` variable. The right plot of Figure 2.57 on the next page shows a scatterplot of home runs hit versus year faceted on `team`. The right plot of Figure 2.57 on the following page shows how Babe Ruth started out hitting very few home runs for the Bos-A, but in 1918 started to hit more home runs per season and was traded to the NY-A where he spent most of his career hitting between 20 and 60 home runs per season. Babe Ruth's home run production started a steady decline in 1930; and in 1934, Babe was traded to the Bos-N for the 1935 season, which was his last.

R Code 2.69

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> p <- ggplot(data = BABERUTH, aes(x = hr, fill = team))
> p1 <- p + geom_dotplot() +
+   facet_grid(team ~ .) +
+   scale_fill_grey()
> p1                                         # left dotplots
> p <- ggplot(data = BABERUTH, aes(x = year, y = hr, color = team))
```

```
> p2 <- p + geom_point(size = 2.5) +
+   scale_color_grey() +
+   facet_grid(team ~ .)
> p2
# right scatterplots
> theme_set(previous_theme)
# restore original theme
```

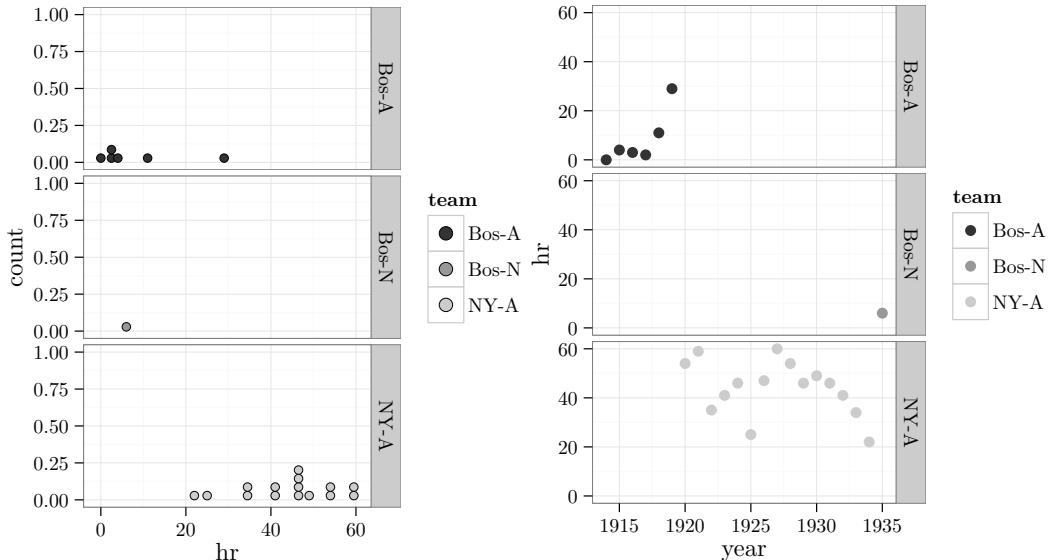


FIGURE 2.57: The left plot shows dotplots of the number of home runs Babe Ruth hit while playing for three different teams. The right plot shows a scatterplot of the number of home runs Babe Ruth hit versus the year for three different teams.

2.9.5.3 Adding a Smoothed Line

Different techniques for fitting lines to bivariate data with base R functions were discussed in Section 2.7.6 on page 147 and in Section 2.9.4 on page 167 for lattice graphics. Different smoothing methods such as `lm` for linear smoothing, `glm` for generalized linear smoothing, and `loess` for local smoothing can be passed to `stat_smooth()`, which will display the result with or without confidence bands on an appropriate scatterplot. For data sets with $n < 1000$, the default smoothing method is `loess`. The `loess()` function fits a polynomial surface determined by the predictors in a `formula`. For more information, see the details section of the `loess` help file. The default arguments for `stat_smooth()` are:

```
function (mapping = NULL, data = NULL, geom = "smooth",
         position = "identity", method = "auto", formula = y ~ x,
         se = TRUE, n = 80, fullrange = FALSE, level = 0.95,
         na.rm = FALSE, ...)
```

The argument `fullrange = FALSE` means that the smoother does not extend outside the range of the data. If the user wants the smoother to span the full range of the plot, use `fullrange = TRUE`. Confidence bands around the smoothed line are the default (`se =`

TRUE) and have a confidence level controlled by `level=`, set to 0.95 by default. For further explanations of the arguments, see the help file for `stat_smooth`.

Example 2.45 Create a scatterplot of body mass index values versus height (cm) using the information in `EPIDURALF`. Map the levels of `ease` to both `color` and `shape`. Add smoothed lines to the scatterplot based on the levels of `ease`. Use both `loess` and `lm` smoothing with 95% confidence intervals.

Solution: R Code 2.70 is used to create Figure 2.58. The variable `BMI` is created since it does not exist in `EPIDURALF`. The default arguments for `stat_smooth` with `method = "loess"` adds loess lines with 95% confidence intervals to the left scatterplot of Figure 2.58. Specifying `method = "lm"` adds ordinary least squares lines with 95% confidence intervals to the right scatterplot of Figure 2.58.

R Code 2.70

```
> previous_theme <- theme_set(theme_bw())      # set black-and-white theme
> EPIDURALF$BMI <- EPIDURALF$kg/(EPIDURALF$cm/100)^2    # create BMI
> p <- ggplot(data = EPIDURALF, aes(x = cm, y = BMI, color = ease,
+   shape = ease)) +
+   labs(x = "Height (cm)", y = "Body Mass Index")
> p1 <- p + geom_point() +
+   stat_smooth(method = "loess") +
+   scale_color_grey()
> p1           # left scatterplot with loess lines and confidence intervals
> p2 <- p + geom_point() +
+   stat_smooth(method = "lm") +
+   scale_color_grey()
> p2           # right scatterplot with ols lines and confidence intervals
> theme_set(previous_theme)                      # restore original theme
```

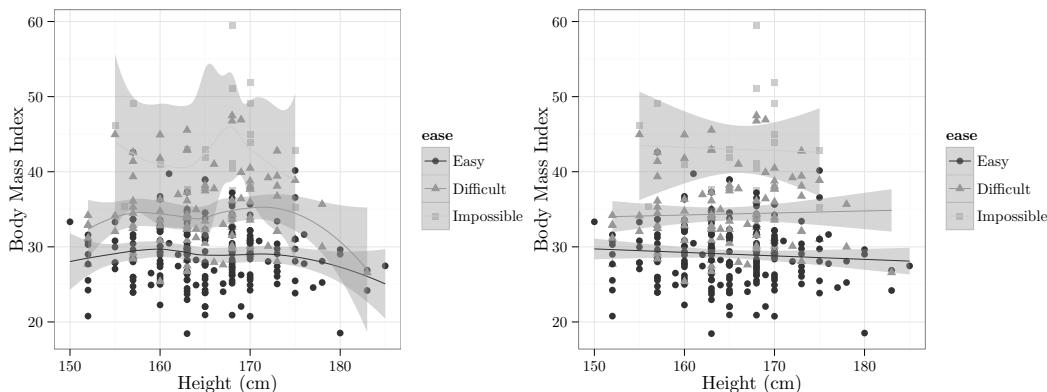


FIGURE 2.58: The left scatterplot shows body mass index versus height with loess lines added based on the physicians' ease of palpating the patient with 95% confidence bands. The right scatterplot shows body mass index versus height with ordinary least squares lines added based on the physicians' ease of palpating the patient with 95% confidence bands.

Example 2.46 Create a scatterplot of body mass index values versus height (cm) using the information in `EPIDURALF`. Map the levels of `ease` to both `color` and `shape`. Add smoothed lines to the scatterplot based on the levels of `ease`. Use both `loess` and `lm` smoothing without showing confidence intervals. Extend the lines for `lm` smoothing to cover the entire range of the plot.

Solution: R Code 2.71 is used to create Figure 2.59. The default arguments for `stat_smooth` with `method = "loess"`, `se = FALSE` adds loess lines without showing confidence intervals to the left scatterplot of Figure 2.59. Specifying `method = "lm"`, `se = FALSE`, `fullrange = TRUE` adds ordinary least squares lines that extend over the entire plot range to the right scatterplot of Figure 2.58 on the preceding page.

R Code 2.71

```
> previous_theme <- theme_set(theme_bw()) # set black-and-white theme
> p <- ggplot(data = EPIDURALF, aes(x = cm, y = BMI, color = ease,
+     shape = ease)) +
+     labs(x = "Height (cm)", y = "Body Mass Index")
> p1 <- p + geom_point() +
+     stat_smooth(method = "loess", se = FALSE) +
+     scale_color_grey()
> p1                                         # Left scatterplot with loess lines
> p2 <- p + geom_point() + stat_smooth(method = "lm", se = FALSE,
+     fullrange = TRUE) + scale_color_grey()
> p2                                         # Right scatterplot with ols lines
> theme_set(previous_theme)                  # Restore original theme
```

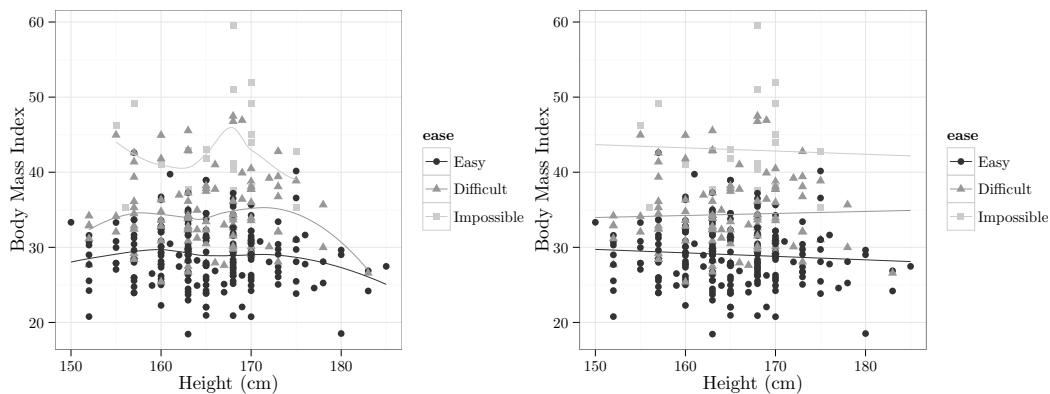


FIGURE 2.59: The left scatterplot shows body mass index versus height with loess lines added based on the physicians' ease of palpating the patient. The right scatterplot shows body mass index versus height with ordinary least squares lines that extend over the full range of the plot added based on the physicians' ease of palpating the patient.

If one has the equation of a line, the line may be added to a `ggplot` object with `geom_abline()`. The function `geom_abline()` adds a line with the arguments `intercept=` and `slope=`. For further details, see the online help. To label graphs with text, the

`geom geom_text()` has three required arguments, `x=`, `y=`, which specify the x and y coordinates, and `label=`, which accepts text strings (specified in quotes). For example, `geom_text(x = 2, y = 3, label = "foo")` adds the text string `foo` to a `ggplot` object at the location (2,3). It is also possible to have the text appear at an angle by providing a value to the `angle=` argument. R Code 2.72 revisits the data frame `Animals` from the MASS package explored previously in Examples 2.30 on page 143 and 2.33 on page 147. Six plots are shown in Figure 2.60 on the following page to illustrate the process of adding layers to a plot. The top left plot of Figure 2.60 on the next page shows a scatterplot of `brain` weight versus `body` weight. Because of the large range in body and brain weights, (0.023 kg to 87,000 kg) and (0.4 g to 5,712 g), respectively, the scatterplot of `brain` weight versus `body` weight is too distorted to reveal any clear pattern. Consequently, the data scale is changed to a log base 10 scale for both axes and shown in the top center plot. The third layer adds customized labels to the x and y axes and is shown in the top left plot of Figure 2.60 on the following page. The fourth layer adds a least squares line while the fifth layer adds a least squares line where dinosaurs have been removed from the computation of the line. Finally, the sixth layer adds angled text strings.

R Code 2.72

```
> previous_theme <- theme_set(theme_bw()) # set black-and-white theme
> library(MASS)
> p <- ggplot(data = Animals, aes(x = body, y = brain)) # Empty plot
> p1 <- p + geom_point()
> p1 # Top left plot
> p2 <- p1 + scale_x_log10() + scale_y_log10()
> p2 # Top center plot log10 axes
> p3 <- p2 + labs(x = "Body weight (kg)", y = "Brain weight (g)")
> p3 # Top right plot with labeled axes
> mod1 <- lm(log10(brain) ~ log10(body), data = Animals)
> mod2 <- lm(log10(brain) ~ log10(body), data = subset(Animals,
+   subset = body < 9400))
> p4 <- p3 + geom_abline(intercept = coef(mod1)[1], slope = coef(mod1)[2],
+   linetype = "dashed")
> p4 # Bottom left plot with ols line
> p5 <- p4 + geom_abline(intercept = coef(mod2)[1], slope = coef(mod2)[2])
> p5 # Bottom center plot with ols line (no dinosaurs)
> p6 <- p5 + geom_text(data = NULL, x = 0.8, y = 2.6, angle = 54,
+   size = 4, label = "Solid line omits dinosaurs", ) +
+   geom_text(data = NULL, x = 2.6, y = 1.9, angle = 41,
+   size = 4, label = "Dashed line includes dinosaurs")
> p6 # Bottom right plot with ols lines labeled with text
> theme_set(previous_theme) # Restore original theme
```

The fifth and sixth plots of Figure 2.60 on the next page are created in R Code 2.73 using `geom_smooth()` in place of `geom_abline()` and are shown in Figure 2.61 on page 189. The second line of code creates a categorical variable `DINO` that has text strings to indicate whether an animal is a dinosaur or not a dinosaur based on its body weight.

R Code 2.73

```
> previous_theme <- theme_set(theme_bw()) # set black-and-white theme
> Animals$DINO <- ifelse(Animals$body < 9400, "Not Dinosaur", "Dinosaur")
```

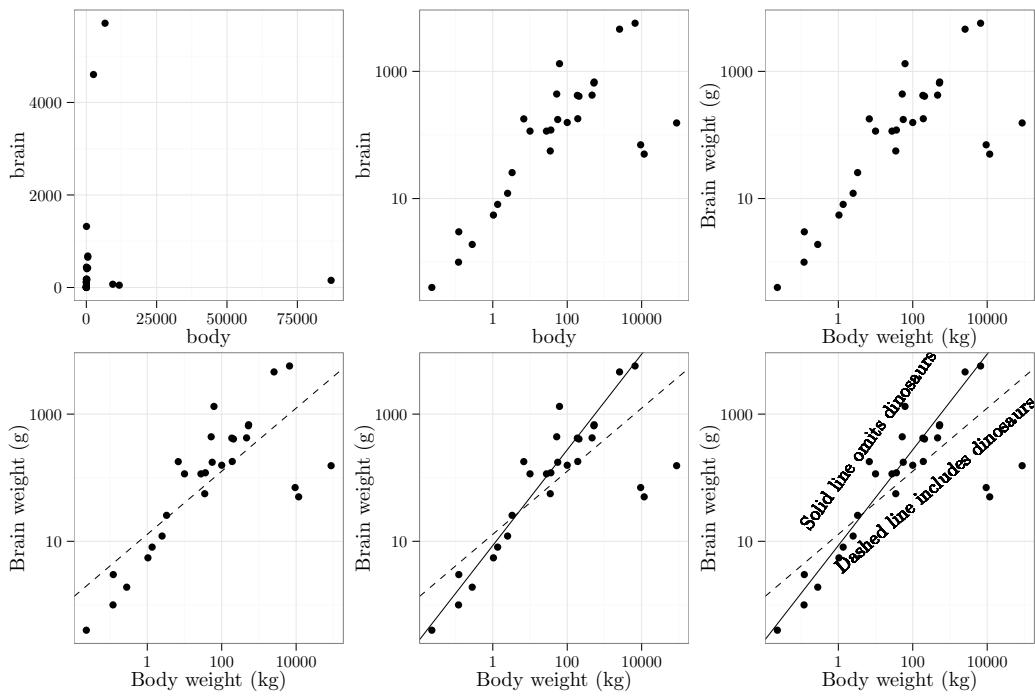


FIGURE 2.60: Six plots illustrating various layers used in the creation of the bottom right scatterplot of brain weight versus body weight on a log base 10 scale with superimposed and labeled least squares regression lines

```
> p1 <- ggplot(data = Animals, aes(x = body, y = brain)) +
+   geom_point(size = 3) +
+   scale_x_log10() +
+   scale_y_log10() +
+   labs(x = "Body weight (kg)", y = "Brain weight (g)") +
+   geom_smooth(data = subset(Animals, subset = DINO == "Not Dinosaur"),
+               method = "lm", se = FALSE, color = "black") +
+   geom_smooth(data = Animals, method = "lm", se = FALSE,
+               linetype = "dashed", color = "black")
> p1 # Left plot with ols lines
> p2 <- p1 + geom_text(data = NULL, x = 1.2, y = 3, angle = 52, size = 4,
+                       label = "Solid line omits dinosaurs") +
+   geom_text(data = NULL, x = 3.1, y = 2.3, angle = 39, size = 4,
+             label = "Dashed line includes dinosaurs")
> p2 # Right plot with ols lines labeled
> theme_set(previous_theme) # Restore original theme
```

2.9.5.4 Choropleth Maps

Choropleth maps are maps shaded in proportion to some statistic being displayed on the map. Since the shaded shapes of a map can be defined by a polygon, the function `geom_polygon()` will be used to create choropleth maps. While there are many ways to create choropleth maps with R, this section will focus exclusively on creating choropleth

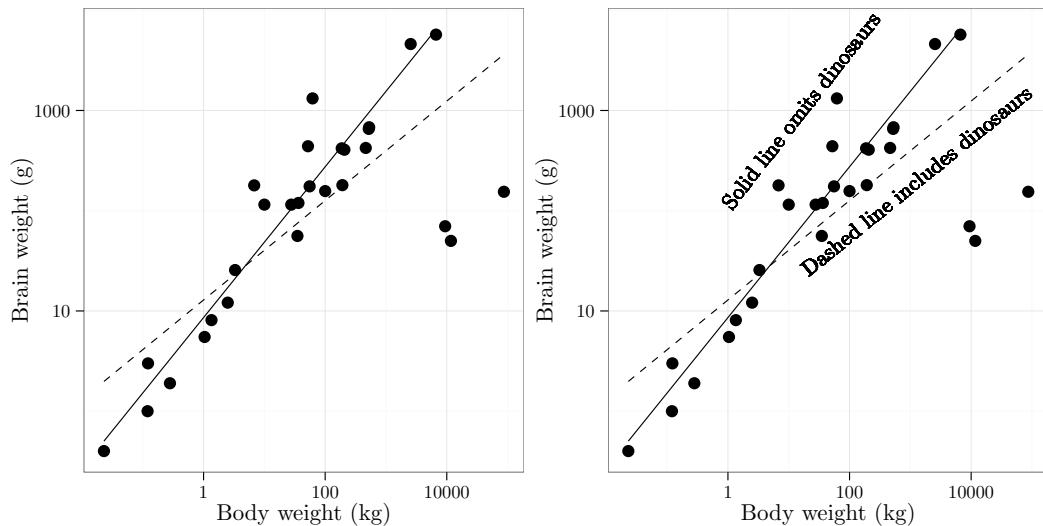


FIGURE 2.61: Graphs are scatterplots of brain weight versus body weight on a log base 10 scale with superimposed least squares regression lines. Right plot has angled text strings indicating which line includes dinosaurs and which line omits dinosaurs.

maps with `ggplot2`. The `maps` package has three USA databases (`usa`, `state`, and `county`) as well a low resolution map of the world. To convert the data in the `maps` package to a data frame, one may use the function `map_data()` from the `ggplot2` package. The `map_data()` function returns a data frame with the following six columns when applied to `map = "county"`: `long`, the longitude; `lat`, the latitude; `group`, a grouping variable for each polygon; `order`, the order to connect points within a group; `region`, generally the state name; `subregion`, generally the county name. There are numerous ways to apply a map projection, and the `mapproj` package provides thirty-three different projections. Further details for projections can be found on the `mapproject()` help page after loading and possibly installing the `mapproj` package if it is not already installed. The different projections from `mapproject()` can be used with `ggplot2` graphics via the `coord_map()` function. A map of the United States of America is created with R Code 2.74 and displayed in Figure 2.62 on the next page. States are shaded based on their alphabetical order.

R Code 2.74

```
> library(maps)      # package has maps
> library(mapproj)   # used for different projections
> STATESmap <- map_data(map = "state")
> p <- ggplot(data = STATESmap, aes(x = long, y = lat,
+                                     group = group, fill = region))
> p + geom_polygon(color = "black", alpha = 0.5) +
+     theme_bw() +
+     coord_map("polyconic") +
+     theme(legend.position = "none") +
+     scale_fill_grey()
```

Example 2.47 \triangleright **North Carolina Choropleth** \lhd Use the data base in the package `maps` to create a data frame named `NCmap` of the counties in North Carolina. Use the

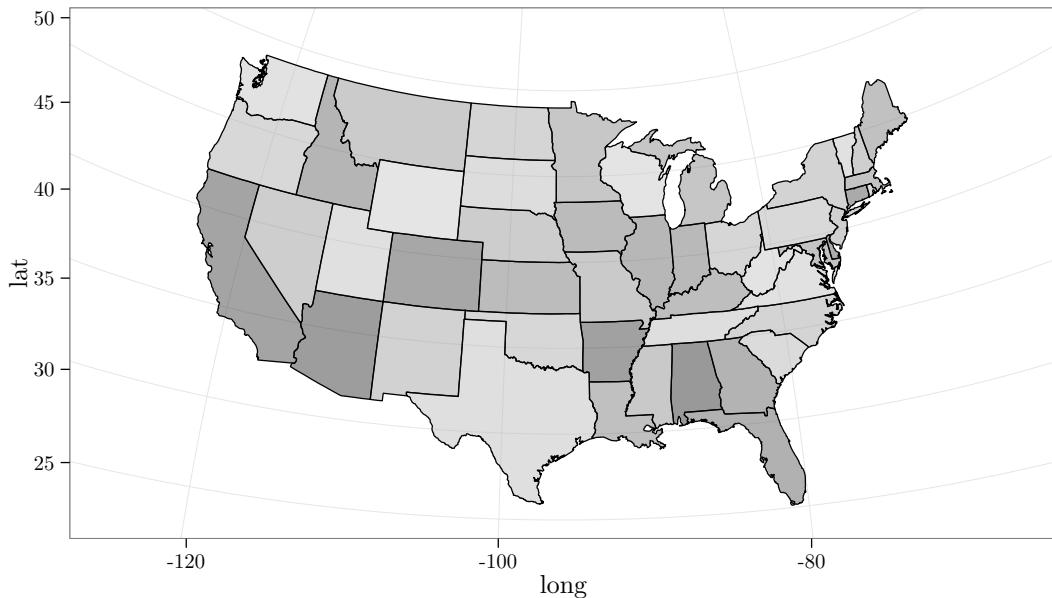


FIGURE 2.62: Map of the United States of America

function `merge()` to create a single data frame named `MERGED` by uniting the data frames `NCmap` and `NC2010DMG` based on county names. Create a choropleth map of the counties of North Carolina shaded according to the evangelical adherence rate (`evanrate`) of each county.

Solution: The first two lines of R Code 2.75 load the packages `maps` and `mapproj`, which are needed for the North Carolina data base and the polyconic projection function, respectively. Note that the names of the counties are stored in the variables `subregion` for the data frame `NCmap` and `countyName` for the data frame `NC2010DMG`, respectively. The remainder of R Code 2.75 is used to create a polyconic projection of North Carolina counties shaded according to evangelical adherence rates as shown in Figure 2.63 on the facing page.

R Code 2.75

```
> library(maps)           # package has maps
> library(mapproj)        # used for different projections
> NCmap <- map_data(map = "county", region = "north carolina")
> MERGED <- merge(x = NCmap, y = NC2010DMG, by.x = "subregion",
+                  by.y = "countyName")
> p <- ggplot(data = MERGED, aes(x = long, y = lat, group = group,
+                                    fill = evanrate)) +
+      labs(fill = "Evangelical\\nAdherence\\nRate")
> p + geom_polygon(color = "black") +
+      theme_bw() +
+      coord_map("polyconic") +
+      scale_fill_gradient2()
```



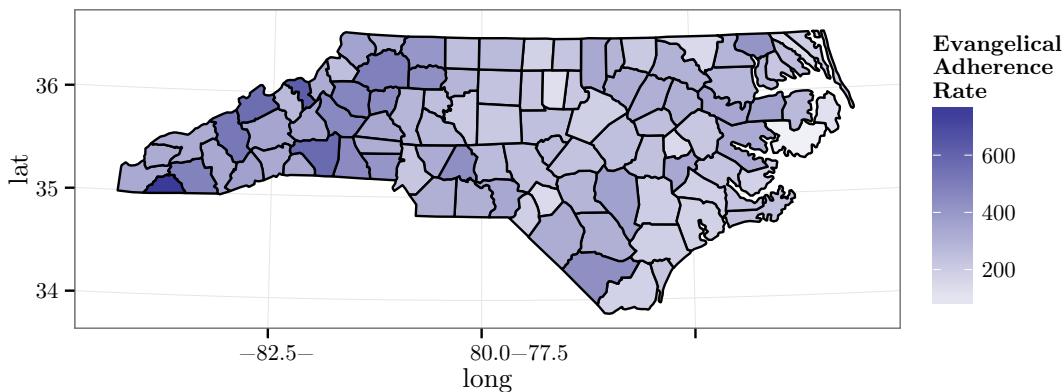


FIGURE 2.63: Choropleth map of North Carolina, where counties are shaded according to the evangelical adherence rate per 1,000 inhabitants

2.9.6 Arranging Several ggplot Graphs on a Single Page

The arrangement of `ggplot` graphs on a single page is different from the arrangement of traditional graphs on a single page. Just as with lattice graphs, approaches involving `par(mfrow=c())`, `par(mfcoll=c())`, `split.screen()`, or `layout()` will not work with `ggplot` graphs since `ggplot` graphs are based on `grid` graphics. Although complex arrangements of `ggplot` graphs can be accomplished using the `grid` graphics system, for pedagogical reasons, the function `multiplot()` is introduced, which has an argument `layout`, a matrix specifying the division of the graphics device. The function `multiplot()`, written by Winston Chang, is part of the `PASWR2` package.

Example 2.48 Use the function `multiplot()` to split the graphics device into four plotting regions as was done in Example 2.36 on page 153. Use the data frame `HSWRESTLER`, and place a scatterplot of `hwfat` versus `tanfat` in the top left of the graphics device using a `ggplot` graph. Directly below the x -axis of the scatterplot, place a horizontal boxplot of the variable `tanfat` using a `ggplot` graph. To the right of the scatterplot, place a vertical boxplot of the variable `hwfat` using a `ggplot` graph. Leave the bottom right plotting region empty.

Solution: R Code 2.76 splits the graphics device according to the values in `mat44`. The requested graphs are shown in Figure 2.64 on the next page. The `ggplot` objects `p1`, `p3`, and `p2` are passed to the function `multiplot()` and arranged on the graphics device according to the values in `mat44`.

R Code 2.76

```
> mat44 <- matrix(c(1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1, 2, 3, 3, 3, 0),
+                   byrow = TRUE, nrow = 4)
> mat44

 [,1] [,2] [,3] [,4]
[1,]    1    1    1    2
[2,]    1    1    1    2
[3,]    1    1    1    2
[4,]    3    3    3    0
```

```

> p1 <- ggplot(data = HSWRESTLER, aes(x = tanfat, y = hwfat)) +
+   geom_point(color = "red") + theme_bw()
> p2 <- ggplot(data = HSWRESTLER, aes(x = 1, y = tanfat)) +
+   geom_boxplot(fill = "purple") + coord_flip() + labs(x = "") +
+   theme_bw()
> p3 <- ggplot(data = HSWRESTLER, aes(x = 1, y = hwfat)) +
+   geom_boxplot(fill = "blue") +
+   labs(x = "") + theme_bw()
> multiplot(p1, p3, p2, layout = mat44)

```

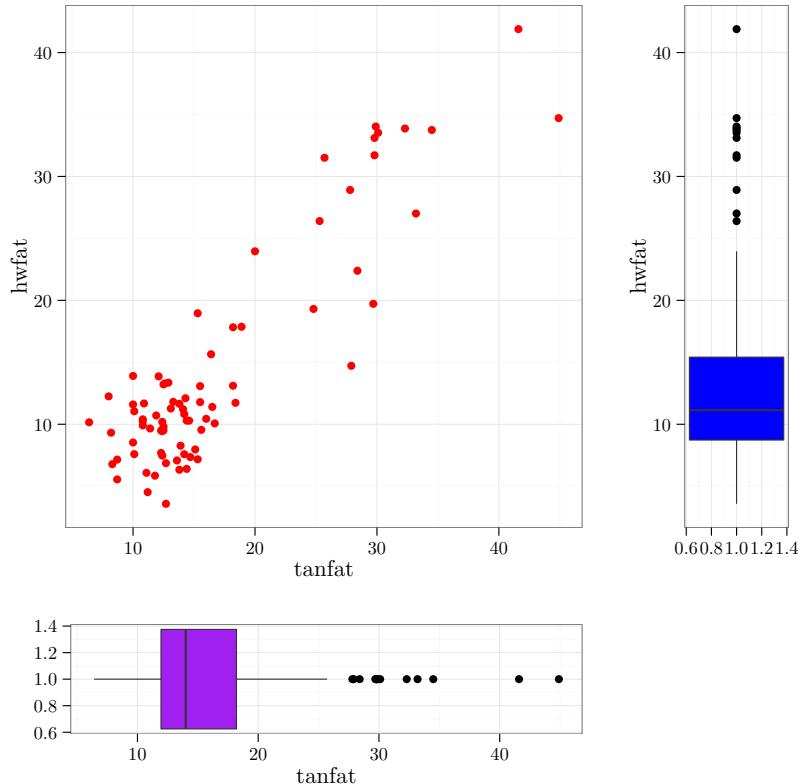


FIGURE 2.64: Scatterplot and boxplots for Example 2.48

2.10 Problems

1. Load the MASS package.
 - (a) Enter the command `help(package="MASS")` and read about the functions and data contained in this package.
 - (b) What does the description in the help file say about the function `lqs()`? Enter `help(lqs, package="MASS")` to obtain information about the command `lqs`.
 - (c) What command shows the loaded packages?
2. Load `Cars93` from the MASS package.
 - (a) Create density histograms for the variables `Min.Price`, `Max.Price`, `Weight`, and `Length` variables using a different color for each histogram.
 - (b) Superimpose estimated density curves over the histograms.
 - (c) Use the `bwplot()` function from `lattice` to create a box and whiskers plot of `Price` for every type of vehicle according to the drive train. Do you observe any differences between prices?
 - (d) Create a graph similar to the one created in (c) using functions from `ggplot2`.
3. Load the data frame `WHEATSPAIN` from the PASWR2 package.
 - (a) Find the quantiles, deciles, mean, maximum, minimum, interquartile range, variance, and standard deviation of the variable `hectares`. Comment on the results. What was Spain's 2004 total harvested wheat area in hectares?
 - (b) Create a function that calculates the quantiles, the mean, the variance, the standard deviation, the total, and the range of any variable.
 - (c) Which communities are below the 10th percentile in `hectares`? Which communities are above the 90th percentile? In which percentile is Navarra?
 - (d) Create and display in the same graphics device a frequency histogram of the variable `acres` and a density histogram of the variable `acres`. Superimpose a density curve over the second histogram.
 - (e) Explain why using breaks of 0; 100,000; 250,000; 360,000; and 1,550,000 automatically results in a density histogram when using `hist()` from base graphics.
 - (f) Create and display in the same graphics device a barplot of `acres` and a density histogram of `acres` using break points of 0; 100,000; 250,000; 360,000; and 1,550,000.
 - (g) Add vertical lines to the density histogram of `acres` to indicate the locations of the mean and the median, respectively.
 - (h) Create a boxplot of `hectares` and label the communities that appear as outliers in the boxplot. (Hint: Use `identify()`.)

- (i) Determine the community with the largest harvested wheat surface area using either acres or hectares. Remove this community from the data frame and compute the mean, median, and standard deviation of `hectares`. How do these values compare to the values for these statistics computed in (a)?
4. Load the **WHEATUSA2004** data frame from the **PASWR2** package.
- Find the quantiles, deciles, mean, maximum, minimum, interquartile range, variance, and standard deviation for the variable `acres`. Comment on what the most appropriate measures of center and spread would be for this variable. What is the USA's 2004 total harvested wheat surface area?
 - Which states are below the 20th percentile? Which states are above the 80th percentile? In which quantile is WI (Wisconsin)?
 - Create a frequency and a density histogram in the same graphics device using square plotting regions of the values in `ACRES`.
 - Add vertical lines to the density histogram from (c) to indicate the location of the mean and the median.
 - Create a boxplot of the `acres` and locate the outliers' communities and their values.
 - Determine the state with the largest harvested wheat surface in acres. Remove this state from the data frame and compute the mean, median, and standard deviation of `acres`. How do these values compare to the values for these statistics computed in (a)?
5. The data frame **VIT2005** in the **PASWR2** package contains descriptive information and the appraised total price (in euros) for apartments in Vitoria, Spain.
- Create a frequency table, a piechart, and a barplot showing the number of apartments grouped by the variable `out`. For you, which method conveys the information best?
 - Characterize the distribution of the variable `totalprice`.
 - Characterize the relationship between `totalprice` and `area`.
 - Create a Trellis plot of `totalprice` versus `area` conditioning on `toilets`. Create the same graph with `ggplot2` graphics. Are there any outliers? Ignoring any outliers, between what two values of `area` do apartments have both one and two bathrooms?
 - Use the `area` values reported in (d) to create a subset of apartments that have both one and two bathrooms. By how much does an additional bathroom increase the appraised value of an apartment? Would you be willing to pay for an additional bathroom if you lived in Vitoria, Spain?
6. Consider the data frame **PAMTEMP** from the **PASWR2** package, which contains temperature and precipitation for Pamplona, Spain, from January 1, 1990, to December 31, 2010.
- Create side-by-side violin plots of the variable `tmean` for each `month`. Make sure the level of `month` is correct. Hint: Look at the examples for `PAMTEMP`. Characterize the pattern of side-by-side violin plots.

- (b) Create side-by-side plots of the variable `tmean` for each `year`. Characterize the pattern of side-by-side violin plots.
- (c) Find the date for the minimum value of `tmean`.
- (d) Find the date for the maximum value of `tmean`.
- (e) Find the date for the maximum value of `precip`.
- (f) How many days have reported a `tmax` value greater than 38 °C?
- (g) Create a barplot showing the total precipitation by `month` for the period January 1, 1990, to December 31, 2010. Based on your barplot, which month had the least amount of precipitation? Which month had the greatest amount of precipitation? Hint: Use the `plyr` package to create an appropriate data frame.
- (h) Create a barplot showing the total precipitation by `year` for the period January 1, 1990, to December 31, 2010. Based on your barplot, which `year` had the least amount of precipitation? Which `year` had the greatest amount of precipitation? Hint: Use the `plyr` package to create an appropriate data frame.
- (i) Create a graph showing the maximum temperature versus `year` and the minimum temperature versus `year`. Does the graph suggest temperatures are becoming more extreme over time?

7. Access the data from url

<http://www.stat.berkeley.edu/users/statlabs/data/babies.data>

and store the information in an object named `BABIES` using the function `read.table()`. A description of the variables can be found at

<http://www.stat.berkeley.edu/users/statlabs/labs.html>.

These data are a subset from a much larger study dealing with child health and development.

- (a) Create a “clean” data set that removes subjects if any observations on the subject are “unknown.” Note that `bwt`, `gestation`, `parity`, `age`, `height`, `weight`, and `smoke` use values of 999, 999, 9, 99, 99, 999, and 9, respectively, to denote “unknown.” Store the modified data set in an object named `CLEAN`.
- (b) Use the information in `CLEAN` to create a density histogram of the birth weights of babies whose mothers have never smoked (`smoke=0`) and another histogram placed directly below the first in the same graphics device for the birth weights of babies whose mothers currently smoke (`smoke=1`). Make the range of the *x*-axis 30 to 180 (ounces) for both histograms. Superimpose a density curve over each histogram.
- (c) Based on the histograms in (b), characterize the distribution of baby birth weight for both non-smoking and smoking mothers.
- (d) What is the mean weight difference between babies of smokers and non-smokers? Can you think of any reasons not to use the mean as a measure of center to compare birth weights in this problem?
- (e) Create side-by-side boxplots to compare the birth weights of babies whose mothers never smoked and those who currently smoke. Use traditional graphics (`boxplot()`), lattice graphics (`bwplot()`), and `ggplot` graphics to create the boxplots.

- (f) What is the median weight difference between babies who are firstborn and those who are not?
- (g) Create a single graph of the densities for pre-pregnancy `weight` for mothers who have never smoked and for mothers who currently smoke. Make sure both densities appear on the same graphics device and use an appropriate legend.
- (h) Characterize the pre-pregnancy distribution of `weight` for mothers who have never smoked and for mothers who currently smoke.
- (i) What is the mean pre-pregnancy weight difference between mothers who do not smoke and those who do? Can you think of any reasons not to use the mean as a measure of center to compare pre-pregnancy weights in this problem?
- (j) Compute the body mass index (BMI) for each mother in `CLEAN`. Recall that BMI is defined as kg/m^2 ($0.0254 \text{ m} = 1 \text{ in.}$, and $0.45359 \text{ kg} = 1 \text{ lb.}$). Add the variables `weight` in kg, height in m, and `BMI` to `CLEAN` and store the result in `CLEANP`.
- (k) Characterize the distribution of `BMI`.
- (l) Group pregnant mothers according to their `BMI` quartile. Find the mean and standard deviation for baby birth weights in each quartile for mothers who have never smoked and those who currently smoke. Find the median and IQR for baby birth weights in each quartile for mothers who have never smoked and those who currently smoke. Based on your answers, would you characterize birth weight in each group as relatively symmetric or skewed? Create histograms and densities of `bwt` conditioned on `BMI` quartiles and whether the mother smokes to verify your previous assertions about the shape.
- (m) Create side-by-side boxplots of `bwt` based on whether the mother smokes conditioned on `BMI` quartiles. Does this graph verify your findings in (l)?
- (n) Does it appear that `BMI` is related to the birth weight of a baby? Create a scatterplot of birth weight (`bwt`) versus `BMI` while conditioning on `BMI` quartiles and whether the mother smokes to help answer the question.
- (o) Replace baby birth weight (`bwt`) with gestation length (`gestation`) and answer questions (l), (m), and (n).
- (p) Create a scatterplot of `bwt` versus `gestation` conditioned on `BMI` quartiles and whether the mother smokes. Fit straight lines to the data using `lm()`, `lqs()`, and `r1m()`; and display the lines in the scatterplots. What do you find interesting about the resulting graphs?
- (q) Create a table of `smoke` by `parity`. Display the numerical results in a graph. What percent of mothers did not smoke during the pregnancy of their first child?

8. Some claim the final hours aboard the RMS Titanic were marked by class warfare; others claim it was characterized by male chivalry. The data frame `TITANIC3` from the `PASWR2` package contains information pertaining to class status (`pclass`), survival of passengers (`survived`), and gender (`sex`), to name but a few. Based on the information in `TITANIC3`:

- (a) Determine the fraction of survivors (`survived`) according to class (`pclass`).
- (b) Compute the fraction of survivors according to class and gender. Did men in first class or women in third class have a higher survival rate?

- (c) How would you characterize the distribution of `age`?
 - (d) Were the median and mean ages for females who survived higher or lower than for females who did not survive? Report the median and mean ages as well as an appropriate measure of spread for each statistic.
 - (e) Were the median and mean ages for males who survived higher or lower than for males who did not survive? Report the median and mean ages as well as an appropriate measure of spread for each statistic.
 - (f) What was the age of the youngest female in first class who survived?
 - (g) Do the data suggest class warfare, male chivalry, or some combination of both characterized the final hours aboard the Titanic? Feel free to explore other relationships based on the numbers in `TITANIC3` in answering this question.
9. Use the `CARS2004` data frame from the `PASWR2` package, which contains the numbers of cars per 1000 inhabitants (`cars`), the total number of known mortal accidents (`deaths`), and the country population/1000 (`population`) for the 25 member countries of the European Union for the year 2004.
- (a) Compute the total number of cars per 1000 inhabitants in each country, and store the result in an object named `total.cars`. Determine the total number of known automobile fatalities in 2004 divided by the total number of cars for each country and store the result in an object named `death.rate`.
 - (b) Create a barplot showing the automobile death rate for each of the European Union member countries. Make the bars increase in magnitude so that the countries with the smallest automobile death rates appear first.
 - (c) Which country has the lowest automobile death rate? Which country has the highest automobile death rate?
 - (d) Create a scatterplot of `population` versus `total.cars`. How would you characterize the relationship?
 - (e) Find the least squares estimates for regressing `population` on `total.cars`. Superimpose the least squares line on the scatterplot from (d). What population does the least squares model predict for a country with a `total.cars` value of 19224.630? Find the difference between the population predicted from the least squares model and the actual population for the country with a `total.cars` value of 19224.630.
 - (f) Create a scatterplot of `total.cars` versus `death.rate`. How would you characterize the relationship between the two variables?
 - (g) Compute Spearman's rank correlation coefficient of `total.cars` and `death.rate`. (Hint: Use `cor(x, y, method="spearman")`.) What is this coefficient measuring?
 - (h) Plot the logarithm of `total.cars` versus the logarithm of `death.rate`. How would you characterize the relationship?
 - (i) What are the least squares estimates for the regression of `log(total.cars)` on `log(death.rate)`. Superimpose the least squares line on the scatterplot from (h). What total number of cars does the least squares model predict for a country with a `log(death.rate)` value of -3.769252? Make sure you express your answer in the same units as those used for `total.cars`.

10. The data frame **SURFACESPAIN** in the **PASWR2** package contains the surface area (km^2) for seventeen autonomous Spanish communities.
- (a) Use the function `merge()` to combine the data frames **WHEATSPAIN** (from Problem 2.10) and **SURFACESPAIN** into a new data frame named **DataSpain**.
 - (b) Create a variable named **surface.h** containing the surface area of each autonomous community in hectares. (Note: 100 hectares = 1 km^2 .) Create a variable named **wheat.p** containing the percent surface area in each autonomous community dedicated to growing wheat. Add the newly created variables to the data frame **DataSpain** and store the result as a data frame with the name **DataSpain.m**.
 - (c) Assign the names of the autonomous communities as row names for **DataSpain.m** and remove the variable **community** from the data frame.
 - (d) Create a barplot showing the percent surface area dedicated to growing wheat for each of the seventeen Spanish autonomous communities. Arrange the communities by decreasing percentages.
 - (e) Display the percent surface area dedicated to growing wheat for each of the seventeen Spanish autonomous communities using the function `dotchart()`. To read about `dotchart()`, type `?dotchart` at the command prompt. Do you prefer the barchart or the dotchart? Explain your answer.
 - (f) Describe the relationship between the surface area in an autonomous community dedicated to growing wheat (**hectares**) and the total surface area of the autonomous community (**surface.h**).
 - (g) Describe the relationship between the surface area in an autonomous community dedicated to growing wheat (**hectares**) and the percent of surface area dedicated to growing wheat out of the communities' total surface area (**wheat.p**).
 - (h) Develop a model to predict the surface area in an autonomous community dedicated to growing wheat (**hectares**) based on the total surface area of the autonomous community (**surface.h**).

Chapter 3

General Probability and Random Variables

3.1 Introduction

One of the main objectives of statistics is to help make “good” decisions under conditions of uncertainty. Probability is one way to quantify outcomes that cannot be predicted with certainty. For example, when throwing two dice, the outcome of the experiment cannot be known before the dice are thrown. Random variables, as well as counting techniques, will facilitate the analysis of problems such as the example of throwing two dice. This chapter provides a brief introduction to counting techniques, axiomatic probability, random variables, and moment generating functions.

3.2 Counting Techniques

One of the fundamental questions surrounding any experiment is how to know the number of possible outcomes of the experiment.

DEFINITION 3.1: Basic principle of counting — Suppose k experiments are to be performed such that the first can result in any one of n_1 outcomes; and if for each of these n_1 outcomes, there are n_2 possible outcomes of the second experiment; and if for each of the possible outcomes of the first two experiments, there are n_3 possible outcomes of the third experiment; and if . . . , then there are $n_1 \times n_2 \times \cdots \times n_k$ possible outcomes for the k experiments.

Example 3.1 A computer store sells three brands of laptops. Each laptop is sold with a carrying case and four different options for upgrading RAM. Suppose the store only carries two styles of carrying cases. How many different combinations of laptop, carrying case, and RAM are possible?

Solution: According to the basic principle of counting, there are $3 \cdot 2 \cdot 4 = 24$ different combinations of laptop, carrying case, and RAM. ■

3.2.1 Sampling with Replacement

When working with finite samples, it is critical to distinguish between **sampling with replacement** and **sampling without replacement**. Sampling with replacement occurs when an object is selected and subsequently replaced before the next object is selected. Consequently, when sampling with replacement, the number of possible ordered samples of size k taken from a set of n objects is n^k .

Example 3.2 How many different license plates can be made from four digits?

Solution: First, note that there is no restriction forbidding repeated digits. That is, 0001, 0002, 0003, ..., 9999 are all permissible. In essence, this translates to sampling with replacement. Since there are 10 choices for each of the four license plate digits, there are a total of $10 \times 10 \times 10 \times 10 = 10^4 = 10,000$ possible license plates. ■

3.2.2 Sampling without Replacement

Sampling without replacement occurs when an object is not replaced after it has been selected. When sampling without replacement, the number of possible ordered samples of size k taken from a set of n objects is

$$P_{k,n} = n(n-1)(n-2) \cdots (n-k+1) = \frac{n!}{(n-k)!}.$$

Any ordered sequence of k objects taken from n distinct objects is called a **permutation** and is denoted $P_{k,n}$.

Example 3.3 How many different ways can the first three places be decided in a race with four runners?

Solution: The number of ways the first three places can be decided using the basic principle of counting is by reasoning as follows:

Any one of the four runners might arrive in first place (four outcomes for the first experiment). After the first runner crosses the finish line, there are three possible choices for second place (three outcomes for the second experiment). Then, after second place is decided, there are only two runners left (two outcomes for the third experiment). Consequently, there are $4 \cdot 3 \cdot 2 = 24$ possible ways to award the first three places. The problem may also be solved by applying the permutation formula:

$$P_{3,4} = \frac{4!}{(4-3)!} = \frac{4!}{1!} = 4 \cdot 3 \cdot 2 = 24.$$

```
> factorial(4)/factorial(4 - 3)
[1] 24
```

■

Example 3.4 How many ways can seven students form a line?

Solution: First, note that once a student is selected for a place in line, the number of students for subsequent orderings is diminished by one. That is, this is a problem where sampling is done without replacement. A useful strategy for this type of problem is actually to think through assigning the students to positions before using a formula (permutation in this case). If seven slots are drawn, then the reasoning is as follows:

There are seven ways a student can be assigned to the first slot. Once the first slot has been assigned, there are six possible ways to pick a student for the next slot. Continue with this logic until all of the students have been assigned a slot. Appealing to the basic principle of counting, it is seen that there are $7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 7! = 5040$ possible ways to form a line with seven students. This is the same number calculated by considering a permutation of seven things taken seven at a time $P_{7,7} = \frac{7!}{(7-7)!} = \frac{7!}{0!} = 5040$. Note that $0! = 1$.

```
> factorial(7)/factorial(7 - 7)
[1] 5040
```



When a subset of the n objects is indistinguishable, clearly the number of permutations is diminished. More to the point, when n objects have n_1 that are the same, n_2 that are the same, and so on until there are n_k that are the same, there are a total of

$$\frac{n!}{n_1! \cdot n_2! \cdots n_k!}$$

permutations of the n objects.

Example 3.5 How many different letter arrangements can be formed using the letters *DATA*?

Solution: Note that there are $4!$ permutations of the letters $D_1A_1T_1A_2$ when the two A 's are distinguished from each other. However, since the A 's are indistinguishable, there are only $\frac{4!}{2! \cdot 1! \cdot 1!} = 12$ possible permutations of the letters *DATA*.

```
> factorial(4)/(factorial(2) * factorial(1) * factorial(1))
[1] 12
```



3.2.3 Combinations

In many problems, selecting k objects from n total objects without regard to order is the scenario of interest. For example, when selecting a committee, the order of the committee is rarely important. That is, a committee consisting of John, Mary, and Paul is considered the same committee if the members are listed as Mary, Paul, and John. An arrangement of k objects taken from n objects without regard to order is called a **combination**. The number of combinations of n distinct objects taken k at a time is denoted as $C_{k,n}$ or $\binom{n}{k}$ and is calculated as

$$C_{k,n} = \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

The R function to compute the number of combinations of n distinct objects taken k at a time is `choose(n, k)`.

Example 3.6 A committee of three people is to be formed from a group of eight people. How many different committees are possible?

Solution: There are $C_{3,8} = \binom{8}{3} = \frac{8!}{3!(8-3)!} = 56$ possible committees.

```
> choose(n = 8, k = 3)
[1] 56
```



Example 3.7 How many different three-letter sequences can be formed from the letters A, B, C, and D if

- (a) letter repetition is not permissible and order matters?
- (b) letter repetition is permissible and order matters?
- (c) letter repetition is not permissible and sequences containing the same letters are considered equal regardless of letter order?

Solution: The answers are as follows:

- (a) There are $P_{3,4} = 4 \cdot 3 \cdot 2 = 24$ possible sequences.
- (b) Since letters may be used more than once, there are $4^3 = 64$ possible sequences.
- (c) Since order does not matter, there are $C_{3,4} = \binom{4}{3} = 4$ possible sequences.



Example 3.8 If nine people are to be assigned into three committees of sizes two, three, and four, respectively, how many possible assignments exist?

Solution: There are $\binom{9}{2}$ ways to pick the first committee. Once that committee is selected, there are seven members left from which a committee of size three is selected. So, there are $\binom{7}{3}$ ways to pick the second committee. Using the same logic, there are finally four members left from which one committee of size four must be selected. There is only one way to select the remaining committee, which is to select all of the remaining members to be on the committee. Using the basic rule of multiplication, there are a total of $\binom{9}{2} \times \binom{7}{3} \times \binom{4}{4} = 1260$ ways to form the three committees. To compute the final answer, the R commands `choose()`, `factorial()`, or a combination of the two can be used as shown in R Code 3.1. Note that the following are all equivalent:

$$\binom{9}{2} \times \binom{7}{3} \times \binom{4}{4} = \frac{9!}{2!7!} \times \frac{7!}{3!4!} \times \frac{4!}{4!} = \frac{9!}{2!3!4!} = \frac{9!}{2!7!} \times \frac{7!}{3!4!} = 1260.$$

R Code 3.1

```
> choose(9, 2) * choose(7, 3) * choose(4, 4)
[1] 1260

> factorial(9)/(factorial(2) * factorial(3) * factorial(4))
[1] 1260

> choose(9, 2) * (factorial(7)/(factorial(3) * factorial(4)))
[1] 1260
```



3.3 Axiomatic Probability

When dealing with problems of uncertainty, one will often want to gain some idea of how likely something is to happen. For anything from flipping coins to rolling dice to more

important topics such as how likely one is to be in an accident on any given day, probability is how this likeliness is described. Definitions and axioms associated with probability will formalize concepts associated with possible outcomes in any non-determinate situation.

3.3.1 Sample Space and Events

A **random experiment** is any action or process that generates an outcome whose value cannot be known with certainty. The **sample space** of an experiment, denoted by Ω , is the set of all of the possible outcomes of an experiment. Although the outcome of an experiment cannot be known before the experiment has taken place, it is possible to define the sample space for a given experiment. The sample space may be either finite or infinite. For example, the number of unoccupied seats in a train corresponds to a finite sample space. The number of passengers arriving at an airport also produces a finite sample space, assuming a one-to-one correspondence between arriving passengers and the natural numbers. The sample space for the lifetime of light bulbs, however, is infinite, since lifetime may be any positive value.

An **event** is any subset of the sample space, which is often denoted with the letter E . Events are said to be **simple** when they contain only one outcome; otherwise, events are considered to be **compound**. Consider an experiment where a single die is thrown. Since the die might show any one of six numbers, the sample space is written $\Omega = \{1, 2, 3, 4, 5, 6\}$; and any subset of Ω , such as $E_1 = \{\text{even numbers}\}$, $E_2 = \{2\}$, $E_3 = \{1, 2, 4\}$, $E_4 = \Omega$, or $E_5 = \emptyset$, is considered an event. Specifically, E_2 is considered a simple event while all of the remaining events are considered to be compound events. Event E_5 is known as the **empty set** or the **null set**, the event that does not contain any outcomes. In many problems, the events of interest will be formed through a combination of two or more events by taking **unions**, **intersections**, and **complements**.

3.3.2 Set Theory

The following definitions review some basic notions from set theory and some basic rules of probability that are not unlike the rules of algebra. For any two events E and F of a sample space Ω , define the new event $E \cup F$ (read E union F) to consist of all outcomes that are either in E or in F or in both E and F . In other words, the event $E \cup F$ will occur if either E or F occurs. In a similar fashion, for any two events E and F of a sample space Ω , define the new event $E \cap F$ (read E intersection F) to consist of all outcomes that are both in E and in F . Finally, the complement of an event E (written E^c) consists of all outcomes in Ω that are not contained in E .

Given events E, F, G, E_1, E_2, \dots , the commutative, associative, distributive, and DeMorgan's laws work as follows with the union and intersection operators:

1. Commutative laws

- for the union
$$E \cup F = F \cup E$$
- for the intersection
$$E \cap F = F \cap E$$

2. Associative laws

- for the union
$$(E \cup F) \cup G = E \cup (F \cup G)$$
- for the intersection
$$(E \cap F) \cap G = E \cap (F \cap G)$$

3. Distributive laws

- $(E \cap F) \cup G = (E \cup G) \cap (F \cup G)$
- $(E \cup F) \cap G = (E \cap G) \cup (F \cap G)$

4. DeMorgan's laws

- $\left(\bigcup_{i=1}^{\infty} E_i \right)^c = \bigcap_{i=1}^{\infty} E_i^c$
- $\left(\bigcap_{i=1}^{\infty} E_i \right)^c = \bigcup_{i=1}^{\infty} E_i^c$

3.3.3 Interpreting Probability

Probability can be considered both conceptually and mathematically. Conceptually, the probability of an event occurring is how many items in the sample space are like what is described divided by the number of items in the sample space. For example, the probability of drawing a club from a standard deck of cards will be the 13 clubs divided by the 52 cards. Mathematically, various truths about probability are assumed, and conclusions are drawn from those assumed truths. Specifically, all probabilities are between zero and one; the probability of landing in the sample space of an experiment is one; and the probability of the union of disjoint events is the sum of their probabilities. Consider rolling a single die, the probability of any combinations of the values one to six must be between zero and one; the probability of rolling one of the numbers between one and six inclusive must be one; and the probability of rolling a number less than three must be the sum of the probabilities of rolling a one and rolling a two.

3.3.3.1 Relative Frequency Approach to Probability

Suppose an experiment can be performed n times under the same conditions with sample space Ω . Let $n(E)$ denote the number of times (in n experiments) that the event E occurs. The relative frequency approach to probability defines the probability of the event E , written $\mathbb{P}(E)$, as

$$\mathbb{P}(E) = \lim_{n \rightarrow \infty} \frac{n(E)}{n}.$$

Although the preceding definition of probability is intuitively appealing, it has a serious drawback. There is nothing in the definition to guarantee $\frac{n(E)}{n}$ converges to a single value. Instead of assuming $\frac{n(E)}{n}$ converges, which is a very complex assumption, the simpler and more self-evident axioms about probability commonly referred to as the **three axioms of probability** are used.

3.3.3.2 Axiomatic Approach to Probability

The Three Axioms of Probability

Consider an experiment with sample space Ω . For each event E of the sample space Ω , assume that a number $\mathbb{P}(E)$ is defined that satisfies the following three axioms:

1. $0 \leq \mathbb{P}(E) \leq 1$
2. $\mathbb{P}(\Omega) = 1$
3. For any sequence of mutually exclusive events E_1, E_2, \dots (that is $E_i \cap E_j = \emptyset$) for all $i \neq j$,

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(E_i).$$

The following results are all easily derived using some combination of the three axioms of probability:

1. $\mathbb{P}(E^c) = 1 - \mathbb{P}(E)$

Proof: Note that E and E^c are always mutually exclusive. Since $E \cup E^c = \Omega$, by probability axioms 2 and 3, $1 = \mathbb{P}(\Omega) = \mathbb{P}(E \cup E^c) = \mathbb{P}(E) + \mathbb{P}(E^c)$.

2. $\mathbb{P}(E \cup F) = \mathbb{P}(E) + \mathbb{P}(F) - \mathbb{P}(E \cap F)$

Proof: Note that $E \cup F$ can be represented as the union of two mutually exclusive events, E and $(E^c \cap F)$. That is, $E \cup F = E \cup (E^c \cap F)$. Event F can also be represented as the union of two mutually exclusive events, $(E \cap F)$ and $(E^c \cap F)$. By probability axiom 3, $\mathbb{P}(E \cup F) = \mathbb{P}(E) + \mathbb{P}(E^c \cap F)$ as well as $\mathbb{P}(F) = \mathbb{P}(E \cap F) + \mathbb{P}(E^c \cap F)$. By solving for $\mathbb{P}(E^c \cap F)$ in the second equation and substituting the answer into the first equation, the desired result of $\mathbb{P}(E \cup F) = \mathbb{P}(E) + \mathbb{P}(F) - \mathbb{P}(E \cap F)$ is obtained.

3. $\mathbb{P}(\emptyset) = 0$

Proof: Consider two events, E_1 and E_2 , where $E_1 = \Omega$ and $E_2 = \emptyset$. Note that $\Omega = E_1 \cup E_2$ and $E_1 \cap E_2 = \emptyset$. By probability axioms 2 and 3, $1 = \mathbb{P}(\Omega) = \mathbb{P}(E_1) + \mathbb{P}(E_2) = 1 + \mathbb{P}(\emptyset) \implies \mathbb{P}(\emptyset) = 0$.

4. If $E \subset F$, then $\mathbb{P}(E) \leq \mathbb{P}(F)$

Proof: Since $E \subset F$, it follows that $F = E \cup (E^c \cap F)$. Note that E and $(E^c \cap F)$ are mutually exclusive events. Consequently, appealing to probability axiom 3, $\mathbb{P}(F) = \mathbb{P}(E) + \mathbb{P}(E^c \cap F)$. Since $\mathbb{P}(E^c \cap F) \geq 0$ by probability axiom 1, it follows that $\mathbb{P}(F) \geq \mathbb{P}(E)$.

Example 3.9 ▷ Law of Complement: Birthday Problem ◁ Suppose that a room contains m students. What is the probability that at least two of them have the same birthday? This is a famous problem with a counterintuitive answer. Assume that every day of the year is equally likely to be a birthday, and disregard leap years. That is, assume there are always $n = 365$ days to a year.

Solution: Let the event E denote two or more students with the same birthday. In this problem, it is easier to find E^c , as there are a number of ways that E can take place. There are a total of 365^m possible outcomes in the sample space. E^c can occur in $365 \times 364 \times \cdots \times (365 - m + 1)$ ways. Consequently,

$$\mathbb{P}(E^c) = \frac{365 \times 364 \times \cdots \times (365 - m + 1)}{365^m}$$

and

$$\mathbb{P}(E) = 1 - \frac{365 \times 364 \times \cdots \times (365 - m + 1)}{365^m}.$$

R Code 3.2 can be used to create or modify a table such as Table 3.1 on the facing page, which gives $\mathbb{P}(E)$ for $m = 10, 15, \dots, 50$.

R Code 3.2

```
> m <- seq(10, 50, 5)
> P.E <- function(m){
+   c(Students = m, ProbAtL2SB = 1 - prod((365:(365 - m + 1)/365)))
+ }
> t(sapply(m, P.E))

  Students ProbAtL2SB
[1,]      10  0.1169482
[2,]      15  0.2529013
[3,]      20  0.4114384
[4,]      25  0.5686997
[5,]      30  0.7063162
[6,]      35  0.8143832
[7,]      40  0.8912318
[8,]      45  0.9409759
[9,]      50  0.9703736
```

R Code 3.3 can be used to create or modify a graph such as the one shown in Figure 3.1 on the facing page that plots the probability of at least two students with the same birthday versus the number of students in the room. The dashed horizontal line in Figure 3.1 on the next page is drawn at 0.5 so that the reader can obtain an idea of the number of students required for the probability to exceed 0.5, which is 23. A dashed vertical line is drawn at 23 so the reader can see that the probability of at least two students having the same birthday is greater than 0.5 for $m \geq 23$.

R Code 3.3

```
> m <- 1:60          # vector of number of students
> p <- numeric(60)  # initialize vector to 0's
> for(i in m){       # index values for loop
+   q = prod((365:(365 - i + 1))/365) # P(No Match) if i people in room
+   p[i] = 1 - q}
> plot(m, p, col = "skyblue3", pch = 19,
+       ylab = "P(at least 2 students with the same birthday)",
+       xlab = "m = Number of students in the room")
> abline(h = 0.5, lty = 2, col = "red")      # Add horizontal line
> abline(v = 23, lty = 2, col = "red")        # Add vertical line
```

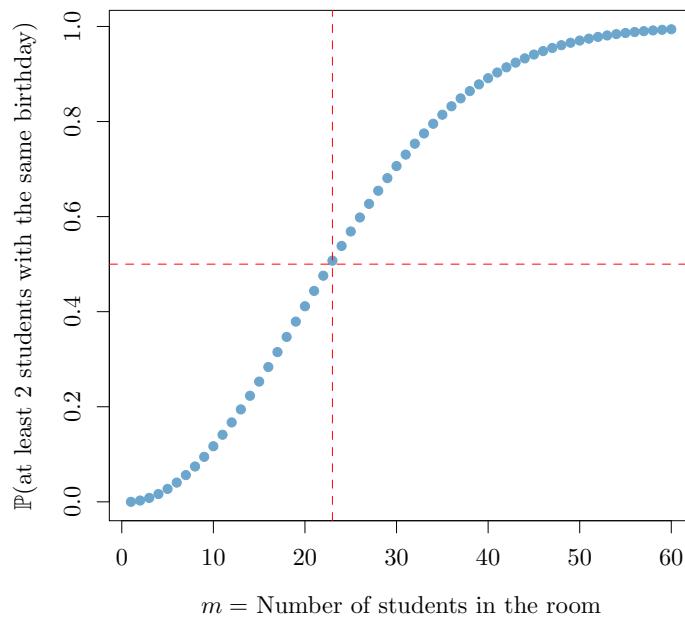


FIGURE 3.1: Probability of two or more students having the same birthday

Table 3.1: Probability of two or more students having the same birthday

m	$\mathbb{P}(E)$
10	0.1169482
15	0.2529013
20	0.4114384
25	0.5686997
30	0.7063162
35	0.8143832
40	0.8912318
45	0.9409759
50	0.9703736

■

Example 3.10 In a class using this text, suppose that 30% of the students are taking a computer science course, 45% of the students are taking a mathematics course, and 10% are taking both a computer science and a mathematics course.

- (a) What is the probability a randomly selected student is taking a computer science course or a mathematics course?
- (b) What is the probability a randomly selected student is taking a computer science course but not a mathematics course?
- (c) What is the probability a randomly selected student is taking a mathematics course but not a computer science course?

- (d) What is the probability a randomly selected student is taking neither a mathematics course nor a computer science course?

Solution: Let the events M and C represent taking a mathematics and a computer science course, respectively. Then, $\mathbb{P}(C) = 0.30$, $\mathbb{P}(M) = 0.45$, and $\mathbb{P}(C \cap M) = 0.10$.

(a) Since $\mathbb{P}(C \cup M) = \mathbb{P}(C) + \mathbb{P}(M) - \mathbb{P}(C \cap M)$, $\mathbb{P}(C \cup M) = 0.3 + 0.45 - 0.10$. Thus, $\mathbb{P}(C \cup M) = 0.65$.

(b) Since $\mathbb{P}(C \cap M^c) = \mathbb{P}(C) - \mathbb{P}(C \cap M)$, $\mathbb{P}(C \cap M^c) = 0.3 - 0.10$. Thus, $\mathbb{P}(C \cup M^c) = 0.20$.

(c) Since $\mathbb{P}(C^c \cap M) = \mathbb{P}(M) - \mathbb{P}(C \cap M)$, $\mathbb{P}(C^c \cap M) = 0.45 - 0.10$. Thus, $\mathbb{P}(C^c \cup M) = 0.35$.

(d) Since $\mathbb{P}(C^c \cap M^c) = \mathbb{P}[(C \cup M)^c] = 1 - \mathbb{P}(C \cup M)$, $\mathbb{P}(C^c \cap M^c) = 1 - 0.65$. Thus, $\mathbb{P}(C^c \cap M^c) = 0.35$.



3.3.4 Conditional Probability

In this section, conditional probability is introduced, which is one of the more important concepts in probability theory. Quite often, one is interested in calculating probabilities when only partial information obtained from an experiment is available. In such situations, the desired probabilities are said to be conditional. Even when partial information is unavailable, often the desired probabilities can be computed using conditional probabilities. If E and F are any two events in a sample space Ω and $\mathbb{P}(E) \neq 0$, the **conditional probability** of F given E is defined as

$$\mathbb{P}(F|E) = \frac{\mathbb{P}(F \cap E)}{\mathbb{P}(E)}. \quad (3.1)$$

It is left as an exercise for the reader to verify that $\mathbb{P}(F|E)$ satisfies the three axioms of probability.

Example 3.11 Suppose two fair dice are rolled where each of the 36 possible outcomes is equally likely to occur. Knowing that the first die shows a 4, what is the probability that the sum of the two dice equals 8?

Solution: The sample space for this experiment is given as $\Omega = \{(i, j), i = 1, 2, \dots, 6, j = 1, 2, \dots, 6\}$, where each pair (i, j) has a probability $1/36$ of occurring. Define “the sum of the dice equals 8” to be event H and “a 4 on the first toss” to be event G . Since $H \cap G$ corresponds to the outcome $(4, 4)$ with probability $\mathbb{P}(H \cap G) = 1/36$ and there are six outcomes with a 4 on the first toss, $(4, 1), (4, 2), \dots, (4, 6)$, the probability of event H , $\mathbb{P}(H) = 6/36 = 1/6$ and the answer is calculated as

$$\mathbb{P}(H|G) = \frac{\mathbb{P}(H \cap G)}{\mathbb{P}(G)} = \frac{1/36}{1/6} = \frac{1}{6}.$$

The R function `expand.grid()` is used to enumerate the possible outcomes in the sample space, Ω , which is stored in the R object `Omega` in R Code 3.4.

R Code 3.4

```
> library(MASS) # used for fractions function
> Omega <- expand.grid(roll1 = 1:6, roll2 = 1:6)
```

```

> H <- subset(Omega, roll1 + roll2 == 8)
> H

  roll1 roll2
12      6     2
17      5     3
22      4     4
27      3     5
32      2     6

> G <- subset(Omega, roll1 == 4)
> G

  roll1 roll2
4      4     1
10     4     2
16     4     3
22     4     4
28     4     5
34     4     6

> PG <- dim(G)[1]/dim(Omega)[1]  # P(G)
> fractions(PG)

[1] 1/6

> HaG <- subset(Omega, roll1 == 4 & roll2 == 4)  # event H and G
> HaG

  roll1 roll2
22      4     4

> PHaG <- dim(HaG)[1]/dim(Omega)[1]  # P(H and G)
> fractions(PHaG)

[1] 1/36

> PHgG <- PHaG/PG  # P(H|G)
> fractions(PHgG)

[1] 1/6

```

It is often the case that it is easier to solve a conditional probability problem, $\mathbb{P}(H|G)$, by enumerating the number of outcomes in the conditioning event G , then enumerating the number of outcomes in H in the reduced sample space G . Consider R Code 3.5, which takes this approach to solve $\mathbb{P}(H|G)$.

R Code 3.5

```

> library(MASS)
> Omega <- expand.grid(roll1 = 1:6, roll2 = 1:6)
> G <- subset(Omega, roll1 == 4)           # event G
> G

```

```

roll1 roll2
4      4     1
10     4     2
16     4     3
22     4     4
28     4     5
34     4     6

> HgG <- subset(G, roll1 + roll2 == 8)  # event H/G
> HgG

roll1 roll2
22     4     4

> HgG <- subset(G, roll2 == 4)           # event H/G
> HgG

roll1 roll2
22     4     4

> HgG <- subset(G, roll2 %in% 4)         # event H/G
> HgG

roll1 roll2
22     4     4

> PHgG <- dim(HgG)[1]/dim(G)[1]          # P(H/G)
> fractions(PHgG)

[1] 1/6

```



Example 3.12 Suppose a box contains 50 defective light bulbs, 100 partially defective light bulbs (last only 3 hours), and 250 good light bulbs. If one of the bulbs from the box is used and it does not immediately go out, what is the probability the light bulb is actually a good light bulb?

Solution: The conditional probability the light bulb is good given that the light bulb is not defective is desired. Using (3.1), write

$$\mathbb{P}(\text{Good}|\text{Not Defective}) = \frac{\mathbb{P}(\text{Good})}{\mathbb{P}(\text{Not Defective})} = \frac{250/400}{350/400} = \frac{5}{7}.$$



3.3.5 The Law of Total Probability and Bayes' Rule

At times, it is much easier to calculate the conditional probabilities $\mathbb{P}(E|F_i)$ for an appropriately selected F_i than it is to compute $\mathbb{P}(E)$ directly. An important tool for solving probability problems where the sample space can be considered a union of mutually exclusive events is the **Law of Total Probability**. **Law of Total Probability** — Let F_1, F_2, \dots, F_n be such that $\bigcup_{i=1}^n F_i = \Omega$ and $F_i \cap F_j = \emptyset$ for all $i \neq j$, with

$\mathbb{P}(F_i) > 0$ for all i . Then, for any event E ,

$$\mathbb{P}(E) = \sum_{i=1}^n \mathbb{P}(E \cap F_i) = \sum_{i=1}^n \mathbb{P}(E|F_i)\mathbb{P}(F_i). \quad (3.2)$$

In other situations, we are interested in computing $\mathbb{P}(F_j|E)$. When this happens, **Bayes' Rule** is used, which is derived using (3.1), to find the answer. **Bayes' Rule** — Let F_1, F_2, \dots, F_n be such that $\bigcup_{i=1}^n F_i = \Omega$ and $F_i \cap F_j = \emptyset$ for all $i \neq j$, with $\mathbb{P}(F_i) > 0$ for all i . Then,

$$\mathbb{P}(F_j|E) = \frac{\mathbb{P}(E \cap F_j)}{\mathbb{P}(E)} = \frac{\mathbb{P}(E|F_j)\mathbb{P}(F_j)}{\sum_{i=1}^n \mathbb{P}(E|F_i)\mathbb{P}(F_i)}. \quad (3.3)$$

Example 3.13 ▷ **Conditional Probability: Car Batteries** ◁ A car manufacturer purchases car batteries from two different suppliers. Supplier A provides 55% of the batteries and supplier B provides the rest. If 5% of all batteries from supplier A are defective and 4% of the batteries from supplier B are defective, determine the probability that a randomly selected battery is not defective.

Solution: Let C correspond to the event “the battery does not work properly,” A to the event “the battery was supplied by A ,” and B to the event “the battery was supplied by B .” The Venn diagram in Figure 3.2 provides a graphical illustration of the sample space for this example. Since a working battery might come from either supplier A or B , A and B are disjoint events. Consequently, $\mathbb{P}(C) = \mathbb{P}(C \cap A) + \mathbb{P}(C \cap B)$. Given that

$$\begin{aligned} \mathbb{P}(A) &= 0.55, \mathbb{P}(C|A) = 0.05, & \mathbb{P}(C \cap A) &= \mathbb{P}(C|A)\mathbb{P}(A), \\ \mathbb{P}(B) &= 0.45, \mathbb{P}(C|B) = 0.04, & \mathbb{P}(C \cap B) &= \mathbb{P}(C|B)\mathbb{P}(B), \end{aligned}$$

write $\mathbb{P}(C) = (0.05)(0.55) + (0.04)(0.45) = 0.0455$. Then, the probability that the battery works properly is $1 - \mathbb{P}(C) = 0.9545$.

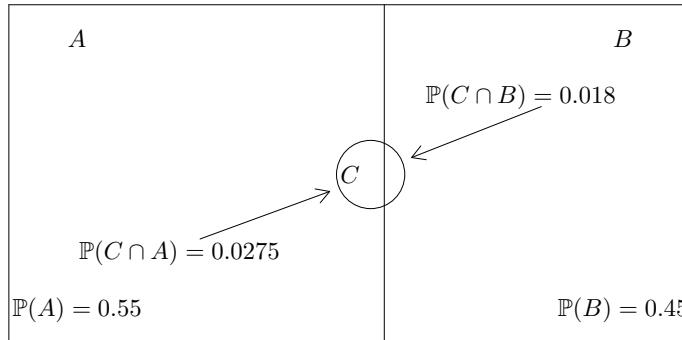


FIGURE 3.2: Sample space for Example 3.13



Example 3.14 Suppose a student answers all of the questions on a multiple-choice test. Let p be the probability the student actually knows the answer and $1 - p$ be the probability the student is guessing for a given question. Assume students that guess have a $1/a$ probability of getting the correct answer, where a represents the number of possible responses to the question. What is the conditional probability a student knew the answer to a question given that he answered correctly?

Solution: Let the events E , F_1 , and F_2 represent the events “question answered correctly,” “student knew the correct answer,” and “student guessed,” respectively. Using (3.3), write

$$\mathbb{P}(F_1|E) = \frac{\mathbb{P}(F_1 \cap E)}{\mathbb{P}(E)} = \frac{\mathbb{P}(F_1)}{\mathbb{P}(E|F_1)\mathbb{P}(F_1) + \mathbb{P}(E|F_2)\mathbb{P}(F_2)} = \frac{p}{p + (1-p)/a}.$$

As a special case, if $a = 4$ and $p = 1/2$, then the probability a student actually knew the answer given their response was correct is $4/5$. ■

Example 3.15 ▷ Bayes' Rule: Choose a Door ◁ The television show *Let's Make a Deal*, hosted by Monty Hall, gave contestants the opportunity to choose one of three doors. Contestants hoped to choose the one that concealed the grand prize. Behind the other two doors were much less valuable prizes. After the contestant chose one of the doors, say Door 1, Monty opened one of the other two doors, say Door 3, containing a much less valuable prize. The contestant was then asked whether he or she wished to stay with the original choice (Door 1) or switch to the other closed door (Door 2). What should the contestant do? Is it better to stay with the original choice or to switch to the other closed door? Or does it really matter? The answer, of course, depends on whether contestants improve their chances of winning by switching doors. In particular, what is the probability of winning by switching doors when given the opportunity; and what is the probability of winning by staying with the initial door selection? First, simulate the problem with R to provide approximate probabilities for the various strategies. Following the simulation, show how Bayes' Rule can be used to solve the problem exactly.

Solution: To simulate the problem, generate a random vector named `actual` of size 10,000 containing the numbers 1, 2, and 3. In the vector `actual`, the numbers 1, 2, and 3 represent the door behind which the grand prize is contained. Then, generate another vector named `aguess` of size 10,000 containing the numbers 1, 2, and 3 to represent the contestant's initial guess. If the i^{th} values of the vectors `actual` and `aguess` agree, the contestant wins the grand prize by staying with his initial guess. On the other hand, if the i^{th} values of the vectors `actual` and `aguess` disagree, the contestant wins the grand prize by switching. Consider R Code 3.6 and the results that suggest the contestant is twice as likely to win the grand prize by switching doors.

R Code 3.6

```
> set.seed(2) # done for reproducibility
> actual <- sample(x = 1:3, size = 10000, replace = TRUE)
> aguess <- sample(x = 1:3, size = 10000, replace = TRUE)
> equals <- (actual == aguess)
> PNoSwitch <- sum>equals)/10000
> not.eq <- (actual != aguess)
> PSwitch <- sum(not.eq)/10000
> Probs <- c(PNoSwitch, PSwitch)
> names(Probs) <- c("P(Win no Switch)", "P(Win Switch)")
> Probs

P(Win no Switch)      P(Win Switch)
        0.3348            0.6652
```

To solve with Bayes' Rule, start by assuming the contestant initially guesses Door 1 and that Monty opens Door 3. Let the event $D_i = \text{Door } i \text{ conceals the prize}$ and $O_j = \text{Monty opens door } j \text{ after the contestant selects Door 1}$. When a contestant initially selects a door,

$\mathbb{P}(D_1) = \mathbb{P}(D_2) = \mathbb{P}(D_3) = 1/3$. (Similar reasoning works independent of the originally selected door.) Once Monty shows the grand prize is not behind Door 3, the probability of winning the grand prize is now one of $\mathbb{P}(D_1|O_3)$ or $\mathbb{P}(D_2|O_3)$. Note that $\mathbb{P}(D_1|O_3)$ corresponds to the strategy of sticking with the initial guess and $\mathbb{P}(D_2|O_3)$ corresponds to the strategy of switching doors. Based on how the show is designed, the following are known:

- $\mathbb{P}(O_3|D_1) = 1/2$ since Monty can open one of either Door 3 or Door 2 without revealing the grand prize.
- $\mathbb{P}(O_3|D_2) = 1$ since the only door Monty can open without revealing the grand prize is Door 3.
- $\mathbb{P}(O_3|D_3) = 0$ since Monty will not open Door 3 if it contains the grand prize.

$$\begin{aligned}\mathbb{P}(D_1|O_3) &= \frac{\mathbb{P}(O_3|D_1)\mathbb{P}(D_1)}{\mathbb{P}(O_3|D_1)\mathbb{P}(D_1) + \mathbb{P}(O_3|D_2)\mathbb{P}(D_2) + \mathbb{P}(O_3|D_3)\mathbb{P}(D_3)} \\ &= \frac{1/2 \times 1/3}{1/2 \times 1/3 + 1 \times 1/3 + 0 \times 1/3} = \frac{1}{3}.\end{aligned}$$

$$\begin{aligned}\mathbb{P}(D_2|O_3) &= \frac{\mathbb{P}(O_3|D_2)\mathbb{P}(D_2)}{\mathbb{P}(O_3|D_1)\mathbb{P}(D_1) + \mathbb{P}(O_3|D_2)\mathbb{P}(D_2) + \mathbb{P}(O_3|D_3)\mathbb{P}(D_3)} \\ &= \frac{1 \times 1/3}{1/2 \times 1/3 + 1 \times 1/3 + 0 \times 1/3} = \frac{2}{3}.\end{aligned}$$

Therefore, it is always to the contestant's benefit to switch doors. ■

3.3.6 Independent Events

Conditional probability allows for an alteration in the probability of an event when additional information is present. That is, $\mathbb{P}(E|F)$ is sometimes different from $\mathbb{P}(E)$ when some knowledge of the event F is available. Note that $\mathbb{P}(E|F)$ is *sometimes* different from $\mathbb{P}(E)$, not that it is *always* different. When $\mathbb{P}(E|F) = \mathbb{P}(E)$, clearly knowledge of the event F does not alter the probability of obtaining E . When this happens, event E is **independent** of event F . More formally, two events E and F are independent if and only if $\mathbb{P}(E|F) = \mathbb{P}(E)$ and $\mathbb{P}(F|E) = \mathbb{P}(F)$. An equivalent way to define independence between two events is to use (3.1) and to show that $\mathbb{P}(E \cap F) = \mathbb{P}(E) \cdot \mathbb{P}(F)$. Independence of two events is really a special case of independence among n events. Define events E_1, \dots, E_n to be independent if, for every k where $k = 2, \dots, n$ and every subset of indices i_1, i_2, \dots, i_k , $\mathbb{P}(E_{i_1} \cap E_{i_2} \cap \dots \cap E_{i_k}) = \mathbb{P}(E_{i_1}) \cdot \mathbb{P}(E_{i_2}) \cdot \dots \cdot \mathbb{P}(E_{i_k})$. It is important to point out that events in any subset of the original independent events of size r , where $r \leq k$, are also independent. Further, if events E_1, \dots, E_n are independent, then so are E_1^c, \dots, E_n^c .

Example 3.16 ▷ **Law of Probability: Components** ◁ A system consists of three components as illustrated in Figure 3.3 on the following page. The entire system will work if either both components 1 and 2 work or if component 3 works. Components 1 and 2 are connected in series, while component 3 is connected in parallel with components 1 and 2. If all of the components function independently, and the probability each component works is 0.9, what is the probability the entire system functions?

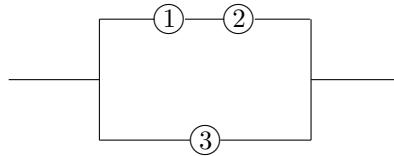


FIGURE 3.3: Circuit system diagram for Example 3.16

Solution: Let A_i ($i = 1, 2, 3$) be the event the i^{th} component works, and E the event the entire system works. Consequently, event $E = (A_1 \cap A_2) \cup A_3$, and $\mathbb{P}(E) = \mathbb{P}[(A_1 \cap A_2) \cup A_3]$.

$$\begin{aligned}\mathbb{P}(E) &= \mathbb{P}[(A_1 \cap A_2) \cup A_3] \\ &= \mathbb{P}(A_1 \cap A_2) + \mathbb{P}(A_3) - \mathbb{P}(A_1 \cap A_2 \cap A_3) \\ &= \mathbb{P}(A_1)\mathbb{P}(A_2) + \mathbb{P}(A_3) - \mathbb{P}(A_1)\mathbb{P}(A_2)\mathbb{P}(A_3) \\ &= (0.9)(0.9) + 0.9 - (0.9)(0.9)(0.9) \\ &= 0.981.\end{aligned}$$

The probability that the system works is 0.981. ■

3.4 Random Variables

In many experiments, it is easier to study some function of the outcomes than it is to study the original outcomes. For example, suppose 20 students are asked whether they favor legislation to reduce ozone emissions. Note that there are $2^{20} = 1,048,576$ possible outcomes in the sample space. However, it would make more sense to study the number of students who favor (equivalently, oppose) legislation out of 20 by defining a variable, say X , that equals the number of students favoring (or opposing) the legislation. Note that the sample space for X is the set of integers from 0 to 20, which is much easier to deal with than the original sample space. In general, a **random variable** is a function from a sample space Ω into the real numbers. Random variables will always be denoted with uppercase letters, for example, X or Y , and the realized values of the random variable will be denoted with lowercase letters, for example, x or y . Here are some examples of random variables:

1. Toss two dice. X = the sum of the numbers on the dice.
2. A surgeon performs 20 heart transplants. X = the number of successful transplants.
3. Individual 40-kilometer cycling time trial. X = the time to complete the course.

Random variables may be either **discrete** or **continuous**. A random variable is said to be discrete if its set of possible outcomes is finite or at most countable. If the random variable can take on a continuum of values, it is continuous. Note that the random variables in examples 1 and 2 are discrete, while the variable in example 3 is continuous. If a random variable X has a distribution $DIST$ with parameter(s) θ , write $X \sim DIST(\theta)$. If Y is a random variable that is distributed approximately $DIST$ with parameter(s) θ , write $Y \approx DIST(\theta)$.

3.4.1 Discrete Random Variables

A discrete random variable assumes each of its values with a certain probability. When two dice are tossed, the probability the sum of two dice is 7, written $\mathbb{P}(X = 7)$, equals 1/6. The function that assigns probability to the values of the random variable is called the probability density function, **pdf**. Many authors also refer to the **pdf** as the probability mass function (**pmf**) when working with discrete random variables. Denote the **pdf** as $p(x) = \mathbb{P}(X = x)$ for each x . All **pdfs** must satisfy the following two conditions:

1. $p(x) \geq 0$ for all x .
2. $\sum_{\forall x} p(x) = 1$.

An estimate of $f(x) = p(x) = \mathbb{P}(X = x)$ is the number of sample points equal to x divided by n , the sample size. This estimate is called the **empirical probability density function**, **epdf** or the **empirical probability mass function**, **epmf**, and is defined as

$$\hat{f}_n(t) = \sum_{i=1}^n \mathbf{I}\{x_i = t\}/n. \quad (3.4)$$

Here, $\mathbf{I}\{x_i = t\}$ is the indicator function that returns a value of 1 when $x_i = t$ and 0 when $x_i \neq t$.

The cumulative distribution function, **cdf**, is defined as

$$F(x) = \mathbb{P}(X \leq x) = \sum_{k \leq x} p(k).$$

The **cdfs** of discrete random variables have the following properties:

1. $0 \leq F(x) \leq 1$.
2. If $a < b$, then $F(a) \leq F(b)$ for any real numbers a and b . In other words, $F(x)$ is a non-decreasing function of x .
3. $\lim_{x \rightarrow \infty} F(x) = 1$.
4. $\lim_{x \rightarrow -\infty} F(x) = 0$.
5. $F(x)$ is a step function, and the height of the step at x is equal to $f(x) = \mathbb{P}(X = x)$.

An estimate of $F(x) = \mathbb{P}(X \leq x)$ is the proportion of sample values that fall in the interval $(-\infty, x]$. This estimate is called the **empirical cumulative distribution function**, **ecdf**, and is defined as

$$\hat{F}_n(t) = \sum_{i=1}^n \mathbf{I}\{x_i \leq t\}/n. \quad (3.5)$$

Here, $\mathbf{I}\{x_i \leq t\}$ is the indicator function that returns a value of 1 when $x_i \leq t$ and 0 when $x_i > t$. The R function `ecdf()` will compute $\hat{F}_n(t)$ when a numeric vector of sample values is supplied to the argument `x=`. If the values supplied to `x=` are values of a discrete random variable, one may use `plot(ecdf(x))` to graph the **cdf**.

Example 3.17 Toss a fair coin three times and let the random variable X represent the number of heads in the three tosses. Produce graphical representations of both the **pdf** and **cdf** for the random variable X .

Solution: The sample space for the experiment is

$$\Omega = \{TTT, HTT, THT, HHT, TTH, HTH, THH, HHH\}$$

The random variable X can take on the values 0, 1, 2, and 3 with probabilities $\frac{1}{8}$, $\frac{3}{8}$, $\frac{3}{8}$, and $\frac{1}{8}$, respectively. Define the **cdf** for X , $F(x) = \mathbb{P}(X \leq x)$ as follows:

$$F(x) = \begin{cases} 0 & \text{if } x < 0, \\ 1/8 & \text{if } 0 \leq x < 1, \\ 4/8 & \text{if } 1 \leq x < 2, \\ 7/8 & \text{if } 2 \leq x < 3, \text{ and} \\ 1 & \text{if } x \geq 3. \end{cases}$$

The code for producing a graph similar to Figure 3.4 on the next page with placement of specific values along the axes for both the **pdf** and **cdf** using the function **axis()** is given in R Code 3.7.

R Code 3.7

```
> opar <- par(no.readonly = TRUE)
> library(MASS) # used for fractions function
> par(mfrow=c(1, 2), pty = "s")
> Omega <- expand.grid(coin1 = 0:1, coin2 = 0:1, coin3 = 0:1)
> n.heads <- apply(Omega, 1, sum)
> cbind(Omega, n.heads)

  coin1 coin2 coin3 n.heads
1      0      0      0      0
2      1      0      0      1
3      0      1      0      1
4      1      1      0      2
5      0      0      1      1
6      1      0      1      2
7      0      1      1      2
8      1      1      1      3

> T1 <- table(n.heads)/length(n.heads)
> fractions(T1)

n.heads
0   1   2   3
1/8 3/8 3/8 1/8

> plot(T1, xlab = "x", ylab="P(X = x)", yaxt = "n", main = "PDF for X")
> axis(2, at = c(1/8, 3/8), labels = c("1/8", "3/8"), las = 1)
```

```

> plot(ecdf(n.heads), main = "CDF for X", ylab = "F(x)", xlab = "x",
+       yaxt = "n")
> axis(2, at = c(1/8, 4/8, 7/8, 1), labels = c("1/8", "4/8", "7/8", "1"),
+       las = 1)
> segments(1, 1/8, 1, 4/8, lty = 2)
> text(2.6, 2.5/8, "P(X = 1) = F(1) - F(0)")
> par(opar)

```

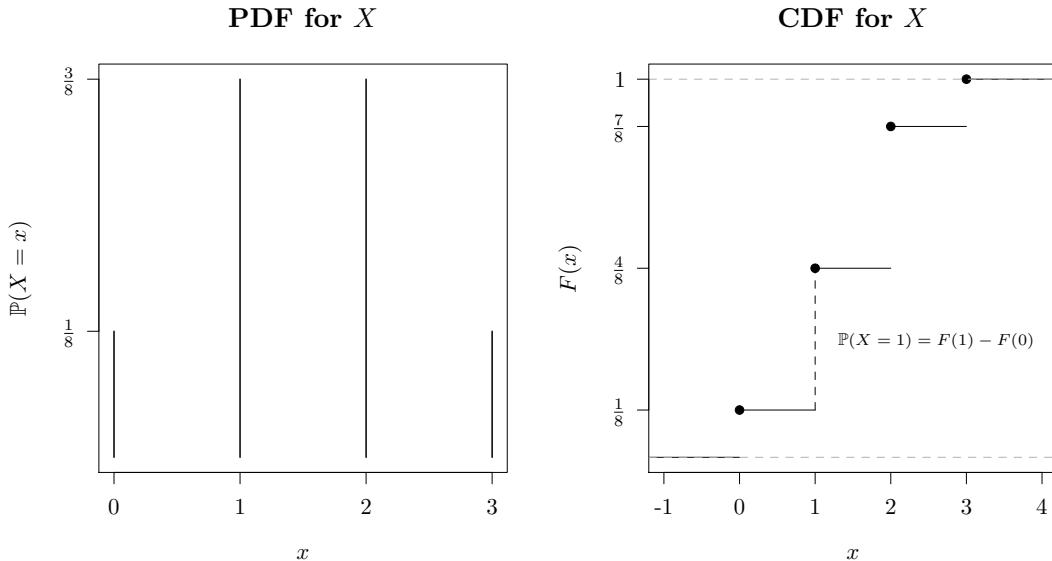


FIGURE 3.4: The **pdf** and **cdf** for the random variable X , the number of heads in three tosses of a fair coin

3.4.1.1 Mode, Median, and Percentiles

The **mode** of a probability distribution is the x -value most likely to occur. If more than one such x value exists, the distribution is multimodal. The **median** of a distribution is the value m such that $\mathbb{P}(X \leq m) \geq 1/2$ and $\mathbb{P}(X \geq m) \geq 1/2$. The j^{th} **percentile** of a distribution is the value x_j such that $\mathbb{P}(X \leq x_j) \geq \frac{j}{100}$ and $\mathbb{P}(X \geq x_j) \geq 1 - \frac{j}{100}$. The m value that satisfies the definition for the median is not unique. If Example 3.17 on page 215 is considered, the modes are 1 and 2; and any value m between 1 and 2 inclusive satisfies the definition for the median. The 25th percentile of the distribution of X is 1 because $\mathbb{P}(X \leq 1) = \frac{4}{8} \geq \frac{25}{100}$ and $\mathbb{P}(X \geq 1) = \frac{7}{8} \geq 1 - \frac{25}{100}$.

3.4.1.2 Expected Values

One of the more important ideas about summarizing the information provided in a **pdf** is that of expected value. Given a discrete random variable X with **pdf** $p(x)$, the **expected value** of the random variable X , written $E[X]$, is

$$E[X] = \sum_x x \cdot p(x). \quad (3.6)$$

$E[X]$ can also be denoted as μ_X , because $E[X]$ is the mean of the random variable X . In this definition, it is assumed the sum exists; otherwise, the expectation is undefined. It can be helpful to think of $E[X]$ as the fulcrum on a balanced beam as illustrated in Figure 3.5.

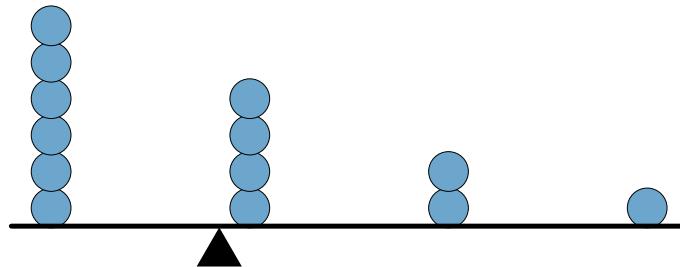


FIGURE 3.5: Fulcrum illustration of $E[X]$

Example 3.18 A particular game is played where the contestant spins a wheel that can land on the numbers 1, 5, or 30 with probabilities of 0.50, 0.45, and 0.05, respectively. The contestant pays \$5 to play the game and is awarded the amount of money indicated by the number where the spinner lands. Is this a fair game?

Solution: By fair, it is meant that the contestant should have an expected return equal to the price she pays to play the game. To answer the question, the expected (average) winnings from playing the game need to be computed. Let the random variable X represent the player's winnings:

$$E[X] = \sum_x x \cdot p(x) = (1 \times 0.50) + (5 \times 0.45) + (30 \times 0.05) = 4.25.$$

Therefore, this game is not fair, as the house makes an average of 75 cents each time the game is played. Another interpretation of the expected value of the random variable X is to view it as a weighted mean. Code to compute the expected value using (3.6) and using the function `weighted.mean()` is illustrated in R Code 3.8.

R Code 3.8

```
> x <- c(1, 5, 30)
> px <- c(0.5, 0.45, 0.05)
> EX <- sum(x * px)
> WM <- weighted.mean(x, px)
> c(EX, WM)

[1] 4.25 4.25
```



Often, a random variable itself is not of interest, but rather some function of the random variable X is important, say $g(X)$. The expected value of a function $g(X)$ of the random variable X with pdf $p(x)$ is

$$E[g(X)] = \sum_x g(x) \cdot p(x). \quad (3.7)$$

Example 3.19 Consider Example 3.18, for which the random variable Y is defined to be the player's net return. That is, $Y = X - 5$ since the player spends \$5 to play the game. What is the expected value of Y ?

Solution: The expected value of Y is

$$E[Y] = \sum_x (x - 5) \cdot p(x) = (-4 \times 0.50) + (0 \times 0.45) + (25 \times 0.05) = -0.75.$$

R Code 3.9 uses both (3.7) and the `weighted.mean()` function to compute the $E[Y]$.

R Code 3.9

```
> x <- c(1, 5, 30)
> px <- c(0.5, 0.45, 0.05)
> EgX <- sum((x - 5) * px)
> WgM <- weighted.mean((x - 5), px)
> c(EgX, WgM)

[1] -0.75 -0.75
```



Rules of Expected Value The function $g(X)$ is often a linear function $a + bX$, where a and b are constants. When this occurs, $E[g(X)]$ is easily computed from $E[X]$. In Example 3.19, a and b were -5 and 1, respectively, for the linear function $g(X)$. The following rules for expected value, when working with a random variable X and constants a and b , are true:

1. $E[bX] = bE[X]$.
2. $E[a + bX] = a + bE[X]$.

Unfortunately, if $g(X)$ is not a linear function of X , such as $g(X) = X^2$, the $E[X^2] \neq (E[X])^2$. In general, $E[g(X)] \neq g(E[X])$.

Though knowing the expected value of a random variable is important, the mean of the distribution does not tell the whole story. Several distributions may have the same mean. In this case, additional information, such as the spread of the distribution and the symmetry of the distribution, is helpful in distinguishing among various distributions.

3.4.1.3 Moments

There exist special quantities that measure mean, spread, and symmetry, called **moments**. The r^{th} **moment about the origin** of a random variable X , denoted α_r , is defined as $E[X^r]$. Note that $\alpha_1 = E[X^1]$ is the mean of the distribution of X , which can also be denoted μ_X or simply μ . The r^{th} **moment about the mean** of a random variable X , denoted μ_r , is the expected value of $(X - \mu)^r$. (Note that this quantity does not always exist.) For the r^{th} moment about the origin of a discrete random variable to be well-defined, $\sum_{i=1}^{\infty} |x_i^r| \mathbb{P}(X = x_i)$ must be less than ∞ .

Moments about 0 and μ

$$E[X^r] = \alpha_r \tag{3.8}$$

$$E[(X - \mu)^r] = \mu_r$$

Variance The second moment about the mean is called the **variance** of the distribution of X , or simply the variance of X :

$$\text{Var}[X] = \sigma_X^2 = E[(X - \mu)^2] = E[X^2] - \mu^2. \quad (3.9)$$

The positive square root of the variance is called the **standard deviation** and is denoted σ_X . The units of measurement for standard deviation are always the same as those for the random variable X . One way to avoid this unit dependency is to use the **coefficient of variation**, a unitless measure of variability. A unitless measure of variability is helpful when comparing distributions measured on different scales. It is also useful when comparing distributions with means that may be an order or more of magnitude different since the coefficient of variation expresses the variability of a distribution in relation to the mean of the distribution.

DEFINITION 3.2: Coefficient of variation — When $E[X] \neq 0$,

$$CV_X = \frac{\sigma_X}{E[X]}. \quad (3.10)$$

When only a sample of data from a population of interest is available, the coefficient of variation is estimated as

$$\widehat{CV}_X = \frac{S}{\bar{X}}. \quad (3.11)$$

Rules of Variance If X is a random variable with mean μ and a and b are constants, then

1. $\text{Var}[b] = 0$.
2. $\text{Var}[aX] = a^2 \text{Var}[X]$.
3. $\text{Var}[aX + b] = a^2 \text{Var}[X]$.

Note that once $\text{Var}[aX + b] = a^2 \text{Var}[X]$ is proved, $\text{Var}[b] = 0$ and $\text{Var}[aX] = a^2 \text{Var}[X]$ have been implicitly shown.

Proof:

$$\begin{aligned} \text{Var}[aX + b] &= E[(aX + b - E[aX + b])^2] = E[(aX + b - a\mu - b)^2] \\ &= E[(aX - a\mu)^2] = a^2 E[(X - \mu)^2] = a^2 \text{Var}[X]. \end{aligned}$$

Example 3.20 ▷ Craps Example: Probability Computations ◁ Pass line betting in bank/casino craps works as follows: A player places a bet anywhere inside the area of the table marked PASS LINE and waits for the shooter (the person rolling the dice) to roll the dice. The first roll the shooter takes is called the come out roll. If the come out roll (sum of the two dice) is either a 7 or an 11, all bettors win the amount of money they have placed in the PASS section of the table. If the come out roll is a 2, 3, or 12 (crapping out), all bettors lose the amount of money they have placed in the PASS LINE section of the table (see Figure 1.19 on page 77). If the come out roll is any other number (4, 5, 6, 8, 9, or 10), then that number is called the shooter's point, and the shooter rolls the dice repeatedly until either a 7 or the shooter's point is rolled again. If the shooter rolls a 7 before rolling point, all bets in the PASS LINE are lost. If the shooter rolls point before rolling a 7, the bank/casino pays all bets in the PASS LINE to the players. Compute the probability of winning at craps.

Solution: The probability of winning on a second roll when the come out roll was 4, 5, 6, 8, 9, or 10 is the probability that the first roll was one of those numbers multiplied by the chance the second number matches the first. The probability of winning on a third roll when the come out roll was 4, 5, 6, 8, 9, or 10 is the probability that the first roll was one of those numbers multiplied by the chance the second number did not match the come out roll OR equal a seven, which would mean a loss, multiplied by the chance the third roll matches the come out roll. For fourth and following rolls, the failure to win is raised to one higher a power than the roll before, still multiplying by the probability that the first roll was the come out roll value and the last roll matches the come out roll. The contribution to win probabilities are the results of the sums of the infinite geometric series of win probabilities. Recall from calculus that $\sum_{k=0}^{\infty} r^k = (1 - r)^{-1}$, provided $|r| < 1$.

Sum	Probability &/or Result of Come Out Roll	Winning on Roll Probability GIVEN Come Out Roll				Contribution to Win Probability
		2	3	4	k	
		2	3	4	k	
2	LOSS					0
3	LOSS					0
4	$\frac{3}{36} = \frac{1}{12}$	$(\frac{1}{12})^2$	$\frac{1}{12} \cdot \frac{27}{36} \cdot \frac{1}{12}$	$\frac{1}{12} \cdot (\frac{27}{36})^2 \cdot \frac{1}{12}$	$\frac{1}{12} \cdot (\frac{27}{36})^{k-2} \cdot \frac{1}{12}$	$(\frac{1}{12})^2 \sum_{i=0}^{\infty} (\frac{27}{36})^i = \frac{1}{36}$
5	$\frac{4}{36} = \frac{1}{9}$	$(\frac{1}{9})^2$	$\frac{1}{9} \cdot \frac{26}{36} \cdot \frac{1}{9}$	$\frac{1}{9} \cdot (\frac{26}{36})^2 \cdot \frac{1}{9}$	$\frac{1}{9} \cdot (\frac{26}{36})^{k-2} \cdot \frac{1}{9}$	$(\frac{1}{9})^2 \sum_{i=0}^{\infty} (\frac{26}{36})^i = \frac{2}{45}$
6	$\frac{5}{36}$	$(\frac{5}{36})^2$	$\frac{5}{36} \cdot \frac{25}{36} \cdot \frac{5}{36}$	$\frac{5}{36} \cdot (\frac{25}{36})^2 \cdot \frac{5}{36}$	$\frac{5}{36} \cdot (\frac{25}{36})^{k-2} \cdot \frac{5}{36}$	$(\frac{5}{36})^2 \sum_{i=0}^{\infty} (\frac{25}{36})^i = \frac{25}{396}$
7	$\frac{6}{36} = \frac{1}{6}$ WIN					$\frac{1}{6}$
8	$\frac{5}{36}$	$(\frac{5}{36})^2$	$\frac{5}{36} \cdot \frac{25}{36} \cdot \frac{5}{36}$	$\frac{5}{36} \cdot (\frac{25}{36})^2 \cdot \frac{5}{36}$	$\frac{5}{36} \cdot (\frac{25}{36})^{k-2} \cdot \frac{5}{36}$	$(\frac{5}{36})^2 \sum_{i=0}^{\infty} (\frac{25}{36})^i = \frac{25}{396}$
9	$\frac{4}{36} = \frac{1}{9}$	$(\frac{1}{9})^2$	$\frac{1}{9} \cdot \frac{26}{36} \cdot \frac{1}{9}$	$\frac{1}{9} \cdot (\frac{26}{36})^2 \cdot \frac{1}{9}$	$\frac{1}{9} \cdot (\frac{26}{36})^{k-2} \cdot \frac{1}{9}$	$(\frac{1}{9})^2 \sum_{i=0}^{\infty} (\frac{26}{36})^i = \frac{2}{45}$
10	$\frac{3}{36} = \frac{1}{12}$	$(\frac{1}{12})^2$	$\frac{1}{12} \cdot \frac{27}{36} \cdot \frac{1}{12}$	$\frac{1}{12} \cdot (\frac{27}{36})^2 \cdot \frac{1}{12}$	$\frac{1}{12} \cdot (\frac{27}{36})^{k-2} \cdot \frac{1}{12}$	$(\frac{1}{12})^2 \sum_{i=0}^{\infty} (\frac{27}{36})^i = \frac{1}{36}$
11	$\frac{2}{36} = \frac{1}{18}$ WIN					$\frac{1}{18}$
12	LOSS					0

$$\begin{aligned}\mathbb{P}(\text{Win}|4 \text{ as come out roll}) &= \mathbb{P}(\text{Win}|10 \text{ as come out roll}) \\ &= \left(\frac{1}{12}\right)^2 \left(\frac{1}{1 - \frac{27}{36}}\right) = \left(\frac{3}{36}\right)^2 \left(\frac{36}{9}\right) = \frac{1}{36}.\end{aligned}$$

$$\begin{aligned}\mathbb{P}(\text{Win}|5 \text{ as come out roll}) &= \mathbb{P}(\text{Win}|9 \text{ as come out roll}) \\ &= \left(\frac{1}{9}\right)^2 \left(\frac{1}{1 - \frac{26}{36}}\right) = \left(\frac{4}{36}\right)^2 \left(\frac{36}{10}\right) = \frac{2}{45}.\end{aligned}$$

$$\begin{aligned}\mathbb{P}(\text{Win}|6 \text{ as come out roll}) &= \mathbb{P}(\text{Win}|8 \text{ as come out roll}) \\ &= \left(\frac{5}{36}\right)^2 \left(\frac{1}{1 - \frac{25}{36}}\right) = \left(\frac{5}{36}\right)^2 \left(\frac{36}{11}\right) = \frac{25}{396}.\end{aligned}$$

Thus, the total theoretical probability of winning at craps is

$$\frac{1}{6} + \frac{1}{18} + 2 \cdot \left(\frac{1}{36} + \frac{2}{45} + \frac{25}{396} \right) = \frac{244}{495},$$

which is approximately 0.4929. ■

3.4.2 Continuous Random Variables

Recall that discrete random variables can only assume a countable number of outcomes. When a random variable has a set of possible values that is an entire interval of numbers, X is a **continuous random variable**. For example, if a 12-ounce can of beer is randomly selected and its actual fluid contents X is measured, then X is a continuous random variable because any value for X between 0 and the capacity of the beer can is possible.

Continuous Probability Density Functions' Properties

The function $f(x)$ is a **pdf** for the continuous random variable X , defined over the set of real numbers \mathbb{R} , if

1. $f(x) \geq 0, -\infty < x < \infty,$
 2. $\int_{-\infty}^{\infty} f(x) dx = 1,$ and
 3. $\mathbb{P}(a \leq X \leq b) = \int_a^b f(x) dx.$
- (3.12)

Condition 3 from (3.12) for the definition of a **pdf** for a continuous random variable is illustrated in Figure 3.6.

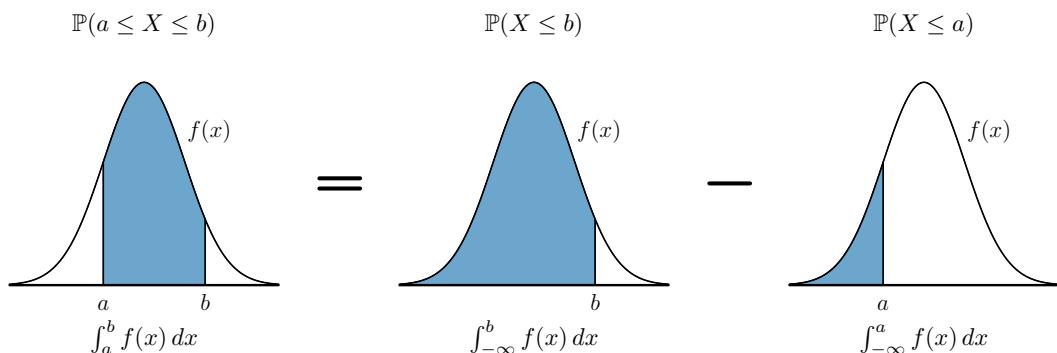


FIGURE 3.6: Illustration of $\mathbb{P}(a \leq X \leq b) = \mathbb{P}(X \leq b) - \mathbb{P}(X \leq a)$

DEFINITION 3.3: Cumulative Distribution Function — The **cdf**, $F(x)$, of a continuous random variable X with **pdf** $f(x)$ is

$$F(x) = \mathbb{P}(X \leq x) = \int_{-\infty}^x f(t) dt, \quad -\infty < x < \infty. \quad (3.13)$$

According to Definition 3.3, the **cdf** is derived from an existing **pdf**. Further, according to the fundamental theorem of calculus, the **pdf** can also be derived from the **cdf** since $F'(x) = f(x)$ for all values of x for which the derivative $F'(x)$ exists.

Continuous Cumulative Distribution Functions' Properties

Continuous **cdfs** have the following properties:

1. $0 \leq F(x) \leq 1$.
 2. If $a < b$, then $F(a) \leq F(b)$ for any real numbers a and b . In other words, $F(x)$ is a non-decreasing function of x .
 3. $\lim_{x \rightarrow \infty} F(x) = 1$.
 4. $\lim_{x \rightarrow -\infty} F(x) = 0$.
- (3.14)

Example 3.21 ▷ *Calculations of pdf and cdf* ◁ Suppose X is a continuous random variable with **pdf** $f(x)$, where

$$f(x) = \begin{cases} k(1-x^2) & \text{if } -1 < x \leq 1, \\ 0 & \text{otherwise.} \end{cases}$$

- (a) Find the constant k so that $f(x)$ is a **pdf** of the random variable X .
- (b) Find the **cdf** for X .
- (c) Compute $\mathbb{P}(-0.5 \leq X \leq 1)$.
- (d) Graph the **pdf** and **cdf** of X using both base graphics and ggplot2.

Solution: The answers are as follows:

- (a) Using property 2 from (3.12) for the **pdf** of a continuous random variable, write

$$\begin{aligned} 1 &= \int_{-\infty}^{\infty} f(x) dx = \int_{-1}^1 k(1-x^2) dx \\ &= k \left[x - \frac{x^3}{3} \right]_{-1}^1 = k \left[\left(1 - \frac{1}{3} \right) - \left(-1 - \frac{-1}{3} \right) \right] \\ &= k \left[\frac{2}{3} - \frac{-2}{3} \right] = k \frac{4}{3} \Rightarrow k = \frac{3}{4}. \end{aligned}$$

(b) Using (3.3) it is left to the reader to verify that

$$F(x) = \begin{cases} 0 & \text{if } x \leq -1, \\ \int_{-1}^x \frac{3}{4}(1-t^2)dt = \frac{-x^3}{4} + \frac{3x}{4} + \frac{1}{2} & \text{if } -1 < x \leq 1, \text{ and} \\ 1 & \text{if } x > 1. \end{cases}$$

(c) Using property 3 from (3.12) for the **pdf** of a continuous random variable, write

$$\begin{aligned} \mathbb{P}(-0.5 \leq X \leq 1) &= F(1) - F(-0.5) \\ &= \left(\frac{-1^3}{4} + \frac{3 \cdot 1}{4} + \frac{1}{2} \right) - \left(-\left(\frac{-1}{2}\right)^3 + \frac{3 \cdot -\frac{1}{2}}{4} + \frac{1}{2} \right) \\ &= \left(\frac{-1}{4} + \frac{3}{4} + \frac{1}{2} \right) - \left(\frac{1}{32} + \frac{-3}{8} + \frac{1}{2} \right) \\ &= 1 - \frac{5}{32} = \frac{27}{32} = 0.84375. \end{aligned}$$

(d) R Code 3.10 can be used to create graphs similar to Figure 3.7 on the facing page, which depicts the **pdf** and **cdf** of X .

R Code 3.10

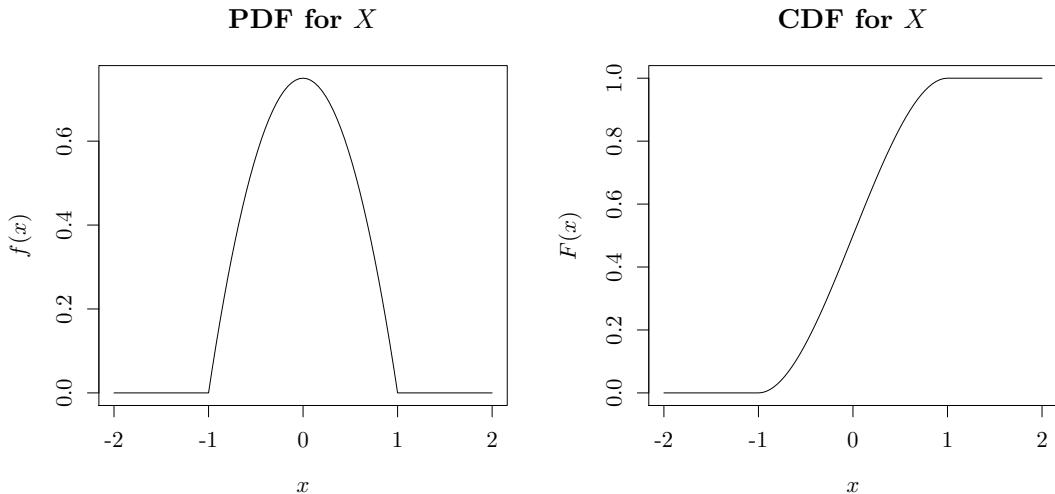
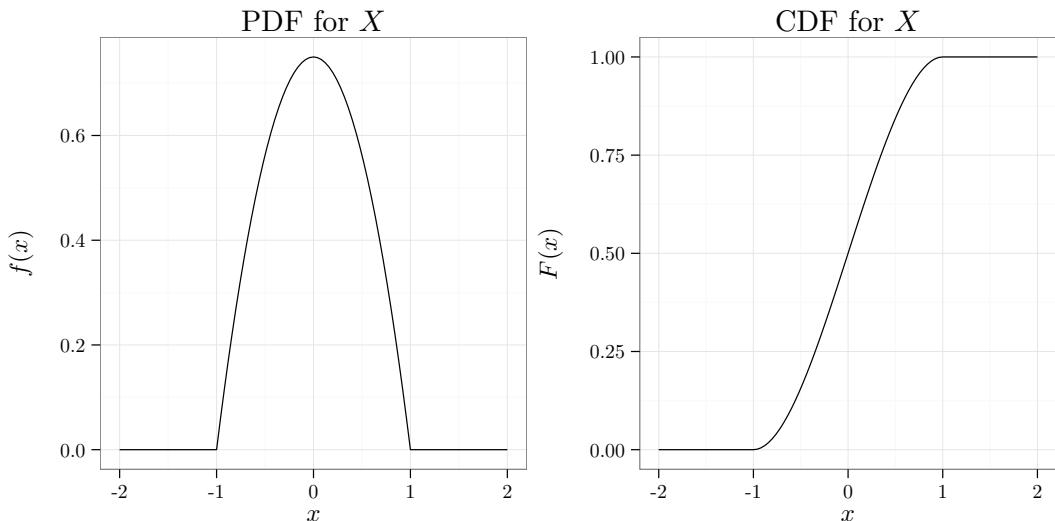
```
> opar <- par(no.readonly = TRUE) # read current parameters
> par(mfrow = c(1, 2)) # split device 1 row 2 columns
> f <- function(x) {
+   y <- 3/4 * (1 - x^2)
+   y[x < -1 | x > 1] <- 0
+   return(y)
+ }
> curve(f, -2, 2, xlab = "x", ylab = "f(x)", main = "PDF for X")
> F <- function(x) {
+   y <- -x^3/4 + 3 * x/4 + 1/2
+   y[x <= -1] <- 0
+   y[x > 1] <- 1
+   return(y)
+ }
> curve(F, -2, 2, xlab = "x", ylab = "F(x)", main = "CDF for X")
> par(opar) # reset graphical parameters
```

To create graphs similar to Figure 3.7 on the next page with **ggplot2** (Figure 3.8 on the facing page), one might use code R Code 3.11 once the functions **f** and **F** from R Code 3.10 are created.

R Code 3.11

```
> p <- ggplot(data.frame(x = c(-2, 2)), aes(x = x))
> p + stat_function(fun=f) + labs(x = "x", y = "f(x)", title = "PDF for X")
> p + stat_function(fun=F) + labs(x = "x", y = "F(x)", title = "CDF for X")
```



FIGURE 3.7: Illustration of **pdf** and **cdf** for Example 3.21FIGURE 3.8: Illustration of **pdf** and **cdf** for Example 3.21 using ggplot2

3.4.2.1 Numerical Integration with R

The R function `integrate()` approximates the integral of functions of one variable over a finite or infinite interval and estimates the absolute error in the approximation. To use `integrate()`, the user must specify `f()`, the function; `lower`, the lower limit of integration; and `upper`, the upper limit of integration. The function `f()` must be a real-valued R function of the form $f(x)$, where x is the variable of integration. In addition to using property 3 from (3.12) for the **pdf** of a continuous random variable to solve (c) of Example 3.21 on page 223, the problem could be solved directly by integrating the original probability $\mathbb{P}(-0.5 \leq X \leq 1)$. That is,

$$\mathbb{P}(-0.5 \leq X \leq 1) = \int_{-0.5}^1 \frac{3}{4} (1 - x^2) dx = \frac{3x}{4} - \frac{x^3}{4} \Big|_{-0.5}^1 = \frac{27}{32} = 0.84375.$$

R Code 3.12 computes $\mathbb{P}(-0.5 \leq X \leq 1)$ using the function `integrate()`.

R Code 3.12

```
> fx <- function(x) {
+   3/4 - 3/4 * x^2
+ } # define function fx
> integrate(fx, lower = -0.5, upper = 1) # gives value and tolerance
0.84375 with absolute error < 9.4e-15

> ans <- integrate(fx, lower = -0.5, upper = 1)$value # just the value
> ans

[1] 0.84375

> library(MASS) # for fractions() function
> fractions(ans) # find closest fraction

[1] 27/32
```

3.4.2.2 Mode, Median, and Percentiles

The **mode** of a continuous probability distribution, just like the mode of a discrete probability distribution, is the x value that maximizes the probability density function. If more than one such x value exists, the distribution is multimodal. The **median** of a continuous distribution is the value m such that

$$\int_{-\infty}^m f(x) dx = \int_m^{\infty} f(x) dx = \frac{1}{2}.$$

The j^{th} **percentile** of a continuous distribution is the value x_j such that

$$\int_{-\infty}^{x_j} f(x) dx = \frac{j}{100}.$$

Example 3.22 Given a random variable X with **pdf**

$$f(x) = \begin{cases} 2e^{-2x} & \text{if } x > 0 \\ 0 & \text{if } x \leq 0, \end{cases}$$

- (a) Find the median of the distribution.
- (b) Find the 25th percentile of the distribution.
- (c) Find the 60th percentile of the distribution.

Solution: The answers are as follows:

- (a) The median is the value m such that $\int_0^m 2e^{-2x} dx = 0.5$, which implies

$$\begin{aligned}-e^{-2x}\Big|_0^m &= 0.5 \\ -e^{-2m} + 1 &= 0.5 \\ -e^{-2m} &= 0.5 - 1 \\ \ln(e^{-2m}) &= \ln(0.5) \\ m &= \frac{\ln(0.5)}{-2} = 0.3466.\end{aligned}$$

- (b) The 25th percentile is the value x_{25} such that $\int_0^{x_{25}} 2e^{-2x} dx = 0.25$, which implies

$$\begin{aligned}-e^{-2x}\Big|_0^{x_{25}} &= 0.25 \\ -e^{-2x_{25}} + 1 &= 0.25 \\ -e^{-2x_{25}} &= 0.25 - 1 \\ \ln(e^{-2x_{25}}) &= \ln(0.75) \\ x_{25} &= \frac{\ln(0.75)}{-2} = 0.1438.\end{aligned}$$

- (c) The 60th percentile is the value x_{60} such that $\int_0^{x_{60}} 2e^{-2x} dx = 0.60$, which implies

$$\begin{aligned}-e^{-2x}\Big|_0^{x_{60}} &= 0.60 \\ -e^{-2x_{60}} + 1 &= 0.60 \\ -e^{-2x_{60}} &= 0.60 - 1 \\ \ln(e^{-2x_{60}}) &= \ln(0.40) \\ x_{60} &= \frac{\ln(0.40)}{-2} = 0.4581.\end{aligned}$$



Example 3.23 Given a random variable X with **pdf**

$$f(x) = \begin{cases} 2 \cos(2x) & \text{if } 0 < x < \pi/4 \\ 0 & \text{otherwise,} \end{cases}$$

- (a) Find the mode of the distribution.
- (b) Find the median of the distribution.
- (c) Draw the **pdf** and add a dashed vertical line at the median of the distribution.

Solution: The answers are as follows:

- (a) The function $2 \cos 2x$ does not have a maximum in the open interval $(0, \pi/4)$ since the derivative $f'(x) = -4 \sin 2x$ does not equal 0 in the open interval $(0, \pi/4)$.

(b) The median is the value m such that

$$\int_0^m 2 \cos 2x \, dx = 0.5$$

$$\Downarrow$$

$$\sin 2x \Big|_0^m = \sin 2m = 0.5$$

$$2m = \arcsin(0.5)$$

$$m = \frac{\pi}{12}.$$

(c) R Code 3.13 can be used to create a graph similar to Figure 3.9. After the function f is defined, the next two lines of R Code 3.13 produce the requested graph with base graphics. The last three lines of R Code 3.13 produce the requested graph with ggplot2.

R Code 3.13

```
> f <- function(x){2*cos(2*x)}
> curve(f, 0, pi/4, , xlab = "x", ylab = "2cos(2x)")
> abline(v = pi/12, lty = 2, lwd = 2)
> # ggplot2 now
> p <- ggplot(data.frame(x = c(0, pi/4)), aes(x = x))
> p + stat_function(fun = f) + labs(x = "x", y = "2cos(2x)") +
+   geom_vline(xintercept = pi/12, lty = "dashed")
```

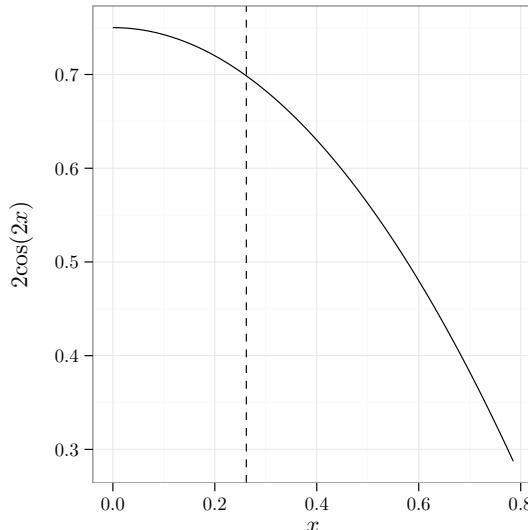


FIGURE 3.9: Graph of $2 \cos(2x)$ from 0 to $\frac{\pi}{4}$ with R



3.4.2.3 Expected Values

For continuous random variables, the definitions associated with the expectation of a random variable X or a function, say $g(X)$, of X are identical to those for discrete random variables, except the summations are replaced with integrals and the probability density functions are represented with $f(x)$ instead of $p(x)$. The **expected value** of a continuous random variable X is

$$E[X] = \mu_X = \int_{-\infty}^{\infty} x \cdot f(x) dx. \quad (3.15)$$

When the integral in (3.15) does not exist, neither does the expectation of the random variable X . The expected value of a function of X , say $g(X)$, is

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) \cdot f(x) dx. \quad (3.16)$$

Using the definitions for moments about 0 and μ given in (3.8), which relied strictly on expectation in conjunction with (3.16), the **variance** of a continuous random variable X is written as

$$Var[X] = \sigma_X^2 = E[(X - \mu)^2] = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx. \quad (3.17)$$

Example 3.24 Given the function

$$f(x) = k, \quad -1 < x < 1$$

of the random variable X ,

- (a) Find the value of k to make $f(x)$ a **pdf**. Use this k for parts (c) and (d).
- (b) Graph the **pdf** with `ggplot2`.
- (c) Find the mean of the distribution using (3.15).
- (d) Find the variance of the distribution using (3.17).

Solution: The answers are as follows:

- (a) Since $\int_{-\infty}^{\infty} f(x) dx$ must equal 1 for $f(x)$ to be a **pdf**, set $\int_{-1}^1 k dx$ equal to one and solve for k :

$$\begin{aligned} \int_{-1}^1 k dx &= 1 \\ kx \Big|_{-1}^1 &= 1 \\ 2k = 1 \Rightarrow k &= \frac{1}{2}. \end{aligned}$$

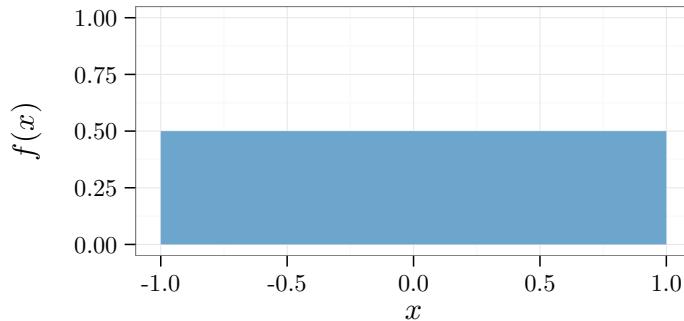
- (b) R Code 3.14 on the next page is used to create Figure 3.10 on the following page.

R Code 3.14

```

> x <- seq(-1, 1, length = 500)
> y <- dunif(x, -1, 1)
> DF <- data.frame(fx = y)
> previous_theme <- theme_set(theme_bw()) # set black and white theme
> ggplot(data = DF, aes(x = x, y = fx)) +
+   geom_area(fill = "skyblue3") +
+   labs(x = "x", y = "f(x)\n") +
+   ylim(c(0, 1)) +
+   theme_set(previous_theme)           # Restore original theme

```

FIGURE 3.10: Graph of $X \sim Unif(-1, 1)$

(c) The mean of the distribution using (3.15) is

$$\begin{aligned} E[X] = \mu_X &= \int_{-1}^1 \frac{1}{2}x \, dx \\ &= \left. \frac{x^2}{4} \right|_{-1}^1 = 0 \end{aligned}$$

(d) The variance of the distribution using (3.17) is

$$\begin{aligned} Var[X] = \sigma_X^2 &= E[(X - \mu)^2] = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) \, dx \\ &= \int_{-1}^1 (x - 0)^2 \frac{1}{2} \, dx \\ &= \left. \frac{x^3}{6} \right|_{-1}^1 = \frac{1}{3} \end{aligned}$$



3.4.3 Markov's Theorem and Chebyshev's Inequality

Theorem 3.1 ▷ Markov's Theorem ◁ If X is a random variable and $g(X)$ is a function of X such that $g(X) \geq 0$, then, for any positive K ,

$$\mathbb{P}(g(X) \geq K) \leq \frac{E[g(X)]}{K}. \quad (3.18)$$

Proof:

Step 1. Let $I(g(X))$ be a function such that

$$I(g(X)) = \begin{cases} 1 & \text{if } g(X) \geq K, \\ 0 & \text{otherwise.} \end{cases}$$

Step 2. Since $g(X) \geq 0$ and $I(g(X)) \leq 1$, when the first condition of $I(g(X))$ is divided by K ,

$$I(g(X)) \leq \frac{g(X)}{K}.$$

Step 3. Taking the expected value,

$$E[I(g(X))] \leq \frac{E[g(X)]}{K}.$$

Step 4. Clearly

$$\begin{aligned} E[I(g(X))] &= \sum_x I(g(x)) \cdot p(x) \\ &= [1 \cdot \mathbb{P}(I(g(X)) = 1)] + [0 \cdot \mathbb{P}(I(g(X)) = 0)] \\ &= [1 \cdot \mathbb{P}(g(X) \geq K)] + [0 \cdot \mathbb{P}(g(X) < K)] \\ &= \mathbb{P}(g(X) \geq K). \end{aligned}$$

Step 5. Rewriting,

$$\mathbb{P}(g(X) \geq K) \leq \frac{E[g(X)]}{K},$$

which is the inequality from (3.18) to be proven.

If $g(X) = (X - \mu)^2$ and $K = k^2\sigma^2$ in (3.18), it follows that

$$\mathbb{P}((X - \mu)^2 \geq k^2\sigma^2) \leq \frac{E[(X - \mu)^2]}{k^2\sigma^2} = \frac{\sigma^2}{k^2\sigma^2} = \frac{1}{k^2}. \quad (3.19)$$

Working inside the probability on the left side of the inequality in (3.19), note that

$$\begin{aligned} ((X - \mu)^2 \geq k^2\sigma^2) &\Rightarrow (X - \mu \geq \sqrt{k^2\sigma^2}) \text{ or } (X - \mu \leq -\sqrt{k^2\sigma^2}) \\ &\Rightarrow (|X - \mu| \geq \sqrt{k^2\sigma^2}) \\ &\Rightarrow (|X - \mu| \geq k\sigma). \end{aligned}$$

Using this, rewrite (3.19) to obtain

$$\mathbb{P}(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}, \quad (3.20)$$

which is known as **Chebyshev's Inequality**.

DEFINITION 3.4: Chebyshev's Inequality — Can be stated as any of

- (a) $\mathbb{P}(|X - \mu| \geq k) \leq \frac{\sigma^2}{k^2}$.
- (b) $\mathbb{P}(|X - \mu| < k) \geq 1 - \frac{\sigma^2}{k^2}$.
- (c) $\mathbb{P}(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$.
- (d) $\mathbb{P}(|X - \mu| < k\sigma) \geq 1 - \frac{1}{k^2}$.

Version (d) of Chebyshev's Inequality is the complement of (c), the version derived in (3.20). Version (b) is the complement of (a), both of which can be obtained by setting $g(X) = (X - \mu)^2$ and $K = k^2$ in (3.18). A verbal interpretation of version (c) is that the probability any random variable X with finite variance, irrespective of the distribution of X , is k or more standard deviations from its mean is less than or equal to $1/k^2$. Likewise, version (d) states that the probability X is within k standard deviations from the mean is at least $1 - \frac{1}{k^2}$. Clearly, Chebyshev's Inequality can be used as a bound for certain probabilities; however, in many instances, the bounds provided by the inequality are very conservative. One reason for this is that there are no restrictions on the underlying distribution.

Example 3.25 Consider Example 3.17 on page 215, where X was defined to be the number of heads in three tosses of a fair coin. Chebyshev's Inequality guarantees at least what fraction of the distribution of X is within $k = 2$ standard deviations from its mean? What is the actual fraction of the distribution of X that is within $k = 2$ standard deviations from its mean?

Solution: Using version (d) of Chebyshev's Inequality, $\mathbb{P}(|X - \mu| < k\sigma) \geq 1 - \frac{1}{k^2}$, compute the first answer to be $1 - \frac{1}{2^2} = \frac{3}{4}$. To answer the second question, first find the mean and variance of X :

$$E[X] = \sum_x x p(x) = 0 \times \frac{1}{8} + 1 \times \frac{3}{8} + 2 \times \frac{3}{8} + 3 \times \frac{1}{8} = \frac{3}{2} = 1.5,$$

$$E[X^2] = \sum_x x^2 p(x) = 0^2 \times \frac{1}{8} + 1^2 \times \frac{3}{8} + 2^2 \times \frac{3}{8} + 3^2 \times \frac{1}{8} = \frac{24}{8} = 3, \text{ and}$$

$$Var[X] = E[X^2] - (E[X])^2 = 3 - 1.5^2 = 0.75.$$

For this example,

$$\begin{aligned} \mathbb{P}(|X - \mu| < k\sigma) &= \mathbb{P}(|X - 1.5| < 2\sqrt{0.75}) \\ &= \mathbb{P}(|X - 1.5| < 1.732) \\ &= \mathbb{P}(-0.232 < X < 3.232) = 1. \end{aligned}$$

Chebyshev's Inequality guaranteed at least 75% of the distribution of X would be within $k = 2$ standard deviations from its mean. The fact that all of the distribution of X is within

$k = 2$ standard deviations from the mean illustrates the conservative nature of Chebyshev's Inequality. R Code 3.15 computes the mean and variance of X as well as the interval $|X - \mu| < 2\sigma$.

R Code 3.15

```
> x <- 0:3
> px <- c(1/8, 3/8, 3/8, 1/8)
> EX <- weighted.mean(x, px)
> EX2 <- weighted.mean(x^2, px)
> VX <- EX2 - EX^2
> sigmaX <- sqrt(VX)
> MUSIG <- c(EX, VX)
> names(MUSIG) <- c("E(X)", "V(X)")
> MUSIG

E(X) V(X)
1.50 0.75

> Int <- 2 * sigmaX * c(-1, 1) + 1.5
> Int

[1] -0.2320508 3.2320508
```



3.4.4 Weak Law of Large Numbers

An important application of Chebyshev's Inequality is proving the **Weak Law of Large Numbers**. The Weak Law of Large Numbers provides proof of the notion that if n independent and identically distributed random variables, X_1, X_2, \dots, X_n , from a distribution with finite variance are observed, then the sample mean, \bar{X} , should be very close to μ provided n is large. Mathematically, the Weak Law of Large Numbers states that if n independent and identically distributed random variables, X_1, X_2, \dots, X_n are observed from a distribution with finite variance, then, for all $\epsilon > 0$,

$$\lim_{n \rightarrow \infty} \mathbb{P}\left(\left|\frac{X_1 + \dots + X_n}{n} - \mu\right| \geq \epsilon\right) = 0. \quad (3.21)$$

Proof: Consider the random variables X_1, \dots, X_n such that the mean of each one is μ and the variance of each one is σ^2 . Since

$$E\left[\frac{\sum_{i=1}^n X_i}{n}\right] = \mu \quad \text{and} \quad \text{Var}\left[\frac{\sum_{i=1}^n X_i}{n}\right] = \frac{\sigma^2}{n},$$

use version (a) of Chebyshev's Inequality with $k = \epsilon$ to write

$$\mathbb{P}\left(\left|\frac{X_1 + \dots + X_n}{n} - \mu\right| \geq \epsilon\right) \leq \frac{\sigma^2}{n\epsilon^2},$$

which proves (3.21) since

$$\lim_{n \rightarrow \infty} \mathbb{P}\left(\left|\frac{X_1 + \dots + X_n}{n} - \mu\right| \geq \epsilon\right) \leq \lim_{n \rightarrow \infty} \frac{\sigma^2}{n\epsilon^2} = 0.$$

3.4.5 Skewness

Earlier it was discussed that the second moment about the mean of a random variable X is the same thing as the variance of X . Now, the third moment about the mean of a random variable X is used in the definition of the skewness of X . To facilitate the notation used with skewness, first define a **standardized** random variable X^* to be:

$$X^* = \frac{X - \mu}{\sigma},$$

where μ is the mean of X and σ is the standard deviation of X . Using the standardized form of X , it is easily shown that $E[X^*] = 0$ and $Var[X^*] = 1$. Define the skewness of a random variable X , denoted γ_1 , to be the third moment about the origin of X^* :

$$\gamma_1 = E[(X^*)^3] = \frac{E[(X - \mu)^3]}{\sigma^3}. \quad (3.22)$$

Positive values for γ_1 indicate a distribution that is skewed to the right while negative values for γ_1 indicate a distribution that is skewed to the left. If the distribution of X is symmetric with respect to its mean, then its skewness is zero. That is, $\gamma_1 = 0$ for distributions that are symmetric about their mean. Examples of distributions with various γ_1 coefficients are shown in Figure 3.11.

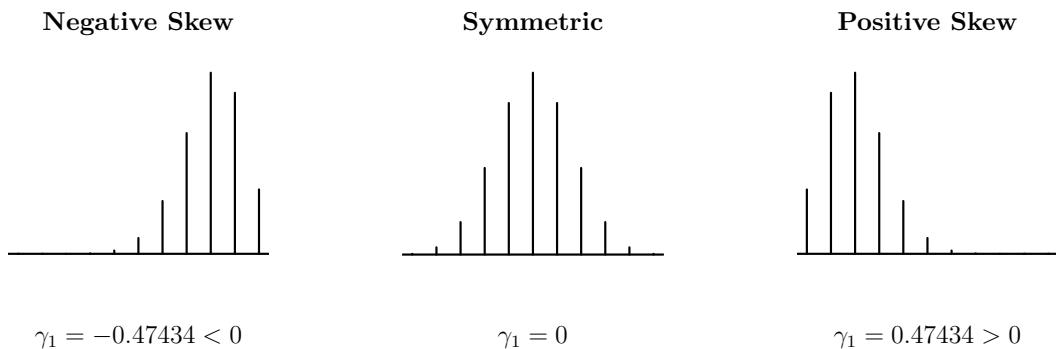


FIGURE 3.11: Distributions with γ_1 (skewness) coefficients that are negative, zero, and positive, respectively

Example 3.26 Let the **pdf** of X be defined by $p(x) = x/15, x = 1, 2, 3, 4, 5$. Compute γ_1 for the given distribution.

Solution: The value of γ_1 is computed to be

$$\gamma_1 = E[(X^*)^3] = \frac{E[(X - \mu)^3]}{\sigma^3} = -0.588$$

which means the distribution has a negative skew. R Code 3.16 on the facing page uses the following facts to compute the answer:

1. $\mu = E[X]$.
2. $\sigma = \sqrt{E[X^2] - E[X]^2}$.

$$3. X^* = \frac{X - \mu}{\sigma}.$$

$$4. \gamma_1 = E[(X^*)^3].$$

R Code 3.16

```
> x <- 1:5
> px <- x/15
> plot(x, px, xlab = "x", ylab = "P(X=x)", type = "h")
> EX <- sum(x * px)
> sigmaX <- sqrt(sum(x^2 * px) - EX^2)
> X.star <- (x - EX)/sigmaX
> skew <- sum(X.star^3 * px)
> skew

[1] -0.5879747
```

This random variable is skewed to the left as seen in Figure 3.12. ■

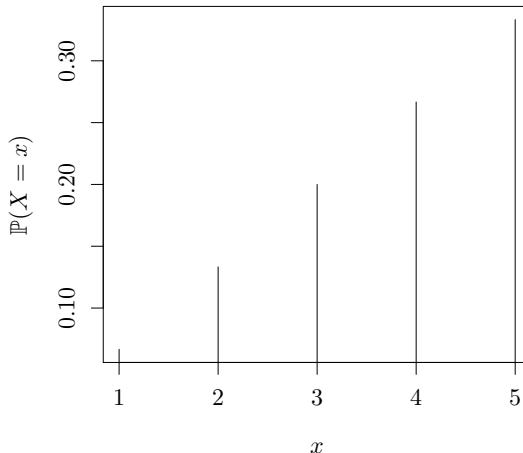


FIGURE 3.12: Graph of the **pdf** for Example 3.26

3.5 Moment Generating Functions

Finding the first, second, and higher moments about the origin using the definition $\alpha_r = E[X^r]$ is not always an easy task, but one may define a function of a real variable t called the moment generating function, **mgf**, that can be used to find moments with relative ease provided the **mgf** exists. Given a random variable X with **pdf** $f(x)$ (continuous) or

$p(x)$ (discrete), the **mgf** of X , written $M_X(t)$, is defined as

$$M_X(t) = E[e^{tX}] = \int_{-\infty}^{\infty} e^{tx} f(x) dx, \quad -h < t < h, \quad (3.23)$$

provided there is a positive number h such that, for $-h < t < h$, the expectation of e^{tX} exists. If X is discrete, then $E[e^{tX}] = \sum_x e^{tx} p(x)$. When the **mgf** exists, it is unique and completely determines the distribution of the random variable. Consequently, if two random variables have the same **mgf**, they have the same distribution.

Example 3.27 Given the function

$$f(x) = k, \quad -1 < x < 1$$

of the random variable X , find the **mgf** of the distribution using (3.23).

Solution: The reader may verify that a value of $k = \frac{1}{2}$ produces a valid **pdf**. The **mgf** of the distribution will then be

$$\begin{aligned} M_X(t) &= E[e^{tX}] = \int_{-\infty}^{\infty} e^{tx} f(x) dx, \quad -h < t < h \\ &= \int_{-1}^1 e^{tx} \frac{1}{2} dx = \frac{e^{tx}}{2t} \Big|_{-1}^1 \\ &= \frac{e^t - e^{-t}}{2t}, \quad t \neq 0. \end{aligned}$$

Note that if $t = 0$, then $M_X(t) = 1$ since $M_X(t) = E[e^{tX}] = E[e^0] = 1$. Therefore, the **mgf** is written

$$M_X(t) = \begin{cases} \frac{e^t - e^{-t}}{2t} & \text{if } t \neq 0 \text{ and} \\ 1 & \text{if } t = 0. \end{cases}$$

Theorem 3.2 If X has **mgf** $M_X(t)$, then the derivatives of $M_X(t)$ of all orders exist at $t = 0$, and

$$E[X^r] = \frac{d^r}{dt^r} M_X(t)|_{t=0}.$$

A proof of the last theorem is beyond the scope of this text; however, assuming the distribution is discrete and summation and differentiation may be interchanged, note that the r^{th} moment about the origin, $\alpha_r = E[X^r]$, is equal to the r^{th} derivative of the moment

generating function evaluated at $t = 0$.

$$\begin{aligned} E[X^1] &= \frac{d^1}{dt^1} M_X(t)|_{t=0} = \frac{d^1}{dt^1} \sum_x e^{tx} p(x)|_{t=0} = \sum_x \frac{d^1}{dt^1} e^{tx} p(x)|_{t=0} \\ &= \sum_x x e^{tx} p(x)|_{t=0} = \sum_x x p(x) = \alpha_1 = E[X^1], \\ E[X^2] &= \frac{d^2}{dt^2} M_X(t)|_{t=0} = \frac{d^2}{dt^2} \sum_x e^{tx} p(x)|_{t=0} = \sum_x \frac{d^2}{dt^2} e^{tx} p(x)|_{t=0} \\ &= \sum_x x^2 e^{tx} p(x)|_{t=0} = \sum_x x^2 p(x) = \alpha_2 = E[X^2], \text{ and} \\ E[X^r] &= \frac{d^r}{dt^r} M_X(t)|_{t=0} = \frac{d^r}{dt^r} \sum_x e^{tx} p(x)|_{t=0} = \sum_x \frac{d^r}{dt^r} e^{tx} p(x)|_{t=0} \\ &= \sum_x x^r e^{tx} p(x)|_{t=0} = \sum_x x^r p(x) = \alpha_r = E[X^r]. \end{aligned}$$

Example 3.28 Let X be a random variable with probability distribution

$$P(X = x|n, \pi) = \frac{n!}{(n-x)!x!} \pi^x (1-\pi)^{(n-x)} \quad x = 0, 1, \dots, n.$$

Using the moment generating function, check that $E[X] = n\pi$ and $\text{Var}[X] = n\pi(1 - \pi)$.
(Hint: $(a+b)^n = \sum_{x=0}^n \binom{n}{x} b^x a^{n-x}$.)

Solution: First, the moment generating function is calculated:

$$\begin{aligned} M(t) &= E[e^{tx}] = \sum_{x=0}^n \binom{n}{x} e^{tx} \pi^x (1-\pi)^{n-x} \\ &= \sum_{x=0}^n \binom{n}{x} (\pi e^t)^x (1-\pi)^{n-x} \\ &= [(1-\pi) + \pi e^t]^n. \end{aligned}$$

The first and second derivatives of $M(t)$ at $t = 0$ give $E[X]$ and $E[X^2]$, respectively, which are used to calculate the mean and variance of X :

$$M'(t) = n[(1-\pi) + \pi e^t]^{n-1}(\pi e^t)$$

and, using the product and chain rules

$$M''(t) = n(n-1)[(1-\pi) + \pi e^t]^{n-2}(\pi e^t)^2 + n[(1-\pi) + \pi e^t]^{n-1}(\pi e^t).$$

This yields

$$E[X] = M'(0) = n\pi \quad \text{and}$$

$$\text{Var}[X] = E[X^2] - E[X]^2 = M''(0) - [M'(0)]^2 = n(n-1)\pi^2 + n\pi - (n\pi)^2 = n\pi(1-\pi). \quad \blacksquare$$

Theorem 3.3 If a and b are real-valued constants, then

$$(1) \quad M_{X+a}(t) = E[e^{(X+a)t}] = e^{at} \cdot M_X(t).$$

$$(2) \quad M_{bX}(t) = E[e^{bXt}] = M_X(bt).$$

$$(3) \quad M_{\frac{X+a}{b}}(t) = E[e^{(\frac{X+a}{b})t}] = e^{\frac{a}{b}t} \cdot M_X\left(\frac{t}{b}\right).$$

The proof of Theorem 3.3 is left as an exercise for the reader.

3.6 Problems

1. How many ways can a host randomly choose 8 people out of 90 in the audience to participate in a TV game show?
2. How many different six-place license plates are possible if the first two places are letters and the remaining places are numbers?
3. How many different six-place license plates are possible (first two places letters, remaining places numbers) if repetition among letters and numbers is not permissible?
4. Susie has 25 books she would like to arrange on her desk. Of the 25 books, 7 are statistics books, 6 are biology books, 5 are English books, 4 are history books, and 3 are psychology books. If Susie arranges her books by subject, how many ways can she arrange her books?
5. A hat contains 20 consecutive numbers (1 to 20). If four numbers are drawn at random, how many ways are there for the largest number to be a 16 and the smallest number to be a 5?
6. A university committee of size 10, consisting of 2 faculty from the college of fine and applied arts, 2 faculty from the college of business, 3 faculty from the college of arts and sciences, and 3 administrators, is to be selected from 6 fine and applied arts faculty, 7 college of business faculty, 10 college of arts and sciences faculty, and 5 administrators. How many committees are possible?
7. How many different letter arrangements can be made from the letters BIOLOGY, PROBABILITY, and STATISTICS, respectively?
8. A doll house must be painted and assembled before it can be given as a gift. If there are 12 equal-sized rooms in the doll house and there is enough white paint for 4 rooms, enough pink paint for 3 rooms, and enough blue paint for 5 rooms, in how many ways can the 12 rooms be painted?
9. A shipment of 50 laptops includes 3 that are defective. If an instructor purchases 4 laptops from the shipment to use in his class, how many ways are there for the instructor to purchase at least 2 of the defective laptops?
10. A multiple-choice test consists of 10 questions. Each question has 5 answers (only one is correct). How many different ways can a student fill out the test?
11. How many ways can five politicians stand in line? In how many ways can they stand in line if two of the politicians refuse to stand next to each other?
12. There are five different colored jerseys worn throughout the Tour de France. The yellow jersey is worn by the rider with the least accumulated time; the green jersey is worn by the best sprinter; the red and white polka dot jersey is worn by the best climber. The white jersey is worn by the best youngest rider, and the red jersey is worn by the rider with the most accumulated time still in the race. If 150 riders finish the Tour, how many different ways can the yellow, green, and red and white polka dot jerseys be awarded if (a) a rider can receive any number of jerseys and (b) each rider can receive at most one jersey?

13. A president, treasurer, and secretary, all different, are to be chosen from among the 10 active members of a university club. How many different choices are possible if
- There are no restrictions.
 - A will serve only if she is the treasurer.
 - B and C will not serve together.
 - D and E will serve together or not at all.
 - F must be an officer.
14. On a multiple-choice exam with three possible answers for each of the five questions, what is the probability that a student would get four or more correct answers just by guessing?
15. Suppose four balls are chosen at random without replacement from an urn containing six black balls and four red balls. What is the probability of selecting two balls of each color?
16. What is the probability that a hand of five cards chosen randomly and without replacement from a standard deck of 52 cards contains the ace of hearts, exactly one other ace, and exactly two kings?
17. In the New York State lottery game, six of the numbers 1 through 54 are chosen by a customer. Then, in a televised drawing, six of these numbers are selected. If all six of a customer's numbers are selected, then that customer wins a share of the first prize. If five or four of the numbers are selected, the customer wins a share of the second or the third prize. What is the probability that any customer will win a share of the first prize, the second prize, and the third prize, respectively?
18. An office supply store is selling packages of 100 CDs at a very affordable price. However, roughly 10% of all packages are defective. If a package of 100 CDs containing exactly 10 defective CDs is purchased, find the probability that exactly 2 of the first 5 CDs used are defective.
19. A box contains six marbles, two of which are black. Three are drawn with replacement. What is the probability two of the three are black?
20. The ASU triathlon club consists of 11 women and 7 men. What is the probability of selecting a committee of size four with exactly three women?
21. Four golf balls are to be placed in six different containers. One ball is red; one, green; one, blue; and one, yellow.
- In how many ways can the four golf balls be placed into six different containers? Assume that any container can contain any number of golf balls (as long as there are a total of four golf balls).
 - In how many ways can the golf balls be placed if container one remains empty?
 - In how many ways can the golf balls be placed if no two golf balls can go into the same container?

(d) What is the probability that no two golf balls are in the same container, assuming that the balls are randomly tossed into the containers?

22. Three dice are thrown. What fraction of the time does a sum of 9 appear on the faces? What percent of the time does a sum of 10 appear?

23. Assume that $\mathbb{P}(A) = 0.5$, $\mathbb{P}(A \cap C) = 0.2$, $\mathbb{P}(C) = 0.4$, $\mathbb{P}(B) = 0.4$, $\mathbb{P}(A \cap B \cap C) = 0.1$, $\mathbb{P}(B \cap C) = 0.2$, and $\mathbb{P}(A \cap B) = 0.2$. Calculate the following probabilities:

- (a) $\mathbb{P}(A \cup B \cup C)$
- (b) $\mathbb{P}(A^c \cap (B \cup C))$
- (c) $\mathbb{P}((B \cap C)^c \cup (A \cap B)^c)$
- (d) $\mathbb{P}(A) - \mathbb{P}(A \cap C)$

24. In a 10k race where three runners, Susie, Mike, and Anna, enter the race with identical personal best times, assume they all have an equal chance of winning today's 10k. Consider the events:

- E_1 : Susie wins the 10k.
- E_2 : Susie places second in the 10k.
- E_3 : Susie places third in the 10k.
- W : Susie places higher than Mike.

Is W independent of E_1 , E_2 , and E_3 ?

25. Verify that $\mathbb{P}(F|E)$ satisfies the three axioms of probability.

26. If A and B are independent events, show that A^c and B^c are also independent events.

27. Let A and B be events where $0 < \mathbb{P}(A) < 1$ and $0 < \mathbb{P}(B) < 1$. Is $\mathbb{P}(A|B) + \mathbb{P}(A^c|B^c) = 1$ true when A and B are

- (a) mutually exclusive?
- (b) independent?

If $\mathbb{P}(A|B) + \mathbb{P}(A^c|B^c) = 1$ is not true for either (a) or (b), provide a counterexample.

28. A family has three cars, all with electric windows. Car A's windows always work. Car B's windows work 30% of the time, and Car C's windows work 75% of the time. The family uses Car A $\frac{2}{3}$ of the time; Car B, $\frac{2}{9}$ of the time; and Car C, the remaining fraction.

- (a) On a particularly hot day, when the family wants to roll the windows down, compute the probability the windows will work.
- (b) If the electric windows work, find the probability the family is driving Car C.

29. A new drug test being considered by the International Olympic Committee can detect the presence of a banned substance when it has been taken by the subject in the last 90 days 98% of the time. However, the test also registers a “false positive” in 2% of the population that has never taken the banned substance. If 2% of the athletes in question are taking the banned substance, what is the probability a person that has a positive drug test is actually taking the banned substance?

30. The products of an agricultural firm are delivered by four different transportation companies, A, B, C, and D. Company A transports 40% of the products; company B, 30%; company C, 20%; and, finally, company D, 10%. During transportation, 5%, 4%, 2%, and 1% of the products spoil with companies A, B, C, and D, respectively. If one product is randomly selected,

- (a) Obtain the probability that it is spoiled.
- (b) If the chosen product is spoiled, derive the probability that it has been transported by company A.

31. Two lots of large glass beads are available (A and B). Lot A has four beads, two of which are chipped; and lot B has five beads, two of which are chipped. Two beads are chosen at random from lot A and passed to lot B. Then, one bead is randomly selected from lot B. Find:

- (a) The probability that the selected bead is chipped.
- (b) The probability that the two beads selected from lot A were not chipped if the bead selected from lot B is not chipped.

32. A box contains 5 defective bulbs, 10 partially defective (they start to fail after 10 hours of use), and 25 perfect bulbs. If a bulb is tested and it does not fail immediately, find the probability that the bulb is perfect.

33. A salesman in a department store receives household appliances from three suppliers: I, II, and III. From previous experience, the salesman knows that 2%, 1%, and 3% of the appliances from suppliers I, II, and III, respectively, are defective. The salesman sells 35% of the appliances from supplier I, 25% from supplier II, and 40% from supplier III. If an appliance randomly selected is defective, find the probability that it comes from supplier III.

34. Last year, a new business purchased 25 tablets, 25 laptops, and 50 desktops, all with three year warranties. The probability a tablet has had warranty work is four times the probability a desktop has had warranty work. The probability a laptop has had warranty work is twice the probability a desktop has had warranty work. Given that 10 computers have used the warranty,

- (a) If a computer is a laptop, what is the probability it has had warranty work?
- (b) If a computer has had warranty work, what is the probability it was a laptop?
- (c) If a computer has had warranty work, what is the probability it was a tablet?

35. An urn contains 14 balls; 6 of them are white, and the others are black. Another urn contains 9 balls; 3 are white, and 6 are black. A ball is drawn at random from the first urn and is placed in the second urn. Then, a ball is drawn at random from the second urn. If this ball is white, find the probability that the ball drawn from the first urn was black.

36. Previous to the launching of a new flavor of yogurt, a company has conducted taste tests with four new flavors: lemon, strawberry, peach, and cherry. It obtained the following probabilities of a successful launch: $\mathbb{P}(\text{lemon}) = \frac{2}{10}$, $\mathbb{P}(\text{strawberry}) = \frac{3}{10}$, $\mathbb{P}(\text{peach}) = \frac{4}{10}$, and $\mathbb{P}(\text{cherry}) = \frac{5}{10}$. Let X be the random variable “number of successful flavors launched.” Obtain its probability mass function.

37. John and Peter play a game with a coin such that $\mathbb{P}(\text{head}) = p$. The game consists of tossing a coin twice. John wins if the same result is obtained in the two tosses, and Peter wins if the two results are different.

(a) At what value of p is neither of them favored by the game?

(b) If p is different from your answer in (a), who is favored?

38. A bank is going to place a security camera in the ceiling of a circular hall of radius r . What is the probability that the camera is placed nearer the center than the outside circumference if the camera is placed at random?

39. Let the random variable X be the sum of the numbers on two fair dice. Find an upper bound on $\mathbb{P}(|X - 7| \geq 4)$ using Chebyshev's Inequality as well as the exact probability for $\mathbb{P}(|X - 7| \geq 4)$.

40. Two colleagues, Terry and Kenneth, have flights that will debark between 11:00 a.m. and 11:30 a.m. at the same terminal. They are aware of each others' arrival times and would like to meet in the terminal.

(a) Find the probability Terry and Kenneth will be able to meet in the terminal if neither colleague can afford to wait more than 10 minutes for the other colleague.

(b) Plot the probability Terry and Kenneth will meet within a time t versus the time one has to spend waiting for the other to arrive if whoever arrives first will wait up to 30 minutes.

41. Two independently wealthy philatelists, Alvin and Bob, are interested in buying rare stamps at a private auction. For each stamp up for auction, given that the previous bid did not win, Alvin or Bob wins on their i^{th} bid with probability p . Assume that Alvin always makes the first bid.

(a) Find the probability that Alvin wins the first auction.

(b) If two stamps are actually auctioned, find the probability that they are purchased by the same bidder.

(Hint: $\sum_{i=0}^{\infty} r^i = \frac{1}{1-r}$ if $|r| < 1$.)

42. Anthony and Mark make a bet at the beginning of the school year. If Anthony passes one exam, Mark will pay him €10, but if Anthony fails the exam, he will give €10 to Mark. If Anthony takes 10 exams and the probability of passing an exam is 0.5, find the probability that

- (a) Anthony wins €60.
 (b) Anthony wins €30.

43. Louis and Joseph have decided to play a beach volleyball match. Each of them put €50 into a pot, so the winner will get €100. The first one to reach 21 points wins. When the score was 19 points for Louis and 18 for Joseph, the match was rained out, and they decided to share the prize so that each one received winnings proportional to the probability of winning the match given their current points. How much money did each receive?

44. Consider tossing three fair coins. The eight possible outcomes are

$$HHH, HHT, HTH, HTT, THH, THT, TTH, TTT.$$

Define X as the random variable “number of heads showing when three coins are tossed.” Obtain the mean and the variance of X . Simulate tossing three fair coins 10,000 times. Compute the simulated mean and variance of X . Are the simulated values within 2% of the theoretical answers?

45. Every month, a family must decide what to do on Sundays. If they do not stay at home, they do one of two things with equal probability: have lunch in a restaurant, which costs €100, or go to the park, which is free. Assuming four weeks in a month, compute the probability distribution of expenditures.

46. In a lottery game, one can win €10,000 with probability 0.01 and €1000 with probability 0.05. How much should one pay for a lottery ticket to make the game fair?

47. Ante €100 to play a game and win €100 each round with a probability of $\frac{1}{2}$. Suppose a person plays the game until he loses once. Then, he leaves the game.

- (a) Find the probability that he plays more than four rounds.
 (b) Find the probability that he leaves the game having won exactly €600.
 (c) Calculate the expected winnings of this game.

48. Consider the random variable X , which takes the values 1, 2, 3, and 4 with probabilities 0.2, 0.3, 0.1, and 0.4, respectively. Calculate $E[X]$, $1/E[X]$, $E[1/X]$, $E[X^2]$, and $E[X]^2$, and check empirically that $E[X]^2 \neq E[X^2]$ and $E[1/X] \neq 1/E[X]$.

49. Show that the following distribution is a probability function. Construct a plot of the probability density function and obtain the cumulative distribution function.

$$\begin{aligned}\mathbb{P}(X = -2) &= 0.2, & \mathbb{P}(1 < X \leq 3) &= 0.1, & \mathbb{P}(X = 4) &= 0.2, \\ \mathbb{P}(5 < X \leq 5.5) &= 0.2, & \mathbb{P}(X = 6) &= 0.15, & \mathbb{P}(7 < X \leq 8) &= 0.15.\end{aligned}$$

50. Find the values of k such that the following functions are probability density functions:

- (a) $f(x) = kx^4/5$, $0 < x < 1$.
 (b) $f(x) = kx^2$, $0 < x < 2$.
 (c) $f(x) = k\sqrt{x}/2$, $0 < x < 1$.

Construct plots of these functions and their corresponding cumulative density functions.

51. Given the function

$$f(x) = k, \quad -1 < x < 1$$

of the random variable X , find the coefficient of skewness for the distribution.

52. Consider an experiment where two dice are rolled. Let the random variable X equal the sum of the two dice and the random variable Y be the difference of the two dice.

- (a) Find the mean of X .
- (b) Find the variance of X .
- (c) Find the skewness of X .
- (d) Find the mean of Y .
- (e) Find the variance of Y .
- (f) Find the skewness of Y .

53. The number of hits on a faculty member's homework solutions page has an average of 100 hits per day.

- (a) Give an upper bound for the probability the faculty member's homework solutions page has more than 112 hits per day.
- (b) Suppose the variance of the number of hits is known to be 36. Now, give an upper bound for the probability the faculty member's homework solutions page has more than 112 hits per day.
- (c) The probability that the number of hits is between 88 and 112 must be at least what?
- (d) How many days must visits to the site be recorded so that the average number of hits is within 6 of 100 with a probability of at least 0.9?

54. Given the following cumulative density function,

$$F(x) = \begin{cases} 0 & x < 0 \\ \frac{x^2}{4} & 0 \leq x \leq 2 \\ 1 & 2 < x, \end{cases}$$

derive the probability density function $f(x)$. Calculate the median of the distribution.

55. Consider the following function:

$$f(x) = \frac{2}{25}(x - 5), \quad 5 \leq x \leq 10.$$

- (a) Show that $f(x) \geq 0$, $-\infty < x < \infty$ and that $\int_{-\infty}^{\infty} f(x) dx = 1$.
- (b) Plot $f(x)$.
- (c) Derive and plot $f(x)$'s cumulative probability function, $F(x)$.

- (d) Calculate $\mathbb{P}(X \leq 8)$, $\mathbb{P}(X \geq 6)$, and $\mathbb{P}(7 \leq X \leq 8)$ by hand.
- (e) Calculate $\mathbb{P}(X \leq 8)$, $\mathbb{P}(X \geq 6)$, and $\mathbb{P}(7 \leq X \leq 8)$ using the function `integrate()`.

56. The number of bottles of milk that a dairy farm fills per day is a random variable with mean 5000 and standard deviation 100. Assume the farm always has a sufficient number of glass bottles to be used to store the milk. However, for a bottle of milk to be sent to a grocery store, it must be hermetically sealed with a metal cap that is produced on site. Calculate the minimum number of metal caps that must be produced on a daily basis so that all filled milk bottles can be shipped to grocery stores with a probability of at least 0.9.

57. Define X as the space occupied by certain device in a 1 m^3 container. The probability density function of X is given by

$$f(x) = \frac{630}{56}x^4(1-x^4), \quad 0 < x < 1.$$

- (a) Graph the probability density function.
- (b) Calculate the mean of X by hand.
- (c) Calculate the variance X by hand.
- (d) Calculate $\mathbb{P}(0.20 < X < 0.80)$ by hand.
- (e) Calculate the mean of X using `integrate()`.
- (f) Calculate the variance of X using `integrate()`.
- (g) Calculate $\mathbb{P}(0.20 < X < 0.80)$ using `integrate()`.

58. Consider the probability density function

$$f(x) = \frac{1}{36}xe^{-x/6}, \quad x > 0.$$

Derive the moment generating function, and calculate the mean and the variance.

59. Suppose the random variable X can take on the values 3, 5, 7, and 8 such that the probability of each value is twice its predecessor.

- (a) Write the **pdf** for X .
- (b) Write the **cdf** for X .
- (c) Compute the mean and variance of X .

60. Prove that if a and b are real-valued constants, then

- (1) $M_{X+a}(t) = E[e^{(X+a)t}] = e^{at} \cdot M_X(t).$
- (2) $M_{bX}(t) = E(e^{bXt}) = M_X(bt).$
- (3) $M_{\frac{X+a}{b}}(t) = E\left[e^{\left(\frac{X+a}{b}\right)t}\right] = e^{\frac{a}{b}t} \cdot M_X\left(\frac{t}{b}\right).$

61. The time, in minutes, that a car is parked in a mall has the following density function:

$$f(x) = \begin{cases} \frac{1}{50}e^{-x/50} & x > 0 \\ 0 & x \leq 0. \end{cases}$$

Using R,

- (a) Find the probability that a car stays more than 1 hour.
- (b) Let $Y = 0.5 + 0.03X$ be the cost in dollars that the mall has to pay a security service per parked car. Find the mean parking cost for 1000 cars.
- (c) Find the variance and skewness coefficient of Y .

62. A high technology company manufactures circular mirrors used in certain satellites. The radius of any mirror in inches is a random variable R with density function

$$f(r) = \begin{cases} \frac{24}{11}(2r - r^2) & 1 \leq r \leq \frac{3}{2} \\ 0 & \text{otherwise.} \end{cases}$$

To place the mirrors in the satellites without any problems, the mirror area, given by πR^2 , cannot be greater than 6.5 inches². Using R,

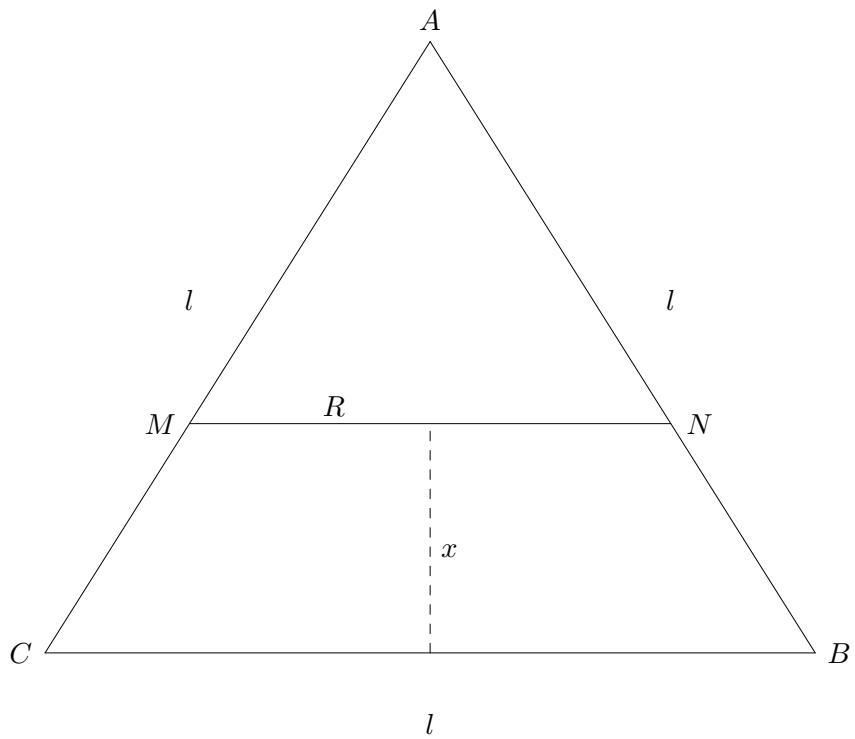
- (a) Verify that $\int_{-\infty}^{\infty} f(r) dr = 1$.
- (b) Find the mean area of the mirrors.
- (c) Find the probability that a mirror's area does not surpass 6.5 inches².

63. The time, in hours, a child practices his musical instrument on Saturdays has **pdf**

$$f(x) = \begin{cases} k(1-x) & \text{for } 0 \leq x \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

- (a) Find k to make $f(x)$ a valid **pdf**.
- (b) Write the **cdf** and find the probability the child practices more than 48 minutes on a Saturday.

64. Consider the equilateral triangle ABC with side l . Given a randomly chosen point R in the triangle, calculate the cumulative and the probability density functions for the distance from R to the side BC . Construct a graph of the cumulative density function for different values of l . (Hint: The equation of the line CA is $y = \sqrt{3}x$.)



Chapter 4

Univariate Probability Distributions

4.1 Introduction

This chapter examines univariate (single variable) probability distributions that are used frequently to model random phenomena. Discrete probability distributions are introduced first, followed by continuous probability distributions. Discrete distributions can be used to model the number of failures until a successful rocket launch, the number of passing students in a class, or the number of taxis that pass a street corner, as well as many other phenomena with countable outcomes. Continuous distributions are used to model measurement variables such as weight, height, and time. Joint distributions will be introduced in Chapter 5.

4.2 Discrete Univariate Distributions

When there are a countable number of elements in a sample space for a single experiment, a discrete univariate distribution is the result. The probability of any given value of the random variable occurring can be the same for every value or be more or less likely for certain values, depending on the structure of the experiment. Discrete distributions model everything from the likelihood of getting a four when a die is rolled to the number of free throws a basketball player might make in a game to how many batteries need to be pulled from a drawer before a good one is found. If the possible outcomes can be counted, even if that count could be infinite, a discrete univariate distribution will be needed to solve such probability problems.

4.2.1 Discrete Uniform Distribution

The random variable X is said to follow a discrete uniform distribution with parameter n (where $n \in \mathbb{N}$) if the probability X takes on the value x is the same for all x , where $x = x_1, x_2, \dots, x_n$:

Discrete Uniform Distribution

$$\begin{aligned}\mathbb{P}(X = x_i | n) &= \frac{1}{n}, \quad i = 1, 2, \dots, n. \\ E[X] &= \frac{1}{n} \sum_{i=1}^n x_i \\ Var[X] &= \frac{1}{n} \sum_{i=1}^n (x_i - E[X])^2 \\ M_X(t) &= \frac{1}{n} \sum_{i=1}^n e^{tx_i}\end{aligned}\tag{4.1}$$

When $x_i = i$ for $i = 1, \dots, n$, it can be shown that $E[X] = \frac{n+1}{2}$ and that $Var[X] = \frac{n^2-1}{12}$, respectively.

Example 4.1 One light bulb is randomly selected from a box that contains a 40-watt light bulb, a 60-watt light bulb, a 75-watt light bulb, a 100-watt light bulb, and a 120-watt light bulb. Write the probability function for the random variable that represents the wattage of the randomly selected light bulb, and determine the mean and variance of that random variable.

Solution: The random variable X can assume the set of values $\Omega = \{40, 60, 75, 100, 120\}$. The probability density function for the random variable X is

$$\mathbb{P}(X = x | n = 5) = 1/5 \quad \text{for } x = 40, 60, 75, 100, 120.$$

The arithmetic is performed in R Code 4.1.

R Code 4.1

```
> Watts <- c(40, 60, 75, 100, 120)
> n <- length(Watts)
> meanWatts <- (1/n) * sum(Watts)
> varWatts <- (1/n) * sum((Watts - meanWatts)^2)
> ans <- c(meanWatts, varWatts)
> ans
[1] 79 804
```

The expected value of X is $E[X] = 79$, and the variance of X is $Var[X] = 804$.

4.2.2 Bernoulli and Binomial Distributions

When the same coin is tossed n times by the same person under the same experimental conditions, it stands to reason that each toss of the coin will result in one of two outcomes (heads or tails), that the outcome on any given trial will not influence the outcome of any other trial, and that the probability of getting a head assuming a fair coin on any trial is a constant $\frac{1}{2}$. Tossing a coin a single time is an example of a **Bernoulli** trial. A Bernoulli trial is a random experiment with only two possible outcomes. The outcomes are mutually

exclusive and exhaustive; for example, success or failure, true or false, alive or dead, male or female, etc. A Bernoulli random variable, X , can take on two values, where $X(\text{success}) = 1$ and $X(\text{failure}) = 0$. The probability that X is a success is π , and the probability that X is a failure is $\varrho = 1 - \pi$. The **pdf**, mean, variance, and **mgf** of a Bernoulli random variable are shown in (4.2).

Bernoulli Distribution

$$X \sim \text{Bernoulli}(\pi)$$

$$\begin{aligned}\mathbb{P}(X = x|\pi) &= \pi^x(1 - \pi)^{1-x}, x = 0, 1 \\ E[X] &= \pi \\ \text{Var}[X] &= \pi(1 - \pi) \\ M_X(t) &= \pi e^t + \varrho\end{aligned}\tag{4.2}$$

When a sequence of Bernoulli trials conforms to the following list of requirements, it is called a **binomial experiment**:

1. The experiment consists of a fixed number (n) of Bernoulli trials.
2. The probability of success for each trial, denoted by π , is constant from trial to trial. The probability of failure is $\varrho = (1 - \pi)$.
3. The trials are independent.
4. The random variable of interest, X , is the number of observed successes during the n trials.

The probability that X is equal to x can be found in the following fashion. Any particular sequence of x successes occurs with probability $\pi^x(1 - \pi)^{(n-x)}$ since there are x successes and $(n - x)$ failures. However, there are $\binom{n}{x} = \frac{n!}{x!(n-x)!}$ possible sequences of x successes. Write $X \sim \text{Bin}(n, \pi)$ to indicate the random variable X follows a binomial distribution with parameters n and π . The probability X is equal to x , the mean, the variance, and the moment-generating function of a binomial random variable are shown in (4.3).

Binomial Distribution

$$X \sim \text{Bin}(n, \pi)$$

$$\begin{aligned}\mathbb{P}(X = x|n, \pi) &= \binom{n}{x} \pi^x(1 - \pi)^{n-x}, x = 0, 1, 2, \dots, n. \\ E[X] &= n\pi \\ \text{Var}[X] &= n\pi(1 - \pi) \\ M_X(t) &= (\pi e^t + \varrho)^n\end{aligned}\tag{4.3}$$

It is left as an exercise for the student to verify that $E[X] = n\pi$, $\text{Var}[X] = n\pi(1 - \pi)$, and that the moment generating function of a binomial random variable is $M_X(t) = (\pi e^t + \varrho)^n$.

Code to create graphs that represent the probability density function and the cumulative distribution function for a $\text{Bin}(8, 0.3)$ random variable is shown in R Code 4.2. The graphs R Code 4.2 creates are similar to those in Figure 4.1.

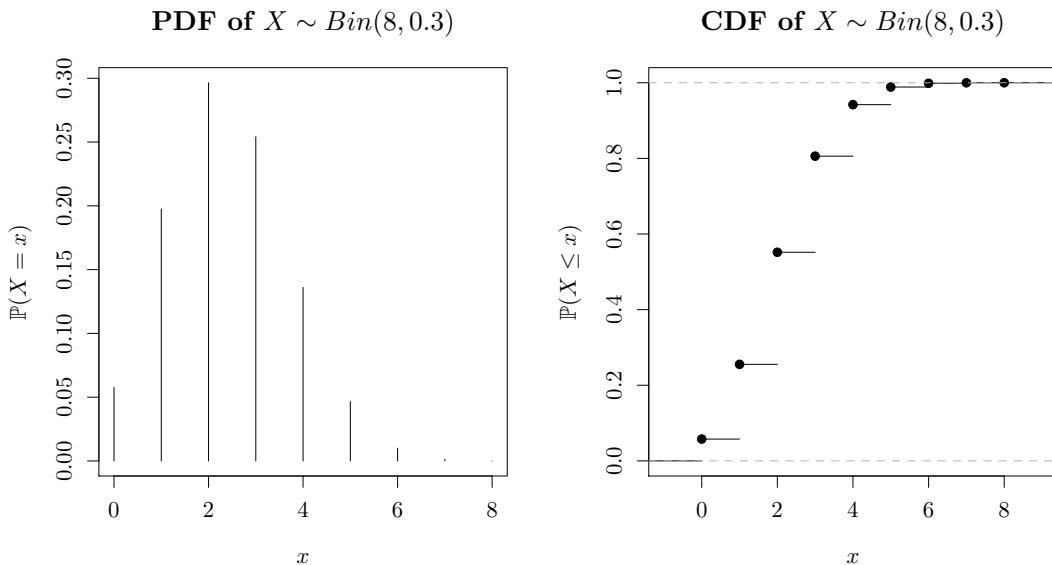


FIGURE 4.1: Left graph is the probability density function (**pdf**) of a binomial random variable with $n = 8$ and $\pi = 0.3$. Right graph is the cumulative distribution function (**cdf**) of a binomial random variable with $n = 8$ and $\pi = 0.3$.

R Code 4.2

```
> opar <- par(no.readonly = TRUE)
> par(mfrow=c(1, 2), pty = "s")
> x <- 0:8
> px <- dbinom(x, 8, 0.3)
> plot(x, px, type = "h", xlab = "x", ylab="P(X = x)",
+       main = "PDF of X~Bin(8, 0.3)")
> xs <- rep(0:8, round(dbinom(0:8, 8, .3)*100000, 0))
> plot(ecdf(xs), main = "CDF of X~Bin(8, 0.3)",
+       ylab = expression(P(X<=x)), xlab = "x")
> par(opar)
```

Figure 4.2 on the next page shows the **pdfs** for three different values of π with an n value of 10. Note how the distribution is skewed to the right when the value of π is close to zero, symmetric when π is 0.5, and skewed to the left when the value of π approaches one.

Example 4.2 \triangleright *Simulating Bernoulli* \lhd Consider the problem of simulating m repeated samples of n Bernoulli trials.

- Write a function that will generate m repeated samples of n Bernoulli trials each with probability of success π . The function should create a histogram with theoretical values superimposed over simulated values.
- Use the function to generate 1000 samples of size $n = 5$ with $\pi = 0.5$ to simulate the binomial distribution. Have the function create frequency tables for both the simulated and the theoretical random variable so that comparisons can be made between the two.

Solution: The answers are

$$X \sim Bin(n = 10, \pi)$$

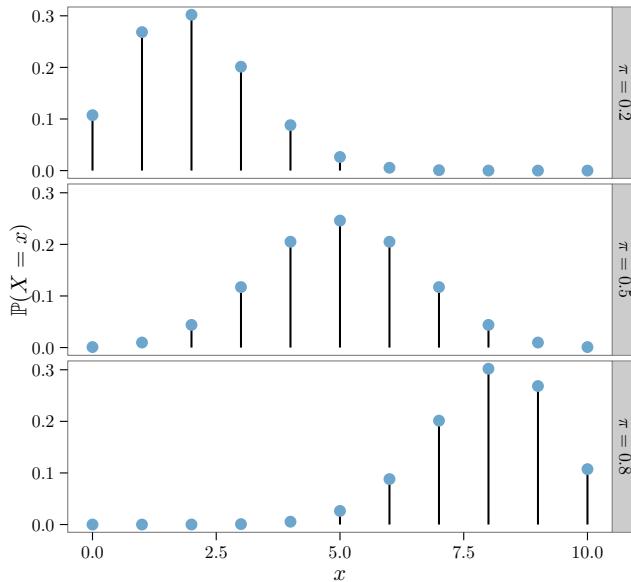


FIGURE 4.2: $Bin(10, \pi)$ pmfs for three different values of π

(a) The function `bino.gen()` shown in R Code 4.3 is written to solve Example 4.2 in general. Note that the function `bino.gen()` is part of the `PASWR2` package.

R Code 4.3

```
> bino.gen <- function (samples = 10000, n = 20, pi = 0.5)
+ {
+   values <- sample(c(0, 1), samples * n, replace = TRUE,
+                     prob = c(1 - pi, pi))
+   value.mat <- matrix(values, ncol = n)
+   Successes <- apply(value.mat, 1, sum)
+   a1 <- round((table(Successes)/samples), 3)
+   b1 <- round(dbinom(0:n, n, pi), 3)
+   names(b1) <- 0:n
+   hist(Successes, breaks = c((-0.5 + 0):(n + 0.5)), freq = FALSE,
+         ylab = "", col = 13, ylim = c(0, max(a1, b1)),
+         main = " Theoretical Values Superimposed
+             Over Histogram of Simulated Values")
+   x <- 0:n
+   fx <- dbinom(x, n, pi)
+   lines(x, fx, type = "h")
+   lines(x, fx, type = "p", pch = 16)
+   list(simulated.distribution = a1, theoretical.distribution = b1)
+ }
```

(b) The results shown in R Code 4.4 on the next page from using `bino.gen()` to generate 1000 samples where $n = 5$ and $\pi = 0.5$ answer this question. The resulting histogram from R Code 4.4 is shown in Figure 4.3.

R Code 4.4

```
> set.seed(31)
> bino.gen(samples = 1000, n = 5, pi = 0.5)

$simulated.distribution
Successes
  0      1      2      3      4      5
0.023 0.174 0.311 0.308 0.153 0.031

$theoretical.distribution
  0      1      2      3      4      5
0.031 0.156 0.312 0.312 0.156 0.031
```

**Theoretical Values Superimposed
Over Histogram of Simulated Values**

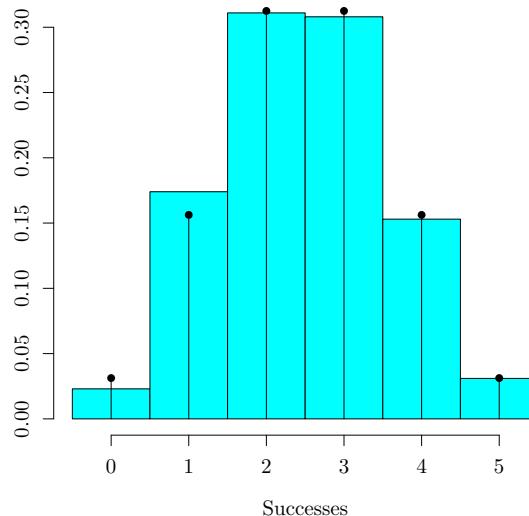


FIGURE 4.3: Histogram of 1000 simulated samples where $n = 5$ and $\pi = 0.5$ superimposed on the theoretical distribution for a random variable following a $Bin(5, 0.5)$ distribution

Using the function `rbinom()`, one can generate 1000 samples of a $Bin(n = 5, \pi = 0.5)$ as shown in R Code 4.5.

R Code 4.5

```
> set.seed(123)
> x <- rbinom(1000, 5, 0.5)
> table(x)/1000 # Empirical distribution

x
  0      1      2      3      4      5
0.029 0.163 0.315 0.314 0.148 0.031
```

If one wants to generate the same numbers at a later date, the command `set.seed()` can be used. The graph in Figure 4.3 on the preceding page was created with `set.seed(31)`. ■

Example 4.3 ▷ Binomial Calculation ◁ Consider the problem of calculating the probability of obtaining 6 or more heads in 10 tosses of a weighted coin, where the probability of obtaining a head in any given trial is 0.33.

Solution: Let the random variable X equal the number of trials that result in a head. Consequently, $X \sim \text{Bin}(10, 0.33)$, and the sum of the individual probabilities of obtaining 6, 7, 8, 9, and 10 heads needs to be found. Mathematically, this is written $\mathbb{P}(X \geq 6) = \mathbb{P}(X = 6) + \mathbb{P}(X = 7) + \dots + \mathbb{P}(X = 10)$, where

$$\mathbb{P}(X = 6) = \frac{10!}{6!(10-6)!} \times 0.33^6 \times (1 - 0.33)^{(10-6)} = 0.0547,$$

$$\mathbb{P}(X = 7) = \frac{10!}{7!(10-7)!} \times 0.33^7 \times (1 - 0.33)^{(10-7)} = 0.0154,$$

$$\mathbb{P}(X = 8) = \frac{10!}{8!(10-8)!} \times 0.33^8 \times (1 - 0.33)^{(10-8)} = 0.0028,$$

$$\mathbb{P}(X = 9) = \frac{10!}{9!(10-9)!} \times 0.33^9 \times (1 - 0.33)^{(10-9)} = 3e-04, \text{ and}$$

$$\mathbb{P}(X = 10) = \frac{10!}{10!(10-10)!} \times 0.33^{10} \times (1 - 0.33)^{(10-10)} = 0.$$

Thus,

$$\begin{aligned}\mathbb{P}(X \geq 6) &= \mathbb{P}(X = 6) + \mathbb{P}(X = 7) + \mathbb{P}(X = 8) + \mathbb{P}(X = 9) + \mathbb{P}(X = 10) \\ &= 0.0547 + 0.0154 + 0.0028 + 3e-04 + 0 \\ &= 0.0732.\end{aligned}$$

There are several approaches one might take to solve the problem with R. One should realize that the following are all equivalent statements:

$$\begin{aligned}\mathbb{P}(X \geq 6) &= \mathbb{P}(X = 6) + \mathbb{P}(X = 7) + \mathbb{P}(X = 8) + \mathbb{P}(X = 9) + \mathbb{P}(X = 10) \\ &= 1 - \mathbb{P}(X \leq 5) \\ &= 1 - [\mathbb{P}(X = 5) + \mathbb{P}(X = 4) + \dots + \mathbb{P}(X = 0)].\end{aligned}$$

To find $\mathbb{P}(X \geq 6)$ with R, compute the individual probabilities with `dbinom(6:10, 10, 0.33)` and then sum them with the command `sum()` by typing `sum(dbinom(6:10, 10, 0.33))`. Another solution is to find $1 - \mathbb{P}(X \leq 5)$, which is accomplished with `1 - pbinom(5, 10, 0.33)` or `1 - sum(dbinom(5:0, 10, 0.33))`. Note that `dbinom()` computes $\mathbb{P}(X = x)$, the **pdf**, while `pbinom()` gives $\mathbb{P}(X \leq x)$, the **cdf**.

```
> sum(dbinom(6:10, 10, 0.33))          # P(X >= 6)
[1] 0.07320046

> 1 - pbinom(5, 10, 0.33)            # 1 - P(X <= 5)
[1] 0.07320046

> pbinom(5, 10, 0.33, lower = FALSE) # P(X >= 6)
[1] 0.07320046

> 1 - sum(dbinom(5:0, 10, 0.33))    # 1 - P(X <= 5)
[1] 0.07320046
```

4.2.3 Poisson Distribution

The Poisson distribution is very popular for modeling the number of times particular events occur in given times or on defined spaces. For example, one might count the number of phone calls to 911 between 1 a.m. and 2 a.m., the number of accidents at a busy street corner during a 24-hour period, or the number of typographical errors on a single page of this book. Unfortunately, the derivation of the Poisson distribution is not straightforward. Instead of deriving the Poisson distribution directly, it is shown that the limiting distribution of the binomial distribution is the Poisson distribution. Actual derivation of the Poisson distribution function is beyond the scope of the current text.

When the number of outcomes in a given continuous interval are counted, an approximate **Poisson process** with parameter $\lambda > 0$ results if the following conditions are satisfied:

- (1) The number of outcomes in non-overlapping intervals are independent. In other words, the number of outcomes in the interval of time $(0, t]$ are independent from the number of outcomes in the interval of time $(t, t + h]$ for any $h > 0$.
- (2) The probability of two or more outcomes in a sufficiently short interval is virtually zero. In other words, provided h is sufficiently small, the probability of obtaining two or more outcomes in the interval $(t, t + h]$ is negligible compared to the probability of obtaining one or zero outcomes in the same interval of time.
- (3) The probability of exactly one outcome in a sufficiently short interval or small region is proportional to the length of the interval or region. In other words, the probability of one outcome in an interval of length h is λh .

When an experiment satisfies the conditions for the Poisson process, the resulting random variable, X , the number of outcomes, is called a Poisson random variable. The probability distribution of the Poisson random variable X , representing the number of outcomes in a given time interval or space region denoted by t , is

$$\mathbb{P}(X = x | \lambda t) = \frac{e^{-\lambda t} (\lambda t)^x}{x!} \quad x = 0, 1, \dots, \quad \lambda > 0. \quad (4.4)$$

Although the Poisson distribution is typically used for problems involving time or space, it can be viewed as the limiting form of the binomial distribution. Suppose there is an experiment that satisfies the three criteria for an approximate Poisson process. Let X represent the number of outcomes in an interval of length 1 ($t = 1$). To find $\mathbb{P}(X = x)$, divide the interval of length 1 into n subintervals of equal length. Provided n is much larger than x , the probability of one outcome in any given interval of length $1/n$ is approximately λ/n by criterion (3) of the Poisson process on this page. Substituting $\pi = \lambda/n$ into the binomial probability distribution gives

$$\begin{aligned} \binom{n}{x} \left(\frac{\lambda}{n}\right)^x \left(1 - \frac{\lambda}{n}\right)^{n-x} &= \frac{n(n-1)\cdots(n-x+1)}{x!} \frac{\lambda^x}{n^x} \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-x} \\ &= \frac{\lambda^x}{x!} \left[\frac{n}{n} \frac{(n-1)}{n} \cdots \frac{(n-x+1)}{n} \right] \left(1 - \frac{\lambda}{n}\right)^n \left(1 - \frac{\lambda}{n}\right)^{-x}. \end{aligned}$$

Now, if x is fixed and $n \rightarrow +\infty$ and $\pi \rightarrow 0$, so that $\lambda = n\pi$ remains constant, the expression between the braces goes to 1 and $\left(1 - \frac{\lambda}{n}\right)^{-x}$ is also 1. Using the fact that $\lim_{n \rightarrow \infty} (1 - \lambda/n)^n = e^{-\lambda}$, obtain $\frac{\lambda^x e^{-\lambda}}{x!}$. The Poisson distribution can be used to approximate binomial probabilities with $\lambda = n\pi$ provided $\pi \leq 0.1$ and $n\pi \leq 5$. See Example 4.8

on page 262 for an example of how the Poisson distribution is used to approximate the probabilities of a binomial distribution.

Poisson Distribution
 $X \sim Pois(\lambda)$

$$\mathbb{P}(X = x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x = 0, 1, 2, \dots \quad (4.5)$$

$$E[X] = \lambda$$

$$Var[X] = \lambda$$

$$M_X(t) = e^{\lambda(e^t - 1)}$$

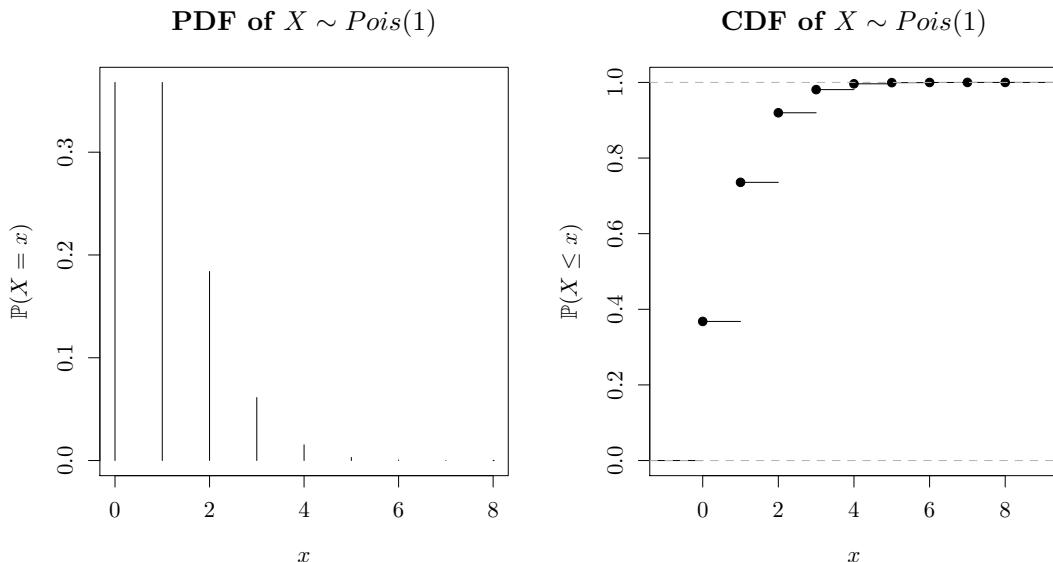


FIGURE 4.4: Left graph is the probability density function (**pdf**) of a Poisson random variable with $\lambda = 1$. Right graph is the cumulative distribution function (**cdf**) of a Poisson random variable with $\lambda = 1$.

Code to represent a probability density function and cumulative distribution function for a $Pois(\lambda = 1)$ random variable similar to the one shown in Figure 4.4 is given in R Code 4.6. Although the values x can take on with the Poisson distribution are $0, 1, 2, \dots$, the probability a Poisson random variable has a value of eight or greater when $\lambda = 1$ is extremely small (< 0). Consequently, the **pdf**, and **cdf** in Figure 4.4 do not extend beyond eight. Figure 4.5 on the next page shows how the shape of a Poisson distribution changes with increasing λ values.

R Code 4.6

```
> opar <- par(no.readonly = TRUE)
```

```

> par(mfrow=c(1, 2), pty = "s")
> x <- 0:8
> px <- dpois(x, 1)
> plot(x, px, type = "h", xlab = "x", ylab="P(X = x)",
+       main = "PDF of X ~ Pois(1)")
> xs <- rep(0:8, round(dpois(0:8, 1)*100000, 0))
> plot(ecdf(xs), main = "CDF of X ~ Pois(1)",
+       ylab = expression(P(X <=x)), xlab = "x")
> par(opar)

```

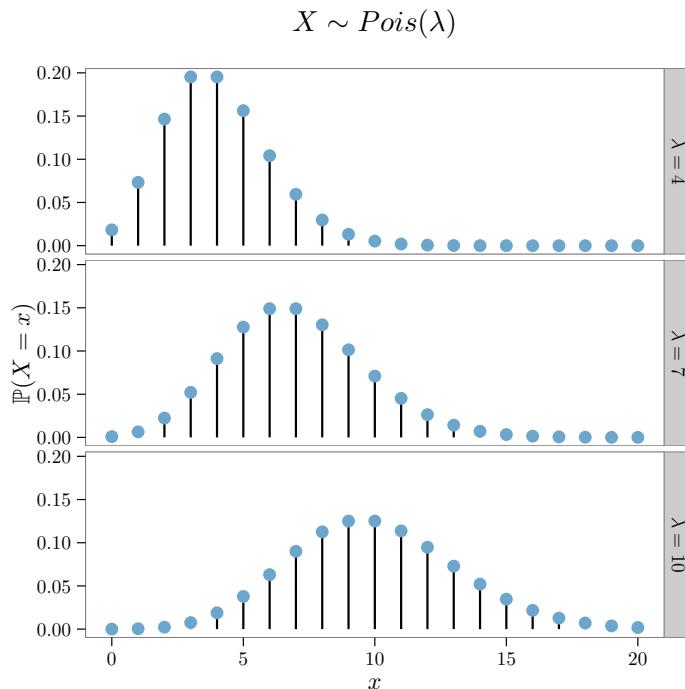


FIGURE 4.5: $Pois(\lambda)$ pdfs for three different values of λ

Note that the parameter λ , referred to as the intensity parameter, represents the mean number of outcomes in either a fixed time interval or a fixed spatial region. The Poisson distribution is particularly appropriate for modeling “rare” phenomena or outcomes where the probability of success is small. Note that whether or not data can be viewed as Poisson data depends on whether the proportions of 0’s, 1’s, 2’s, and so on, are similar to those predicted by the Poisson pdf given in (4.5). Given n independent Poisson random variables X_1, X_2, \dots, X_n with parameters $\lambda_1, \lambda_2, \dots, \lambda_n$, respectively, $Y = \sum_{i=1}^n X_i \sim Pois(\sum_{i=1}^n \lambda_i = \lambda)$.

Example 4.4 \triangleright **Poisson: World Cup Soccer** \triangleleft The World Cup is played once every four years. National teams from all over the world compete. In 2002 and in 1998, 36 teams were invited; whereas, in 1994 and in 1990, only 24 teams participated. The data frame **SOCCKER** contains three columns: **CGT**, **Game**, and **Goals**. All of the information contained in **SOCCKER** is indirectly available from the FIFA World Cup website, located at

<http://fifaworldcup.yahoo.com/>. The numbers of goals scored in the regulation 90-minute periods of World Cup soccer matches from 1990 to 2002 are listed in column `goals`. There were a total of 575 goals scored during regulation time. The game in which the goals were scored is in column `game`. There were 232 World Cup soccer games played from 1990 to 2002. There were 64 games played in each of 2002 and 1998 and 54 games played in each of 1994 and 1990. The cumulative goal time is provided in column `cgt`. For example, the first goal was scored at the 67th minute of the first game and the second goal was scored at the 42nd minute of the second game. Consequently, the times listed in `CGT` for the first two goals are 67, and $132 = 90 + 42$. For consistency, all goals scored during injury time are recorded in either the 45th or 90th minute, depending on the half when the injury occurred. Analyze the number of goals scored during regulation play (90 minutes) of World Cup soccer matches to verify that the scores follow an approximate Poisson distribution (Chu, 2003).

Solution: To investigate whether criterion (1) of the Poisson process on page 256 is reasonable, the one, two, three, four, and five game lagged correlation coefficients are examined in R Code 4.7.

R Code 4.7

```
> L1 <- SOCCER$goals[1:228]
> L2 <- SOCCER$goals[2:229]
> L3 <- SOCCER$goals[3:230]
> L4 <- SOCCER$goals[4:231]
> L5 <- SOCCER$goals[5:232]
> LAG <- cbind(L1, L2, L3, L4, L5)
> # or more succinctly
> LAG <- sapply(1:5, function(x) {
+   SOCCER$goals[x:(x + 227)]
+ })
> round(cor(LAG), 3)

      [,1]   [,2]   [,3]   [,4]   [,5]
[1,] 1.000 -0.049  0.055 -0.138 -0.008
[2,] -0.049  1.000 -0.046  0.044 -0.138
[3,]  0.055 -0.046  1.000 -0.054  0.045
[4,] -0.138  0.044 -0.054  1.000 -0.057
[5,] -0.008 -0.138  0.045 -0.057  1.000
```

Independence seems reasonable due to the small correlation coefficients (near zero) but should also be computed with time periods smaller than 90 minutes. Criterion (2) of the Poisson process on page 256, appears satisfied since two goals are never registered during the same one minute period. One way to investigate this is to create a table of the interarrival goal times and note that 0 is not in the table. Whether criterion (3) of the Poisson process on page 256 is satisfied is addressed in Problem 4.4 on page 313 at the end of the chapter. Next, examine the data to see how well they conform to the Poisson distribution. To calculate the observed number of goals scored during regulation time for the 232 World Cup soccer matches, use either `xtabs()` or `table()`.

```
> xtabs(~goals, data = SOCCER)

goals
 0  1  2  3  4  5  6  7  8
19 49 60 47 32 18  3  3  1
```

```
> table(SOCCER$goals)

 0 1 2 3 4 5 6 7 8
19 49 60 47 32 18 3 3 1
```

Since there are NA values in the `goals` column, use the `na.rm = TRUE` argument for the R functions `mean()` and `var()`, respectively. To verify that the mean and the variance of `goals` are approximately equal, type

```
> mean(SOCCER$goals, na.rm = TRUE)
[1] 2.478448
> var(SOCCER$goals, na.rm = TRUE)
[1] 2.458408
```

Because the mean and variance of `Goals` are approximately equal, it is reasonable to proceed in analyzing the frequencies of `goals` in comparison to those of a Poisson distribution with $\lambda = 2.4784$. The rationale for using the sample mean as an estimate for λ is that the sample mean of a Poisson distribution is both an unbiased estimator and the maximum likelihood estimator (see Chapter 7) of λ . A table is created in R Code 4.8 to facilitate comparing the observed values (`OBS`) to the expected values (`EXP`) as well as the empirical proportions (`Empir`) to the theoretical proportions (`TheoP`) for a Poisson distribution with $\lambda = 2.4784$, the mean number of goals per game. The empirical proportions are merely the number of goals in each category divided by the total number of goals.

R Code 4.8

```
> OBS <- xtabs(~goals, data = SOCCER)
> Empir <- round(OBS/sum(OBS), 3)
> TheoP <- round(dpois(0:(length(OBS) - 1), mean(SOCCER$goals,
+   na.rm = TRUE)), 3)
> EXP <- round(TheoP * 232, 0)
> ANS <- cbind(OBS, EXP, Empir, TheoP)
> ANS

  OBS EXP Empir TheoP
0 19 19 0.082 0.084
1 49 48 0.211 0.208
2 60 60 0.259 0.258
3 47 49 0.203 0.213
4 32 31 0.138 0.132
5 18 15 0.078 0.065
6 3 6 0.013 0.027
7 3 2 0.013 0.010
8 1 1 0.004 0.003
```

Since the observed values are close to the expected values, the empirical proportions will be close to the theoretical probabilities. This, in conjunction with the fact that the sample mean (2.4784) is roughly equal to the sample variance (2.4584), implies that modeling the number of goals scored during World Cup soccer games with a Poisson distribution is reasonable. ■

Example 4.5 Given a random variable X that follows a Poisson distribution with parameter λ , find the mean and variance of X . Use the fact that

$$e^\lambda = \sum_{r=0}^{\infty} \frac{\lambda^r}{r!} = 1 + \frac{\lambda}{1!} + \frac{\lambda^2}{2!} + \dots$$

Solution:

$$\begin{aligned} E[X] &= \sum_{r=0}^{\infty} r \frac{\lambda^r}{r!} e^{-\lambda} = \lambda e^{-\lambda} \sum_{r=1}^{\infty} \frac{\lambda^{r-1}}{(r-1)!} = \lambda \text{ and} \\ \text{Var}[X] &= \sum_{r=0}^{\infty} (r - \lambda)^2 \frac{\lambda^r}{r!} e^{-\lambda}. \end{aligned}$$

Rearranging terms,

$$\begin{aligned} \text{Var}[X] &= e^{-\lambda} \left\{ \sum_{r=0}^{\infty} r^2 \frac{\lambda^r}{r!} + \sum_{r=0}^{\infty} \lambda^2 \frac{\lambda^r}{r!} - 2\lambda \sum_{r=0}^{\infty} r \frac{\lambda^r}{r!} \right\} \\ &= e^{-\lambda} \left\{ \sum_{r=1}^{\infty} r \frac{\lambda^r}{(r-1)!} + \lambda^2 e^\lambda - 2\lambda \cdot \lambda \sum_{r=1}^{\infty} \frac{\lambda^{r-1}}{(r-1)!} \right\} \\ &= e^{-\lambda} \left\{ \sum_{r=1}^{\infty} (r-1+1) \frac{\lambda^r}{(r-1)!} + \lambda^2 e^\lambda - 2\lambda^2 e^\lambda \right\} \\ &= e^{-\lambda} \left\{ \sum_{r=1}^{\infty} (r-1) \frac{\lambda^r}{(r-1)!} + \sum_{r=1}^{\infty} \frac{\lambda^r}{(r-1)!} + \lambda^2 e^\lambda - 2\lambda^2 e^\lambda \right\} \\ &= e^{-\lambda} \{ \lambda^2 + \lambda + \lambda^2 - 2\lambda^2 \} e^\lambda = \lambda. \end{aligned}$$



Example 4.6 More accidents are registered in auto body repair shops during the months of May and June than in the rest of the year. Suppose a particular auto body repair shop has an average of four accidents per month. What is the probability there will be more than seven accidents in this auto body shop during the month of May? What is the probability no more than three accidents will occur during the months of May and June?

Solution: Assuming accidents in the auto body shop follow an approximate Poisson process, the probability of x accidents in one month is

$$\mathbb{P}(X = x) = \frac{4^x e^{-4}}{x!} \quad \text{for } x = 0, 1, 2, \dots$$

The probability more than seven accidents occur during the month of May is

$$\mathbb{P}(X > 7) = 1 - \mathbb{P}(X \leq 7) = 1 - \sum_{i=0}^7 \frac{4^i e^{-4}}{i!} = 0.0511.$$

```
> 1 - ppois(q = 7, lambda = 4)                                # P(X > 7/4)
[1] 0.05113362
> ppois(q = 7, lambda = 4, lower = FALSE)                      # P(X > 7/4)
[1] 0.05113362
```

Since the expected number of accidents during May and June is $\lambda' = 2 \cdot 4 = 8$, the probability no more than three accidents occur for the two months in question is calculated as

$$\mathbb{P}(X \leq 3) = \sum_{i=0}^3 \frac{8^i e^{-8}}{i!} = 0.0424.$$

```
> ppois(q = 3, lambda = 8) # P(X <= 3/8)
[1] 0.04238011
```



Example 4.7 Telephone calls to a local 911 number are known to follow a Poisson distribution with an average of two calls per minute. Compute the probability that

- (a) There will be zero calls during a one minute period.
- (b) There will be less than five calls in a one minute period.
- (c) There will be less than six calls in one hour.

Solution: The answers are as follows:

$$(a) \mathbb{P}(X = 0 | \lambda = 2) = \frac{\lambda^0 e^{-\lambda}}{0!} = \frac{2^0}{0!} e^{-2} = 0.1353.$$

```
> dpois(x = 0, lambda = 2)
```

```
[1] 0.1353353
```

$$(b) \text{Note that the } \mathbb{P}(X < 5) = \mathbb{P}(X \leq 4).$$

$$\mathbb{P}(X \leq 4 | \lambda = 2) = \sum_{r=0}^4 \frac{\lambda^r e^{-\lambda}}{r!} = e^{-2} \left(1 + 2 + \frac{2^2}{2!} + \frac{2^3}{3!} + \frac{2^4}{4!} \right) = 0.9473.$$

```
> ppois(q = 4, lambda = 2)
```

```
[1] 0.947347
```

(c) Note that the time period changes from one minute to one hour (60 minutes). Consequently, the average number of calls in one hour is $\lambda' = 2 \times (60) = 120$.

$$\begin{aligned} \mathbb{P}(X \leq 5 | \lambda' = 120) &= \sum_{r=0}^5 \frac{\lambda'^r e^{-\lambda'}}{r!} \\ &= e^{-120} \left(1 + 120 + \frac{120^2}{2!} + \frac{120^3}{3!} + \frac{120^4}{4!} + \frac{120^5}{5!} \right) = 0. \end{aligned}$$

```
> ppois(q = 5, lambda = 120)
[1] 1.658476e-44
```



Example 4.8 Numerically show the results from approximating a $Bin(n = 100, \pi = 0.04)$ distribution with a $Pois(\lambda = 4)$.

Solution: The probability distribution function for a $\text{Bin}(100, 0.04)$ random variable is

$$\mathbb{P}_{\text{Bin}}(X = x) = \binom{100}{x} (0.04)^x (0.96)^{100-x}, \quad x = 0, 1, 2, \dots$$

Since $\pi < 0.1$ and $\lambda = n\pi = 100(0.04) = 4 < 5$, the Poisson distribution can be used to obtain reasonable approximations to the binomial distribution. The probability distribution for a $\text{Pois}(4)$ is

$$\mathbb{P}_{\text{Pois}}(X = x) = \frac{e^{-4} 4^x}{x!}, \quad x = 0, 1, 2, \dots$$

The first eight values of x for $\mathbb{P}_{\text{Bin}}(X = x)$ and $\mathbb{P}_{\text{Pois}}(X = x)$ are given in Table 4.1. Note

Table 4.1: Comparison of binomial and Poisson probabilities

x	0	1	2	3	4	5	6	7	8
$\mathbb{P}_{\text{Bin}}(X = x)$	0.017	0.070	0.145	0.197	0.199	0.160	0.105	0.059	0.029
$\mathbb{P}_{\text{Pois}}(X = x)$	0.018	0.073	0.147	0.195	0.195	0.156	0.104	0.060	0.030

that the results of $\mathbb{P}_{\text{Bin}}(X = x)$ and $\mathbb{P}_{\text{Pois}}(X = x)$ are virtually identical out to two decimal places. The values in Table 4.1 were generated using R commands as follows:

```
> x <- 0:8
> round(dbinom(x, 100, 0.04), 3)
[1] 0.017 0.070 0.145 0.197 0.199 0.160 0.105 0.059 0.029
> round(dpois(x, 4), 3)
[1] 0.018 0.073 0.147 0.195 0.195 0.156 0.104 0.060 0.030
```



4.2.4 Geometric Distribution

The geometric distribution, like the binomial distribution, is based on Bernoulli trials; however, the geometric distribution does not fix the number of trials prior to the experiment. The geometric distribution computes the probability that the first success occurs after r failures instead of computing the probability of observing x successes in n trials. A random variable X that counts the number of Bernoulli trials that result in failure before the first success is called a **geometric** random variable. Clearly, the probability of a success after r failures is $\pi \times (1 - \pi)^r$, which leads to the geometric probability distribution function where $\varrho = 1 - \pi$ is the probability of failure as it was for the Bernoulli and binomial distributions. The **pdf**, mean, variance, and **mgf** for a geometric random variable are shown in (4.6).

Geometric Distribution
 $X \sim Geo(\pi)$

$$\mathbb{P}(X = x | \pi) = \pi \varrho^x, x = 0, 1, \dots$$

$$E[X] = \frac{\varrho}{\pi} \quad (4.6)$$

$$Var[X] = \frac{\varrho}{\pi^2}$$

$$M_X(t) = \frac{\pi}{1 - \varrho e^t}$$

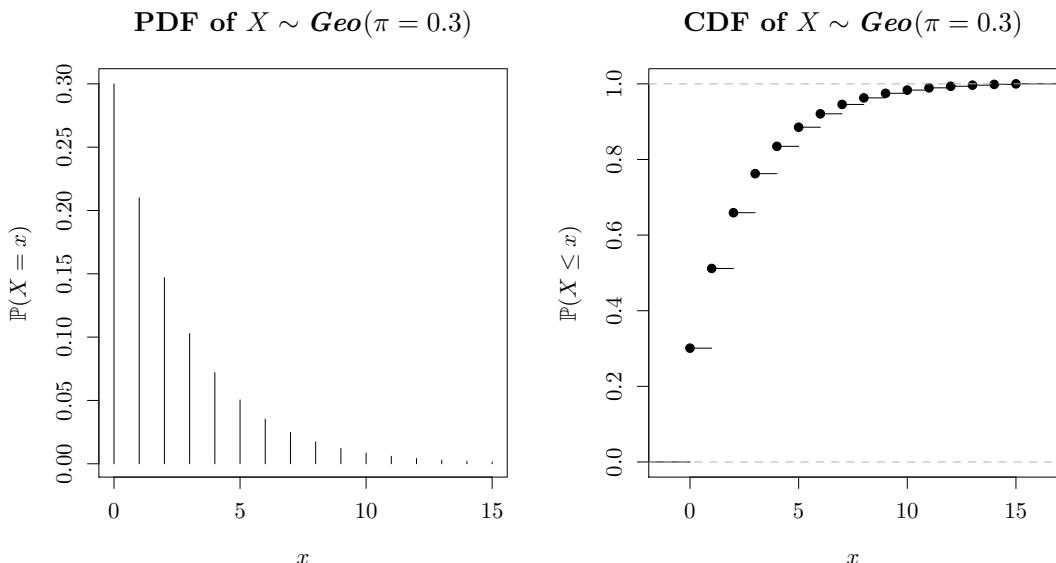


FIGURE 4.6: Left graph is the probability density function (**pdf**) of a geometric random variable with $\pi = 0.3$. Right graph is the cumulative distribution function (**cdf**) of a geometric random variable with $\pi = 0.3$.

R Code 4.9 can be used to create graphs that represent the probability density function and the cumulative distribution function for a $Geo(\pi = 0.3)$ random variable. The graphs from R Code 4.9 are similar to those in Figure 4.6. Like the Poisson distribution, the values x can assume with the Geometric distribution are $0, 1, 2, \dots$; however, since the $\mathbb{P}(X = 15 | \pi = 0.3) = 0.0014$ is extremely small, Figure 4.6 only shows x values out to 15. Figure 4.7 on the facing page shows how the shape of the distribution changes as π gets larger.

R Code 4.9

```
> opar <- par(no.readonly = TRUE)
> par(mfrow=c(1, 2), pty = "s")
> x <- 0:15
```

$$X \sim Geo(\pi)$$

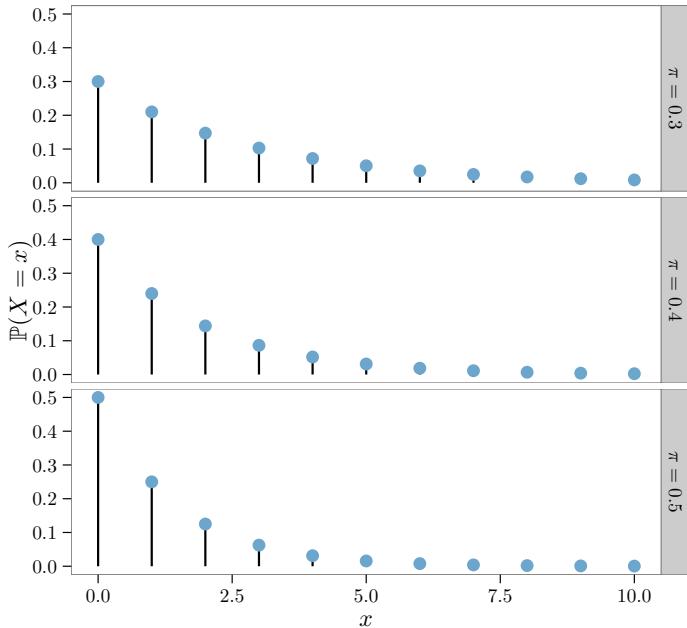


FIGURE 4.7: $Geo(\pi)$ pmfs for three different values of π

```
> px <- dgeom(x, .3)
> plot(x, px, type = "h", xlab = "x", ylab="P(X = x)",
+       main = "PDF of X ~ Geom(0.3)")
> xs <- rep(0:15, round(dgeom(0:15, 0.3)*100000, 0))
> plot(ecdf(xs), main = "CDF of X ~ Geom(0.3)",
+       ylab = expression(P(X <=x)), xlab = "x")
> par(opar)
```

Example 4.9 \triangleright **Geometric Distribution: Hiring a CPA** \triangleleft It is known that 20% of all applicants for an overseas position with an international accounting firm speak a foreign language and have passed the CPA (certified public accountant) exam. If applicants are selected at random and interviewed one at a time for the position,

- (a) Compute the probability that the first applicant who speaks a foreign language and has passed the CPA exam is the fourth applicant interviewed.
- (b) Suppose the first applicant that speaks a foreign language who has passed the CPA exam is offered the position and that the applicant accepts the offer. If the accounting firm spends 200 dollars for each interview, what are the expected value and standard deviation of the firm's cost for filling the position.

Solution: The answers are as follows:

- (a) Let the random variable X represent the number of applicants interviewed who either do not speak a foreign language or have not passed the CPA exam before the first applicant who both speaks a foreign language and has passed the CPA exam is interviewed. The random variable $X \sim Geo(\pi = 0.2)$ and the $P(X = 3)$ is computed using the **pdf** from (4.6)

as

$$\mathbb{P}(X = 3) = \pi\varrho^3 = 0.2(0.8)^3 = 0.1024.$$

When $X \sim Geo(\pi = 0.2)$, the $\mathbb{P}(X = 3)$ can be found with R using the command `dgeom(3, 0.2)`:

```
> dgeom(3, 0.2) # P(X = 3|0.2)
```

```
[1] 0.1024
```

(b) Be careful with this problem! The expected value and standard deviation of the cost for filling the position are not the same as the expected value and standard deviation of the random variable X as defined in the solution for part (a). Since the question asks for the expected value and standard deviation of the cost for filling the position (r failures and one success),

$$\begin{aligned} E[200(X + 1)] &= 200E[(X + 1)] \\ &= 200(E[X] + 1) \\ &= 200 \left(\frac{0.8}{0.2} + 1 \right) = 1000 \text{ dollars.} \end{aligned}$$

$$\begin{aligned} Var[200(X + 1)] &= 40,000 Var[(X + 1)] \\ &= 40,000 Var[X] \\ &= 40,000 \left(\frac{0.8}{0.2^2} \right) = 800,000 \text{ dollars}^2. \end{aligned}$$

$$\Rightarrow \sigma_{200(X+1)} = \sqrt{Var[200(X + 1)]} = 894.43 \text{ dollars.}$$



4.2.5 Negative Binomial Distribution

The geometric random variable counted the number of failures prior to the first success. Quite often, the number of Bernoulli trials required to achieve some fixed number (r) of successes is the quantity of interest. When the random variable X is defined as the number of failures prior to the r^{th} success, X has a **negative binomial** distribution written $X \sim NB(r, \pi)$. To find the $\mathbb{P}(X = x)$, first find the probability of $r - 1$ successes in the first $x + r - 1$ trials, and then multiply by the probability of success on the $(x + r)^{\text{th}}$ trial, $\binom{x+r-1}{r-1}\pi^{r-1}(1-\pi)^x \times \pi$. Combining like terms leads to the probability distribution for the negative binomial given in (4.7). The mean, variance, and mgf are also shown in (4.7):

Negative Binomial Distribution
$X \sim NB(r, \pi)$
$\mathbb{P}(X = x r, \pi) = \binom{x+r-1}{r-1} \pi^r \varrho^x, x = 0, 1, 2, \dots$
$E[X] = r \frac{\varrho}{\pi}$
$Var[X] = r \frac{\varrho}{\pi^2}$
$M_X(t) = \pi^r (1 - \varrho e^t)^{-r}$

(4.7)

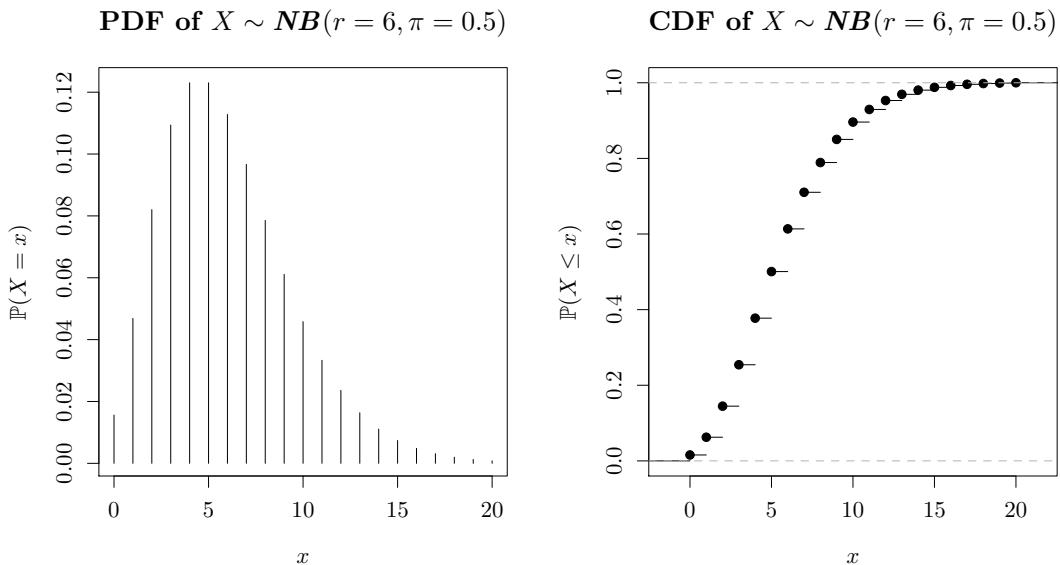


FIGURE 4.8: Left graph is the probability density function (**pdf**) of a negative binomial random variable with $r = 6$ and $\pi = 0.5$. Right graph is the cumulative distribution function (**cdf**) of a negative binomial random variable with $r = 6$ and $\pi = 0.5$.

Code to create graphs that represent the probability density function and the cumulative distribution function for a $NB(r, \pi)$ random variable is shown in R Code 4.10. The graphs created with R Code 4.10 are similar to those in Figure 4.8. The probability a negative binomial random variable with $r = 6$ and $\pi = 0.5$ for x values of 20 or greater is small ($< 8e-04$), so the values for x in Figure 4.8 extend only to 20. Figure 4.9 on the next page shows **pdfs** of the negative binomial distribution for the six combinations of $r = 1, 3, 5$ and $\pi = 0.4, 0.6$.

R Code 4.10

```
> opar <- par(no.readonly = TRUE)
> par(mfrow=c(1, 2), pty = "s")
> x <- 0:20
> r <- 6
> px <- dnbinom(x, r, .5)
> plot(x, px, type = "h", xlab = "x", ylab="P(X = x)",
+       main = "PDF of X ~ NB(6, 0.5)")
> xs <- rep(0:20, round(dnbinom(0:20, r, 0.5)*100000, 0))
> plot(ecdf(xs), main = "CDF of X ~ NB(6, 0.5)",
+       ylab = expression(P(X <= x)), xlab = "x")
> par(opar)
```

Useful Relationships

1. If n independent random variables, X_1, \dots, X_n , have a geometric distribution with parameter π , then the sum of the n independent random variables follows a negative binomial distribution with parameters (n, π) .
2. If n independent random variables, X_1, \dots, X_n , have a negative binomial distribution with parameters r_i and π , then the sum of the n random variables is $NB(\sum_{i=1}^n r_i, \pi)$.

$$X \sim NB(r, \pi)$$

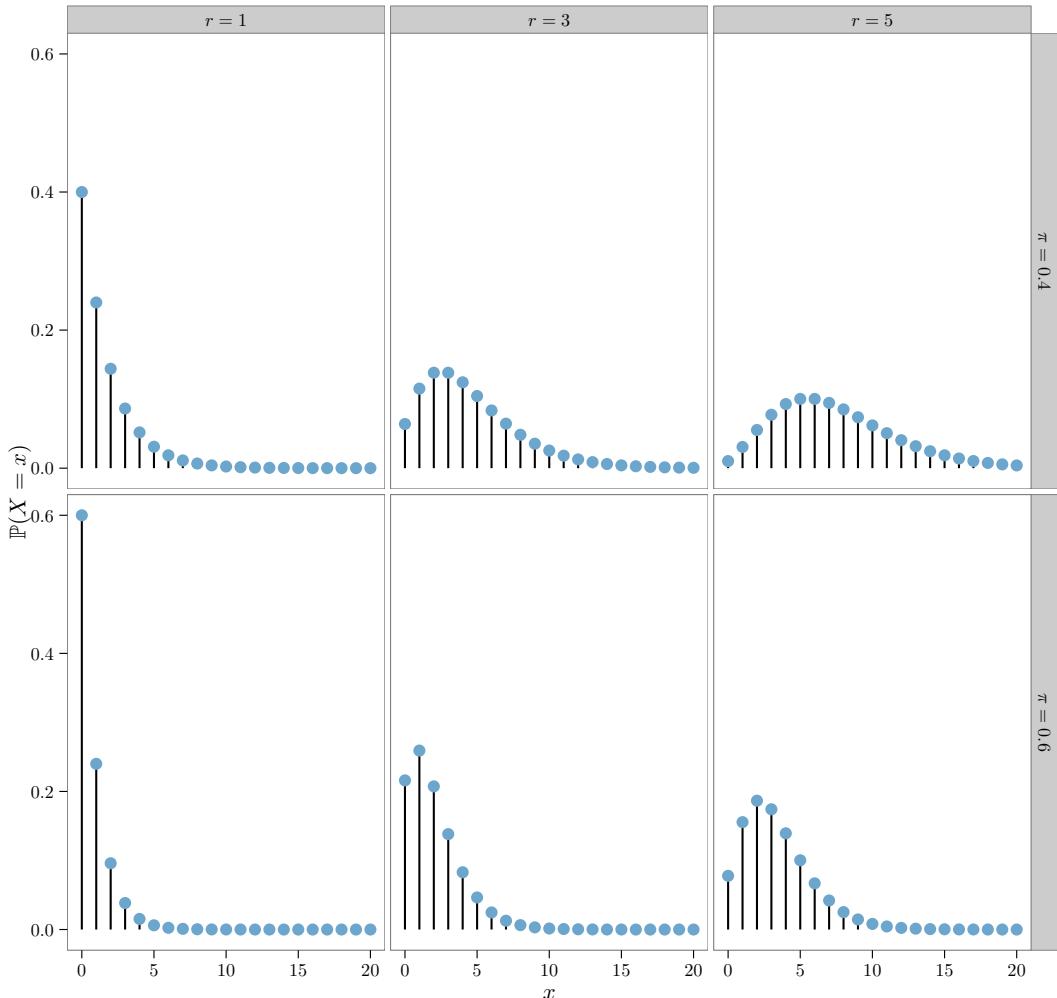


FIGURE 4.9: $NB(r, \pi)$ pdfs for three different values of r and two different values of π

3. When $X \sim NB(r, \pi)$ and $r = 1$, a negative binomial random variable is the same as a geometric random variable with parameter π .

Example 4.10 In a particular lot of white wall tires, 10% are missing their white wall. What is the probability one will have to examine six tires before finding four tires with white walls?

Solution: Let the random variable X represent the number of tires without white walls examined before obtaining four tires with white walls. In other words, $X \sim NB(4, 0.90)$ and it follows that

$$\begin{aligned} \mathbb{P}(X = 2|4, 0.9) &= \binom{2+4-1}{4-1} (0.9)^4 (0.1)^2 \\ &= \frac{5!}{3!(2!)} (0.9)^4 (0.1)^2 = 0.0656. \end{aligned}$$

To compute the answer with R use the command `dnbino(x, r, π)`.

```
> dnbino(2, 4, 0.9) # P(X = 2 | 4, 0.9)
[1] 0.06561
```



4.2.6 Hypergeometric Distribution

When working with finite populations, the binomial model often becomes untenable. Specifically, when sampling without replacement, the assumption of constant probability from trial to trial is no longer satisfied; however, deriving the exact distribution for a finite sample of dichotomous objects is not difficult. Given a dichotomous population of objects such that m are good and n are bad, the probability of selecting exactly x good items and $k - x$ bad items from a sample of size k is $\binom{m}{x} \binom{n}{k-x} / \binom{m+n}{k}$. Consequently, the random variable X that represents the number of good items selected from a total of m good items in a sample of size k is a **hypergeometric** random variable.

Hypergeometric Distribution

$$X \sim \text{Hyper}(m, n, k)$$

$$\mathbb{P}(X = x|m, n, k) = \frac{\binom{m}{x} \binom{n}{k-x}}{\binom{N}{k}}, \quad (4.8)$$

for $x = \max\{0, k - n\}, \dots, \min\{m, k\}$, where $N = m + n$

$$E[X] = \frac{m \times k}{N}$$

$$Var[X] = \frac{m \times n \times k \times (N - k)}{N^2 \times (N - 1)}$$

Code to create graphs that represent the probability density function and the cumulative distribution function for a $\text{Hyper}(m, n, k)$ random variable is provided in R Code 4.11. The graphs created from R Code 4.11 are similar to those in Figure 4.10 on the following page. Figure 4.11 on page 271 shows six hypergeometric probability mass functions for three different values of m and two different values of k .

R Code 4.11

```
> opar <- par(no.readonly = TRUE)
> par(mfrow=c(1, 2), pty = "s")
> x <- 0:10
> m <- 15
> n <- 10
> k <- 10
> px <- dhyper(x, m, n, k)
> plot(x, px, type = "h", xlab = "x", ylab="P(X = x)",
+       main = "PDF of X ~ Hyper(15, 10, 10)")
> xs <- rep(x, round(dhyper(x, m, n, k)*10000000, 0))
> plot(ecdf(xs), main = "CDF of X ~ Hyper(15, 10, 10)",
```

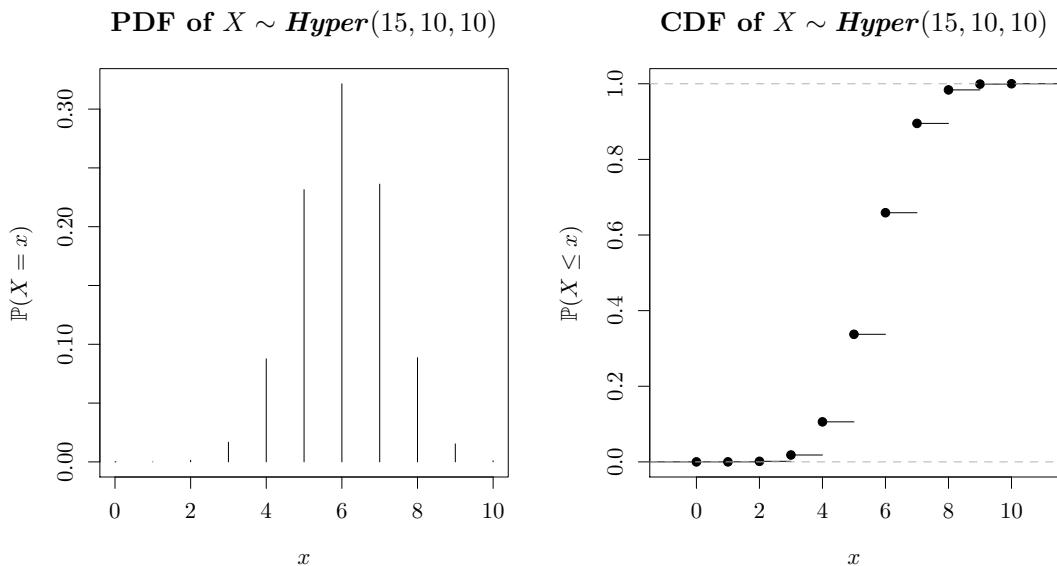


FIGURE 4.10: Left graph is the probability density function (**pdf**) of a hypergeometric random variable with $m = 15$, $n = 10$, and $k = 10$. Right graph is the cumulative distribution function (**cdf**) of a hypergeometric random variable with $m = 15$, $n = 10$, and $k = 10$.

```
+     ylab = expression(P(X <=x)), xlab = "x")
> par(opar)
```

One should note that when $\frac{k}{N}$ is small (≤ 0.10), the distribution of a hypergeometric random variable does not differ greatly from the distribution of a binomial random variable with parameters $n = k$ and $\pi = \frac{m}{N}$. In particular, consider how the top row of Figure 4.11 on the facing page compares to Figure 4.12 on page 272.

Example 4.11 A computer manufacturer decides to purchase monitors from a new start-up company claiming strict quality control standards. The manufacturer orders 150 monitors and decides to accept the lot provided a random sample of size 25 reveals no defective monitors. If the lot of 150 monitors contains three defective monitors, what is the probability the lot will be accepted?

Solution: Let the random variable X represent the number of non-defective monitors in the sample. Since $X \sim \text{Hyper}(147, 3, 25)$, the $\mathbb{P}(X = 25 | m = 147, n = 3, k = 25)$ is computed as

$$\mathbb{P}(X = 25 | m = 147, n = 3, k = 25) = \frac{\binom{147}{25} \binom{3}{0}}{\binom{150}{25}} = 0.5764.$$

To compute the answer in R use the command `dhyper(x, m, n, k)`:

```
> dhyper(25, 147, 3, 25) # P(X = 25 | m = 147, n = 3, k = 25)
[1] 0.576365
```



$$X \sim \text{Hyper}(m, 10, k)$$

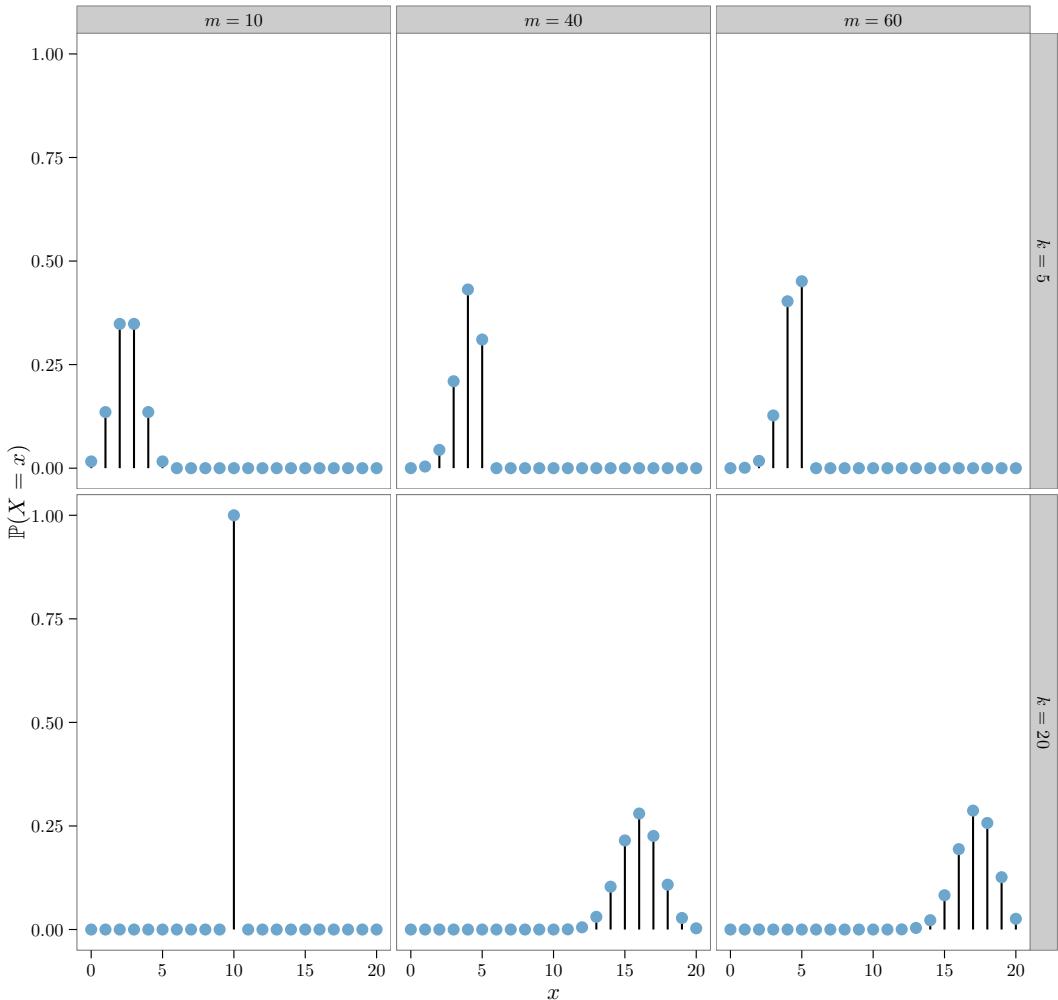


FIGURE 4.11: $\text{Hyper}(m, n = 10, k)$ pdfs for three different values of m and two different values of k

4.3 Continuous Univariate Distributions

If the possible results of a single experiment fall in a range of values, a continuous univariate distribution will be needed to model the results. Essentially any non-negative function whose integral over the domain of possible values is one can be the probability density of a random variable. Acceptable functions for a distribution can be lines or curves, and the probability of a particular range of values is the integral from the lower limit to the upper limit of the probability density function.

$$X \sim Bin(n = 5, \pi)$$

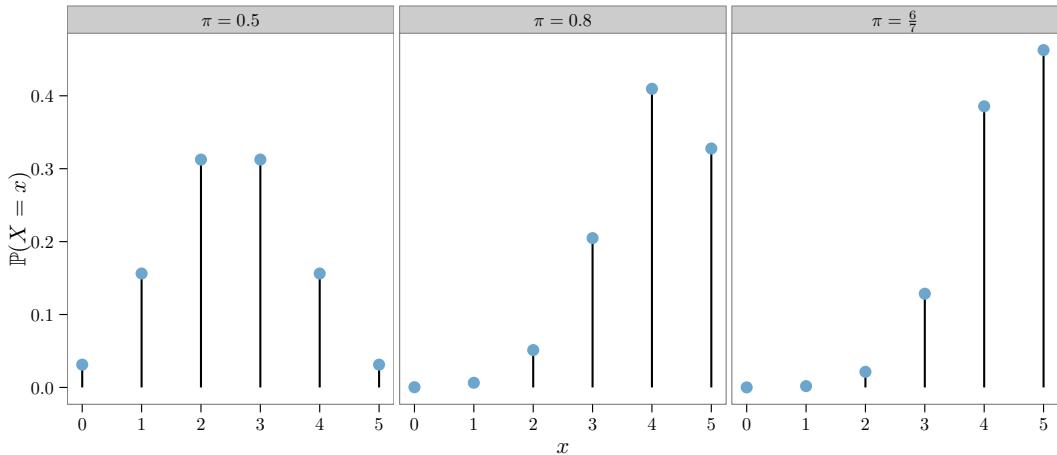


FIGURE 4.12: $Bin(n = 5, \pi)$ pmfs for three different values of π

4.3.1 Uniform Distribution (Continuous)

The continuous uniform distribution has a **pdf** that is constant over a closed interval. An important application of the uniform distribution includes random number generation. Random number generation with the inverse transformation technique is illustrated so the reader can generate values from a defined distribution that may not be programmed in R.

X is a **uniform** random variable defined on the interval $[a, b]$ if its **pdf** is given by

$$f(x|a, b) = \frac{1}{b - a}, \quad a \leq x \leq b.$$

The **pdf**, mean, variance, and **mgf** for a uniform random variable are found in (4.9).

Uniform Distribution $X \sim Unif(a, b)$ $f(x a, b) = \frac{1}{b - a}, \quad a \leq x \leq b$ $E[X] = \frac{b + a}{2}$ $Var[X] = \frac{(b - a)^2}{12}$ $M_X(t) = \begin{cases} \frac{e^{tb} - e^{ta}}{t(b - a)} & \text{if } t \neq 0 \\ 1 & \text{if } t = 0 \end{cases}$
--

(4.9)

Figure 4.13 on the facing page displays both the **pdf** and **cdf** for two $Unif(a, b)$ random variables. The **pdf** for the $Unif(0, 8)$ is shown in the top left of Figure 4.13 on the next page, while the **cdf** for the $Unif(0, 8)$ is shown in the bottom left of Figure 4.13 on the facing page. The **pdf** for the $Unif(4, 8)$ is shown in the top right of Figure 4.13 on the next page, while the **cdf** for the $Unif(4, 8)$ is shown in the bottom right of Figure 4.13 on the facing

page. Note that the area beneath each **pdf** is clearly one since the **pdf** forms a rectangle whose area is height \times length, $\frac{1}{(b-a)} \times (b-a) = 1$.

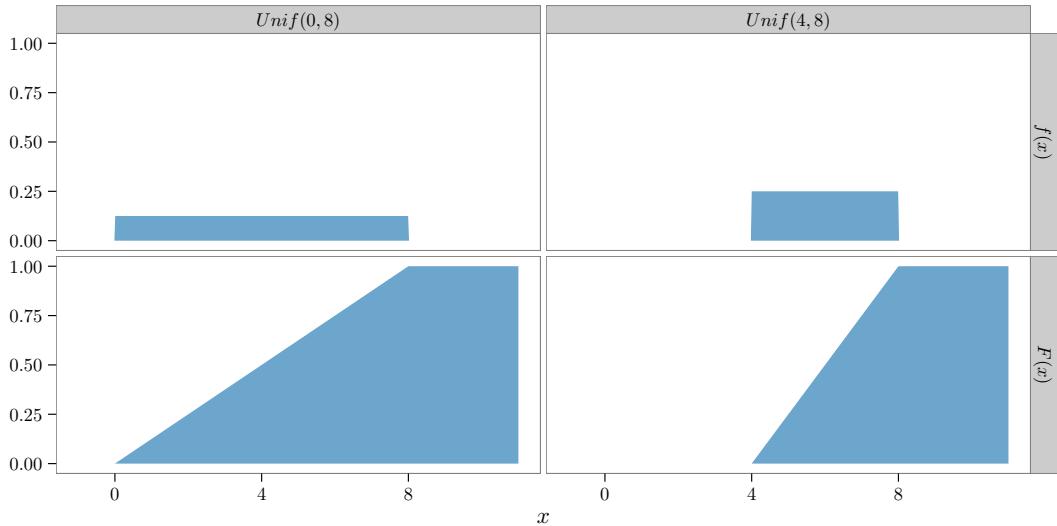


FIGURE 4.13: The **pdfs** and **cdfs** for a $Unif(0, 8)$ and a $Unif(4, 8)$

Example 4.12 Given a continuous random variable X defined over $[a, b]$ with **pdf** $f(x|a, b) = \frac{1}{b-a}$, $a \leq x \leq b$, find the expected value and the variance of X .

Solution: Using the definition for a continuous random variable from (3.15), write

$$\begin{aligned} E[X] &= \int_a^b x \cdot f(x) dx = \int_a^b \frac{x}{b-a} dx = \frac{1}{b-a} \cdot \frac{x^2}{2} \Big|_a^b \\ &= \frac{b^2 - a^2}{2(b-a)} = \frac{(b+a)(b-a)}{2(b-a)} = \frac{b+a}{2}. \end{aligned}$$

Next find $E[X^2]$ to use in computing the variance since $Var[X] = E[(X - \mu)^2] = E[X^2] - (E[X])^2$:

$$E[X^2] = \int_a^b x^2 \cdot \frac{1}{b-a} dx = \frac{1}{b-a} \cdot \frac{x^3}{3} \Big|_a^b = \frac{b^3 - a^3}{3(b-a)}$$

$$\begin{aligned} Var[X] &= E[X^2] - (E[X])^2 = \frac{b^3 - a^3}{3(b-a)} - \frac{(b+a)^2}{4} \\ &= \frac{(b-a)(b^2 + ab + a^2)}{3(b-a)} - \frac{(b+a)^2}{4} = \frac{4(b^2 + ab + a^2)}{12} - \frac{3(b+a)^2}{12} \\ &= \frac{4b^2 + 4ab + 4a^2 - (3b^2 + 6ab + 3a^2)}{12} = \frac{b^2 - 2ab + a^2}{12} \\ &= \frac{(b-a)^2}{12}. \end{aligned}$$

Example 4.13 If aerosol particles produced over forested areas have uniformly distributed diameters between 3 and 5 nanometers, compute the average volume of aerosol particles found over forested areas.

Solution: Recall that the volume of a sphere is $\frac{4}{3}\pi r^3$, or expressed in terms of the diameter, $\frac{1}{6}\pi d^3$. Consequently,

$$E\left[\frac{1}{6}\pi d^3\right] = \frac{1}{6}\pi E[d^3] \quad (4.10)$$

needs to be found. Let d represent the diameter of aerosol particles produced over forested areas. Since $d \sim \text{Unif}(3, 5)$,

$$\begin{aligned} E[d^3] &= \int_3^5 \frac{1}{5-3} \cdot x^3 dx = \frac{1}{2} \cdot \frac{x^4}{4} \Big|_3^5 \\ &= \frac{(5)^4}{8} - \frac{(3)^4}{8} = 68. \end{aligned}$$

Using the right side of (4.10), compute the average volume of aerosol particles to be

$$\frac{\pi}{6} \cdot 68 = 35.60472 \text{ nanometers}^3.$$

Estimate $E[d^3]$, denoted by $\widehat{E}[d^3]$, by cubing a large number of values drawn at random from a $\text{Unif}(3, 5)$ distribution and subsequently computing the mean of the cubed values. Then, the estimated mean volume of aerosol particles is computed by substituting $\widehat{E}[d^3]$ for $E[d^3]$ in the right-hand side of (4.10). R Code 4.12 estimates the mean volume of aerosol particles by simulating a sample of size 10,000 from a $\text{Unif}(3, 5)$ distribution:

R Code 4.12

```
> set.seed(13)
> MV <- (pi/6) * mean(runif(10000, 3, 5)^3)
> MV

[1] 35.69406
```

The solution reached with simulation is within 0.09 nanometers³ of the theoretical solution.

Generating Pseudo-Random Numbers The generation of pseudo-random numbers is fundamental to any simulation study. The term “pseudo-random” is used because once one value in such a simulation is known, the next values can be determined without fail, since they are generated by an algorithm. Most major statistical software systems have reputable pseudo-random number generators. When using R, the user can specify one of several different random number generators, including a user-supplied random number generator. For more details, type `?RNG` at the R prompt. Generation of random values from named distributions is accomplished with the R command `rdist`, where `dist` is the distribution name; however, it is helpful to understand some of the basic ideas of random number generation in the event a simulation does not involve a named distribution. When the user wants to generate a sample from a continuous random variable X with cdf F , one approach is to use the *Inverse Transformation Method*. This method simply sets $F_X(X) = U \sim \text{Unif}(0, 1)$ and solves for X , assuming $F_X^{-1}(U)$ actually exists.

Example 4.14 Generate a sample of 1000 random values from a continuous distribution with $\text{pdf } f(x) = \frac{4}{3}x(2 - x^2)$, $0 \leq x \leq 1$. Verify that the mean and variance of the 10,000 random values are approximately equal to the mean and variance of the given **pdf**.

Solution: First, the **cdf** is found. Then, $F_X(x)$ is set equal to u and solved.

$$F_X(x) = \int_0^x \frac{4}{3}t(2 - t^2) dt = \frac{4}{3} \left(x^2 - \frac{x^4}{4} \right) = \frac{1}{3}x^2(4 - x^2), \quad 0 \leq x \leq 1$$

Solving for x in terms of u by setting $u = F_X(x)$:

$$\begin{aligned} u &= \frac{1}{3}x^2(4 - x^2) \\ 3u &= 4x^2 - x^4 && \text{multiply by 3 and distribute } x^2 \\ -3u + 4 &= x^4 - 4x^2 + 4 && \text{multiply by } -1 \text{ and add 4 to complete the square} \\ -3u + 4 &= (x^2 - 2)^2 && \text{factor} \\ \pm\sqrt{-3u + 4} &= x^2 - 2 && \text{take the square root of both sides} \\ 2 \pm \sqrt{-3u + 4} &= x^2 && \text{add 2} \\ \pm\sqrt{2 \pm \sqrt{-3u + 4}} &= x && \text{take the square root of both sides,} \end{aligned}$$

which gives four solutions for x . The only one that is viable is $x = \sqrt{2 - \sqrt{4 - 3u}}$ because $0 \leq x \leq 1$. Provided $U \sim \text{Unif}(0, 1)$, $F_X^{-1}(U) = \sqrt{2 - \sqrt{4 - 3U}}$.

The theoretical mean and variance of X are calculated as

$$\begin{aligned} \mu_X &= E[X] = \int_0^1 x \cdot \frac{4}{3}x(2 - x^2) dx = \frac{84}{135} = 0.6222 \\ E[X^2] &= \int_0^1 x^2 \cdot \frac{4}{3}x(2 - x^2) dx = \frac{4}{9} = 0.4444 \\ \sigma_X^2 &= E[X^2] - E[X]^2 = \frac{4}{9} - \left(\frac{84}{135} \right)^2 = \frac{116}{2025} = 0.0573. \end{aligned}$$

R Code 4.13 computes the simulated mean and variance or 10,000 simulated values after setting the seed with `set.seed(33)`.

R Code 4.13

```
> set.seed(33)
> U <- runif(10000)
> X <- sqrt((2 - sqrt(4 - 3 * U)))
> SM <- mean(X)
> SV <- var(X)
> TM <- 84/135
> TV <- 116/2025
> PE <- c(PercentErrorMean = abs(SM - TM)/TM*100,
+        PercentErrorVariance = abs(SV - TV)/TV*100)
> ANS <- c(SimMean = SM, TheMean = TM, SimVar = SV, TheoVar = TV)
> ANS
```

SimMean	TheMean	SimVar	TheoVar
0.62364649	0.62222222	0.05664695	0.05728395

> PE

PercentErrorMean	PercentErrorVariance
0.2289005	1.1120003

The mean and variance of the 10,000 simulated random values using `set.seed(33)` are 0.6236 and 0.0566, respectively, which are both within 2% of their theoretical values. R Code 4.14 computes the theoretical mean and variance using numerical integration.

R Code 4.14

```
> f <- function(x){(4/3)*x*(2 - x^2)}
> ex <- function(x){x*f(x)}
> ex2 <- function(x){x^2*f(x)}
> EX <- integrate(ex, 0, 1)
> EX2 <- integrate(ex2, 0, 1)
> VX <- EX2$value - EX$value^2
> c(EX$value, EX2$value, VX)

[1] 0.62222222 0.44444444 0.05728395
```



4.3.2 Exponential Distribution

When observing a Poisson process such as that in Example 4.4 on page 258, where the number of outcomes in a fixed interval such as the number of goals scored during 90 minutes of World Cup soccer is counted, the random variable X , which measures the number of outcomes (number of goals), is modeled with the Poisson distribution. Additionally, the waiting time between successive outcomes is a random variable. If W is the waiting time until the first outcome of a Poisson process with mean $\lambda > 0$, then the **pdf** for W is

$$f(w) = \begin{cases} \lambda e^{-\lambda w} & \text{if } w \geq 0 \text{ and} \\ 0 & \text{if } w < 0. \end{cases}$$

Proof: Since waiting time is non-negative, $F(w) = 0$ for $w < 0$. When $w \geq 0$,

$$\begin{aligned} F(w) &= \mathbb{P}(W \leq w) = 1 - \mathbb{P}(W > w) \\ &= 1 - \mathbb{P}(\text{no outcomes in } [0, w]) \\ &= 1 - \frac{(\lambda w)^0 e^{-\lambda w}}{0!} \\ &= 1 - e^{-\lambda w}. \end{aligned}$$

Consequently, when $w > 0$, the **pdf** of W is $F'(w) = f(w) = \lambda e^{-\lambda w}$.

The exponential distribution is characterized by a lack of memory property and is often used to model lifetimes of electronic components as well as waiting times for Poisson processes. A random variable is said to be **memoryless** if

$$\mathbb{P}(X > t_2 + t_1 | X > t_1) = \mathbb{P}(X > t_2) \text{ for all } t_1, t_2 \geq 0. \quad (4.11)$$

The **pdf**, mean, variance, and **mgf** for an exponential random variable are shown in (4.12), while the **pdf** and **cdf** for an exponential random variable are illustrated in Figure 4.14. The **cdf**, $F(x)$, for the exponential distribution is written

$$F(x) = \mathbb{P}(X \leq x) = \int_0^x \lambda e^{-\lambda t} dt = -e^{-\lambda t} \Big|_0^x = 1 - e^{-\lambda x}, \quad x \geq 0.$$

Exponential Distribution
 $X \sim \text{Exp}(\lambda)$

$$f(x | \lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

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

$$\text{Var}[X] = \frac{1}{\lambda^2}$$

$$M_X(t) = (1 - \lambda^{-1}t)^{-1} \text{ for } t < \lambda$$

(4.12)

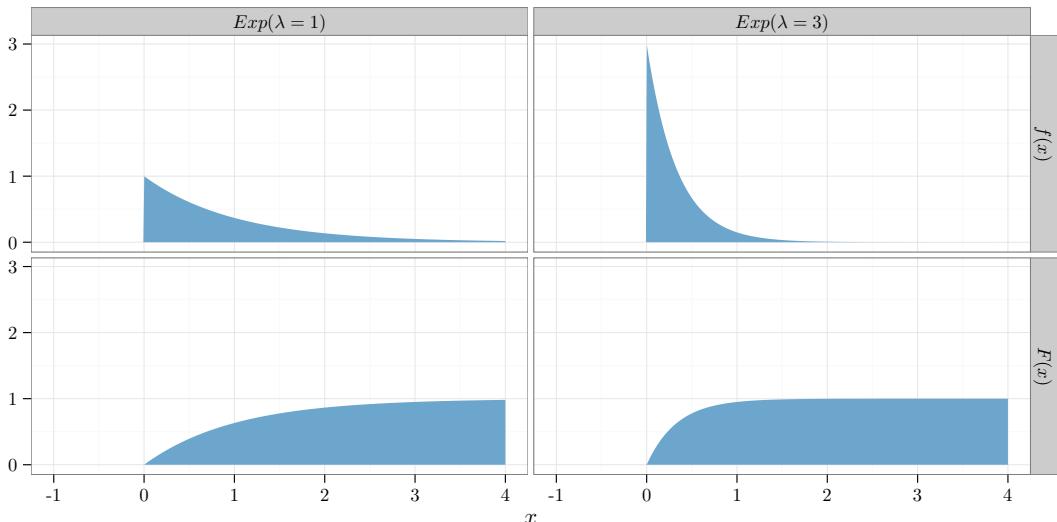


FIGURE 4.14: The **pdfs** and **cdfs** for an $\text{Exp}(\lambda = 1)$ and an $\text{Exp}(\lambda = 3)$

Example 4.15 Show that the function $f(x | \lambda)$ in (4.12) satisfies Condition 2 on page 222 from the properties of all **pdfs**.

Solution: To satisfy Condition 2 on page 222, it must be shown that the integral from

$-\infty$ to $+\infty$ of the function $f(x)$ given in (4.12) is 1:

$$\begin{aligned} \int_{-\infty}^{\infty} \lambda e^{-\lambda x} dx &= \int_{-\infty}^0 0 dx + \int_0^{\infty} \lambda e^{-\lambda x} dx \\ &= \int_0^{\infty} \lambda e^{-\lambda x} dx = -e^{-\lambda x} \Big|_0^{\infty} = 0 - (-1) = 1. \end{aligned}$$



Example 4.16 Given $X \sim \text{Exp}(\lambda)$, find the mean and variance of X .

Solution: Using (3.15), write

$$E[X] = \int_0^{\infty} x \lambda e^{-\lambda x} dx.$$

Integrating by parts where $u = x$ and $dv = \lambda e^{-\lambda x} dx$, obtain

$$\begin{aligned} E[X] &= -xe^{-\lambda x} \Big|_0^{\infty} - \int_0^{\infty} -e^{-\lambda x} dx \\ &= 0 - \frac{1}{\lambda e^{\lambda x}} \Big|_0^{\infty} = \frac{1}{\lambda}. \end{aligned}$$

Before finding the variance of X , find $E[X^2]$ using (3.16) as follows:

$$E[X^2] = \int_0^{\infty} x^2 \lambda e^{-\lambda x} dx. \quad (4.13)$$

Note that $E[X] = \int_0^{\infty} x \lambda e^{-\lambda x} dx \Rightarrow \frac{E[X]}{\lambda} = \int_0^{\infty} x e^{-\lambda x} dx$ and integrate (4.13) by parts where $u = x^2$ and $dv = \lambda e^{-\lambda x} dx$:

$$\begin{aligned} E[X^2] &= -x^2 e^{-\lambda x} \Big|_0^{\infty} - \int_0^{\infty} -2xe^{-\lambda x} dx \\ &= 0 + 2 \frac{E[X]}{\lambda} = \frac{2}{\lambda^2}. \end{aligned}$$

Using the fact that $\text{Var}[X] = E[X^2] - (E[X])^2$, obtain $\text{Var}[X] = \frac{2}{\lambda^2} - \left(\frac{1}{\lambda}\right)^2 = \frac{1}{\lambda^2}$.



Based on the results from Example 4.16, note that the mean and standard deviation of the exponential random variable are identical. Quite often, the **pdf** for the exponential is expressed as

$$f(x) = \frac{1}{\theta} e^{-x/\theta}, \quad x \geq 0, \quad \theta > 0,$$

where $\theta = \frac{1}{\lambda}$. Of course, the **mgf** is then written as $M_X(t) = (1 - \theta t)^{-1}$ and the reparameterized mean and variance are θ and θ^2 , respectively. Note the relationship between the

Poisson mean and the exponential mean. Given a Poisson process with mean λ , the waiting time until the first outcome has an exponential distribution with mean $\frac{1}{\lambda}$. That is, if λ represents the number of outcomes in a unit interval, $\frac{1}{\lambda}$ is the mean waiting time for the first change. If X denotes the lifetime of an electronic component following an exponential distribution with mean $\frac{1}{\lambda}$, (4.11) implies that the probability the component will work for $t_2 + t_1$ hours given that it has worked for t_1 hours is the same as the probability that the component will function for at least t_2 hours. In other words, the component has no memory of having functioned for t_1 hours. Note that (4.11) is equivalent to

$$\frac{\mathbb{P}((X > t_2 + t_1) \cap (X > t_1))}{\mathbb{P}(X > t_1)} = \mathbb{P}(X > t_2),$$

which is equivalent to

$$\mathbb{P}(X > t_2 + t_1) = \mathbb{P}(X > t_2)\mathbb{P}(X > t_1). \quad (4.14)$$

Since $\mathbb{P}(X > t_2 + t_1) = 1 - F(t_2 + t_1) = e^{-\lambda(t_2+t_1)} = e^{-\lambda t_2}e^{-\lambda t_1} = \mathbb{P}(X > t_2)\mathbb{P}(X > t_1)$ for any exponential random variable, exponential random variables are memoryless according to (4.14).

Example 4.17 \triangleright **Exponential Distribution: Light Bulbs** \triangleleft If the life of a certain type of light bulb has an exponential distribution with a mean of 8 months, find

- (a) The probability that a randomly selected light bulb lasts between 3 and 12 months.
- (b) The 95th percentile of the distribution.
- (c) The probability that a light bulb that has lasted for 10 months will last more than 25 months.

Solution: The answers are as follows:

- (a) Since $X \sim \text{Exp}(\lambda = \frac{1}{8})$, the probability that a randomly selected light bulb lasts between 3 and 12 months is

$$\mathbb{P}(3 < X < 12) = \int_{3}^{12} \frac{1}{8} e^{-x/8} dx = -e^{-x/8} \Big|_3^{12} = -0.2231 + 0.6873 = 0.4642.$$

The following code solves the problem with R:

```
> pexp(12, 1/8) - pexp(3, 1/8)
[1] 0.4641591
```

The function `integrate()` can also be used to solve this problem using numerical integration:

```
> f1 <- function(x){(1/8) * exp(-x/8)}           # define f1
> integrate(f1, lower = 3, upper = 12)$value      # integrate f1
[1] 0.4641591
```

(b) The 95th percentile is the value x_{95} such that

$$\begin{aligned} \int_{-\infty}^{x_{95}} f(x) dx &= \int_0^{x_{95}} \frac{1}{8} e^{-x/8} dx = \frac{95}{100} \\ -e^{-x/8} \Big|_0^{x_{95}} &= 1 - e^{-\frac{x_{95}}{8}} = \frac{95}{100} \\ e^{-\frac{x_{95}}{8}} &= \frac{5}{100} \\ x_{95} &= -8 \ln(0.05) = 23.9659. \end{aligned}$$

To find the answer with R, type

```
> qexp(0.95, 1/8)
[1] 23.96586
```

(c) The probability that a light bulb that has lasted for 10 months will last more than 25 months mathematically is written $\mathbb{P}(X > 25 | X > 10) = \mathbb{P}(X > 25)/\mathbb{P}(X > 10)$. Because an exponential distribution is present, (4.11) can be used to say that this is equal to $\mathbb{P}(X > 15) = e^{-15/8} = 0.1534$.

Solve the problem with R as follows:

```
> pexp(25, 1/8, lower = FALSE)/pexp(10, 1/8, lower = FALSE)
[1] 0.153355

> # OR
> 1 - pexp(15, 1/8)
[1] 0.153355

> # OR
> pexp(15, 1/8, lower = FALSE)
[1] 0.153355
```



Example 4.18 ▷ Exponential Distribution: Inter-goal Times ◁ Example 4.4 on page 258 illustrated how the number of goals scored during World Cup games could be modeled with the Poisson distribution. Now, look at the distribution of T , the time between goals. In Example 4.4 on page 258, λ was estimated to be $\frac{575}{232}$. Since one soccer match lasts 90 minutes, the average time (in minutes) before a goal is scored is $\frac{90}{\lambda} = 36.313$ minutes assuming λ is $\frac{575}{232}$. To find the inter-goal times from the cumulative goal times stored in column `cgt` of the `SOCCKER` data frame, compute $cgt_{i+1} - cgt_i$.

- (a) Compute the mean and standard deviation for the time between goals.
- (b) Is it reasonable to model the time between goals with the exponential distribution?
- (c) In particular, is the lack of memory property evident in the data?

Solution: The answers are as follows:

(a) First, attach **Soccer** so that columns can be referenced by their names. Then, use R to calculate both the mean and standard deviation for the time between goals:

```
> inter.times <- with(data = SOCCER, cgt[2:575] - cgt[1:574])
> MEAN <- mean(inter.times)
> SD <- sd(inter.times)
> c(MEAN = MEAN, SD = SD)

MEAN      SD
36.24042 36.67138
```

Note that both the mean (36.2404 minutes) and standard deviation (36.6714 minutes) for time between goals are close to the theoretical time of 36.313 minutes under the assumption that λ is $\frac{575}{232}$.

(b) To assess the fit of the data to an exponential distribution with a mean of 36.313 minutes, first split the data into discrete categories. If the underlying distribution is exponential, then a good bin width is approximately $(\frac{12}{n})^{1/3} \cdot \mu_X$ (Scott, 1992). In our case, the bin width is $(\frac{12}{574})^{1/3} \cdot 36.313 \approx 10$.

```
> rate <- (575/232)*(1/90)          # rate = lambda*t
> nit <- sum(!is.na(inter.times))    # number of inter.times
> OBS <- xtabs(~cut(inter.times, breaks = c(seq(0, 130, 10), 310)))
> EmpiP <- round(OBS/nit, 3)
> TheoP <- round(c((pexp(seq(10, 130, 10), rate) -
+                    pexp(seq(0, 120, 10), rate)),
+                    (1 - pexp(130, rate))), 3)
> EXP <- round(TheoP*nit, 0)
> ANS <- cbind(OBS, EXP, EmpiP, TheoP)
> ANS

      OBS EXP EmpiP TheoP
(0,10] 144 138 0.251 0.241
(10,20] 106 105 0.185 0.183
(20,30] 86 80 0.150 0.139
(30,40] 53 60 0.092 0.105
(40,50] 45 46 0.078 0.080
(50,60] 27 35 0.047 0.061
(60,70] 35 26 0.061 0.046
(70,80] 16 20 0.028 0.035
(80,90] 22 15 0.038 0.027
(90,100] 12 11 0.021 0.020
(100,110] 3 9 0.005 0.015
(110,120] 3 7 0.005 0.012
(120,130] 6 5 0.010 0.009
(130,310] 16 16 0.028 0.028
```

The observed and expected values as well as the empirical and theoretical probabilities are similar.

(c) The lack of memory property is also evident from the data. Empirically, $\mathbb{P}(T > 10) = 1 - \mathbb{P}(T \leq 10) = 1 - \frac{144}{574} = \frac{430}{574} = 0.749$, and $\mathbb{P}(T > 20 | T > 10) = \frac{574-144-106}{574-144} = 0.754$,

which are both roughly the same and similar to the theoretical $\mathbb{P}(T > 10)$, which is 0.7593 under the assumption that the mean is 36.313 minutes. Since the observed data appear to lack memory, the same probability statements could be used to justify independence among the times between goals using (4.14). Finally, the inter-goal times are shown in Figure 4.15 with a superimposed density for an exponential with a mean of 36.313 minutes. R Code 4.15 was used to create Figure 4.15. Based on the analysis and Figure 4.15, it seems reasonable to model the time between goals scored in World Cup competition for the years 1990 to 2002 with an exponential distribution.

R Code 4.15

```
> DF <- data.frame(x = inter.times)
> previous_theme <- theme_set(theme_bw()) # set black and white theme
> p <- ggplot(data = DF, aes(x = x)) +
+   geom_histogram(aes(y = ..density..), fill = "pink", binwidth = 10,
+                 color = "black") +
+   labs(x = "Time Between Goals (minutes)", y = "")
> p + stat_function(fun = dexp, arg = list(rate = rate), size = 1)
> theme_set(previous_theme) # Restore original theme
```

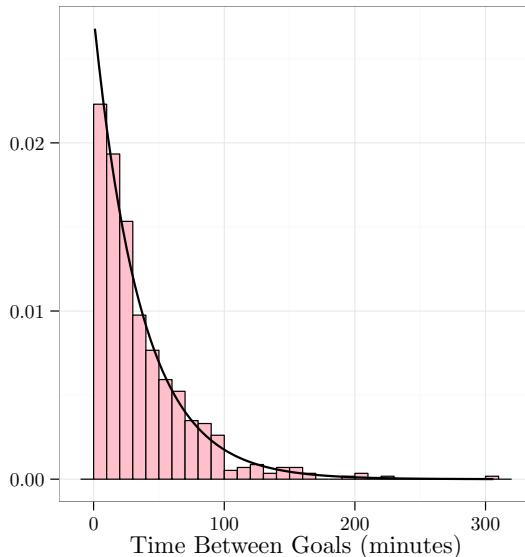


FIGURE 4.15: Histogram of time between goals with superimposed exponential density curve with mean of 36.31 minutes

4.3.3 Gamma Distribution

Some random variables are always non-negative and yield distributions of data that tend to be skewed. The waiting time until a certain number of malfunctions in jet engines, the waiting time until a certain number of accidents at a given intersection, and similar

scenarios where the random variable of interest is the waiting time until a certain number of events takes place yield skewed distributions. The **gamma** distribution is often used to model the waiting time until the α^{th} event in a Poisson process. Before defining the gamma distribution, review the definition of the gamma function from mathematics. The **gamma function**, $\Gamma(\alpha)$, is defined by:

$$\boxed{\Gamma(\alpha) = \int_0^\infty x^{\alpha-1} e^{-x} dx, \quad \alpha > 0} \quad (4.15)$$

Some of the more important properties of the gamma function include:

1. For $\alpha > 0$, $\Gamma(\alpha + 1) = \alpha\Gamma(\alpha)$.
2. For any positive integer, n , $\Gamma(n) = (n - 1)!$
3. $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$.

In Section 4.3.2 on page 276, it was proved that the waiting time until the first outcome in a Poisson process follows an exponential distribution. Now, let W denote the waiting time until the α^{th} outcome and derive the distribution of W in a similar fashion. Since waiting time is non-negative, $F(w) = 0$ for $w < 0$. When $w \geq 0$,

$$\begin{aligned} F(w) &= \mathbb{P}(W \leq w) = 1 - \mathbb{P}(W > w) \\ &= 1 - \mathbb{P}(\text{fewer than } \alpha \text{ outcomes in } [0, w]) \\ &= 1 - \sum_{k=0}^{\alpha-1} \frac{(\lambda w)^k e^{-\lambda w}}{k!}. \end{aligned}$$

Consequently, when $w > 0$, the **pdf** of W is $F'(w) = f(w)$ whenever this derivative exists. It follows then that

$$\begin{aligned} f(w) &= F'(w) = - \sum_{k=0}^{\alpha-1} \frac{(\lambda w)^k e^{-\lambda w} (-\lambda) + e^{-\lambda w} k (\lambda w)^{k-1} \lambda}{k!} \\ &= -e^{-\lambda w} \sum_{k=0}^{\alpha-1} \frac{k \lambda (\lambda w)^{k-1} - \lambda (\lambda w)^k}{k!} \\ &= \lambda e^{-\lambda w} - e^{-\lambda w} \sum_{k=1}^{\alpha-1} \frac{k \lambda (\lambda w)^{k-1} - \lambda (\lambda w)^k}{k!} \\ &= \lambda e^{-\lambda w} - e^{-\lambda w} \sum_{k=1}^{\alpha-1} \left[\frac{\lambda (\lambda w)^{k-1}}{(k-1)!} - \frac{\lambda (\lambda w)^k}{k!} \right] \\ &= \lambda e^{-\lambda w} - e^{-\lambda w} \left[\frac{\lambda (\lambda w)^0}{0!} - \frac{\lambda (\lambda w)^1}{1!} + \frac{\lambda (\lambda w)^1}{1!} - \frac{\lambda (\lambda w)^2}{2!} + \right. \\ &\quad \left. \dots - \frac{\lambda (\lambda w)^{\alpha-1}}{(\alpha-1)!} \right] \\ &= \lambda e^{-\lambda w} - e^{-\lambda w} \left[\lambda - \frac{\lambda (\lambda w)^{\alpha-1}}{(\alpha-1)!} \right] \\ &= \frac{\lambda (\lambda w)^{\alpha-1} e^{-\lambda w}}{(\alpha-1)!} = \frac{\lambda^\alpha w^{\alpha-1} e^{-\lambda w}}{\Gamma(\alpha)}. \end{aligned}$$

From the previous derivation, note that the gamma is a generalization of the exponential distribution. The **pdf**, mean, variance, and **mgf** for a gamma random variable are listed in (4.16). The **pdfs** for $\lambda = 2$ and $\lambda = 1$ with $\alpha = 1, 2$, and 3 , respectively, are illustrated in Figure 4.16. Notice that different shapes are produced in Figure 4.16 for different values of α . For this reason, α is often called the shape parameter associated with the gamma distribution. The parameter λ is referred to as the inverse scale or rate parameter. Varying λ changes the units of measurement (say, from seconds to minutes) and does not affect the shape of the density.

Gamma Distribution
 $X \sim \Gamma(\alpha, \lambda)$

$$f(x | \alpha, \lambda) = \begin{cases} \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (4.16)$$

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

$$Var[X] = \frac{\alpha}{\lambda^2}$$

$$M_X(t) = (1 - \lambda^{-1}t)^{-\alpha} \text{ for } t < \lambda$$

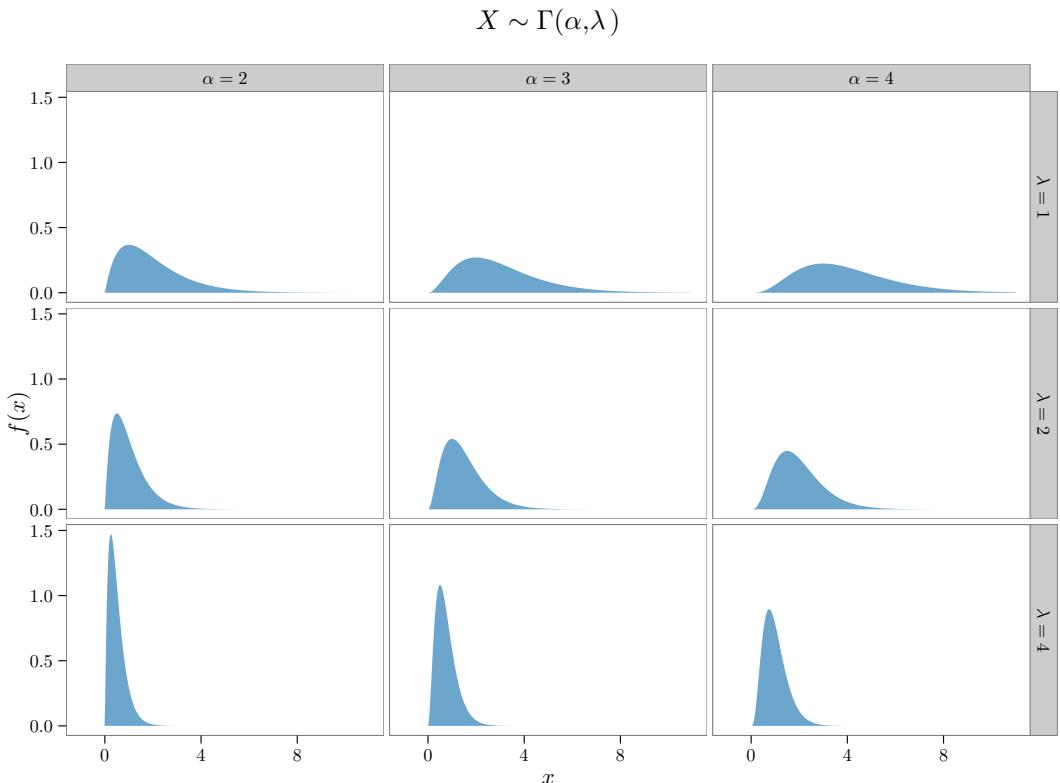


FIGURE 4.16: Graphical illustration of the **pdfs** of $\Gamma(\alpha, \lambda)$ for $\alpha = 2, 3$, and 4 , and $\lambda = 1, 2$, and 4

Useful Relationships

1. Given $X \sim \Gamma(\alpha, \lambda)$. When $\alpha = 1$, the resulting random variable is $X \sim Exp(\lambda)$. That is, the exponential distribution is a special case of the gamma distribution.
2. Given $X \sim \Gamma(\alpha, \lambda)$. When $\alpha = n/2$ and $\lambda = 1/2$, the resulting random variable has a chi-square distribution with n degrees of freedom. (The chi-square is discussed in Section 6.6.1.)
3. Given $X \sim \Gamma(\alpha, \lambda)$. Provided α is a positive integer, the resulting distribution is known as the Erlang. In this case, the Erlang distribution gives the waiting time until the α^{th} occurrence when the number of outcomes in an interval of length t follows a Poisson distribution with parameter λt .
4. Given $X \sim \Gamma(\alpha, \lambda)$. The sampling distribution of \bar{X} for a random sample of size n is $\bar{X} \sim \Gamma(n\alpha, n\lambda)$.

Example 4.19 Given $X \sim \Gamma(\alpha, \lambda)$, find the mean and variance of X .

Solution: Using the **mgf** from (4.16), it is known that the first and second derivatives of the **mgf** evaluated at zero, respectively, yield the $E[X]$ and the $E[X^2]$. Consequently,

$$\begin{aligned} E[X] &= M'_X(t) \Big|_{t=0} \\ &= (-\alpha)(1 - \lambda^{-1}t)^{-\alpha-1}(-\lambda^{-1}) \Big|_{t=0} = \frac{\alpha}{\lambda}, \\ E[X^2] &= M''_X(t) \Big|_{t=0} \\ &= \alpha\lambda^{-1}(-\alpha-1)(1 - \lambda^{-1}t)^{-\alpha-2}(-\lambda^{-1}) \Big|_{t=0} = \frac{\alpha(\alpha+1)}{\lambda^2}, \text{ and} \\ Var[X] &= E[X^2] - (E[X])^2 \\ &= \frac{\alpha(\alpha+1)}{\lambda^2} - \left(\frac{\alpha}{\lambda}\right)^2 = \frac{\alpha}{\lambda^2}. \end{aligned}$$

So the mean of X is $\frac{\alpha}{\lambda}$ and the variance of X is $\frac{\alpha}{\lambda^2}$. ■

Example 4.20 Suppose that the average arrival rate at a local fast food drive-through window is three cars per minute ($\lambda = 3$). Find

- (a) The probability that at least five cars arrive in 120 seconds.
- (b) The probability that more than one minute elapses before the second car arrives.
- (c) If one car has already gone through the drive-through, what is the average waiting time before the third car arrives?

Solution: The answers are as follows:

- (a) If the average number of car arrivals follows a Poisson distribution with a rate of three cars per minute, then the average rate of arrival for 2 minutes is six cars. Given that $X \sim Pois(\lambda = 6)$,

$$\mathbb{P}(X \geq 5) = 1 - \mathbb{P}(X \leq 4) = 1 - \sum_{x=0}^4 \frac{e^{-6} 6^x}{x!} = 1 - 0.2851 = 0.7149.$$

To solve the problem with R, use the command `ppois()`:

```
> 1 - ppois(4, 6)
[1] 0.7149435
> # OR
> ppois(4, 6, lower = FALSE)
[1] 0.7149435
```

(b) Let W represent the waiting time until the α^{th} outcome. It follows that $W \sim \Gamma(\alpha = 2, \lambda = 3)$. Consequently,

$$\begin{aligned}\mathbb{P}(W > 1) &= 1 - \mathbb{P}(W \leq 1) = 1 - \mathbb{P}(\Gamma(2, 3) \leq 1) = 1 - \int_0^1 \frac{3^2}{\Gamma(2)} x^{2-1} e^{-3x} dx \\ &= 1 - \int_0^1 3x e^{-3x} 3 dx.\end{aligned}$$

Using integration by parts where $u = 3x$ and $dv = 3e^{-3x}dx$,

$$\begin{aligned}\int_0^1 3x e^{-3x} 3 dx &= -3xe^{-3x} \Big|_0^1 + \int_0^1 3e^{-3x} dx \\ &= -3e^{-3} + \left[-e^{-3x} \Big|_0^1 \right] = -3e^{-3} + \left[-e^{-3} + 1 \right] \\ &= 1 - 4e^{-3} = 0.8009.\end{aligned}$$

In other words, $\mathbb{P}(W > 1) = 1 - 0.8009 = 0.1991$. To solve the problem with R, use the command `pgamma()` or `integrate()`:

```
> 1 - pgamma(1, 2, 3)
[1] 0.1991483
> gam23 <- function(x){9*x*exp(-3*x)}
> integrate(gam23, 1, Inf)$value
[1] 0.1991483
```

(c) This problem is really asking for the mean of $a\Gamma(\alpha = 2, \lambda = 3)$ random variable. Note: $\alpha = 2$ since one car has already arrived and the problem requests the average waiting time until the third car arrives. Therefore, $E[X] = \frac{\alpha}{\lambda} = \frac{2}{3}$. In other words, there is an average wait of $\frac{2}{3}$ of a minute before the arrival of the third vehicle given one vehicle has already arrived.

4.3.4 Hazard Function, Reliability Function, and Failure Rate

In addition to studying the `pdf` of continuous random variables, at times it is helpful to study other functions related to the `pdf` such as the **reliability function** or the **hazard function**, which is also often called the **failure rate** or **force of mortality**, especially when dealing with lifetime data. Suppose the random variable T represents the useful life

of some component with **pdf** and **cdf** given by $f(t)$ and $F(t)$, respectively. The reliability function $R(t)$ is defined as

$$R(t) = \mathbb{P}(T > t) = 1 - F(t), \quad t > 0 \quad (4.17)$$

and represents the probability that the lifetime of the component exceeds t . The hazard function, $h(t)$, is defined as

$$h(t) = \frac{f(t)}{1 - F(t)} = \frac{f(t)}{R(t)}, \quad t > 0, \quad F(t) < 1. \quad (4.18)$$

Note that the hazard function is often called the conditional failure rate.

The functions $h(t)$, $f(t)$, and $F(t)$ provide mathematically equivalent specifications of the distribution of T . In fact, it can be shown that

$$f(t) = h(t)e^{-\int_0^t h(x) dx}. \quad (4.19)$$

To gain an intuitive understanding of what $h(t)$ is measuring, let dt represent a small unit of measurement. Then, the quantity $h(t)dt$ can be thought of as the approximate probability that T takes on a value in $(t, t + dt)$. Keeping in mind that $1 - F(t) = \mathbb{P}(T > t)$, write

$$h(t)dt = \frac{f(t)dt}{1 - F(t)} \approx \mathbb{P}[T \in (t, t + dt) | T > t].$$

In other words, $h(t)dt$ represents the approximate probability of having a breakdown during the interval $(t, t + dt)$ given that a component has lasted up to time t . In mathematical terms,

$$\lim_{dt \rightarrow 0} \frac{\mathbb{P}(t \leq T \leq t + dt | T > t)}{dt} \quad (4.20)$$

may be written, which represents the instantaneous rate of death or failure at time t , given the individual or component has survived to time t . It may then be noted that the hazard function is a rate rather than a probability. The failure rate for an exponential random variable is a constant λ :

$$h(t) = \frac{f(t)}{1 - F(t)} = \frac{\lambda e^{-\lambda x}}{1 - [1 - e^{-\lambda x}]} = \lambda.$$

Not many components have a constant failure rate. As a matter of fact, it stands to reason that the failure rate should increase as the life of a component ages. For most manufactured items as well as human populations, this is the case after some initial time period. However, there are some instances such as breakdowns when equipment is on a preventative maintenance schedule where it is still reasonable to assume a constant failure rate. A very flexible hazard function is $h(t) = \frac{\alpha t^{\alpha-1}}{\beta^\alpha}$, for all α and β greater than 0, since the function is monotone increasing for $\alpha > 1$, monotone decreasing for $\alpha < 1$, and constant for $\alpha = 1$, as illustrated in Figure 4.17 on the following page. This hazard function corresponds to the Weibull distribution that is discussed in Section 4.3.5

Example 4.21 ▷ Hazard Rate ◁ In an effort to attract more business, a local computer outlet has agreed to replace its laser printers with a brand new laser printer in the event any of its laser printers malfunction within one year of the date of their purchase. According to the manufacturer of the printer, the useful life (in years) of the printer is a random variable T with **pdf** $f(t) = K(2000 - 0.1e^{-2t})$ for $0 < t < 5$ and 0 otherwise.

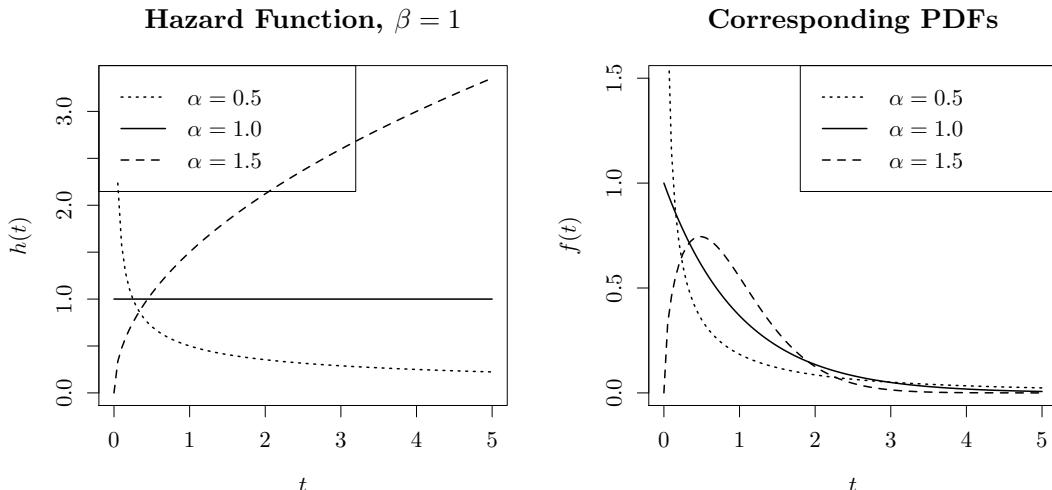


FIGURE 4.17: Illustration of the hazard function $h(t) = \frac{\alpha t^{\alpha-1}}{\beta^\alpha}$ for $\alpha = 0.5$, $\alpha = 1.0$, and $\alpha = 1.5$ with $\beta = 1$ and the corresponding **pdfs**

- Find K so that $f(t)$ is a **pdf**.
- Compute the probability a randomly selected laser printer will have to be replaced due to a malfunction.
- What are the mean and standard deviation for laser printer life?
- If a small business purchases five laser printers from the computer outlet, what is the probability there are no malfunctions during the first year?
- What should the length of guarantee time be for a laser printer if the outlet store wants to replace no more than 5% of the laser printers?
- Compute, graphically represent, and interpret the hazard function.

Solution: The answers are as follows:

- To find K such that $f(t)$ is a **pdf**, the integral over all possible values of t must be one:

$$\begin{aligned} \int_0^5 K(2000 - 0.1e^{-2t}) dt &\stackrel{\text{set}}{=} 1 \\ K \left[(2000t + 0.05e^{-2t}) \Big|_0^5 \right] &= 1 \\ K [10000 + 0.05e^{-10} - 0.05] &= 1 \\ K = \frac{1}{9999.95 + 0.05e^{-10}} &= 1e-04 \end{aligned}$$

The value of K is computed numerically in R Code 4.16.

R Code 4.16

```
> f <- function(x){2000 - 0.1*exp(-2*x)}
> K <- 1 / (integrate(f, lower = 0, upper = 5)$value)
> K
```

```
[1] 0.0001000005
```

$$(b) P(T < 1) = \int_0^1 K(2000 - 0.1e^{-2t}) dt = 0.2$$

The numerical answer computed with R using the `f` from part (a) is

```
> ansB <- K*integrate(f, lower = 0, upper = 1)$value
> ansB
```

```
[1] 0.1999967
```

$$(c) E(T) = \int_0^5 tK(2000 - 0.1e^{-2t}) dt = 2.5$$

```
> et <- function(x){x*K*f(x)}
> ET <- integrate(et, lower = 0, upper = 5)$value
> ET
```

```
[1] 2.50001
```

$$\sigma_T = \sqrt{\sigma_T^2} = \sqrt{E(T^2) - E(T)^2} = 1.4434$$

```
> et2 <- function(x) {
+   x^2 * K * f(x)
+ }
> ET2 <- integrate(et2, lower = 0, upper = 5)$value
> VX <- ET2 - ET^2
> SX <- sqrt(VX)
> SX
```

```
[1] 1.443372
```

The mean printer life is 2.5 years and the standard deviation for printer life is 1.4434 years.

(d) Assuming the useful lives of laser printers are independent, the probability none of the five printers have to be replaced is

$$P(T_1 > 1) \times P(T_2 > 1) \times \cdots \times P(T_5 > 1) = (1 - 0.2)^5 = 0.3277.$$

If the random variable X is defined to be the number of printers that need to be replaced during the first year of operation, then $X \sim Bin(n = 5, \pi = 0.2)$ and the problem is solved by computing $P(X = 0) = \binom{5}{0}(0.2)^0(1 - 0.2)^5 = 0.3277$. The problem is solved with R using the value for π computed from part (b), which was stored in the object `ansB`.

```
> ansB
```

```
[1] 0.1999967
```

```
> dbinom(0, 5, ansB)
```

```
[1] 0.3276868
```

(e) The length of guarantee time for a laser printer if the outlet store wants to replace no more than 5% of the laser printers will be the roots of the equation

$$\mathbb{P}(T < t) = \int_0^t K(2000 - 0.1e^{-2x}) dx = 0.05.$$

$$\begin{aligned} \int_0^t K(2000 - 0.1e^{-2x}) dx &= 0.05 \\ (2000x + 0.05e^{-2x})|_0^t &= 0.05/K \\ 2000t + 0.05e^{-2t} - 0.05 &= 0.05/K \\ \text{Find roots of } 2000t + 0.05e^{-2t} - 0.05 - 0.05/K &= 0. \end{aligned}$$

Use the function `uniroot()` to solve for t numerically:

```
> fr <- function(x){2000*x + 0.05*exp(-2*x) - 0.05 - 0.05/K}
> ansE <- uniroot(fr, c(0, 5))$root
> ansE
[1] 0.25
```

Since t is given in years, multiplying $0.25 \times 365 = 91.25$ days. In other words, the computer outlet will have to replace less than 5% of their laser printers if they use a guarantee period of 91 days.

(f) Note that the reliability (survival) function is

$$P(T > t) = 1 - F(t) = 1 - K(2000t + 0.05e^{-2t} - 0.05), \quad 0 < t < 5.$$

Using the reliability function, the hazard function can be written as

$$h(t) = \frac{f(t)}{1 - F(t)} = \frac{K(2000 - 0.1e^{-2t})}{1 - K(2000t + 0.05e^{-2t} - 0.05)}, \quad 0 < t < 5.$$

This particular hazard function (Figure 4.18 on the facing page) represents the instantaneous rate of failure given that a printer has lasted until time t . R Code 4.17 can be used to create a graph similar to Figure 4.18. After the function `h` is defined, `curve()` can be used to create the requested graph with base graphics. The last two lines of R Code 4.17 can be used to create the requested graph with `ggplot2`. Recall that the value of $K = 1e-04$ from part (a).

R Code 4.17

```
> h <- function(x) {
+   1e-04 * (2000 - 0.1 * exp(-2 * x))/(1 - 1e-04 * (2000 *
+     x + 0.05 * exp(-2 * x) - 0.05))
+ }
> # Base graphics
> curve(h, from = 0, to = 4.99, n = 1000, xlab = "year", ylab = "h(year)")
> # ggplot2 now
> p <- ggplot(data.frame(x = c(0, 4.99)), aes(x = x))
> p + stat_function(fun = h) + labs(x = "year", y = "h(year)")
```

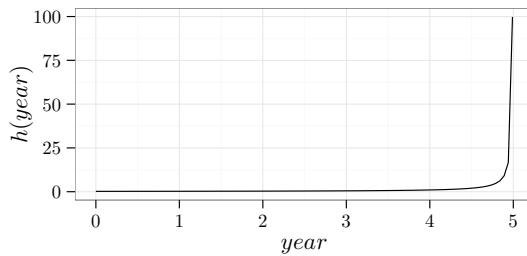


FIGURE 4.18: Hazard function for printer failure



4.3.5 Weibull Distribution

The gamma distribution provides an adequate model for some systems' lifetime distributions. However, since the hazard function for the gamma does not have a closed form expression, and its failure rate approaches λ from both above (when $\alpha < 1$) and below (when $\alpha > 1$) as t gets large, distributions with closed form expressions for the hazard function such as the Weibull tend to be favored by practitioners who deal with lifetime distributions. In particular, the hazard function for the Weibull distribution has a failure rate that varies with time. The hazard rate for the Weibull distribution is $h(t) = \frac{\alpha t^{\alpha-1}}{\beta^\alpha}$, for all α and β greater than 0. Using (4.19), derive the **pdf** for the Weibull distribution as follows:

$$f(t) = h(t)e^{-\int_0^t h(x) dx} = \frac{\alpha t^{\alpha-1}}{\beta^\alpha} e^{-\int_0^t \frac{\alpha x^{\alpha-1}}{\beta^\alpha} dx} = \frac{\alpha t^{\alpha-1}}{\beta^\alpha} e^{-(t/\beta)^\alpha}.$$

The **pdf**, mean, variance, and hazard function for a Weibull random variable ($\alpha > 0$ and $\beta > 0$) are shown in (4.21), while the **pdfs** for $Weib(\alpha, \beta)$ with $\alpha = 1, 2$, and 3 , and $\beta = 1, 2$, and 3 , are illustrated in Figure 4.19 on the next page. The first parameter in the Weibull distribution, α , is the shape parameter; and the second parameter, β , is the scale parameter. If $X \sim Weib(\alpha, \beta)$ and $\alpha = 1$, then $X \sim Exp(\lambda = 1/\beta)$.

Weibull Distribution
$X \sim Weib(\alpha, \beta)$
$f(x \alpha, \beta) = \begin{cases} \alpha \beta^{-\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (4.21)$ $E[X] = \beta \Gamma(1 + \alpha^{-1})$ $Var[X] = \beta^2 \left\{ \Gamma(1 + 2\alpha^{-1}) - [\Gamma(1 + \alpha^{-1})]^2 \right\}$ $h(x) = \alpha \beta^{-\alpha} x^{\alpha-1} \text{ for } x \geq 0$

Example 4.22 The useful life (in thousands of hours) of a certain type of transistor follows a Weibull distribution with $\alpha = 2$ and $\beta = 8$. Find the probability that a randomly selected transistor lasts more than 8000 hours. What is the average life for this type of transistor?

$$X \sim Weib(\alpha, \beta)$$

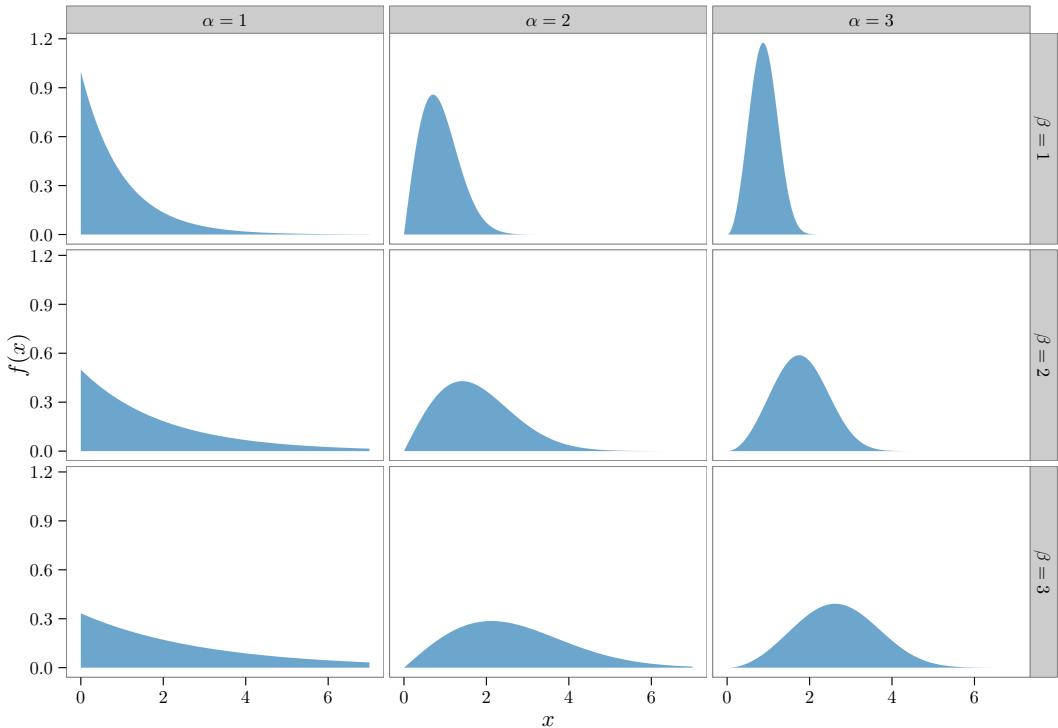


FIGURE 4.19: Graphical illustration of the **pdfs** of a $Weib(\alpha, \beta)$ for $\alpha = 1, 2, 3$, and $\beta = 1, 2, 3$,

Solution: First, find the **cdf** for $X \sim Weib(\alpha, \beta)$:

$$F(x) = \int_0^x \alpha \beta^{-\alpha} t^{\alpha-1} e^{-(t/\beta)^\alpha} dt = -e^{-(t/\beta)^\alpha} \Big|_0^x = 1 - e^{-(x/\beta)^\alpha}.$$

Using the **cdf** for the Weibull, the probability a randomly selected transistor lasts more than 8000 hours is

$$\mathbb{P}(X > 8) = 1 - F(8) = 1 - \left[1 - e^{-(8/\beta)^2}\right] = e^{-1} = 0.3679.$$

The expected value of X (in thousands of hours) is

$$E[X] = \beta \Gamma(1 + \alpha^{-1}) = 8 \Gamma\left(1 + \frac{1}{2}\right) = 8 \frac{1}{2} \Gamma\left(\frac{1}{2}\right) = 4\sqrt{\pi} = 7.0898.$$

To solve the first question and to compute $\Gamma\left(\frac{3}{2}\right)$ with R, use the functions `pweibull()` and `gamma()`, respectively:

```
> 1 - pweibull(8, 2, 8)
[1] 0.3678794
> # OR
> pweibull(8, 2, 8, lower = FALSE)
```

```
[1] 0.3678794
> 8 * gamma(3/2)
[1] 7.089815
> # OR
> 4 * sqrt(pi)
[1] 7.089815
```



4.3.6 Beta Distribution

The continuous distributions discussed up to this point, with the exception of the continuous uniform, have positive densities over unbounded intervals. To model phenomena restricted to a finite interval, another type of distribution is needed, such as the beta (β) distribution, whose density function is positive only over the interval (A, B) . The standard beta distribution, $(A = 0, B = 1)$, is often used to model proportions, especially in Bayesian analysis, where parameters are treated as random variables. For example, π from the binomial distribution can be modeled with the standard β distribution as it takes on only non-zero values in the interval $(0, 1)$. The distribution can take on a wide variety of shapes, as depicted in Figure 4.20 on the following page. The **pdf**, mean, and variance for a general beta random variable ($\alpha > 0$ and $\beta > 0$) are shown in (4.22). When working with the standard β distribution, that is, $A = 0$ and $B = 1$, a β random variable X is denoted simply $X \sim \beta(\alpha, \beta)$. The β distribution available in R is the standard β distribution rather than the general β distribution.

Beta Distribution

$$X \sim \beta(\alpha, \beta, A, B)$$

$$f(x | \alpha, \beta, A, B) = \begin{cases} \frac{1}{B-A} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \left(\frac{x-A}{B-A}\right)^{\alpha-1} \left(\frac{B-x}{B-A}\right)^{\beta-1} & \text{if } A \leq x \leq B \\ 0 & \text{otherwise} \end{cases} \quad (4.22)$$

$$E[X] = A + (B - A) \frac{\alpha}{\alpha + \beta}$$

$$Var[X] = \frac{(B - A)^2 \alpha \beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}$$

Example 4.23 ▷ **Beta Distribution: Selling Computers** ◁ A wholesale computer distributor has a fixed amount of storage space in his warehouse. The warehouse is restocked with computers every 15 days. The distributor would like more information on the proportion of computers in the warehouse that are sold every 15 days. The warehouse manager claims that the proportion of computers sold can be modeled with a standard beta distribution where $\alpha = 4$ and $\beta = 2$. Compute the expected value for the proportion of computers sold every 15 days. How likely is it that at least 80% of the computers in stock will be sold during a 15-day period?

$$X \sim \beta(\alpha, \beta)$$

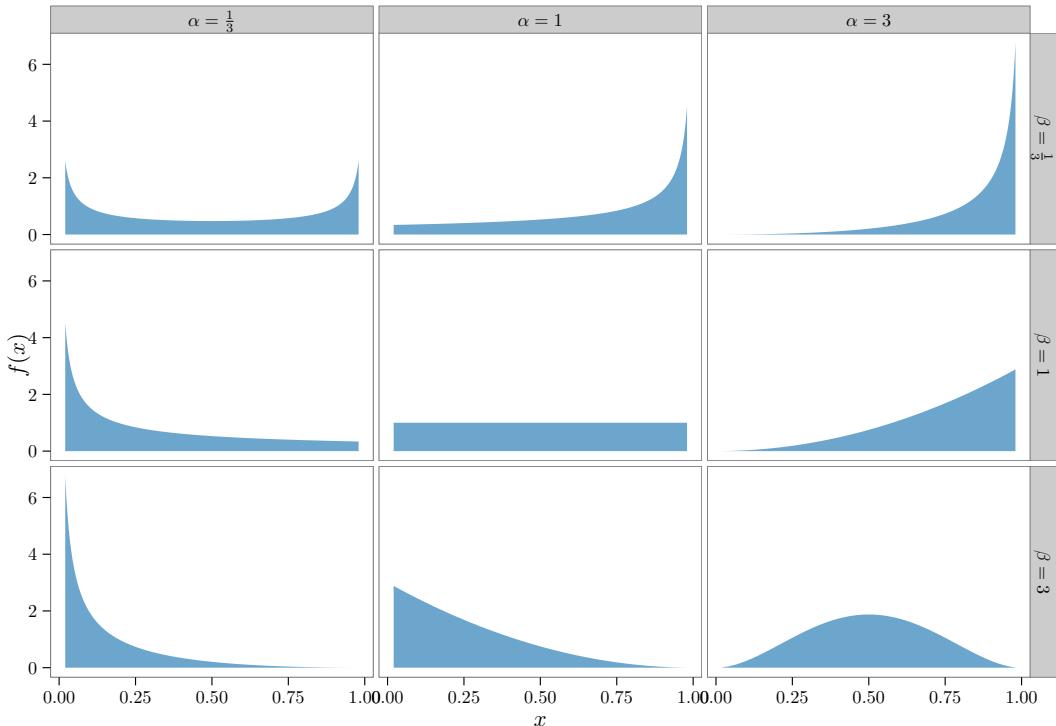


FIGURE 4.20: Graphical illustration of the pdfs of a $\beta(\alpha, \beta)$ for $\alpha = 1/3, 1$, and 3 , and $\beta = 1/3, 1$, and 3

Solution: Let the random variable X represent the proportion of computers sold in a 15-day period. Since $X \sim \beta(4, 2)$, the expected value from (4.22) yields

$$E[X] = \frac{\alpha}{\alpha + \beta} = \frac{2}{3}.$$

The probability that at least 80% of the computers in the warehouse are sold is

$$\mathbb{P}(X \geq 0.8) = \int_{0.8}^1 \frac{\Gamma(4+2)}{\Gamma(4)\Gamma(2)} x^3(1-x) dx = 20 \int_{0.8}^1 (x^3 - x^4) dx = 0.2627.$$

To compute the last answer with R, use the command `pbeta()` or `integrate()` as shown in R Code 4.18.

R Code 4.18

```
> pbeta(0.8, 4, 2, lower = FALSE)
[1] 0.26272

> b42 <- function(x){(gamma(6)/(gamma(4)*gamma(2)))*x^3*(1 - x)}
> integrate(b42, lower = 0.8, upper = 1)$value
[1] 0.26272
```

Example 4.24 ▷ Beta Distribution: Roof My House ◁ Project managers often use a Program Evaluation and Review Technique (PERT) to manage large-scale projects. PERT was actually developed by the consulting firm of Booz, Allen, & Hamilton in conjunction with the United States Navy as a tool for coordinating the activities of several thousands of contractors working on the Polaris missile project. A standard assumption in PERT analysis is that the time to complete any given activity follows a general beta distribution, where A is the optimistic time to complete an activity and B is the pessimistic time to complete the activity. Suppose the time X (in hours) it takes a three-man crew to re-roof a single-family house has a beta distribution with $A = 8$, $B = 16$, $\alpha = 2$, and $\beta = 3$. The crew will complete the re-roofing in a single day provided the total time to complete the job is no more than 10 hours. If this crew is contracted to re-roof a single-family house, what is the chance that they will finish the job in the same day?

Solution: To answer the question, find $\mathbb{P}(X \leq 10)$:

$$\begin{aligned}\mathbb{P}(X \leq 10) &= \int_8^{10} \frac{1}{8} \cdot \frac{\Gamma(5)}{\Gamma(2)\Gamma(3)} \left(\frac{x-8}{8}\right) \left(\frac{16-x}{8}\right)^2 dx \\ &= \frac{\Gamma(5)}{8^4\Gamma(2)\Gamma(3)} \int_8^{10} (x-8)(16-x)^2 dx \\ &= \frac{24}{4096 \cdot 1 \cdot 2} \int_8^{10} (512x - 40x^2 + x^3 - 2048) dx \\ &= \frac{3}{1024} \cdot \left(256x^2 - \frac{40}{3}x^3 + \frac{x^4}{4} - 2048x\right) \Big|_8^{10} \\ &= \frac{3}{1024} \cdot \frac{268}{3} = 0.2617.\end{aligned}$$

To compute the last answer with R, use the command `integrate()`:

```
> GB <- function(x){(1/8) * (gamma(5)/(gamma(2) * gamma(3))) *  
+ ((x - 8)/8) * ((16 - x)/8)^2}  
> integrate(GB, lower = 8, upper = 10)$value  
  
[1] 0.2617187
```

To solve the problem with `pbeta()`, enter

```
> A <- 8  
> B <- 16  
> x <- 10  
> ans <- pbeta((x - A)/(B - A), 2, 3)  
> ans  
  
[1] 0.2617188
```

The chance the crew will finish the job on the same day is 26.1719%.

4.3.7 Normal (Gaussian) Distribution

The **normal** or **Gaussian** distribution is more than likely the most important distribution in statistical applications. This is due to the fact that many numerical populations have distributions that can be approximated with the normal distribution. Examples of distributions following an approximate normal distribution include physical characteristics such as the height and weight of a particular species. Further, certain statistics, such as the mean, follow an approximate normal distribution when certain conditions are satisfied. The **pdf**, mean, variance, and **mgf** for a normal random variable X with mean μ and variance σ^2 are provided in (4.23). The **pdf** for a normal random variable has an infinite number of centers and spreads, depending on both μ and σ , respectively. Although there are an infinite number of possible normal distributions, all normal distributions have a bell shape that is symmetric around the distribution's mean. Figure 4.21 illustrates three normal distributions with identical means, μ , and increasing variances as the distributions are viewed from left to right. The standard deviation in a normal distribution is the horizontal distance from the center of the distribution to the point on the density curve where the curve changes from concave down to concave up (point of inflection). Small values of σ produce distributions that are relatively close to the distribution's mean. On the other hand, values of σ that are large produce distributions that are quite spread out around the distribution's mean.

Normal Distribution

$$X \sim N(\mu, \sigma)$$

$$f(x | \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty,$$

where $-\infty < \mu < \infty$ and $0 < \sigma < \infty$. (4.23)

$$E[X] = \mu$$

$$Var[X] = \sigma^2$$

$$M_X(t) = e^{\mu t + \frac{\sigma^2 t^2}{2}}$$

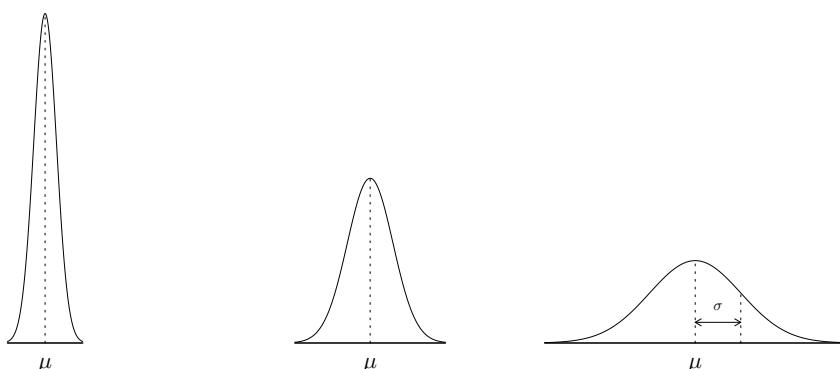


FIGURE 4.21: Three normal distributions, each with an increasing σ value as read from left to right

The **cdf** for a normal random variable, X , with mean, μ , and standard deviation, σ , is

$$F(x) = \mathbb{P}(X \leq x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt. \quad (4.24)$$

A normal random variable with $\mu = 0$ and $\sigma = 1$, often denoted Z , is called a **standard normal** random variable. The **cdf** for the standard normal distribution, given in (4.26), is computed by first standardizing the random variable X , where $X \sim N(\mu, \sigma)$, using the change of variable formula,

$$Z = \frac{X - \mu}{\sigma} \sim N(0, 1). \quad (4.25)$$

$$F(x) = \mathbb{P}(X \leq x) = \mathbb{P}\left(Z \leq \frac{x - \mu}{\sigma}\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{(x-\mu)}{\sigma}} e^{-\frac{z^2}{2}} dz. \quad (4.26)$$

Neither the integral for (4.26) nor the integral for (4.24) can be computed with standard techniques of integration; however, (4.26) has been numerically evaluated and tabled. Further, any normal random variable can be converted to a standard normal random variable using (4.25). The process of computing $\mathbb{P}(a \leq X \leq b)$, where $X \sim N(\mu, \sigma)$, is graphically illustrated in Figure 4.22 on the next page.

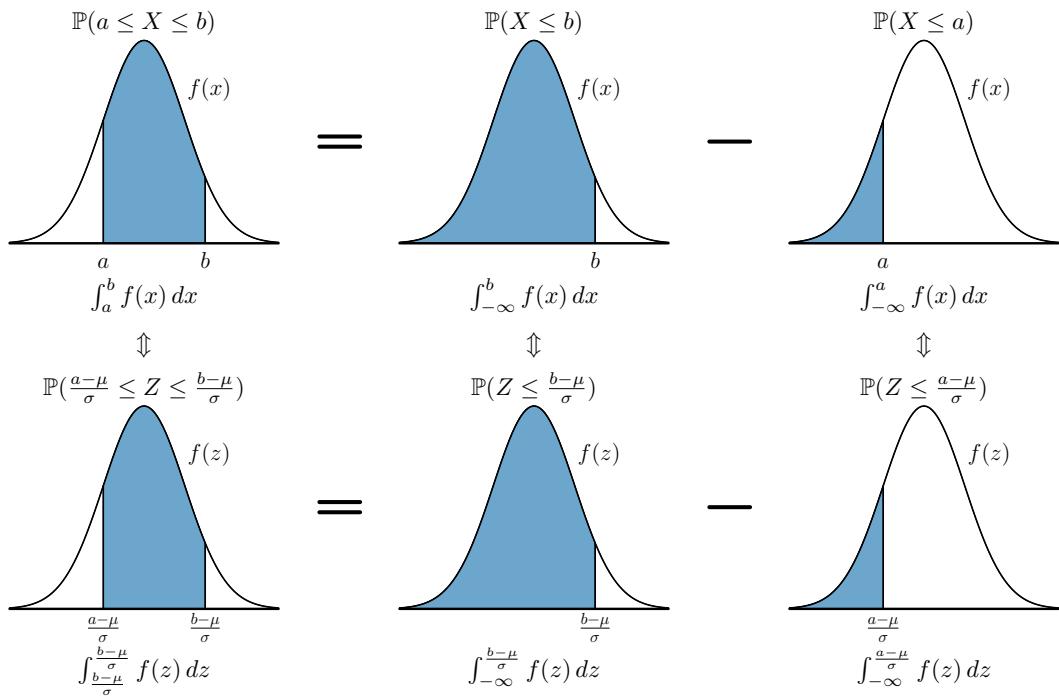
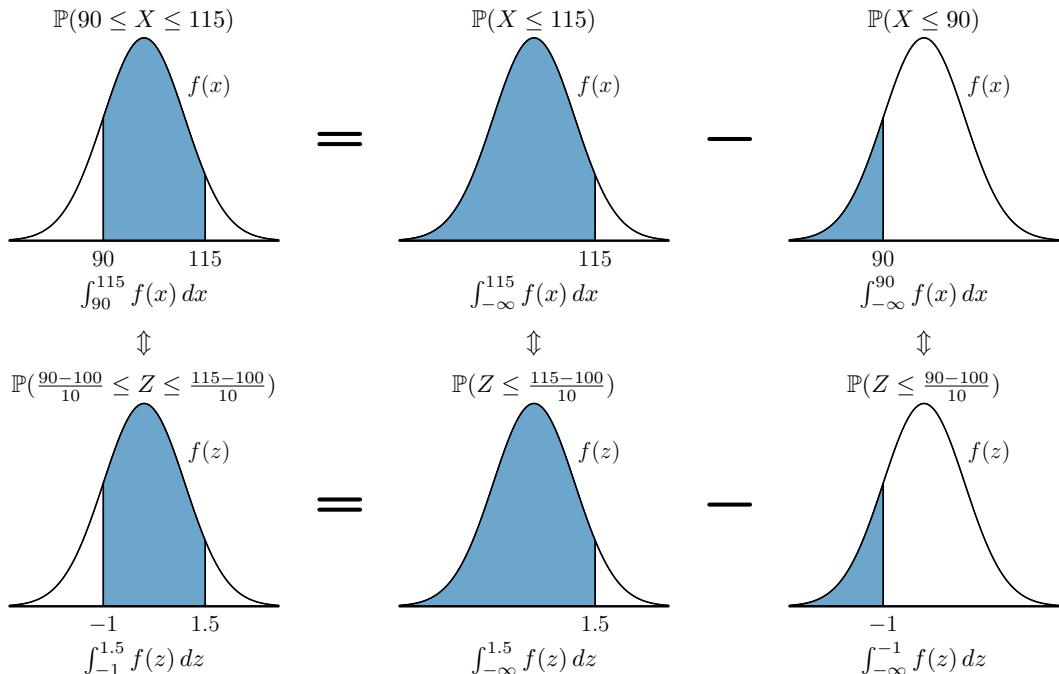
Throughout the text, the convention z_α is used to represent the value of the standard normal random variable Z that has α of its area to the left of said value. In other words, $\mathbb{P}(Z < z_\alpha) = \alpha$. Another notation that is also used in the text is $\Phi(z_\alpha) = \alpha$. Basically, the Φ (value) is the same as $\mathbb{P}(Z < \text{value})$. That is, Φ is the **cdf** of the standard normal distribution. Likewise, $\Phi^{-1}(\alpha) = z_\alpha$. The Φ notation for the **cdf** and inverse **cdf** is used more in Chapter 10.

To find the numerical value of X_α , where $X \sim N(\mu, \sigma)$ and α is the area (or probability) to the left of the value X_α , use the R command `qnorm(p, mean=MValue, sd=SValue)`, where `p` is the area or probability (this is equivalent to α) to the left of X_α , `MValue` is the value of the mean, and `SValue` is the value of the standard deviation. Note that if one is dealing with the standard normal distribution, the `mean=MValue` or `sd=SValue` arguments are not needed.

Example 4.25 Scores on a particular standardized test follow a normal distribution with a mean of 100 and standard deviation of 10.

- (a) What is the probability that a randomly selected individual will score between 90 and 115?
- (b) What score does one need to be in the top 10%?
- (c) Find the constant c such that $\mathbb{P}(105 \leq X \leq c) = 0.10$.

Solution: Historically, normal distributions had to be standardized and the values of probabilities looked up in tables. Though this is no longer the case, this example shows how to standardize X and how to use the R command `pnorm()` with a standard normal random variable to “look up” probabilities to the left of given values. Understanding the standard normal, $Z \sim N(0, 1)$, and the probabilities associated with different values from this distribution gives the student an intuition about other normal distributions whose mean and standard deviation are something other than 0 and 1.

FIGURE 4.22: Graphical representation for computing $\mathbb{P}(a \leq X \leq b)$ given $X \sim N(\mu, \sigma)$ FIGURE 4.23: Graphical representation for computing $\mathbb{P}(a \leq X \leq b)$ given $X \sim N(100, 10)$

(a) To find $\mathbb{P}(90 \leq X \leq 115)$, first draw a picture representing the desired area such as the one in Figure 4.23 on the facing page. Note that finding the area between 90 and 115 is equivalent to finding the area to the left of 115 and from that area, subtracting the area to the left of 90. In other words,

$$\mathbb{P}(90 \leq X \leq 115) = \mathbb{P}(X \leq 115) - \mathbb{P}(X \leq 90).$$

To find $\mathbb{P}(X \leq 115)$ and $\mathbb{P}(X \leq 90)$, one can standardize using (4.25). That is,

$$\mathbb{P}(X \leq 115) = \mathbb{P}\left(Z \leq \frac{115 - 100}{10}\right) = \mathbb{P}(Z \leq 1.5),$$

and

$$\mathbb{P}(X \leq 90) = \mathbb{P}\left(Z \leq \frac{90 - 100}{10}\right) = \mathbb{P}(Z \leq -1.0).$$

Using the R commands `pnorm(1.5)` and `pnorm(-1)`, find the areas to the left of 1.5 and -1.0 to be 0.9332 and 0.1587, respectively. Consequently,

$$\begin{aligned}\mathbb{P}(90 \leq X \leq 115) &= \mathbb{P}(-1.0 \leq Z \leq 1.5) \\ &= \mathbb{P}(Z \leq 1.5) - \mathbb{P}(Z \leq -1.0) \\ &= 0.9332 - 0.1587 = 0.7745.\end{aligned}$$

The probability a selected individual will score between 90 and 115 is 0.7745.

```
> pnorm(115, 100, 10) - pnorm(90, 100, 10)
[1] 0.7745375
```

(b) Finding the value c such that 90% of the area is to its left is equivalent to finding the value c such that 10% of its area is to the right. That is, finding the value c that satisfies $\mathbb{P}(X \leq c) = 0.90$ is equivalent to finding the value c such that $\mathbb{P}(X \geq c) = 0.10$. Since the `qnorm()` function refers to areas to the left of a given value by default, solve

$$\mathbb{P}(X \leq c) = \mathbb{P}\left(Z = \frac{X - 100}{10} \leq \frac{c - 100}{10}\right) = 0.90 \text{ for } c.$$

Using `qnorm(.9)`, find the Z value (1.2816) such that 90% of the area in the distribution is to the left of that value. Consequently, to be in the top 10%, one needs to be more than 1.2816 standard deviations above the mean:

$$\frac{c - 100}{10} \stackrel{\text{set}}{=} 1.2816$$

and solve for $c \Rightarrow c = 112.816$.

To be in the top 10%, one needs to score 112.8155 or higher.

```
> qnorm(0.9, 100, 10)
[1] 112.8155
```

(c) $\mathbb{P}(105 \leq X \leq c) = 0.10$ is the same as

$$\mathbb{P}(X \leq c) = 0.10 + \mathbb{P}(X \leq 105) = 0.10 + \mathbb{P}\left(Z \leq \frac{105 - 100}{10}\right).$$

Using `pnorm(.5)`,

$$\mathbb{P}\left(Z \leq \frac{105 - 100}{10}\right) = \mathbb{P}(Z \leq 0.5) = 0.6915.$$

It follows then that $\mathbb{P}(X \leq c) = 0.7915$. Using `qnorm(.7915)` gives 0.8116:

$$\mathbb{P}(X \leq c) = \mathbb{P}\left(Z = \frac{X - 100}{10} \leq \frac{c - 100}{10}\right) = 0.7915$$

$$\text{is found by solving } \frac{c - 100}{10} = 0.8116 \Rightarrow c = 108.116.$$

Note that a Z value of 0.8116 has 79.15% of its area to the left of that value.

```
> qnorm(0.1 + pnorm(105, 100, 10), 100, 10)
[1] 108.1151
```



Example 4.26 ▷ Normal Distribution: Cell Phone Components ◁ Most mobile appliances today allow the consumer to switch from the built-in speaker and microphone to an external source. A manufacturer of cell phones wants to package an external speaker and microphone for hands-free operation. A new company has patented a component that allows the on-resistance flatness for both the microphone and speaker to be lower than ever before. The cell phone company requires that the on-resistance flatness be less than 0.7 ohms (Ω). If it is known that 50% of the components from the new company have an ohm rating of 0.5 Ω or less, 10% have an ohm rating of 0.628 Ω or greater, and the distribution of the ohm ratings is normal, then:

- Find the mean and standard deviation for the distribution of the ohm rating of the components.
- If a component is selected at random, what is the probability that its on-resistance flatness will be less than 0.7 Ω ?
- If 20 components are selected at random, what is the probability that at least 19 components will have on-resistance flatness values less than 0.7 Ω ?

Solution: Let X = the ohm rating of the patented components.

(a) Because a normal distribution is symmetric, the mean equals the median. It is known that 50% of the components have an ohm rating of 0.5 Ω or less, so $\mu_X = 0.5$. To calculate the standard deviation of the components' ohm ratings, use the fact that "10% have an ohm rating of 0.628 Ω or greater."

This means that $\mathbb{P}(X \leq 0.628) = 0.9$,

which implies $\mathbb{P}\left(Z = \frac{X - 0.5}{\sigma} \leq \frac{0.628 - 0.5}{\sigma}\right) = 0.9$.

Because $\mathbb{P}(Z \leq 1.28) = 0.9$, set $\frac{0.628 - 0.5}{\sigma} = 1.28$

and solve for σ . $\frac{0.628 - 0.5}{1.28} = \sigma$

Therefore $\sigma = 0.1$.

(b) Calculate the probability a component has an on-resistance flatness less than 0.7 Ω :

$$\begin{aligned}\mathbb{P}(X \leq 0.7) &= \mathbb{P}\left(Z = \frac{X - 0.5}{0.1} \leq \frac{0.7 - 0.5}{0.1}\right) \\ &= \mathbb{P}(Z \leq 2) \\ &= 0.9772.\end{aligned}$$

The answer computed with R is

```
> p <- pnorm(0.7, 0.5, 0.1)
> p
[1] 0.9772499
```

(c) Calculate the probability that at least 19 of the 20 components will have an on-resistance flatness value less than 0.7 Ω . Let $Y \sim Bin(20, 0.9772)$.

$$\mathbb{P}(Y \geq 19) = \sum_{i=19}^{20} \binom{20}{i} (0.9772)^i (1 - 0.9772)^{20-i} = 0.925.$$

To compute the answer with R, type

```
> sum(dbinom(19:20, 20, p))
[1] 0.9249673

> # OR
> pbinom(18, 20, p, lower = FALSE)
[1] 0.9249673
```

Quantile-Quantile Plots for Normal Distributions Many of the techniques presented later in the book assume the underlying distribution is normal. One of the more useful graphical procedures for assessing distributions is the quantile-quantile plot. (Recall from Section 2.7.3 that this graph is also called a Q-Q plot.) To help determine whether the underlying distribution is normal, use the R function `qqnorm()`.

To understand the `qqnorm()` function, one needs to have some understanding of R's `quantile()` function. Recall that the cumulative distribution function (`cdf`) is $F(x) = P(X \leq x)$. The `quantile()` function is the inverse of the `cdf`, where this exists; that is, $Q(u) = F^{-1}(u)$. The `qqnorm()` function works by first computing the quantiles of the points $(i - 1/2)/n$ for the standard normal distribution when $n > 10$. For $n \leq 10$, the quantiles of the points $(i - 3/8)/n$ for the standard normal are computed. The actual points are computed with the function `ppoints()`. The ordered sample values are then plotted against the quantiles. When the resulting plot is linear, it indicates the sample values have a normal distribution. To help assess the linearity of the `qqnorm()` plot, it is often quite helpful to plot a straight line through the 25th and 75th percentiles, also referred to as the first and third quartiles, using the function `qqline()`, which connects the pair of points (First Quartile Standard Normal, First Quartile Data), (Third Quartile Standard Normal, Third Quartile Data).

For example, consider the values stored in the variable **scores** of the data frame **SCORE** and reported in Table 4.2, which are the scores a random sample of 20 college freshmen received on a standardized test. The points $(i - 1/2)/n$ are calculated as

$$(1 - 1/2)/20 = 0.025, (2 - 1/2)/20 = 0.075, \dots, (20 - 1/2)/20 = 0.975,$$

while the corresponding standard normal quantiles of $\{0.025, 0.075, \dots, 0.975\}$ are computed with **qnorm()** to be $\{-1.96, -1.44, \dots, 1.96\}$, respectively. The function **qqnorm()** plots the quantiles $\{-1.96, -1.44, \dots, 1.96\}$ versus the ordered values in the sample, $\{87, 90, \dots, 119\}$ as shown in Figure 4.24 on the facing page. The pair of points (First Quartile Standard Normal, First Quartile Data), (Third Quartile Standard Normal, Third Quartile Data) is $(-0.637, 96.75)$ and $(0.637, 106.25)$, respectively. Note how the line in Figure 4.24 on the next page created using the function **qqline()** goes through the points $(-0.637, 96.75)$ and $(0.637, 106.25)$. To compute the pairs of values plotted in an R quantile-quantile plot for the variable **scores** of the data frame **SCORE**, see R Code 4.19.

Table 4.2: Standardized scores (data frame **SCORE**)

119	107	96	107	97	103	94	106	87	112
99	99	90	106	110	99	105	100	100	94

R Code 4.19

```
> n <- length(SCORE$scores)
> X <- (1:n - 1/2)/n
> Xs <- qnorm(X)
> Ys <- sort(SCORE$scores)
> plot(Xs, Ys)
> quantile(Xs, c(0.25, 0.75))

      25%      75%
-0.6371739  0.6371739

> quantile(Ys, c(0.25, 0.75))

      25%      75%
96.75 106.25
```

Generally, the command **qqnorm()** is used to generate the pairs of values that are plotted for a normal quantile-quantile plot, while the command **qqline()** adds a line to a normal quantile-quantile plot that passes through the first and third quartiles. The commands **qqnorm(SCORE\$scores)** and **qqline(SCORE\$scores)** were used to create Figure 4.24 on the next page. The graphics packages **lattice** and **ggplot2** may also be used to create quantile-quantile plots. The R Code in 4.20 on the facing page can be used to create graphs similar to Figure 4.25 on the next page using both **lattice** and **ggplot2** functions. The **lattice** function to create a quantile-quantile plot is **qqmath()**, while the **ggplot2** function that adds a quantile-quantile layer is **stat_qq()**. The default distribution for quantile-quantile plots with both **lattice** and **ggplot2** is the normal distribution. To compare a

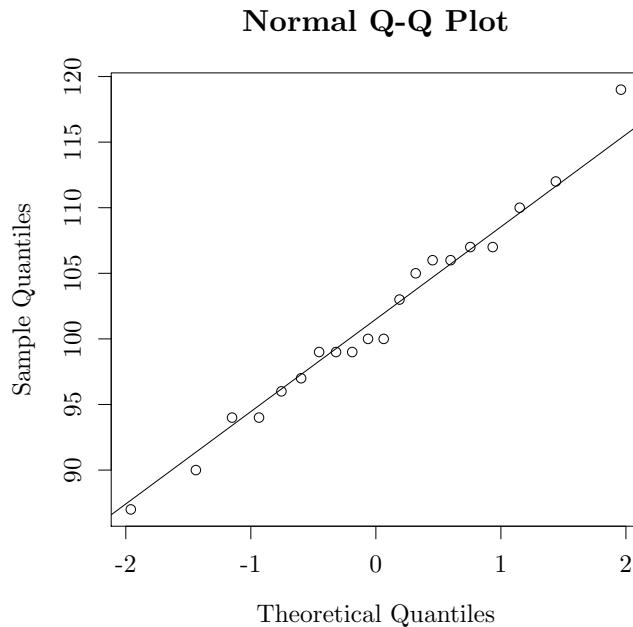


FIGURE 4.24: Quantile-quantile plot of the standardized test scores of 20 randomly selected college freshmen

sample to a distribution other than the normal distribution, see the help files of `qqmath()` or `stat_qq()`, depending on whether one is using `lattice` or `ggplot2`, respectively.

R Code 4.20

```
> qqmath(~scores, data = SCORE)
> ggplot(data = SCORE, aes(sample = scores)) + stat_qq()
```

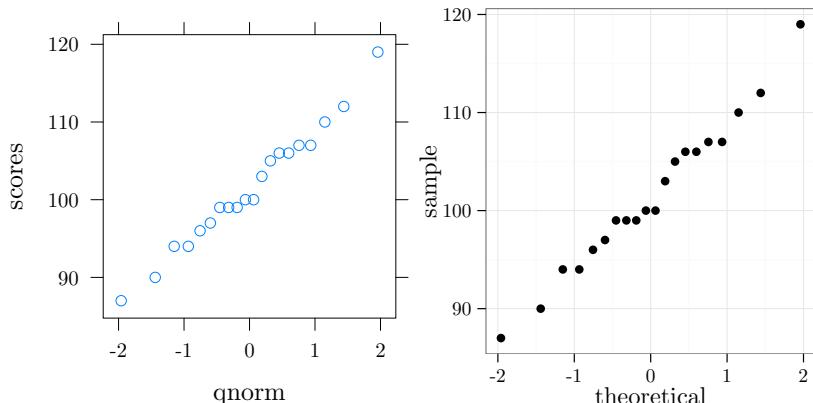


FIGURE 4.25: Quantile-quantile plot of the standardized test scores of 20 randomly selected college freshmen created with the `lattice` package (left graph) and the `ggplot2` package (right graph)

It is possible to tell from a quantile-quantile plot whether the distribution has shorter or longer tails than a normal distribution. In addition, the quantile-quantile plot will show whether a distribution is skewed and in which direction the distribution is skewed. The right quantile-quantile plots in Figure 4.26 illustrate how distributions that have a positive skew will appear as upward opening U shapes in the quantile-quantile plot, while distributions with a negative skew have downward facing U shapes. The left quantile-quantile plots in Figure 4.26 illustrate how distributions that have short tails relative to the normal distribution will have an S shape while distributions with tails longer than the normal distribution will have an inverted S shape.

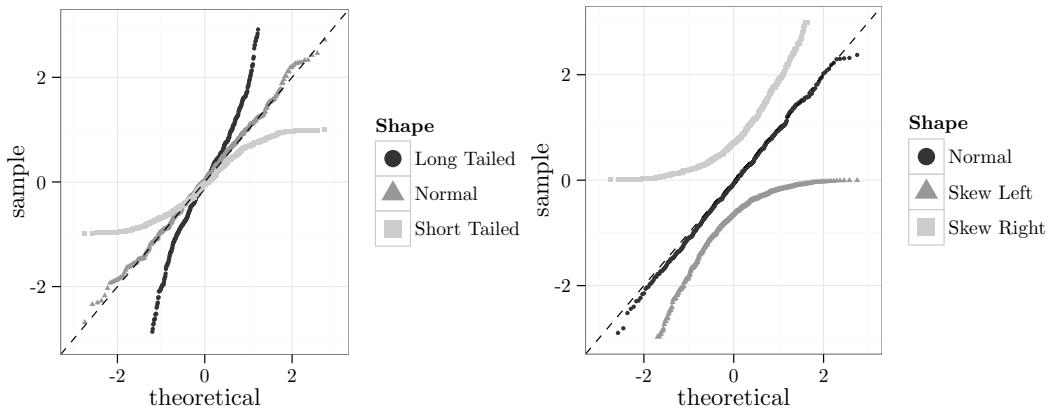


FIGURE 4.26: Superimposed quantile-quantile plots for simulated data from a skew left, skew right, and normal distribution (on the right) and from a short-tailed, long-tailed, and normal distribution (on the left)

The graphs in Figure 4.26 can be slightly misleading in the sense that they were constructed from large data sets ($n = 500$). When n is smaller, reading a quantile-quantile plot is slightly more challenging; however, the plotted values still need to fall close to a straight line. One way to train the eye with the quantile-quantile plot is to use simulation to generate data from a normal distribution for various values of n and observe the resulting quantile-quantile plots. When this is done, what one realizes is that for small values of n , even when sampling from a normal distribution, the resulting quantile-quantile plot is not always linear. The function `ntester()`, available in the `PASWR2` package, demonstrates how samples ($n < 5000$) from a normal distribution that have the same sample size as the actual data can appear in quantile-quantile plots. One is strongly encouraged to run this function before finalizing the assessment about the normality of a smaller-sized sample. The results from using `ntester()` on the standardized test scores from Table 4.2 on page 302 are shown in Figure 4.27 on the next page. Note that the actual data are the center normal quantile-quantile plot and all of the surrounding quantile-quantile plots are for simulated normal data having the same sample size as the center plot. One should pay close attention to how variable the eight surrounding graphs can be even when the data are coming from a normal distribution. If the data are no more variable than the surrounding plots, it should be safe to assume they are normal.

It is often helpful to look at several graphs at once when assessing the general shape of a distribution. The function `eda()` in the `PASWR2` package displays a histogram, a density plot, a boxplot, and a normal quantile-quantile plot of a numeric variable as well as

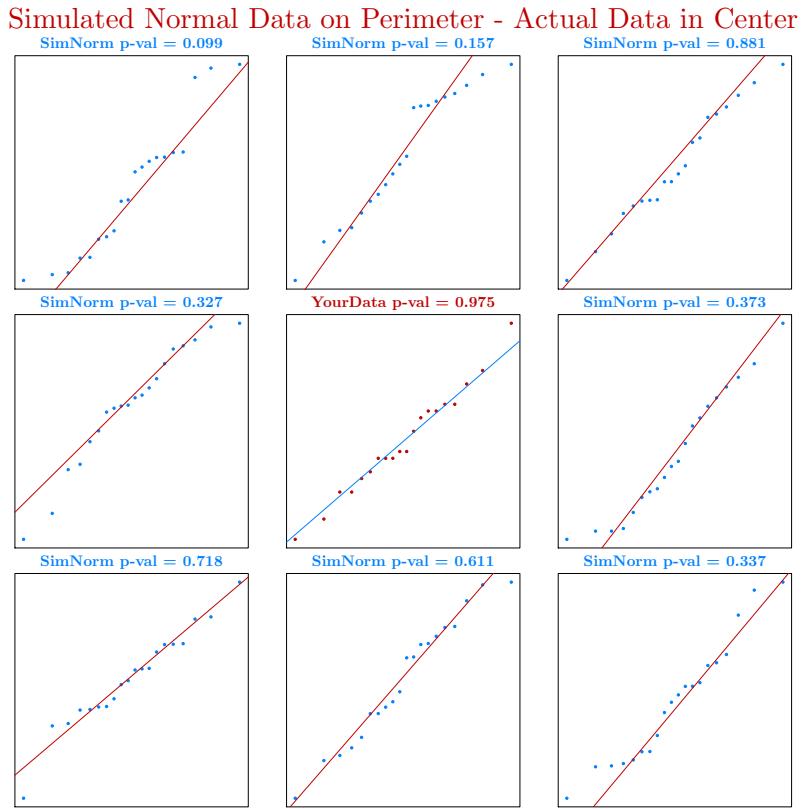
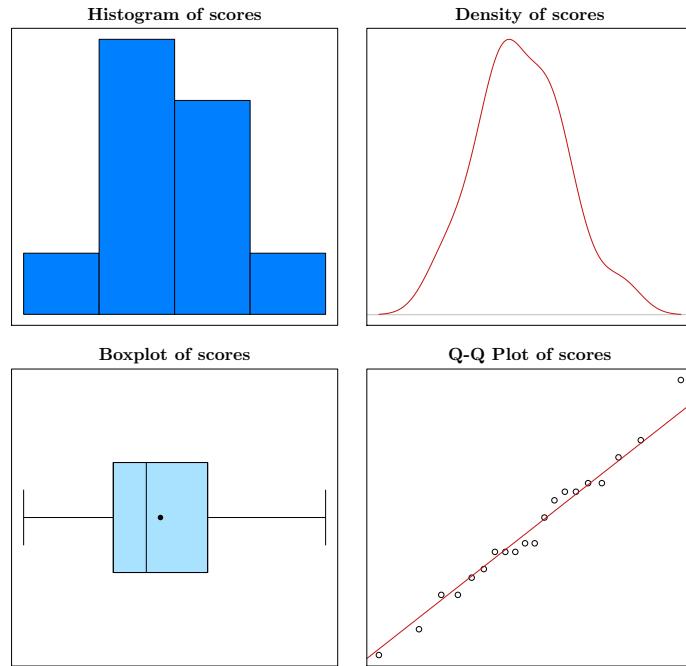


FIGURE 4.27: Resulting quantile-quantile plots using the function `ntester()` on the standardized test scores from Table 4.2 on page 302

computing various numerical summaries that are returned in the console. In order to allow the user to focus strictly on the resulting shapes, no measurement scales are given in the graphical output. Figure 4.28 on the following page shows the graphical results from using `eda(scores)`. All four graphs in Figure 4.28 confirm normality as a reasonable assumption for the distribution of the variable `scores`.

EXPLORATORY DATA ANALYSISFIGURE 4.28: Graphical results from `eda(scores)`

4.4 Problems

1. Let X be a Poisson random variable with mean equal to 2. Find $\mathbb{P}(X = 0)$, $\mathbb{P}(X \geq 3)$, and $\mathbb{P}(X \leq k) \geq 0.70$.
2. Let X be an exponential random variable $Exp(\lambda = 3)$. Find $\mathbb{P}(2 < X < 6)$.
3. Let X be a normal random variable $N(\mu = 7, \sigma = 3)$. Calculate $\mathbb{P}(X > 7.1)$. Find the value of k such that $\mathbb{P}(X < k) = 0.8$.
4. Let X be a normal random variable $N(\mu = 3, \sigma = \sqrt{0.5})$. Calculate $\mathbb{P}(X > 3.5)$.
5. Let X be a gamma random variable $\Gamma(\alpha = 2, \lambda = 6)$. Find the value a such that $\mathbb{P}(X < a) = 0.95$.
6. Construct a plot for the probability mass function and the cumulative probability distribution of a binomial random variable $Bin(n = 8, \pi = 0.3)$. Find the smallest value of k such that $\mathbb{P}(X \leq k) \geq 0.44$ when $X \sim Bin(n = 8, \pi = 0.7)$. Calculate $\mathbb{P}(Y \geq 3)$ if $Y \sim Bin(20, 0.2)$.
7. If X is the number of 3s that appear when 60 dice are tossed, what is the $E(X^2)$?
8. An importing company knows that 80% of its imported Chinese socks are suitable for sale. If a sample of 60 pairs is drawn at random, find the probability that a percentage between 70% and 90% (inclusive) of the sample is suitable for sale.
9. It is known that 3% of the seeds of a certain variety of tomato do not germinate. The seeds are sold in individual boxes that contain 20 seeds per box with the guarantee that at least 18 seeds will germinate. Find the probability that a randomly selected box does not fulfill the aforementioned requirement.
10. A garage has two machines, A and B, to balance the wheels of a car. Suppose that 95% of the wheels are correctly balanced by machine A, while 85% of the wheels are correctly balanced by machine B. A machine is randomly selected to balance 20 wheels, and 3 of them are not properly balanced. What is the probability that machine A was used? What is the probability machine B was used?
11. Traffic volume is an important factor for determining the most cost-effective method to surface a road. Suppose that the average number of vehicles passing a certain point on a road is 2 every 30 seconds.
 - (a) Find the probability that more than 3 cars will pass the point in 30 seconds.
 - (b) What is the probability that more than 10 cars pass the point in 3 minutes?
12. The retaining wall of a dam will break if it is subjected to the pressure of two floods. If the average number of floods in a century is two, find the probability that the retaining wall lasts more than 20 years.
13. A particular competition shooter hits his targets 70% of the time with any pistol. To prepare for shooting competitions, this individual practices with a pistol that holds 5 bullets on Tuesday, Thursday, and Saturday, and a pistol that holds 7 bullets the other days. If

he fires at targets until the pistol is empty, find the probability that he hits only one target out of the bullets shot in the first round of bullets in the pistol he is carrying that day. In this case, what is the probability that he used the pistol with 7 bullets?

14. The lifetime of a certain engine follows a normal distribution with mean and standard deviation of 10 and 3.5 years, respectively. The manufacturer replaces all catastrophic engine failures within the guarantee period free of charge. If the manufacturer is willing to replace no more than 4% of the defective engines, what is the largest guarantee period the manufacturer should advertise?

15. Agronomists are developing an improved variety of green peppers. Supermarket managers have indicated customers are not likely to purchase green peppers weighing less than 45 grams. The current variety of green pepper plants produces green peppers that weigh 48 grams on average, but 13% weigh less than 45 grams. Assume the weight of the current variety of green peppers follows a normal distribution.

- (a) What is the standard deviation of the weights of the current variety of green peppers?
- (b) The agronomists want to reduce the frequency of green peppers weighing less than 45 grams to no more than 5%. One way to reduce the frequency of underweight green peppers is to increase the weight of the green peppers. If the standard deviation remains the same, what mean should the agronomists target as a goal?
- (c) The agronomists produce a new variety of green peppers with a mean weight of 50 grams, which meets the 5% goal. What is the standard deviation of the weights of these new green peppers?
- (d) Does the current variety or the new variety produce a green pepper with a more consistent weight?

16. Given independent random variables Y_1, Y_2, X, W, Z_1, Z_2 , and Z_3 ,

- (a) Compute $\mathbb{P}((Y_2 \geq 3) \cup (Y_1 < 9))$ if $Y_1 \sim \text{Bin}(n = 10, \pi = 0.3)$ and $Y_2 \sim \text{Bin}(n = 5, \pi = 0.1)$.
- (b) Compute $\mathbb{P}(X \geq 2 | X < 6)$ if $X \sim \text{Pois}(\lambda = 4)$.
- (c) If $W \sim N(\mu, \sigma)$, find the value of k that satisfies the equation $\mathbb{P}(\mu < W < \mu + 2k\sigma) = 0.45$.
- (d) If $Z_i \sim N(0, 1)$ for $i = 1, 2, 3$, compute $\mathbb{P}\left(\sqrt{Z_1^2 + Z_2^2 + Z_3^2} > 1.5\right)$.

17. Derive the mean and variance for the discrete uniform distribution.

(Hints: $\sum_{i=1}^n x_i = \frac{n(n+1)}{2}$; $\sum_{i=1}^n x_i^2 = \frac{n(n+1)(2n+1)}{6}$, when $x_i = 1, 2, \dots, n$.)

18. Suppose the percentage of drinks sold from a vending machine are 80% and 20% for soft drinks and bottled water, respectively.

- (a) What is the probability that on a randomly selected day, the first soft drink is the fourth drink sold?
- (b) Find the probability that exactly 1 out of 10 drinks sold is a soft drink.
- (c) Find the probability that the fifth soft drink is the seventh drink sold.

- (d) Verify empirically that $\mathbb{P}(Bin(n,\pi) \leq r-1) = 1 - \mathbb{P}(NB(r,\pi) \leq (n-r))$, with $n = 10$, $\pi = 0.8$, and $r = 4$.
19. The hardness of a particular type of sheet metal sold by a local manufacturer has a normal distribution with a mean of 60 micra and a standard deviation of 2 micra.
- This type of sheet metal is said to conform to specification provided its hardness measure is between 57 and 65 micra. What percent of the manufacturer's sheet metal can be expected to fall within the specification?
 - A building contractor agrees to purchase metal from the local metal manufacturer at a premium price provided four out of four randomly selected pieces of metal test between 57 and 65 micra. What is the probability the building contractor will purchase metal from the local manufacturer and pay a premium price?
 - If an acceptable sheet of metal is one whose hardness is not more than c units away from the mean, find c such that 97% of the sheets are acceptable.
 - Find the probability that at least 10 out of 20 sheets have a hardness greater than 60.
20. The weekly production of a banana plantation can be modeled with a normal random variable that has a mean of 5 tons and a standard deviation of 2 tons.
- Find the probability that, in at most 1 out of the 8 randomly chosen weeks, the production has been less than 3 tons.
 - Find the probability that at least 3 weeks are needed to obtain a production greater than 10 tons.
21. A bank has 50 deposit accounts with €25,000 each. The probability of having to close a deposit account and then refund the money in a given day is 0.01. If account closings are independent events, how much money must the bank have available to guarantee it can refund all closed accounts in a given day with probability greater than 0.95?
22. The mean number of calls a tow truck company receives during a day is 5 per hour. Find the probability that a tow truck is requested more than 4 times per hour in a given hour. What is the probability the company waits for less than 1 hour before the tow truck is requested 3 times?
23. In the printing section of a plastics company, a machine receives on average 6 buckets per minute to be painted and paints them. The machine has been out of service for 90 seconds due to a power failure.
- Find the probability that more than 8 buckets remain unpainted.
 - Find the probability that the first bucket, after the electricity is restored, arrives before 10 seconds have passed.
24. Give a general expression to calculate the quantiles of a Weibull random variable.
25. A used-car salesman offers a guarantee period of 12 months for his cars. He knows that the distribution of the elapsed time (in months) until the first breakdown occurs follows a

Weibull distribution, $Weib(3, 25)$. If the salesman expects to sell 50 cars per year, and the repair cost per car is on average 800 dollars, what is the mean cost of the guarantee?

26. Fix the seed value at 500, and generate a random sample of size $n = 10000$ from a $Unif(0,1)$ distribution. Calculate the sample mean and the sample variance. Are your answers within 1% of the theoretical values for the mean and variance of a $Unif(0,1)$ distribution?

27. Fix the seed value at 50, and generate a random sample of size $n = 10000$ from an exponential distribution with $\lambda = 2$. Create a density histogram and superimpose the histogram with a theoretical $Exp(\lambda = 2)$ distribution. Calculate the sample mean and the sample variance of the randomly generated values. Are your answers within 1% of the theoretical values for the mean and variance of an $Exp(\lambda = 2)$ distribution?

28. An oyster farm harvests pearls with uniformly distributed radii between 2 and 4 cm.

- (a) Find the theoretical mean and variance for the volume of the pearls.
- (b) Set the seed to 961 and simulate the volume for 10,000 pearls. Find the mean and variance for the volume of the simulated pearls. Are your answers within 2% of the theoretical values found in (a)?

29. Let X be a random variable with probability density function

$$f(x) = 3 \left(\frac{1}{x} \right)^4, \quad x \geq 1.$$

- (a) Find the cumulative density function.
- (b) Fix the seed at 98 (`set.seed(98)`), and generate a random sample of size $n = 100,000$ from X 's distribution. Compute the mean, variance, and coefficient of skewness for the random sample.
- (c) Obtain the theoretical mean, variance, and coefficient of skewness of X .
- (d) How close are the estimates in (b) to the theoretical values in (c)?

30. Let X be a random variable with probability density function

$$f(x) = \theta \left(\frac{1}{x} \right)^{\theta+1}, \quad x \geq 1, \theta > 2.$$

- (a) Verify that the area under $f(x)$ is 1.
- (b) Find the cumulative density function.
- (c) What is $\mathbb{P}(X \leq 3)$?
- (d) Fix the seed at 42 (`set.seed(42)`), and generate 100,000 realizations of X with $\theta = 3$. What are the mean and variance of the random sample?
- (e) Calculate the theoretical mean and variance of X .
- (f) How close are the estimates in (d) to the theoretical values in (e)?

31. Let X be a random variable with probability density function

$$f(x) = \frac{4}{3}x(2 - x^2), \quad 0 \leq x \leq 1.$$

- (a) Verify that the area under $f(x)$ is 1.
- (b) Find the cumulative density function.
- (c) What is $\mathbb{P}(X > .75)$?
- (d) Fix the seed at 13 (`set.seed(13)`), and generate 100,000 realizations of X . What are the mean and variance of the random sample?
- (e) Calculate the theoretical mean and variance of X .
- (f) How close are the estimates in (d) to the theoretical values in (e)?

32. Let X be a random variable with probability density function

$$f(x) = (\theta + 1)(1 - x)^\theta, \quad 0 \leq x \leq 1, \theta > 0.$$

- (a) Verify that the area under $f(x)$ is 1.
- (b) Find the cumulative density function.
- (c) What is $\mathbb{P}(X \leq .25 | \theta = 2)$?
- (d) Fix the seed at 80 (`set.seed(80)`), and generate 100,000 realizations of X with $\theta = 2$. What are the mean and variance of the random sample?
- (e) Calculate the theoretical mean and variance of X when $\theta = 2$.
- (f) How close are the estimates in (d) to the theoretical values in (e)?

33. Let X be a random variable with probability density function

$$f(x) = 3\pi\theta x^2 e^{-\theta\pi x^3}, \quad x \geq 0.$$

- (a) Verify that the area under $f(x)$ is 1.
- (b) Find the cumulative density function.
- (c) What is $\mathbb{P}(X > 1)$?
- (d) Fix the seed at 201 (`set.seed(201)`), and generate 100,000 realizations of X with $\theta = 5$. What are the mean and variance of the random sample?
- (e) Calculate the theoretical mean and variance of X .
- (f) How close are the estimates in (d) to the theoretical values in (e)?

34. A copper wire manufacturer produces conductor cables. These cables are of practical use if their resistance lies between 0.10 and 0.13 ohms per meter. The resistance of the cables follows a normal distribution, where 50% of the cables have resistance under 0.11 ohms and 10% have resistance over 0.13 ohms.

- (a) Determine the mean and the standard deviation for cable resistance.
- (b) Find the probability that a randomly chosen cable can be used.
- (c) Find the probability that at least 3 out of 5 randomly chosen cables can be used.

35. A binomial, $\text{Bin}(n,\pi)$, distribution can be approximated by a normal distribution, $N(n\pi, \sqrt{n\pi(1-\pi)})$, when $n\pi > 10$ and $n(1-\pi) > 10$. The Poisson distribution can also be approximated by a normal distribution $N(\lambda, \sqrt{\lambda})$ if $\lambda > 10$. Consider a sequence from 5 to 27 of a variable X (binomial or Poisson) and show that for $n = 80$, $\pi = 0.2$, and $\lambda = 16$ the aforementioned approximations are appropriate. The normal approximation to a discrete distribution can occasionally be improved by adding 0.5 to the normal random variable when finding the area to the left of said random variable. Specifically, create a table showing $\mathbb{P}(X \leq x)$ for the range of X for the four distributions and a graph showing the density of the normal distributions with vertical lines showing $\mathbb{P}(X = x)$ for the binomial and Poisson distributions, respectively.

36. Verify that if k/N is small (≤ 0.1) and $N = m + n$ is large, a hypergeometric distribution, $\text{Hyper}(m, n, k)$, can be adequately approximated by a $\text{Bin}(n = k, \pi = m/N)$ distribution. Compute the probabilities for each distribution using the hypergeometric values $n = 20$, $m = 300$, $k = 10$. Show the numerical results to three decimal places as well as a graph depicting the probabilities of the hypergeometric distribution with a vertical line and the probabilities of the binomial distribution in the same plot with a solid circle.

37. In 1935, Fisher described the following experiment in his book, *Design of Experiments*: A friend of Fisher's said that when she drank tea with milk, she was able to determine if the tea was poured first or if the milk was poured first. Find the probability that Fisher's colleague guesses 3 cups in which milk has been added before tea, given that in 4 out of 8 cups, milk has been added before tea.

38. Consider the function $g(x) = (x - a)^2$, where a is a constant and $E[(X - a)^2]$ is finite. Find a so that $E[(X - a)^2]$ is minimized.

39. Consider the random variable $X \sim \text{Weib}(\alpha, \beta)$.

- (a) Find the **cdf** for X .
- (b) Use the definition of the hazard function to verify that for $X \sim \text{Weib}(\alpha, \beta)$, the hazard function is given by $h(t) = \frac{\alpha t^{\alpha-1}}{\beta^\alpha}$.

40. If $X \sim \text{Bin}(n,\pi)$, use the binomial expansion to find the mean and variance of X . To find the variance, use the second factorial moment $E[X(X - 1)]$ and note that $\frac{x}{x!} = \frac{1}{(x-1)!}$ when $x > 0$.

41. The speed of a randomly chosen gas molecule in a certain volume of gas is a random variable, V , with probability density function

$$f(v) = \sqrt{\frac{2}{\pi}} \left(\frac{M}{RT} \right)^{\frac{3}{2}} v^2 e^{-\frac{Mv^2}{2RT}} \quad \text{for } v \geq 0$$

where R is the gas constant ($= 8.3145 \text{ J/mol} \cdot \text{K}$), M is the molecular weight of the gas, and T is the absolute temperature measured in degrees Kelvin.

(Hints:

$$\int_0^\infty x^k e^{-x^2} dx = \frac{1}{2} \Gamma\left(\frac{k+1}{2}\right), \quad \Gamma(\alpha+1) = \alpha \Gamma(\alpha), \text{ and } \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}.$$

- (a) Derive a general expression for the average speed of a gas molecule.
- (b) If $1 \text{ J} = 1 \text{ kg} \cdot \text{m}^2/\text{s}^2$, what are the units for the answer in part (a)?
- (c) Kinetic energy for a molecule is $E_k = \frac{Mv^2}{2}$. Derive a general expression for the average kinetic energy of a molecule.
- (d) The weight of hydrogen is 1.008 g/mol . Note that there are 6.0221415×10^{23} molecules in 1 mole. Find the average speed of a hydrogen molecule at 300°K using the result from part (a).
- (e) Use numerical integration to verify the result from part (d).
- (f) Show the probability density functions for the speeds of hydrogen, helium, and oxygen on a single graph. The molecular weights for these elements are 1.008 g/mol , 4.003 g/mol , and 16.00 g/mol , respectively.

42. The Laplace distribution, also known as a double exponential, has a **pdf** given by

$$f(x) = \frac{\lambda}{2} \cdot e^{-\lambda|x-\mu|}, \text{ where } -\infty < x < \infty, -\infty < \mu < \infty, \lambda > 0.$$

- (a) Find the theoretical mean and variance of a Laplace distribution. (Hint: Integrals of absolute values should be done as a positive and negative part, in this case, with limits from $-\infty$ to μ and from μ to ∞ .)
- (b) Let X_1 and X_2 be independent exponential random variables, each with parameter λ . The distribution of $Y = X_1 - X_2$ is a Laplace distribution with a mean of zero and a standard deviation of $\sqrt{2}/\lambda$. Set the seed equal to 3, and generate 25,000 X_1 values from an $\text{Exp}(\lambda = \frac{1}{2})$ and 25,000 X_2 values from another $\text{Exp}(\lambda = \frac{1}{2})$ distribution. Use these values to create the simulated distribution of $Y = X_1 - X_2$.
- (i) Superimpose a Laplace distribution over a density histogram of the Y values. (Hint: The R function `curve()` can be used to superimpose the Laplace distribution over the density histogram.)
 - (ii) Is the mean of Y within 0.02 of the theoretical mean?
 - (iii) Is the variance of Y within 2% of the theoretical variance?

43. A tombola is a raffle in which prizes are assigned to winning tickets. In a particular tombola, only 2 tickets out of n win a prize. After the two winning tickets are sold, a new tombola is started. The tickets are sold consecutively, and the prize is immediately announced when one person wins. Two friends have decided to play tombola in the following way: One of them buys the first ticket on sale, and the other one buys the first ticket after the first prize has been announced. Derive the probability that each of them wins a prize. If there are m tombolas during the night in which the two friends participate, what is the probability that each of them wins more than one prize?

44. Consider the World Cup Soccer data stored in the data frame **SOCGER**. The observed and expected number of goals for a 90-minute game were computed in the “Poisson: World Cup

Soccer Example” in this chapter. To verify that the Poisson rate λ is constant, compute the observed and expected number of goals with the time intervals 45, 15, 10, 5, and 1 minute(s). Compute the means and variances for both the observed and expected counts in each time interval. Based on the results, is the probability of exactly one outcome in a sufficiently short interval proportional to the length of the interval?

45. A communication system consists of n components, where the probability that each component works is π . The system will work if at least half of its components work. For what values of π will a system consisting of 5 components have a greater probability of working than a system consisting of 3 components? Plot the probability each system ($n = 5$ and $n = 3$) works for values of π from 0 to 1 in increments of 0.01.

Chapter 5

Multivariate Probability Distributions

5.1 Joint Distribution of Two Random Variables

In Sections 3.4.1 and 3.4.2, respectively, both discrete and continuous random variables were defined; however, it stands to reason that many random variables might be defined over the same sample space. In random variable example 1 on page 214, the random variable X was defined as the sum of the numbers from rolling two dice; furthermore, one might also wish to consider “the product of the numbers rolled with the two dice” or “the absolute value of the difference between the numbers rolled with the two dice” as additional random variables that are defined on the same sample space. Another example might be the verbal (X) and quantitative (Y) scores on a standardized test for incoming freshmen at a private college. In this section, a brief overview for both discrete and continuous **pdfs** and **cdfs** of jointly distributed random variables is provided as well as some important properties associated with jointly distributed random variables.

5.1.1 Joint pdf for Two Discrete Random Variables

If X and Y are discrete random variables, the function given by

$$p_{X,Y}(x,y) = \mathbb{P}(X = x, Y = y) \quad (5.1)$$

for each pair of values (x,y) within the domain of X and Y is called the joint **pdf** of X and Y . Any function $p_{X,Y}(x,y)$ can be used as a joint **pdf** provided the following properties are satisfied:

- (i) $p_{X,Y}(x,y) \geq 0$ for all x and y ,
- (ii) $\sum_x \sum_y p_{X,Y}(x,y) = 1$, and
- (iii) $\mathbb{P}[(X, Y) \in A] = \sum_{(x,y) \in A} p_{X,Y}(x,y).$

Property (iii) states that when A is composed of pairs of (x,y) values, the probability $\mathbb{P}[(X, Y) \in A]$ is obtained by summing the joint **pdf** over pairs in A .

Example 5.1 ▷ **Joint Distribution: Mathematics Grades** ◁ To graduate with a bachelor of science (B.S.) degree in mathematics, all majors must pass Calculus III and Linear Algebra with a grade point of 3 or better. The population of B.S. graduates in mathematics earned grades as given in Table 5.1 on the following page.

- (a) What is the probability of getting a 3 or better in Linear Algebra?
- (b) What is the probability of getting a 3 or better in Calculus III?

Table 5.1: B.S. graduate grades in Linear Algebra and Calculus III

		Linear Algebra			
		4	3	2	Total
Calculus III	4	2	13	6	21
	3	5	85	40	130
	2	7	33	9	49
Total		14	131	55	200

(c) What is the probability of getting a 3 or better in both Calculus III and Linear Algebra?

Solution: The answers are as follows:

(a) Let the random variables X and Y represent the number of points (on a 4 point scale) students earned in Calculus III and Linear Algebra, respectively from a population of 200 students. If E represents the pairs of Calculus III and Linear Algebra values such that the grade in Linear Algebra is a 3 or better, then the probability of getting a 3 or better in Linear Algebra is written

$$\mathbb{P}[(X, Y) \in E] = \sum_{(x,y) \in E} \sum p_{X,Y}(x, y) = \frac{2 + 5 + 7 + 13 + 85 + 33}{200} = \frac{145}{200}.$$

(b) Let the random variables X and Y represent the number of points (on a 4 point scale) students earned in Calculus III and Linear Algebra, respectively from a population of 200 students. If E represents the pairs of Calculus III and Linear Algebra values such that the grade in Calculus III is a 3 or better, then the probability of getting a 3 or better in Calculus III is written

$$\mathbb{P}[(X, Y) \in E] = \sum_{(x,y) \in E} \sum p_{X,Y}(x, y) = \frac{2 + 13 + 6 + 5 + 85 + 40}{200} = \frac{151}{200}.$$

(c) Let the random variables X and Y represent the number of points (on a 4 point scale) students earned in Calculus III and Linear Algebra, respectively from a population of 200 students. If E represents the pairs of Calculus III and Linear Algebra values such that the grade in both Calculus III and Linear Algebra is a 3 or better, then the probability of getting a 3 or better in both Calculus III and Linear Algebra is written

$$\mathbb{P}[(X, Y) \in E] = \sum_{(x,y) \in E} \sum p_{X,Y}(x, y) = \frac{2 + 5 + 13 + 85}{200} = \frac{105}{200}.$$



For any random variables X and Y , the joint **cdf** is defined in (5.2), while the marginal **pdfs** of X and Y , denoted $p_X(x)$ and $p_Y(y)$, respectively, are defined in Equations (5.3) and (5.4):

$$F_{X,Y}(x, y) = \mathbb{P}(X \leq x, Y \leq y), \quad -\infty < x < \infty, \quad -\infty < y < \infty \quad (5.2)$$

$$p_X(x) = \sum_y p_{X,Y}(x, y) \quad (5.3)$$

$$p_Y(y) = \sum_x p_{X,Y}(x, y). \quad (5.4)$$

In (a) of Example 5.1 on page 315, the problem requests the probability of getting a 3 or better in Linear Algebra. Another way to compute the answer is by adding the two marginals $p_Y(4) + p_Y(3) = \frac{14}{200} + \frac{131}{200} = \frac{145}{200}$. Likewise, (b) of Example 5.1 on page 315 can also be solved with the marginal distribution for X : $p_X(4) + p_X(3) = \frac{21}{200} + \frac{130}{200} = \frac{151}{200}$.

5.1.2 Joint pdf for Two Continuous Random Variables

In Section 3.4.2 on page 222, property (3) for continuous **pdfs** states that the probability the observed value for the random variable X falls in the interval (a, b) is the integral of the **pdf** $f(x)$ over the interval (a, b) . In a similar fashion, the probability that the pair of random variables (X, Y) falls in a two-dimensional region (say A) is obtained by integrating the joint **pdf** over the region A . The joint **pdf** of two continuous random variables is any integrable function $f_{X,Y}(x, y)$ with the following properties:

- (1) $f_{X,Y}(x, y) \geq 0$ for all x and y .
- (2) $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx dy = 1$.
- (3) $\mathbb{P}[(X, Y) \in A] = \iint_{(x,y) \in A} f_{X,Y}(x, y) dx dy$.

Property (3) implies that $\mathbb{P}[(X, Y) \in A]$ is the volume of a solid over the region A bounded by the surface $f_{X,Y}(x, y)$.

For any random variables X and Y , the joint **cdf** is defined in (5.5), while the marginal **pdfs** of X and Y , denoted $f_X(x)$ and $f_Y(y)$, respectively, are defined in Equations (5.6) and (5.7):

$$F_{X,Y}(x, y) = \int_{-\infty}^x \int_{-\infty}^y f_{X,Y}(r, s) ds dr, \quad -\infty < x < \infty, \quad -\infty < y < \infty. \quad (5.5)$$

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) dy, \quad -\infty < x < \infty. \quad (5.6)$$

$$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx, \quad -\infty < y < \infty. \quad (5.7)$$

Example 5.2 Given the joint continuous **pdf**

$$f_{X,Y}(x, y) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1, \quad 0 \leq y \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

- (a) Find $F_{X,Y}(x = 0.6, y = 0.8)$.
- (b) Find $\mathbb{P}(0.25 \leq X \leq 0.75, 0.1 \leq Y \leq 0.9)$.
- (c) Find $f_X(x)$.

Solution: The answers are as follows:

(a)

$$F_{X,Y}(x = 0.6, y = 0.8) = \int_0^{0.6} \int_0^{0.8} f_{X,Y}(r, s) ds dr = \int_0^{0.6} \int_0^{0.8} 1 ds dr = \int_0^{0.6} 0.8 dr = 0.48.$$

(b)

$$\mathbb{P}(0.25 \leq x \leq 0.75, 0.1 \leq y \leq 0.9)$$

$$= \int_{0.25}^{0.75} \int_{0.1}^{0.9} f_{X,Y}(r,s) ds dr = \int_{0.25}^{0.75} \int_{0.1}^{0.9} 1 ds dr = \int_{0.25}^{0.75} 0.8 dr = 0.40.$$

(c)

$$f_X(x) = \int_0^1 f_{X,Y}(x,y) dy = 1, \quad 0 \leq x \leq 1. \quad \blacksquare$$

Example 5.3 ▷ **Joint PDF** ◁ Find the value c to make $f_{X,Y}(x,y) = cx$ a valid joint pdf for $x > 0$, $y > 0$, and $2 < x + y < 3$.

Solution: The domain of interest is lightly shaded in Figure 5.1. To solve the problem, first compute the volume bounded by $x = 0$, $y = 0$, and $y = 3 - x$ beneath the surface $f_{X,Y}(x,y) = cx$, which is denoted $V1$. Next, find the volume bounded by $x = 0$, $y = 0$, and $y = 2 - x$ beneath the surface $f_{X,Y}(x,y) = cx$, which is denoted $V2$. For $f_{X,Y}(x,y)$ to be a valid pdf, c must be found such that the difference between $V1$ and $V2$ is one.

$$V1 = \int_0^3 \int_0^{3-x} cx dy dx = c \int_0^3 (3x - x^2) dx = c \left[\frac{3x^2}{2} - \frac{x^3}{3} \Big|_0^3 \right] = \frac{27c}{6}$$

$$V2 = \int_0^2 \int_0^{2-x} cx dy dx = c \int_0^2 (2x - x^2) dx = c \left[x^2 - \frac{x^3}{3} \Big|_0^2 \right] = \frac{8c}{6}$$

$$V1 - V2 = \frac{27c}{6} - \frac{8c}{6} \stackrel{\text{set}}{=} 1 \Rightarrow c = \frac{6}{19}$$

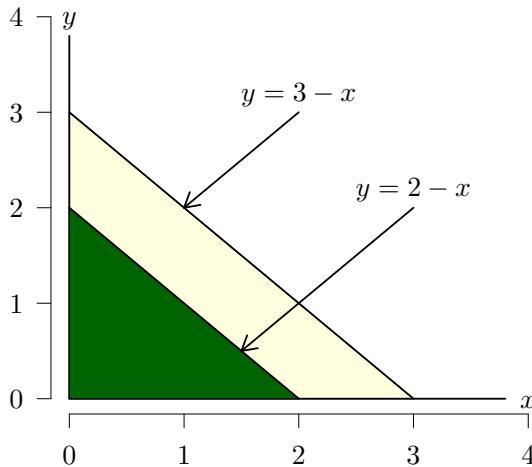


FIGURE 5.1: Graphical representation of the domain of interest for Example 5.3

5.2 Independent Random Variables

In Section 3.3.6 on page 213, it was shown that two events, E and F , are independent if $\mathbb{P}(E \cap F) = \mathbb{P}(E) \cdot \mathbb{P}(F)$. In a similar fashion, two random variables are independent if for every pair of x and y values, $p_{X,Y}(x,y) = p_X(x) \cdot p_Y(y)$, when X and Y are discrete, or $f_{X,Y}(x,y) = f_X(x) \cdot f_Y(y)$ when X and Y are continuous.

Example 5.4 Use Table 5.1 on page 316 to decide if the random variables X , grade in Calculus III, and Y , grade in Linear Algebra, are dependent.

Solution: The random variables X and Y are dependent if $p_{X,Y}(x,y) \neq p_X(x) \cdot p_Y(y)$ for any (x,y) . Consider the pair $(x,y) = (4,4)$, that is, a 4 in both Calculus III and in Linear Algebra.

$$\begin{aligned} p_{X,Y}(4,4) &\stackrel{?}{=} p_X(4) \cdot p_Y(4) \\ \frac{2}{200} &\stackrel{?}{=} \frac{21}{200} \cdot \frac{14}{200} \\ \frac{2}{200} &\neq \frac{21 \times 14}{40,000} \\ 0.01 &\neq 0.00735 \end{aligned}$$

Since $0.01 \neq 0.00735$, the random variables X and Y , the grades in Calculus III and Linear Algebra, respectively, are dependent. It is important to note that the definition of independence requires all the joint probabilities to be equal to the product of the corresponding row and column marginal probabilities. Consequently, if the joint probability of a single entry is not equal to the product of the corresponding row and column marginal probabilities, the random variables in question are said to be dependent. ■

Example 5.5 Are the random variables X and Y in Example 5.2 on page 317 independent? Recall that the **pdf** for Example 5.2 was defined as

$$f_{X,Y}(x,y) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1, \quad 0 \leq y \leq 1 \text{ and} \\ & \\ 0 & \text{otherwise.} \end{cases}$$

Solution: Since the marginal **pdf** for X , $f_X(x) = 1$, and the marginal **pdf** for Y , $f_Y(y) = 1$, it follows that X and Y are independent since $f_{X,Y}(x,y) = f_X(x) \cdot f_Y(y)$ for all x and y . ■

5.3 Several Random Variables

This section examines the joint **pdf** of several random variables by extending the material presented for the joint **pdf** of two discrete random variables and two continuous random variables covered in Section 5.1.1. The joint **pdf** of X_1, X_2, \dots, X_n discrete random variables is any function $p_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = \mathbb{P}(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$ provided the following properties are satisfied:

- (a) $p_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) \geq 0$ for all x_1, x_2, \dots, x_n .
- (b) $\sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} p_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) = 1$.
- (c) $\mathbb{P}[(X_1, X_2, \dots, X_n) \in A] = \sum_{(x_1, x_2, \dots, x_n) \in A} p_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n)$.

The joint **pdf** of X_1, X_2, \dots, X_n continuous random variables is any integrable function $f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n)$ such that following properties are satisfied:

- (a) $f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) \geq 0$ for all x_1, x_2, \dots, x_n .
- (b) $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) dx_1 dx_2 \cdots dx_n = 1$
- (c)

$$\mathbb{P}[(X_1, X_2, \dots, X_n) \in A] = \iint_{(x_1, x_2, \dots, x_n) \in A} \cdots \int f_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n) dx_1 dx_2 \cdots dx_n$$

Independence for several random variables is a generalization of the independence between two random variables. X_1, X_2, \dots, X_n are independent if, for every subset of the random variables, the joint **pdf** of the subset is equal to the product of the marginal **pdfs**. Further, if X_1, X_2, \dots, X_n are independent random variables with respective moment-generating functions $M_{X_1}(t), M_{X_2}(t), \dots, M_{X_n}(t)$, then the moment-generating function of $Y = \sum_{i=1}^n c_i X_i$ is

$$M_Y(t) = M_{X_1}(c_1 t) \times M_{X_2}(c_2 t) \times \cdots \times M_{X_n}(c_n t). \quad (5.8)$$

In the case where X_1, X_2, \dots, X_n are independent normal random variables, a theorem for the distribution of $Y = a_1 X_1 + \cdots + a_n X_n$, where a_1, a_2, \dots, a_n are constants, is stated.

Theorem 5.1 If X_1, X_2, \dots, X_n are independent normal random variables, with means μ_i and standard deviations σ_i for $i = 1, 2, \dots, n$, the distribution of $Y = a_1 X_1 + a_2 X_2 + \cdots + a_n X_n$, where a_1, a_2, \dots, a_n are constants, is normal with mean $E[Y] = a_1 \mu_1 + a_2 \mu_2 + \cdots + a_n \mu_n$ and variance $Var[Y] = a_1^2 \sigma_1^2 + a_2^2 \sigma_2^2 + \cdots + a_n^2 \sigma_n^2$. In other words,

$$Y \sim N\left(\sum_{i=1}^n a_i \mu_i, \sqrt{\sum_{i=1}^n a_i^2 \sigma_i^2}\right).$$

Proof: Since $X_i \sim N(\mu_i, \sigma_i)$, the **mgf** for X_i is $M_{X_i}(t) = e^{\mu_i t + \frac{\sigma_i^2 t^2}{2}}$ using the **mgf** from (4.23). Further, since the X_1, X_2, \dots, X_n are independent,

$$\begin{aligned} M_Y(t) &= M_{X_1}(ta_1) \times M_{X_2}(ta_2) \times \cdots \times M_{X_n}(ta_n) \\ &= e^{t \sum_{i=1}^n a_i \mu_i + t^2 \sum_{i=1}^n \frac{a_i^2 \sigma_i^2}{2}}, \end{aligned}$$

which is the moment-generating function for a normal random variable with mean $\sum_{i=1}^n a_i \mu_i$ and variance $\sum_{i=1}^n a_i^2 \sigma_i^2$.

Example 5.6 A small town in the Pyrenean mountains wants to reduce the bear population because several sheep have recently been killed by bears. Three autonomous communities (Cataluña, Aragón, and Navarra) have made bids to remove 10 bears. The three autonomous communities indicated in their bids that they are willing to spend 5, 7.5, and 10 thousand dollars per bear to capture the bears. Decide which autonomous communities can capture 10 bears with a probability of at least 0.999 knowing that the cost to capture a bear follows a normal distribution with a mean of 5 thousand dollars and a standard deviation of 0.6 thousand dollars.

Solution: Assume that the costs to capture the bears act as independent random variables, such that if X_i is the cost to capture one bear, the total cost to capture 10 bears is also a random variable, given by $Y = X_1 + \dots + X_{10}$. Since $X_i \sim N(5, 0.6)$, it follows using Theorem 5.1 on the facing page that the mean of Y will be $5 \cdot 10 = 50$ and the standard deviation of Y will be $\sqrt{10 \cdot (0.6)^2} = 1.8974$. Mathematically, write $Y \sim N(50, 1.8974)$. Cataluña will be able to capture 10 bears provided $Y \leq 50$, Aragón will be able to capture 10 bears provided $Y \leq 75$, and Navarra will be able to capture 10 bears provided $Y \leq 100$. The probabilities of these events are

$$\begin{aligned}\mathbb{P}(Y \leq 50) &= \mathbb{P}\left(Z \leq \frac{50 - 50}{1.8974}\right) = \mathbb{P}(Z \leq 0) = 0.5, \\ \mathbb{P}(Y \leq 75) &= \mathbb{P}\left(Z \leq \frac{75 - 50}{1.8974}\right) = \mathbb{P}(Z \leq 13.1762) = 1, \\ \text{and } \mathbb{P}(Y \leq 100) &= \mathbb{P}\left(Z \leq \frac{100 - 50}{1.8974}\right) = \mathbb{P}(Z \leq 26.3523) = 1.\end{aligned}$$

The following R code computes the answers directly:

```
> pnorm(50, 50, 1.8974)
[1] 0.5
> pnorm(75, 50, 1.8974)
[1] 1
> pnorm(100, 50, 1.8974)
[1] 1
```

There is only a 50% chance that the Catalan bid would provide sufficient funds to catch 10 bears. On the other hand, the bids from Navarra and Aragón would both have a 100% chance of catching all 10 bears. ■

Example 5.7 Use moment generating functions to show that the sum of two independent Poisson random variables is a Poisson random variable.

Solution: First recall that the **mgf** of a Poisson random variable is $M_X(t) = e^{\lambda(e^t - 1)}$. If X is a Poisson random variable with mean λ and Y is a Poisson random variable with mean μ , then $Z = X + Y$ is also a Poisson random variable with mean $\lambda + \mu$ since

$$M_Z(t) = M_X(t) \times M_Y(t) = e^{\lambda(e^t - 1)} \times e^{\mu(e^t - 1)} = e^{(\lambda + \mu)(e^t - 1)}.$$
■

5.4 Conditional Distributions

Suppose X and Y represent the respective lifetimes (in years) for the male and the female in married couples. If $X = 72$, what is the probability that $Y \geq 75$? In other words, if the male partner of a marriage dies at age 72, how likely is it that the surviving female will live to an age of 75 or more? Questions of this type are answered with conditional distributions. Given two discrete random variables, X and Y , define the conditional **pdf** of X given that $Y = y$ provided that $p_Y(y) > 0$ as

$$p_{X|Y}(x|y) = \mathbb{P}(X = x|Y = y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}. \quad (5.9)$$

If the random variables are continuous, the conditional **pdf** of X given that $Y = y$ provided that $f_Y(y) > 0$ is defined as

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}. \quad (5.10)$$

In addition, if X and Y are jointly continuous over an interval A ,

$$\mathbb{P}(X \in A|Y = y) = \int_A f_{X|Y}(x|y) dx.$$

Example 5.8 Let the random variables X and Y have a joint **pdf**:

$$f_{X,Y}(x,y) = \begin{cases} \frac{12}{5}x(2-x-y) & \text{for } 0 < x < 1, 0 < y < 1 \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Find the **pdf** of X given $Y = y$, for $0 < y < 1$.

Solution: Using the definition for the conditional **pdf** of X given $Y = y$ from (5.10), write

$$\begin{aligned} f_{X|Y}(x|y) &= \frac{f_{X,Y}(x,y)}{f_Y(y)} = \frac{f_{X,Y}(x,y)}{\int_{-\infty}^{\infty} f_{X,Y}(x,y) dx} = \frac{x(2-x-y)}{\int_0^1 x(2-x-y) dx} \\ &= \frac{x(2-x-y)}{2/3 - y/2} = \frac{6x(2-x-y)}{4 - 3y} \text{ for } 0 < x < 1, 0 < y < 1. \end{aligned}$$



Example 5.9 ▷ **Joint Distribution: Radiators** ◁ A local radiator manufacturer subjects his radiators to two tests. The function that describes the percentage of radiators that pass the two tests is

$$f_{X,Y}(x,y) = 8xy, \quad 0 \leq y \leq x \leq 1. \quad (5.11)$$

The random variable X represents the percentage of radiators that pass test A , and Y represents the percentage of radiators that pass test B .

- (a) Is the function given in (5.11) a **pdf**?
- (b) Determine the marginal and conditional **pdfs** for X and Y .
- (c) Are X and Y independent?
- (d) Compute the probability that less than $\frac{1}{8}$ of the radiators will pass test B given that $\frac{1}{2}$ have passed test A .
- (e) Compute the quantities: $E[X]$, $E[X^2]$, $Var(X)$, $E[Y]$, $E[Y^2]$, and $Var(Y)$.
- (f) Use R to represent graphically (5.11).

Solution: The answers are as follows:

- (a) The function (5.11) is a **pdf** since $f_{X,Y}(x,y)$ is non-negative and

$$\int_0^1 \int_0^x 8xy \, dy \, dx = 8 \int_0^1 \left[x \int_0^x y \, dy \right] \, dx = 8 \int_0^1 \frac{x^3}{2} \, dx = 1.$$

- (b) The marginal and conditional **pdfs** are

$$\begin{aligned} f_X(x) &= \int f(x,y) \, dy = \int_0^x 8xy \, dy = 4x^3, \quad 0 \leq x \leq 1. \\ f_Y(y) &= \int f(x,y) \, dx = \int_y^1 8xy \, dx = 4y(1-y^2), \quad 0 \leq y \leq 1. \\ f_{X|Y}(x|y) &= \frac{f_{X,Y}(x,y)}{f_Y(y)} = \frac{8xy}{4y(1-y^2)} = \frac{2x}{1-y^2}, \quad y \leq x \leq 1. \\ f_{Y|X}(y|x) &= \frac{f_{X,Y}(x,y)}{f_X(x)} = \frac{8xy}{4x^3} = \frac{2y}{x^2}, \quad 0 \leq y \leq x. \end{aligned}$$

- (c) The random variables X and Y are dependent since $f_{X,Y}(x,y) = 8xy \neq f_X(x) \cdot f_Y(y) = 16x^3y - 16x^3y^3$.

- (d) The probability that $\mathbb{P}(Y < \frac{1}{8} \mid X = \frac{1}{2})$ is computed as

$$\mathbb{P}(Y < \frac{1}{8} \mid X = \frac{1}{2}) = \int_0^{\frac{1}{8}} f_{Y|X}(y|\frac{1}{2}) \, dy = \int_0^{\frac{1}{8}} \frac{2y}{\frac{1}{4}} \, dy = 4y^2 \Big|_0^{\frac{1}{8}} = \frac{1}{16}.$$

(e) The quantities $E[X]$, $E[X^2]$, $Var(X)$, $E[Y]$, $E[Y^2]$, and $Var(Y)$ are

$$E[X] = \int_0^1 x \cdot 4x^3 dx = 4 \int_0^1 x^4 dx = \frac{4}{5}.$$

$$E[X^2] = \int_0^1 x^2 \cdot 4x^3 dx = 4 \int_0^1 x^5 dx = \frac{2}{3}.$$

$$Var(X) = E[X^2] - [E[X]]^2 = \frac{2}{3} - \frac{16}{25} = \frac{2}{75}.$$

$$E[Y] = \int_0^1 y \cdot 4y(1-y^2) dy = 4 \int_0^1 (y^2 - y^4) dy = \frac{8}{15}.$$

$$E[Y^2] = \int_0^1 y^2 \cdot 4y(1-y^2) dy = 4 \int_0^1 (y^3 - y^5) dy = \frac{1}{3}.$$

$$Var(Y) = E[Y^2] - [E[Y]]^2 = \frac{1}{3} - \frac{64}{225} = \frac{11}{225}.$$

(f) R Code 5.1 can be used to create a graph similar to Figure 5.2 on the facing page.

R Code 5.1

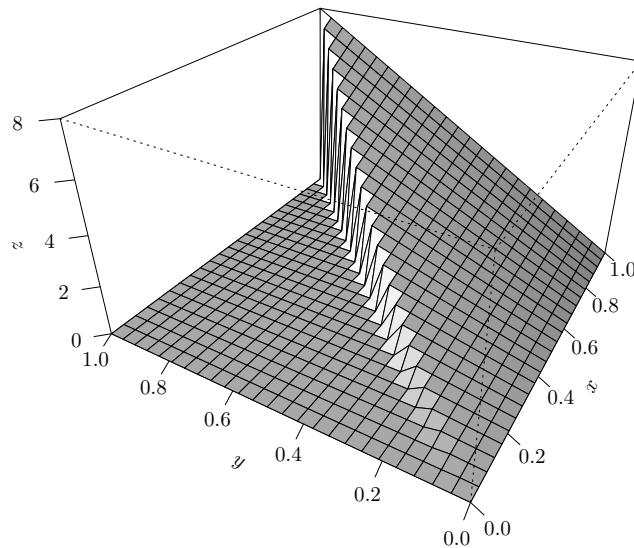
```
> x <- seq(0, 1, length.out = 25)
> y <- x
> f2 <- function(x, y){ifelse(x >= y, 8*x*y, 0)}
> persp(x, y, outer(x, y, f2), shade = 0.6, expand = 0.6,
+         theta = 300, phi = 30, ticktype = "detailed", zlab = "z")
```

R Code 5.1 uses the function `outer()` inside the function `persp()`. The function `outer()` has three arguments: X, Y, and FUN. When the arguments X and Y are passed vectors u and v of length n and m , respectively, and the default value for FUN, `FUN = "*"` is used, the result is an $n \times m$ matrix where each element of u is multiplied by each element of v . This operation is known as an outer product and is denoted by $u \otimes v$. The outer product, $u \otimes v$, is the same as the matrix multiplication uv' , assuming u and v are $n \times 1$ and $m \times 1$ column vectors, respectively. Consider R Code 5.2, which shows how using the outer product and matrix multiplication will return the same matrix.

R Code 5.2

```
> u <- 1:3
> v <- 1:3
> M1 <- outer(X = u, Y = v, FUN = "*")
> M2 <- u %*% t(v)
> M1 == M2

 [,1] [,2] [,3]
[1,] TRUE TRUE TRUE
[2,] TRUE TRUE TRUE
[3,] TRUE TRUE TRUE
```

FIGURE 5.2: Graphical representation of $f_{X,Y}(x,y) = 8xy$, $0 \leq y \leq x \leq 1$ ■

> M1

```
[,1] [,2] [,3]
[1,] 1    2    3
[2,] 2    4    6
[3,] 3    6    9
```

The function `outer()` does not use the default value for `FUN` in R code 5.1 on the preceding page. Instead, Equation (5.11) is defined with function `f2` and `outer()` returns the height of `f2` for each of the 25×25 possible combinations of `x` and `y`. The possible combinations of `x` and `y` in this problem are points in the x - y plane. Consider R Code 5.3, which uses the function `paste` to create a 3×3 grid of points.

R Code 5.3

```
> x <- c("x1", "x2", "x3")
> y <- c("y1", "y2", "y3")
> outer(x, y, paste)

[,1]      [,2]      [,3]
[1,] "x1 y1" "x1 y2" "x1 y3"
[2,] "x2 y1" "x2 y2" "x2 y3"
[3,] "x3 y1" "x3 y2" "x3 y3"

> t(outer(x, y, paste))

[,1]      [,2]      [,3]
[1,] "x1 y1" "x2 y1" "x3 y1"
[2,] "x1 y2" "x2 y2" "x3 y2"
[3,] "x1 y3" "x2 y3" "x3 y3"
```

5.5 Expected Values, Covariance, and Correlation

Three of the important statistics of jointly distributed random variables are expected values, covariance, and correlation. The expected values are measures of the center of the distribution, while covariance and correlation are measures of how two variables are related to one another. Covariance will give both a magnitude and a direction of the relationship including a unit dependence; correlation will measure strength and direction without any dependence on units. All of the statistics in this section can be used to describe important characteristics of jointly distributed random variables.

5.5.1 Expected Values

In Sections 3.4.1.2 on page 217 and 3.4.2.3 on page 229, the expected value for a single random variable for the discrete and continuous cases, respectively, was discussed. Also discussed was the expected value of a function of a random variable. In this section, the expected value of a function of two random variables is examined. When X and Y are jointly distributed random variables with **pdfs** $p_{X,Y}(x,y)$ or $f_{X,Y}(x,y)$, depending on whether the random variables are discrete or continuous, respectively, the expected value of $g(X,Y)$ is

$$E[g(X,Y)] = \begin{cases} \sum_x \sum_y g(x,y) \cdot p_{X,Y}(x,y) & \text{if } X \text{ and } Y \text{ are discrete;} \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x,y) \cdot f_{X,Y}(x,y) dx dy & \text{if } X \text{ and } Y \text{ are continuous.} \end{cases} \quad (5.12)$$

The conditional expectation of X given a value y of Y is written

$$E[X|Y] = \begin{cases} \sum_x x \cdot p_{X|Y}(x|y) & \text{if } X \text{ and } Y \text{ are discrete;} \\ \int_{-\infty}^{\infty} x \cdot f_{X|Y}(x|y) dx & \text{if } X \text{ and } Y \text{ are continuous.} \end{cases} \quad (5.13)$$

Example 5.10 Let the random variables X and Y have a joint **pdf**:

$$f_{X,Y}(x,y) = \frac{e^{-y/x} e^{-x}}{x} \quad x > 0, \quad y > 0.$$

Compute $E[Y|X = x]$.

Solution: First, compute the conditional **pdf** $f_{Y|X}(y|x)$:

$$\begin{aligned} f_{Y|X}(y|x) &= \frac{f_{X,Y}(x,y)}{f_X(x)} = \frac{\frac{e^{-y/x} e^{-x}}{x}}{\int_0^{\infty} \frac{e^{-y/x} e^{-x}}{x} dy} = \frac{\frac{e^{-y/x}}{x}}{\int_0^{\infty} \frac{e^{-y/x}}{x} dy} \\ &= \frac{e^{-y/x}}{x}, \quad x > 0, \quad y > 0. \end{aligned}$$

Using (5.13) for continuous random variables, write

$$E[Y|X = x] = \int_0^{\infty} y \cdot \frac{e^{-y/x}}{x} dy$$

Integrating by parts with $u = y$ and $dv = \frac{e^{-y/x}}{x}$, obtain

$$E[Y|X = x] = -ye^{-y/x} \Big|_0^\infty + \int_0^\infty e^{-y/x} dy = 0 + -xe^{-y/x} \Big|_0^\infty = x, \quad x > 0.$$
■

When two random variables, say X and Y , are independent, recall that $f(x, y) = f_X(x) \cdot f_Y(y)$ for the continuous case and $p_{X,Y}(x, y) = p_X(x) \cdot p_Y(y)$ for the discrete case. Further, $E[XY] = E[X] \cdot E[Y]$. The last statement is true for both continuous and discrete X and Y . A proof for the discrete case is provided. Note that the proof in the continuous case would simply consist of exchanging the summation signs for integral signs.

Proof:

$$\begin{aligned} E[XY] &= \sum_x \sum_y xy p_{X,Y}(x, y) = \sum_x \sum_y xy p_X(x) p_Y(y) \\ &= \sum_y y p_Y(y) \sum_x x p_X(x) = E[Y]E[X]. \end{aligned}$$

Example 5.11 Use the joint pdf provided in Example 5.9 on page 322 and compute $E[XY]$.

Solution:

$$E[XY] = \int_0^1 \int_0^x xy \cdot 8xy dy dx = 8 \int_0^1 \left[x^2 \int_0^x y^2 dy \right] dx = 8 \int_0^1 \frac{x^5}{3} dx = \frac{4}{9}.$$

Note that

$$E[XY] = \frac{4}{9} \neq E[X] \cdot E[Y] = \frac{4}{5} \cdot \frac{8}{15} = \frac{32}{75}$$

further confirming X and Y are dependent as shown in part (c) of Example 5.9 on page 322.

■

5.5.2 Covariance

When two variables, X and Y , are not independent or when it is noted that $E[XY] \neq E[X] \cdot E[Y]$, one is naturally interested in some measure of their dependency. The covariance of X and Y , written $Cov[X, Y]$, provides one measure of the degree to which X and Y tend to move linearly in either the same or opposite directions. The covariance of two random variables X and Y is defined as

$$\begin{aligned} Cov[X, Y] &= E[(X - \mu_X)(Y - \mu_Y)] \\ &= \begin{cases} \sum_x \sum_y (x - \mu_X)(y - \mu_Y)p_{X,Y}(x, y) & X, Y \text{ discrete; and} \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \mu_X)(y - \mu_Y)f(x, y) dx dy & X, Y \text{ continuous.} \end{cases} \end{aligned} \tag{5.14}$$

A $Cov[X, Y] > 0$ indicates that, generally, as X increases, so does Y (that is, X and Y move in the same direction); whereas, a $Cov[X, Y] < 0$ indicates that, generally, as X increases Y decreases (that is, X and Y move in opposite directions). To gain an intuitive understanding of covariance, see Figure 5.3 on the next page, which has both horizontal and vertical dotted lines to indicate μ_{X_i} and μ_{Y_i} in each of the three plots. The first plot

in Figure 5.3 exhibits a strong positive relationship. By this it is meant that large values of X tend to occur with large values of Y and small values of X tend to occur with small values of Y . Consequently, $(x - \mu_{X_1})$ will tend to have the same sign as $(y - \mu_{Y_1})$, so their product will be positive. In the center plot of Figure 5.3, the relationship between the two variables is negative, and note that $(x - \mu_{X_2})$ and $(y - \mu_{Y_2})$ tend to have opposite signs, which makes most of their products negative.

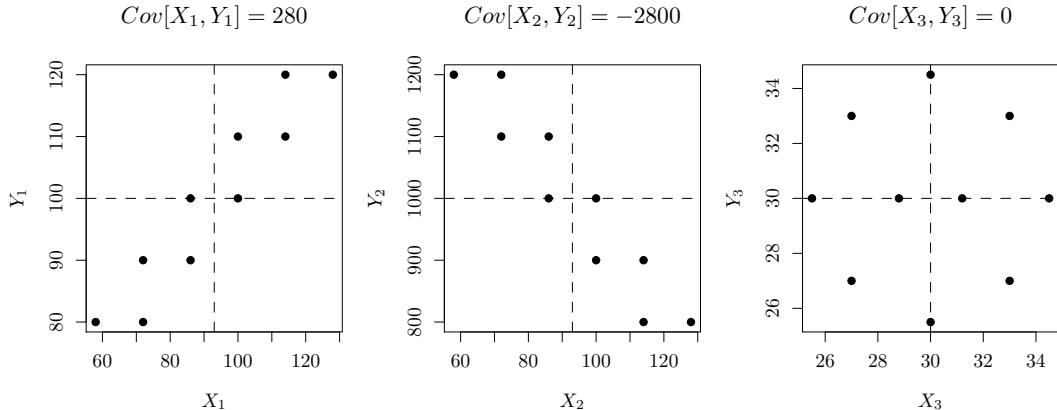


FIGURE 5.3: Scatterplots showing positive, negative, and zero covariance between two random variables where $p_{X,Y}(x,y) = \frac{1}{10}$ for each of the ten pairs of plotted points

Table 5.2: Values used to compute covariance for Figure 5.3

X_1	Y_1	X_2	Y_2	X_3	Y_3
58	80	58	1200	25.5	30.0
72	80	72	1200	27.0	33.0
72	90	72	1100	30.0	34.5
86	90	86	1100	33.0	33.0
86	100	86	1000	34.5	30.0
100	100	100	1000	33.0	27.0
100	110	100	900	30.0	25.5
114	110	114	900	27.0	27.0
114	120	114	800	28.8	30.0
128	120	128	800	31.2	30.0

Example 5.12 Compute the covariance between X_1 and Y_1 for the values provided in Table 5.2 given that $p_{X,Y}(x,y) = \frac{1}{10}$ for each (x,y) pair.

Solution:

$$\begin{aligned}
 p_{X_1}(x) &= \sum_y p_{X_1,Y_1}(x,y) \\
 \mu_{X_1} &= \sum_x x \cdot p_{X_1}(x) = \frac{58 + 72 + \dots + 128}{10} = 93 \\
 \mu_{Y_1} &= \sum_y y \cdot p_{Y_1}(y) = \frac{80 + 80 + \dots + 120}{10} = 100 \\
 Cov[X_1, Y_1] &= \sum_x \sum_y (x - \mu_{X_1})(y - \mu_{Y_1}) p_{X_1,Y_1}(x,y) \\
 &= (58 - 93) \cdot (80 - 100) \cdot \frac{1}{10} + (72 - 93) \cdot (80 - 100) \cdot \frac{1}{10} + \dots \\
 &\quad + (128 - 93) \cdot (120 - 100) \cdot \frac{1}{10} \\
 &= 280.
 \end{aligned}$$

To reduce the arithmetic drudgery, one can solve the problem with R:

```

> X1 <- c(58, 72, 72, 86, 86, 100, 100, 114, 114, 128)
> Y1 <- c(80, 80, 90, 90, 100, 100, 110, 110, 120, 120)
> covar <- function(x, y, f) {
+   sum((x - mean(x)) * (y - mean(y)) * f)
+ }
> covar(X1, Y1, 1/10)

[1] 280

```



At times, it will be easier to work with the shortcut formula $Cov[X, Y] = E[XY] - \mu_X \cdot \mu_Y$ instead of using the definition in (5.14). Additionally, the shortcut formula for the covariance of X with itself yields another expression for the variance of X , $Cov[X, X] = E[X^2] - \mu_x^2 = Var[X]$.

Example 5.13 Compute the covariance between X and Y for Example 5.9 on page 322. In part (e) of Example 5.9, $E[X]$ and $E[Y]$ were computed to be $\frac{4}{5}$ and $\frac{8}{15}$, respectively, and in Example 5.11 on page 327, it was found that $E[XY] = \frac{4}{9}$.

Solution: Using the shortcut formula,

$$Cov[X, Y] = E[XY] - \mu_X \mu_Y = \frac{4}{9} - \frac{4}{5} \cdot \frac{8}{15} = \frac{4}{225}.$$



Rules of Covariance If X and Y are random variables, and a, b, c , and d are constants, then

1. $Cov[X, X] = E[(X - E[X])(X - E[X])] = E[(X - \mu_X)^2] = Var[X]$.
2. $Cov[aX + c, bY + d] = ab \cdot Cov[X, Y]$.

3. If X_i ($1 \leq i \leq n$) and Y_j ($1 \leq j \leq m$) are random variables and a_i ($1 \leq i \leq n$) and b_j ($1 \leq j \leq m$) are constants, then

$$\text{Cov} \left(\sum_{i=1}^n a_i X_i, \sum_{j=1}^m b_j Y_j \right) = \sum_{i=1}^n \sum_{j=1}^m a_i b_j \cdot \text{Cov}[X_i, Y_j]. \quad (5.15)$$

A special case of (5.15) is

$$\begin{aligned} \text{Var} \left[\sum_{i=1}^n a_i X_i \right] &= \text{Cov} \left[\sum_{i=1}^n a_i X_i, \sum_{i=1}^n a_i X_i \right] \\ &= \sum_{i=1}^n a_i^2 \text{Var}[X_i] + 2 \left(\sum_{i=j+1}^n \sum_{j=1}^{m-1} a_i a_j \cdot \text{Cov}[X_i, X_j] \right). \end{aligned}$$

4. $\text{Var}[aX \pm bY] = a^2 \text{Var}[X] + b^2 \text{Var}[Y] \pm 2ab \cdot \text{Cov}[X, Y]$.
5. If X and Y are independent, then $\text{Cov}[X, Y] = 0$ and $\text{Var}[aX \pm bY] = a^2 \text{Var}[X] + b^2 \text{Var}[Y]$.

Be careful not to assume the variance of the sum of two random variables is the sum of the variances of each random variable. If X and Y are independent or $\text{Cov}[X, Y] = 0$ then $\text{Var}[X + Y] = \text{Var}[X] + \text{Var}[Y]$. A simple example to show why this is not true in general is computing $\text{Var}[X + X] \neq \text{Var}[X] + \text{Var}[X]$ since $\text{Var}[X + X] = \text{Var}[2X] = 4\text{Var}[X]$; however, if X_1, X_2, \dots, X_n are n independent random variables with means $\mu_1, \mu_2, \dots, \mu_n$ and variances $\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$, respectively, then the mean and variance of $Y = \sum_{i=1}^n c_i X_i$ where the c_i s are real-valued constants are $\mu_Y = \sum_{i=1}^n c_i \mu_i$ and $\sigma_Y^2 = \sum_{i=1}^n c_i^2 \sigma_i^2$. The proofs of the last two statements are left as exercises for the reader.

Example 5.14 Let X_1, X_2, \dots, X_n be a random sample from a distribution with mean μ and standard deviation σ . Find the mean and variance of $Y = \frac{X_1 + X_2 + \dots + X_n}{n}$.

Solution: In the expression $Y = \frac{X_1 + X_2 + \dots + X_n}{n}$, the c_i values are all $\frac{1}{n}$. Consequently, $\mu_Y = \sum_{i=1}^n \frac{1}{n} \cdot \mu = \mu$ and $\sigma_Y^2 = \sum_{i=1}^n \left(\frac{1}{n}\right)^2 \cdot \sigma^2 = \frac{\sigma^2}{n}$. ■

When one examines the first two plots in Figure 5.3 on page 328, the dependency in the left plot seems to be about as strong as the dependency in the center plot, just in the opposite direction. However, the $\text{Cov}[X, Y] = 280$ in the left plot and $\text{Cov}[X, Y] = -2800$ in the center plot. It turns out that the dependencies are the same (just in opposite directions), but the units of measurement for the Y variable in the center plot are a factor of 10 times larger than those in the left plot. So, it turns out that covariance is unit dependent. If one wishes to eliminate this unit dependency, one can scale the covariance.

5.5.3 Correlation

The **correlation coefficient** between X and Y , denoted $\rho_{X,Y}$, or simply ρ , is a scale independent measure of linear dependency between two random variables. The independence in scale is achieved by dividing the covariance by $\sigma_X \sigma_Y$. Specifically, define the correlation between X and Y as

$$\rho_{X,Y} = \frac{\text{Cov}[X, Y]}{\sigma_X \sigma_Y}. \quad (5.16)$$

The correlation coefficient measures the degree of linear dependency between two random variables and is bounded by -1 and $+1$. The values $\rho = -1$ and $\rho = +1$ indicate perfect negative and positive relationships between two random variables. When $\rho = 0$, there is an absence of linear dependency between X and Y . If X and Y are independent, it is also true that $\rho = 0$; however, $\rho = 0$ does not imply independence. A similar statement is true for the $Cov[X, Y]$. That is, if X and Y are independent, $Cov[X, Y] = 0$; however, $Cov[X, Y] = 0$ does not imply independence.

Example 5.15 Compute $\rho_{X,Y}$ for Example 5.9 on page 322. Recall that $Cov[X, Y] = \frac{4}{225}$ was computed in Example 5.13 on page 329, and $Var[X] = \frac{2}{75}$ and $Var[Y] = \frac{11}{225}$ in part (e) of Example 5.9 on page 322.

Solution:

$$\rho_{X,Y} = \frac{Cov[X, Y]}{\sigma_X \sigma_Y} = \frac{\frac{4}{225}}{\sqrt{\frac{2}{75} \cdot \frac{11}{225}}} = 0.4924$$



Example 5.16 Given the random variables X and Y with their joint probability distribution provided in Table 5.3, verify that although $Cov[X, Y] = 0$, X and Y are dependent.

Table 5.3: Joint probability distribution for X and Y

		Y		
		-1	0	1
X	-1	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
	0	$\frac{1}{8}$	0	$\frac{1}{8}$
	1	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$

Solution: Start by computing the quantities $E[XY]$, $E[X]$, and $E[Y]$ to use in the short-cut formula for the covariance:

$$E[X] = (-1) \cdot \frac{3}{8} + (0) \cdot \frac{2}{8} + (1) \cdot \frac{3}{8} = 0$$

$$E[Y] = (-1) \cdot \frac{3}{8} + (0) \cdot \frac{2}{8} + (1) \cdot \frac{3}{8} = 0$$

$$E[XY] = (-1 \cdot -1) \cdot \frac{1}{8} + \cdots + (1 \cdot 1) \cdot \frac{1}{8} = 0$$

$$Cov[X, Y] = E[XY] - E[X] \cdot E[Y] = 0.$$

The covariance for this problem is 0. However, the random variables are dependent since

$$\mathbb{P}(X = -1, Y = -1) = \frac{1}{8} \neq \mathbb{P}(X = -1) \cdot \mathbb{P}(Y = -1) = \frac{3}{8} \cdot \frac{3}{8} = \frac{9}{64}.$$

This example reinforces the idea that a covariance or correlation coefficient of 0 does not imply independence.



Example 5.17 Compute ρ_{X_1, Y_1} for Example 5.12 on page 328. Recall that $\mu_{X_1} = 93$, $\mu_{Y_1} = 100$, and $Cov[X_1, Y_1] = 280$.

Solution: Start by computing the quantities $E[X_1^2]$, $E[Y_1^2]$, σ_{X_1} , and σ_{X_2} :

$$\begin{aligned} E[X_1^2] &= \sum_x x^2 p_{X_1}(x) \\ &= 58^2 \cdot \frac{1}{10} + 72^2 \cdot \frac{1}{10} + \cdots + 128^2 \cdot \frac{1}{10} = 9090 \\ E[Y_1^2] &= \sum_y y^2 p_{Y_1}(y) \\ &= 80^2 \cdot \frac{1}{10} + 80^2 \cdot \frac{1}{10} + \cdots + 120^2 \cdot \frac{1}{10} = 10200 \\ Var[X_1] &= E[X_1^2] - (E[X_1])^2 = 9090 - 93^2 = 441 \\ \sigma_{X_1} &= \sqrt{Var[X_1]} = \sqrt{441} = 21 \\ Var[Y_1] &= E[Y_1^2] - (E[Y_1])^2 = 10200 - 100^2 = 200 \\ \sigma_{Y_1} &= \sqrt{Var[Y_1]} = \sqrt{200} = 14.1421 \\ \rho_{X_1, Y_1} &= \frac{Cov[X_1, Y_1]}{\sigma_{X_1} \sigma_{Y_1}} = \frac{280}{21 \times 14.1421} = 0.9428. \end{aligned}$$



It is worthwhile to note that $\rho_{X_1, Y_1} = 0.9428$ and $\rho_{X_2, Y_2} = -0.9428$ for the left and center plots, respectively, in Figure 5.3 on page 328. In other words, the correlations have the same absolute magnitude for both plots, even though the absolute values of the covariances differ by a factor of ten.

5.6 Multinomial Distribution

The multinomial distribution is a generalization of the binomial distribution. Recall that each trial in a binomial experiment results in only one of two mutually exclusive outcomes. Experiments where each trial can result in any one of k possible mutually exclusive outcomes A_1, \dots, A_k with probabilities $\mathbb{P}(A_i) = \pi_i$, $0 < \pi_i < 1$, for $i = 1, \dots, k$ such that $\sum_{i=1}^k \pi_i = 1$ can be modeled with the **multinomial distribution**. Specifically, the multinomial distribution computes the probability that A_1 occurs x_1 times, A_2 occurs x_2 times, \dots , A_k occurs x_k times in n independent trials, where $x_1 + x_2 + \cdots + x_k = n$. To derive the probability distribution function, reason in a fashion similar to that done with the binomial. Since the trials are independent, any specified ordering yielding x_1 outcomes for A_1 , x_2 outcomes for A_2, \dots , and x_k outcomes for A_k will occur with probability $\pi_1^{x_1} \pi_2^{x_2} \cdots \pi_k^{x_k}$. The total number of orderings yielding x_1 outcomes for A_1 , x_2 outcomes for A_2, \dots , and x_k outcomes for A_k is $\frac{n!}{x_1! x_2! \cdots x_k!}$. With these two facts in mind, the probability distribution and **mgf** of a multinomial distribution can be derived. Both are found in (5.17).

Multinomial Distribution

$$\mathbf{X} \sim MN(n, \pi_1, \dots, \pi_k)$$

$$\begin{aligned}\mathbb{P}(\mathbf{X} = (x_1, \dots, x_k) | n, \pi_1, \dots, \pi_k) &= \frac{n!}{x_1! x_2! \cdots x_k!} \pi_1^{x_1} \pi_2^{x_2} \cdots \pi_k^{x_k} \\ E[X_i] &= n\pi_i \\ Var[X_i] &= n\pi_i(1 - \pi_i) \\ &\text{given that each } X_i \sim Bin(n, \pi_i) \\ M_{\mathbf{X}}(t) &= (\pi_1 e^{t_1} + \pi_2 e^{t_2} + \cdots + \pi_{k-1} e^{t_{k-1}} + \pi_k e^{t_k})^n\end{aligned}\tag{5.17}$$

Example 5.18 The probability a particular type of light bulb lasts less than 500 hours is 0.5 and the probability the same type of light bulb lasts more than 800 hours is 0.2. In a random sample of ten light bulbs, what is the probability of obtaining exactly four light bulbs that last less than 500 hours and two light bulbs that last more than 800 hours?

Solution: Let the random variables X_1, X_2 , and X_3 denote the number of light bulbs that last less than 500 hours, the number of light bulbs that last between 500 and 800 hours, and the number of light bulbs that last more than 800 hours, respectively. Since $\pi_1 = 0.5$, $\pi_2 = 0.3$, and $\pi_3 = 0.2$, use the first equation in (5.17) and compute $\mathbb{P}(X_1 = 4, X_2 = 4, X_3 = 2)$ as

$$\mathbb{P}(X_1 = 4, X_2 = 4, X_3 = 2 | 10, 0.5, 0.3, 0.2) = \frac{10!}{4!4!2!} (0.5)^4 (0.3)^4 (0.2)^2 = 0.0638.$$

To obtain the answer with R, use the `dmultinom()` function as shown in R Code 5.4.

R Code 5.4

```
> dmultinom(x = c(4, 4, 2), size = 10, prob = c(0.5, 0.3, 0.2))
[1] 0.0637875
```



5.7 Bivariate Normal Distribution

The joint distribution of the random variables X and Y is said to have a **bivariate normal** distribution when its joint density takes the form

$$\begin{aligned}f_{X,Y}(x, y) &= \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 \right.\right. \\ &\quad \left.\left. - 2\rho \left(\frac{x-\mu_X}{\sigma_X}\right) \left(\frac{y-\mu_Y}{\sigma_Y}\right) + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 \right]\right\},\end{aligned}\tag{5.18}$$

for $-\infty < x, y < +\infty$, where $\mu_X = E[X]$, $\mu_Y = E[Y]$, $\sigma_X^2 = \text{Var}[X]$, $\sigma_Y^2 = \text{Var}[Y]$, and ρ is the correlation coefficient between X and Y . An equivalent representation of (5.18) is given in (5.19), where $\mathbf{X} = (X, Y)^T$ is a vector of random variables where T represents the transpose, $\boldsymbol{\mu} = (\mu_X, \mu_Y)^T$, is a vector of constants, and $\boldsymbol{\Sigma}$ is a 2×2 non-singular matrix such that its inverse $\boldsymbol{\Sigma}^{-1}$ exists and the determinant $|\boldsymbol{\Sigma}| \neq 0$, where

$$\boldsymbol{\Sigma} = \begin{pmatrix} \text{Var}[X] & \text{Cov}[X, Y] \\ \text{Cov}[Y, X] & \text{Var}[Y] \end{pmatrix}.$$

$$f(\mathbf{x}) = \frac{1}{(\sqrt{2\pi})^2} |\boldsymbol{\Sigma}|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{X} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{X} - \boldsymbol{\mu}) \right\}. \quad (5.19)$$

The shorthand notation used to denote a multivariate (bivariate being a subset) normal distribution is $\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. In general, $\boldsymbol{\Sigma}$ represents what is called the variance covariance matrix. When $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ and $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_n)^T$, it is defined as

$$\begin{aligned} \boldsymbol{\Sigma} &= E[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^T] = E \left[\begin{pmatrix} X_1 - \mu_1 \\ \vdots \\ X_n - \mu_n \end{pmatrix} (X_1 - \mu_1, \dots, X_n - \mu_n) \right] \\ &= \begin{pmatrix} \sigma_{X_1}^2 & \dots & \text{Cov}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \text{Cov}(X_n, X_1) & \dots & \sigma_{X_n}^2 \end{pmatrix}. \end{aligned}$$

Representations of three bivariate normal distributions, all with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.40, and 0.80, respectively, created with the `persp()` function, are displayed in Figure 5.4 on the next page. R Code 5.5 creates three functions that will be used with `persp()`, `contour()`, and `image()`, to create different representations of the three bivariate normal distributions. The base R functions `persp()`, `contour()`, and `image()`, which draw perspective plots over the x - y plane, contour plots, and heat plots, respectively, all require the data to be formatted in the same fashion. In particular, the `x =`, and `y =` arguments are the locations of grid lines at which the values in `z` are measured. The values provided to `z` must be a matrix. One way to create the matrix of `z` values for each of the values on the x - y plane is to use the function `outer()`.

R Code 5.5

```
> f1 <- function(x, y, p = 0){
+   exp( (x^2 - 2*p*x*y + y^2) / (-2*(1 - p^2)) ) / (2*pi*sqrt(1 - p^2)) }
> f2 <- function(x, y, p = 0.4){
+   exp( (x^2 - 2*p*x*y + y^2) / (-2*(1 - p^2)) ) / (2*pi*sqrt(1 - p^2)) }
> f3 <- function(x, y, p = 0.8){
+   exp( (x^2 - 2*p*x*y + y^2) / (-2*(1 - p^2)) ) / (2*pi*sqrt(1 - p^2)) }
```

R Code 5.6 on the next page can be used to create perspective graphs similar to Figure 5.4 on the facing page.

R Code 5.6

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(1, 3), mar = c(1.1, 1.1, 1.1, 1.1), pty = "s")
> x <- seq(-3, 3, length = 40)
> y <- x
> persp(x, y, outer(x, y, f1), zlab = "z", main = expression(rho == 0),
+        theta = -25, expand = 0.65, phi = 25, shade = 0.4)
> persp(x, y, outer(x, y, f2), zlab = "z", main = expression(rho == 0.4),
+        theta = -25, expand = 0.65, phi = 25, shade = 0.4)
> persp(x, y, outer(x, y, f3), zlab = "z", main = expression(rho == 0.8),
+        theta = -25, expand = 0.65, phi = 25, shade = 0.4)
> par(opar)
```

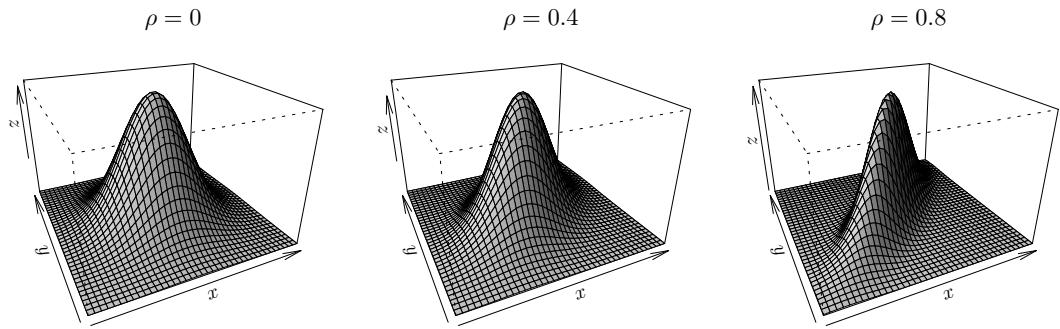


FIGURE 5.4: Bivariate normal densities with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.4, and 0.8, respectively, created with `persp()`

R Code 5.7 creates contour plots similar to those shown in Figure 5.5 on the following page.

R Code 5.7

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(1, 3), mar = c(4.1, 4.1, 4.1, 1.1), pty = "s")
> x <- seq(-3, 3, length = 50)
> y <- x
> contour(x, y, outer(x, y, f1), nlevels = 10, xlab = "x", ylab = "y",
+           main = expression(rho == 0))
> contour(x, y, outer(x, y, f2), nlevels = 10, xlab = "x", ylab = "y",
+           main = expression(rho == 0.4))
> contour(x, y, outer(x, y, f3), nlevels = 10, xlab = "x", ylab = "y",
+           main = expression(rho == 0.8))
> par(opar)
```

R Code 5.8 on the next page creates heat plots similar to those shown in Figure 5.6 on the following page.

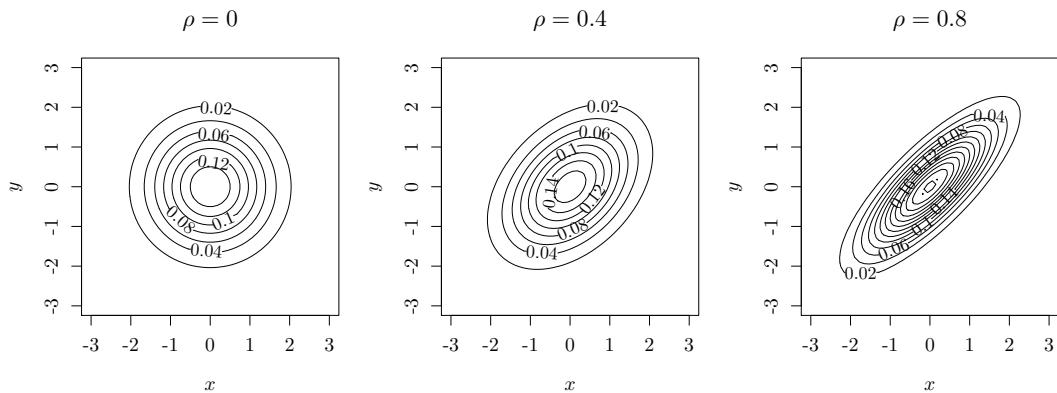


FIGURE 5.5: Bivariate normal densities with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.4, and 0.8, respectively, created with `contour()`

R Code 5.8

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(1, 3), mar = c(4.1, 4.1, 4.1, 1.1), pty = "s")
> x <- seq(-3, 3, length = 50)
> y <- x
> image(x, y, outer(x, y, f1), col = gray((0:100)/100), xlab = "x",
+       ylab = "y", main = expression(rho == 0))
> image(x, y, outer(x, y, f2), col = gray((0:100)/100), xlab = "x",
+       ylab = "y", main = expression(rho == 0.4))
> image(x, y, outer(x, y, f3), col = gray((0:100)/100), xlab = "x",
+       ylab = "y", main = expression(rho == 0.8))
> par(opar)
```

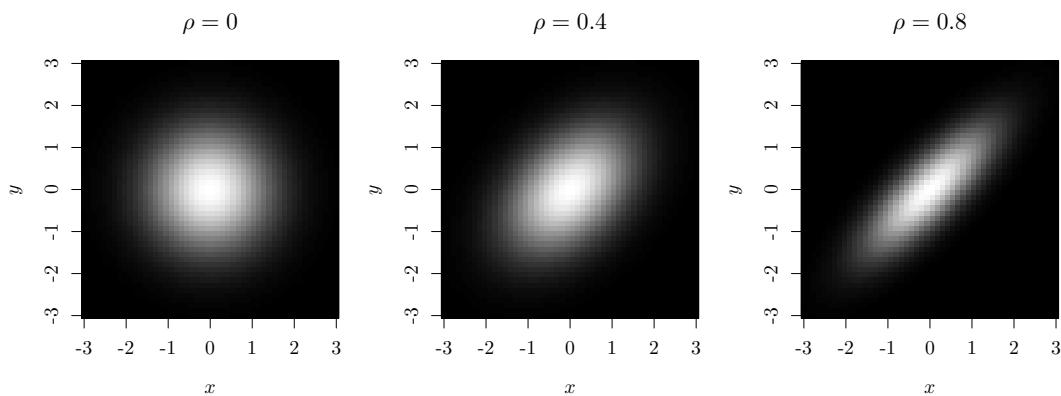


FIGURE 5.6: Bivariate normal densities with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.4, and 0.8, respectively, created with `image()`

The lattice analogs to the base R functions `persp()`, `contour()`, and `image()` are `wireframe()`, `contourplot()`, and `levelplot()`, respectively. The lattice functions

`wireframe()`, `contourplot()`, and `levelplot()` require the data in a slightly different format compared to their base analogs. The function `expand.grid()` is used to create a grid of points where the function will be evaluated. This is similar to using `outer()`, with the exception that `outer()` returned the grid points in a matrix and `expand.grid()` returns the points in a data frame. R Code 5.9 prepares the data to work with `wireframe()`, `contourplot()`, and `levelplot()`.

R Code 5.9

```
> x <- seq(-3, 3, length = 40)
> y <- x
> z1 <- outer(x, y, f1)
> z2 <- outer(x, y, f2)
> z3 <- outer(x, y, f3)
> Grid <- expand.grid(x = x, y = y)
> zp <- c(expression(rho == 0.0), expression(rho == 0.4),
+           expression(rho == 0.8))
```

R Code 5.10 can be used to create wireframe graphs similar to those in Figure 5.7.

R Code 5.10

```
> wireframe( z1 + z2 + z3 ~ x * y, data = Grid, xlab = "x", ylab = "y",
+             zlab = "z", outer = TRUE, layout = c(3, 1),
+             strip = strip.custom(factor.levels = zp) )
```

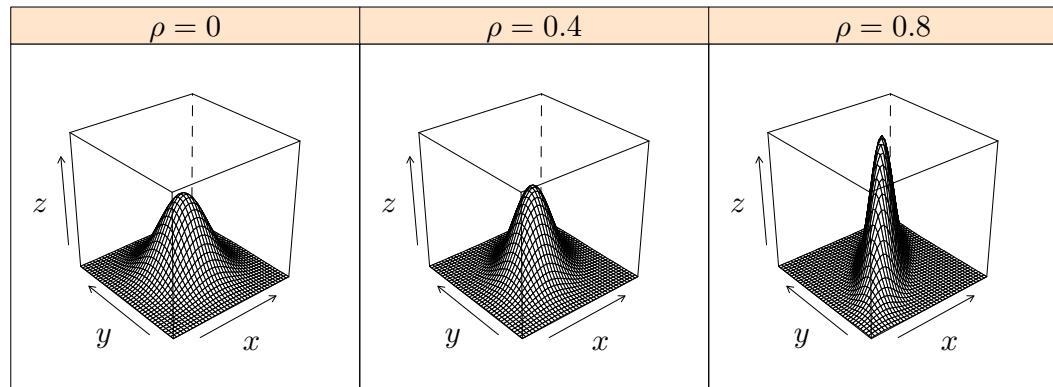


FIGURE 5.7: Bivariate normal densities with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.4, and 0.8, respectively, created with `wireframe()`

R Code 5.11 can be used to create contour graphs similar to those shown in Figure 5.8 on the next page.

R Code 5.11

```
> contourplot(z1 + z2 + z3 ~ x * y, data = Grid, xlab = "x", ylab = "y",
+               outer = TRUE, layout = c(3, 1), aspect = "xy",
+               cuts = 11, strip = strip.custom(factor.levels = zp))
```

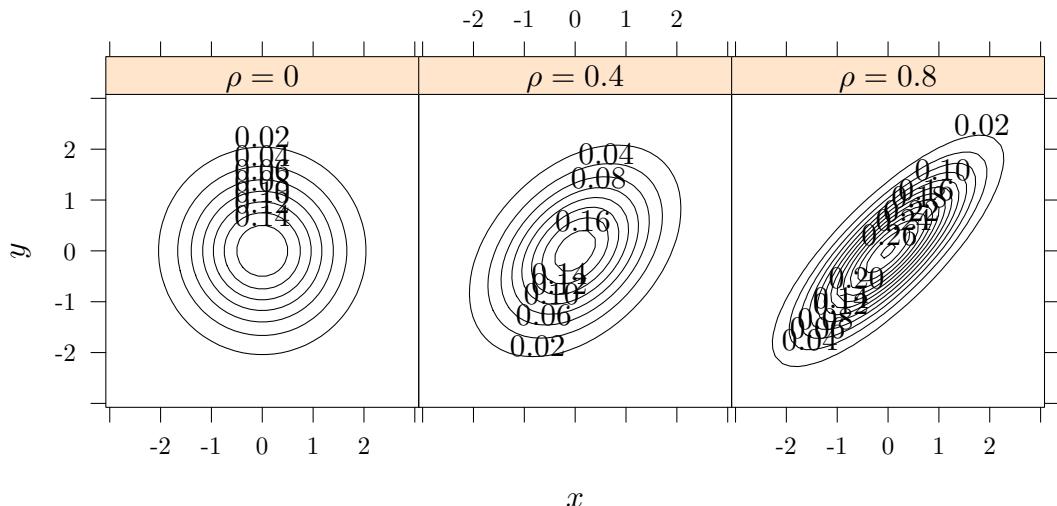


FIGURE 5.8: Bivariate normal densities with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.4, and 0.8, respectively, created with `contourplot()`

R Code 5.12 can be used to create levelplot graphs similar to Figure 5.9.

R Code 5.12

```
> levelplot(z1 + z2 + z3 ~ x * y, data = Grid, xlab = "x", ylab = "y",
+            outer = TRUE, layout = c(3, 1), aspect = "xy",
+            strip = strip.custom(factor.levels = zp))
```

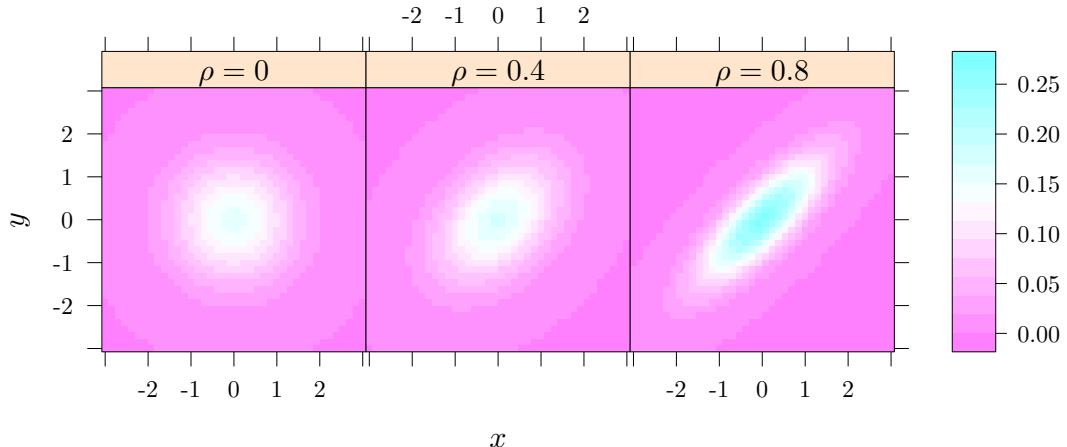


FIGURE 5.9: Bivariate normal densities with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.4, and 0.8, respectively, created with `levelplot()`

The `ggplot2` analogs to the lattice functions `contourplot()` and `levelplot()` are `stat_contour()` and `geom_tile()` or `geom_raster()`. `ggplot2` does not currently render three-dimensional graphs. R Code 5.13 creates the needed data frame to use with `ggplot2`.

R Code 5.13

```
> x <- seq(-3, 3, length = 50)
> y <- x
> z1 <- outer(x, y, f1)
> z2 <- outer(x, y, f2)
> z3 <- outer(x, y, f3)
> Grid <- expand.grid(x = x, y = y)
> DF1 <- data.frame(x = Grid$x, y = Grid$y, z = as.vector(z1))
> DF2 <- data.frame(x = Grid$x, y = Grid$y, z = as.vector(z2))
> DF3 <- data.frame(x = Grid$x, y = Grid$y, z = as.vector(z3))
> DF1$r = "rho == 0.0"
> DF2$r = "rho == 0.4"
> DF3$r = "rho == 0.8"
> BDF <- rbind(DF1, DF2, DF3)
```

R Code 5.14 can be used to create contour graphs similar to Figure 5.10. To get mathematical symbols for the labels of the various facets, the value `label_parsed` is passed to the argument `labeller`. The value `label_parsed` takes strings and treats them as R math expressions.

R Code 5.14

```
> p <- ggplot(data = BDF, aes(x = x, y = y, z = z))
> p + stat_contour(aes(colour = ..level..)) + theme_bw() +
+   scale_colour_gradient(low = "gray10", high = "gray90") +
+   labs(colour = "Density", x = "x", y = "y") +
+   facet_grid(. ~ r, labeller = label_parsed) +
+   coord_fixed(ratio = 1)
```

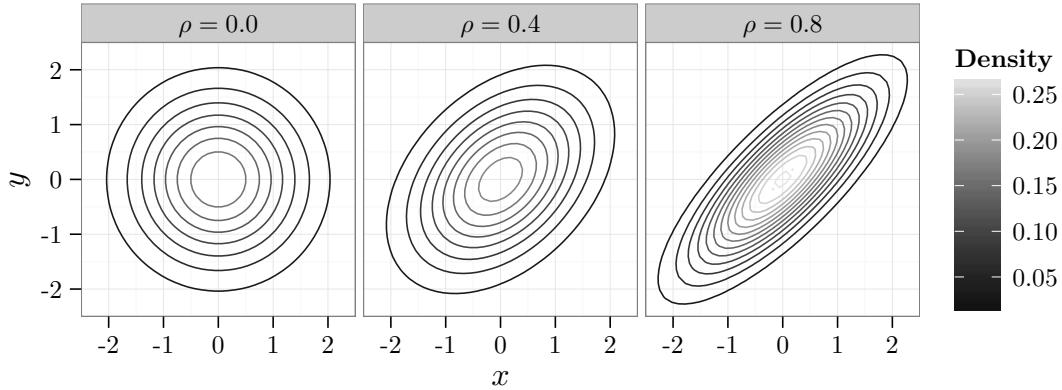


FIGURE 5.10: Bivariate normal densities with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.4, and 0.8, respectively

R Code 5.15 on the next page can be used to create heat maps with the `ggplot2` function `geom_raster()` similar to those shown in Figure 5.11 on the following page.

R Code 5.15

```
> p <- ggplot(data = BDF, aes(x = x, y = y, fill = z))
> p + geom_raster() + theme_bw() +
+   scale_fill_gradient(low = "gray10", high = "gray90") +
+   labs(fill = "Density", x = "x", y = "y") + facet_grid(. ~ r) +
+   coord_fixed(ratio = 1)
```

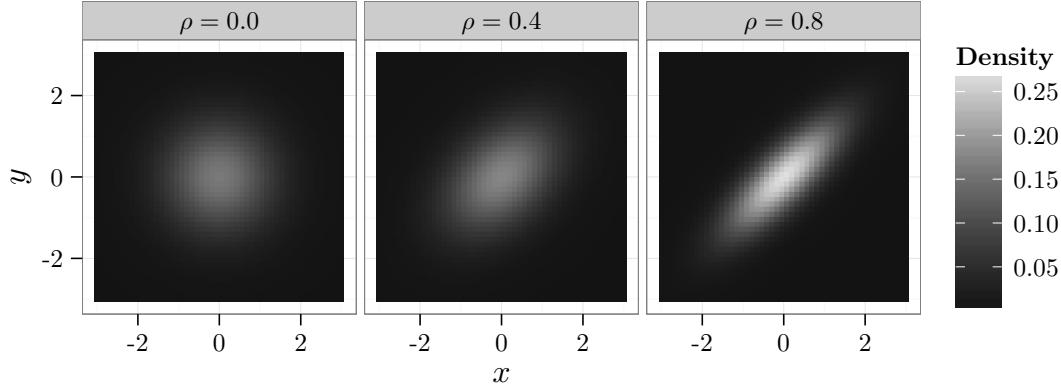


FIGURE 5.11: Bivariate normal densities with parameters $\mu_X = \mu_Y = 0$, $\sigma_X = \sigma_Y = 1$, and ρ values of 0, 0.4, and 0.8, respectively

The following facts about the bivariate normal distribution are listed without proof:

- (a) The marginal distribution of X is $N(\mu_X, \sigma_X)$.
- (b) The marginal distribution of Y is $N(\mu_Y, \sigma_Y)$.
- (c) If X and Y have a bivariate normal distribution, the conditional density of Y given $X = x$ is a normal distribution with mean $\mu_{Y|x} = E(Y|x) = \mu_Y + \rho \frac{\sigma_Y}{\sigma_X}(x - \mu_X)$ and variance $\sigma_{Y|x}^2 = \sigma_Y^2(1 - \rho^2)$.
- (d) Given any two constants a and b , the distribution of $aX + bY$ is

$$N \left(a\mu_X + b\mu_Y, \sqrt{a^2\sigma_X^2 + b^2\sigma_Y^2 + 2ab\rho\sigma_X\sigma_Y} \right).$$

Example 5.19 \triangleright **Bivariate Normal Grades** \triangleleft Assume the distribution of grades for a particular group of students has a bivariate normal distribution with parameters $\mu_X = 3.2$, $\mu_Y = 2.4$, $\sigma_X = 0.4$, $\sigma_Y = 0.6$, and $\rho = 0.6$, where X and Y represent the grade point averages in high school and the first year of college, respectively. Find the following:

- (a) $\mathbb{P}(Y < 1.8)$,
- (b) $\mathbb{P}(Y < 1.8 | X = 2.5)$,
- (c) $\mathbb{P}(Y > 3.0)$, and
- (d) $\mathbb{P}(Y > 3.0 | X = 2.5)$.

Solution: The answers are computed first manually, and then with R.

(a) Using the parameters given in the problem,

$$\mathbb{P}(Y < 1.8) = \mathbb{P}\left(\frac{Y - 2.4}{0.6} < \frac{1.8 - 2.4}{0.6}\right) = \mathbb{P}(Z < -1) = 0.1587.$$

```
> pnorm(1.8, 2.4, 0.6)
[1] 0.1586553
```

(b) First, find the quantities $\mu_{Y|x=2.5}$ and $\sigma_{Y|x=2.5}$:

$$\begin{aligned}\mu_{Y|x=2.5} &= E(Y|x=2.5) = \mu_Y + \rho \frac{\sigma_Y}{\sigma_X} (x - \mu_X) = 2.4 + 0.6 \cdot \frac{0.6}{0.4} \cdot (2.5 - 3.2) = 1.77 \\ \sigma_{Y|x=2.5}^2 &= \sigma_Y^2 (1 - \rho^2) = 0.6^2 \cdot (1 - 0.6^2) = 0.2304 \Rightarrow \sigma_{Y|x=2.5} = 0.48 \\ \mathbb{P}(Y < 1.8|X = 2.5) &= \mathbb{P}\left(\frac{Y-1.77}{0.48} < \frac{1.8-1.77}{0.48}\right) = \mathbb{P}(Z < 0.0625) = 0.5249.\end{aligned}$$

```
> pnorm(1.8, 1.77, 0.48)
[1] 0.5249177
```

(c) Using the parameters given in the problem,

$$\begin{aligned}\mathbb{P}(Y > 3.0) &= 1 - \mathbb{P}(Y \leq 3.0) = 1 - \mathbb{P}\left(\frac{Y - 2.4}{0.6} \leq \frac{3.0 - 2.4}{0.6}\right) \\ &= 1 - \mathbb{P}(Z \leq 1) = 0.1587.\end{aligned}$$

```
> 1 - pnorm(3, 2.4, 0.6)
[1] 0.1586553
```

(d) Using the quantities $\mu_{Y|x}$ and $\sigma_{Y|x}$ from (b),

$$\begin{aligned}\mathbb{P}(Y > 3.0|X = 2.5) &= 1 - \mathbb{P}(Y \leq 3.0 | X = 2.5) \\ &= 1 - \mathbb{P}\left(\frac{Y - 1.77}{0.48} \leq \frac{3.0 - 1.77}{0.48}\right) = 1 - \mathbb{P}(Z \leq 2.5625) \\ &= 0.0052.\end{aligned}$$

```
> 1 - pnorm(3, 1.77, 0.48)
[1] 0.005196079
```



5.8 Problems

1. Let X and Y have the following joint distribution:

Joint Probability Distribution of X and Y

		Y			
		-1	0	1	
X		-1	1/6	0	1/6
		0	1/3	0	0
		1	1/6	0	1/6

- (a) Find the covariance between X and Y .
- (b) Find $\mathbb{P}(X = -1|Y = 1)$.
- (c) Show that X and Y are dependent.

2. Given the random variables X and Y and their joint probability $p_{X,Y}(X, Y)$:

		Y			
		1	2	3	
X		1	0.05	0.05	0.1
		2	0.05	0.1	0.35
		3	0	0.2	0.1

- (a) Show that $\sum_x \sum_y p_{X,Y}(X, Y) = 1$.
- (b) Find the mean of X and the mean of Y .
- (c) Find $\mathbb{P}(X \leq 1, Y \leq 2)$.
- (d) Are X and Y independent?
- (e) Find the variances of X and of Y .
- (f) Find the covariance of X and Y .

3. A particular unfair coin is constructed so that the probability of obtaining a head is $\frac{1}{3}$. The unfair coin is flipped twice. Define two random variables: Z = the number of heads in the first flip and W = the number of heads in two flips.

- (a) Construct a table showing the joint probability distribution of both random variables Z and W including the marginal probabilities.
- (b) Find the covariance between Z and W . Are they independent?
- (c) Suppose the covariance between Z and W were 0. Would this imply that Z and W are independent?

4. An international travel agency translates its promotional fliers each season. Translators are hired to translate the fliers into several languages. The translators are paid either €60 or €90 per page, depending on word density. The fliers are all either 5, 7, or 10 pages in length. The joint density function for X and Y , where X = number of pages and Y = price per page, is

		Y	
		60	90
X	5	0.05	0.4
	7	0.05	0.1
	10	0.35	0.05

- (a) Find the mean and variance of X and Y .
- (b) Find $\text{Cov}[X, Y]$, and explain its meaning.
- (c) Find the probability function of Z (the total translation cost).
- (d) Find the mean of Z .
5. A student uses a free dialup service to access the Internet. Depending on the server to which the Internet service provider connects the student, there are three transmission rates: 1800, 2700, and 3600 bytes per second. Let X be the number of transmitted bytes and Y the transmission rate in bytes per second. The joint probability for X and Y is given by the following table:
- | | | Y | | |
|-----|--------|-------|------|-------|
| | | 1800 | 2700 | 3600 |
| X | 64800 | 0.3 | 0.05 | 0.025 |
| | 324000 | 0.025 | 0.15 | 0.15 |
| | 972000 | 0 | 0.2 | 0.1 |
- (a) Let Z be the random variable indicating the time necessary for transmission. Write down the probability function of Z .
- (b) Find the expected time spent in transmission.
- (c) Find the mean and variance of X and Y and $\text{Cov}(X, Y)$.
6. At the local movie theater, drinks and popcorn come in three sizes: small, medium, and large. The prices for both drinks and popcorn are \$1.50, \$2.50, and \$3.50 for the small, medium, and large sizes, respectively. For a given customer, define the random variables X = amount spent for popcorn and Y = amount spent for drinks. Suppose the joint distribution for X and Y is

		X		
		1.5	2.5	3.5
Y	1.5	0.03	0.07	0.05
	2.5	0.08	0.08	0.30
	3.5	0.00	0.30	0.09

- (a) Find the probability a given customer spends no more than \$2.50 on popcorn. What is the probability a given customer spends at least \$2.50 on popcorn?
- (b) What is the average amount of money spent at the movies for a customer buying both popcorn and a drink, if the cost of the movie ticket is \$5.20?
7. The interior diameter of a particular type of test tube is a random variable with a mean of 5 cm and a standard deviation of 0.03 cm. If the test tube thickness is a random variable with a mean of 0.5 cm and a standard deviation of 0.001 cm and both variables are independent, find the mean and standard deviation of the exterior diameter.
8. The flow of water arriving at an irrigation canal is measured in cubic meters and follows a $N(100, 20)$ distribution. The canal has a flow capacity that follows a $N(120, 30)$ distribution. The sluice gate is opened when the water flow exceeds the canal's capacity. What is the probability that the sluice gate will be opened?
9. Jim is preparing pancakes for breakfast. The recipe calls for 1 cup of flour and $\frac{3}{4}$ cup of milk. He is cooking for a family, which requires him to quadruple the recipe. The mean amount of flour he puts into the batter is 1 cup per multiple with a standard deviation of $\frac{1}{16}$ cup. The mean amount of milk he puts into the recipe per multiple is $\frac{3}{4}$ cup with a standard deviation of $\frac{1}{8}$ cup. Each multiple is independent and normally distributed for each ingredient.
- (a) What are the mean and standard deviation of the total amount of milk and flour in Jim's quadrupled pancake recipe?
- (b) What is the chance he ends up with more than $7\frac{1}{2}$ cups of milk and flour together in his batter?
10. A cereal manufacturer is creating a new cereal that will have a mean of 8.2 ounces of oats per box with a standard deviation of 0.25 ounce, 12.4 ounces of flakes with a standard deviation of 3 ounces, and raisins with a mean of 4.1 ounces per box and a standard deviation of $\frac{1}{2}$ ounce.
- (a) What are the mean and standard deviation of the amount of ingredients in the box if ingredients are independent?
- (b) If the box is advertised as containing 24 ounces, what percent of the time will the actual amount be at least what is advertised if all ingredients have independent normal distributions?
11. Emily and Albert are getting married and are looking at combining their finances. Both work in sales, so they have salaries that vary from month to month. They want to know if they can afford to buy a house with payments of \$1350 per month. They only want to spend 30% of their total income on house payments. If Emily's average sales are \$3000 per month with a standard deviation of \$500 and Albert's average sales are \$2000 with a standard deviation of \$1000, and both sales are normally distributed with a covariance between Emily's sales and Albert's sales of 10000 dollars², what percent of the time will they be able to spend less than 30% of their total income on their house payment?
12. Bob's Gas Station sells unleaded regular and super gasoline. They sell regular for \$2.99 a gallon and super at \$3.28 a gallon. If regular sales per day are normally distributed with

a mean of 300 gallons and a standard deviation of 50 gallons and super sales are normally distributed with a mean of 200 gallons and a standard deviation of 80 gallons, and the covariance between regular sales and super sales is -3000,

- (a) What are the mean and standard deviation of total sales per day?
- (b) What percent of the time will the station make more than \$1700 dollars per day?

13. Let be X be a random variable with standard deviation σ_X and let $Y = aX + b$ where a and b are constants. Show that $\rho = 1$ when $a > 0$ and $\rho = -1$ when $a < 0$.

14. Given the joint density function

$$f(x, y) = 6x, \quad 0 < x < y < 1,$$

find the $E[Y | X]$ that is the regression line resulting from regressing Y on X .

15. A poker hand (5 cards) is dealt from a single deck of well-shuffled cards. If the random variables X and Y represent the number of aces and the number of kings in a hand, respectively,

- (a) Write the joint distribution $p_{X,Y}(x, y)$.
- (b) What is the marginal distribution of X , $p_X(x)$?
- (c) What is the marginal distribution of Y , $p_Y(y)$?

$$\left(\text{Hint: } \sum_{y=0}^{\infty} \binom{a}{x} \binom{b}{n-x} = \binom{a+b}{n}. \right)$$

16. If $f_{X,Y}(x, y) = 5x - y^2$ in the region bounded by $y = 0$, $x = 0$, and $y = 2 - 2x$, find the density function for the marginal distribution of X , for $0 < x < 1$.

17. If $f(x, y) = e^{-(x+y)}$, $x > 0$, and $y > 0$, find $\mathbb{P}(X + 3 > Y | X > \frac{1}{3})$.

18. If $f(x, y) = 1$, $0 < x < 1$, $0 < y < 1$, what is $\mathbb{P}(Y - X > \frac{1}{2} | X + Y > \frac{1}{2})$?

19. If $f(x, y) = k(y - 2x)$ is a joint density function over $0 < x < 1$, $0 < y < 1$, and $y > x^2$, then what is the value of the constant k ?

20. Let X and Y have the joint density function

$$f(x, y) = \begin{cases} \frac{4}{3}x + \frac{2}{3}y & \text{for } 0 < x < 1 \text{ and } 0 < y < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Find $\mathbb{P}(2X < 1 | X + Y < 1)$.

21. Let X and Y have the joint density function

$$f(x, y) = \begin{cases} 6(x - y)^2 & \text{for } 0 < x < 1 \text{ and } 0 < y < 1, \\ 0 & \text{otherwise.} \end{cases}$$

- (a) Find $\mathbb{P}(X < \frac{1}{2} | Y < \frac{1}{4})$.
- (b) Find $\mathbb{P}(X < \frac{1}{2} | Y = \frac{1}{4})$.

22. Let X and Y denote the weight (in kilograms) and height (in centimeters), respectively, of 20-year-old American males. Assume that X and Y have a bivariate normal distribution with parameters $\mu_X = 82$, $\sigma_X = 9$, $\mu_Y = 190$, $\sigma_Y = 10$, and $\rho = 0.8$. Find

- (a) $E[Y | X = 75]$,
- (b) $E[Y | X = 90]$,
- (c) $Var[Y | X = 75]$,
- (d) $Var[Y | X = 90]$,
- (e) $\mathbb{P}(Y \geq 190 | X = 75)$, and
- (f) $\mathbb{P}(185 \leq Y \leq 195 | X = 90)$.

23. Let X and Y denote the heart rate (in beats per minute) and average power output (in watts) for a 10-minute cycling time trial performed by a professional cyclist. Assume that X and Y have a bivariate normal distribution with parameters $\mu_X = 180$, $\sigma_X = 10$, $\mu_Y = 400$, $\sigma_Y = 50$, and $\rho = 0.9$. Find

- (a) $E[Y | X = 170]$,
- (b) $E[Y | X = 200]$,
- (c) $Var[Y | X = 170]$,
- (d) $Var[Y | X = 200]$,
- (e) $\mathbb{P}(Y \leq 380 | X = 170)$, and
- (f) $\mathbb{P}(Y \geq 450 | X = 200)$.

24. A certain group of college students takes both the Scholastic Aptitude Test (SAT) and an intelligence quotient (IQ) test. Let X and Y denote the students' scores on the SAT and IQ tests, respectively. Assume that X and Y have a bivariate normal distribution with parameters $\mu_X = 980$, $\sigma_X = 126$, $\mu_Y = 117$, $\sigma_Y = 7.2$, and $\rho = 0.58$. Find

- (a) $E[Y | X = 1350]$,
- (b) $E[Y | X = 700]$,
- (c) $Var[Y | X = 700]$,
- (d) $\mathbb{P}(Y \leq 120 | X = 1350)$, and
- (e) $\mathbb{P}(Y \geq 100 | X = 700)$.

25. A canning industry uses tins with weight equal to 20 grams. The tin is placed on a scale and filled with red peppers until the scale shows the weight μ . Then, the tin contains Y grams of peppers. If the scale is subject to a random error $X \sim N(0, \sigma = 10)$,

- (a) How is Y related to X and μ ?
- (b) What is the probability distribution of the random variable Y ?
- (c) Calculate the value μ such that 98% of the tins contain at least 400 grams of peppers.

(d) Repeat the exercise assuming the weight of the tins to be a normal random variable $W \sim N(20, \sigma = 5)$ if X and W are independent.

26. Given the joint density function $f_{X,Y}(x, y) = x + y$, $0 \leq x \leq 1, 0 \leq y \leq 1$,

(a) Show that $f_{X,Y}(x, y) \geq 0$ for all x and y and that $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx dy = 1$.

(b) Find the cumulative distribution function.

(c) Find the marginal means of X and Y .

(d) Find the marginal variances of X and Y .

27. The lifetime of two electronic components are two random variables, X and Y . Their joint density function is given by

$$f_{X,Y}(x, y) = \frac{1 + x + y + cxy}{(c + 3)} \exp(-(x + y)) \quad x \geq 0 \text{ and } y \geq 0.$$

(a) Verify that $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x, y) dx dy = 1$.

(b) Find $f_X(x)$.

(c) What value of c makes X and Y independent?

28. Given the joint continuous pdf

$$f_{X,Y}(x, y) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1, \quad 0 \leq y \leq 1 \text{ and} \\ 0 & \text{otherwise} \end{cases}$$

and using the function `adaptIntegrate()` from the package `cubature`,

(a) find $F_{X,Y}(x = 0.6, y = 0.8)$.

(b) find $\mathbb{P}(0.25 \leq X \leq 0.75, 0.1 \leq Y \leq 0.9)$.

(c) find $f_X(x)$.

29. Let X and Y have the joint density function

$$f_{X,Y}(x, y) = \begin{cases} Kxy & 2 \leq x \leq 4 \text{ and } 4 \leq y \leq 6 \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

(a) Find K so that the given function is a valid pdf.

(b) Find the marginal densities of X and Y .

(c) Are X and Y independent? Justify.

30. Given the joint density function of X and Y

$$f_{X,Y}(x, y) = \begin{cases} 1/2 & x + y \leq 2, \quad x \geq 0, \quad y \geq 0 \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

- (a) Find the marginal densities of X and Y .
- (b) Find $E[X]$, $E[Y]$, $Cov[X, Y]$, and $\rho_{X,Y}$.
- (c) Find $\mathbb{P}(X + Y < 1 \mid X > \frac{1}{2})$.

31. Let X and Y have the joint density function

$$f_{X,Y}(x, y) = \begin{cases} Ky & -2 \leq x \leq 2, 1 \leq y \leq x^2 \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

- (a) Find K so that $f_{X,Y}(x, y)$ is a valid **pdf**.
- (b) Find the marginal densities of X and Y .
- (c) Find $\mathbb{P}(Y > \frac{3}{2} \mid X < \frac{1}{2})$.

32. An engineer has designed a new diesel motor that is used in a prototype earth mover. The prototype's diesel consumption in gallons per mile C follows the equation $C = 3 + 2X + \frac{3}{2}Y$, where X is a speed coefficient and Y is the quality diesel coefficient. Suppose the joint density for X and Y is $f_{X,Y}(x, y) = ky$, $0 \leq x \leq 2$, $0 \leq y \leq x$.

- (a) Find k so that $f_{X,Y}(x, y)$ is a valid density function.
- (b) Are X and Y independent?
- (c) Find the mean diesel consumption for the prototype.

33. To make porcelain, kaolin X and feldspar Y are needed to create a soft mixture that later becomes hard. The proportion of these components for every tone of porcelain has the density function $f_{X,Y}(x, y) = Kx^2y$, $0 \leq x \leq y \leq 1$, $x + y \leq 1$.

- (a) Find the value of K so that $f_{X,Y}(x, y)$ is a valid **pdf**.
- (b) Find the marginal densities of X and Y .
- (c) Find the kaolin mean and the feldspar mean by tone.
- (d) Find the probability that the proportion of feldspar will be higher than $\frac{1}{3}$, if the kaolin is more than half of the porcelain.

34. A device can fail in four different ways with probabilities $\pi_1 = 0.2$, $\pi_2 = 0.1$, $\pi_3 = 0.4$, and $\pi_4 = 0.3$. Suppose there are 12 devices that fail independently of one another. What is the probability of 3 failures of the first kind, 4 of the second, 3 of the third, and 2 of the fourth?

35. The wait time in minutes a shopper spends in a local supermarket's checkout line has distribution

$$f(x) = \frac{\exp(-x/2)}{2}, \quad x > 0.$$

On weekends, however, the wait is longer, and the distribution then is given by

$$g(x) = \frac{\exp(-x/3)}{3}, \quad x > 0.$$

Find

- (a) The probability that the waiting time for a customer will be less than 1 minute.
- (b) The probability that, given a waiting time of less than 2 minutes, it will be a weekend.
- (c) The probability that the customer waits less than 2 minutes.
36. An engineering team has designed a lamp with two light bulbs. Let X be the lifetime for bulb 1 and Y the lifetime for bulb 2, both in thousands of hours. Suppose that X and Y are independent and they follow an $\text{Exp}(\lambda = 1)$ distribution.
- (a) Find the joint density function of X and Y . What is the probability neither bulb lasts longer than 1000 hours?
- (b) If the lamp works when at least one bulb is lit, what is the probability that the lamp works no more than 2000 hours?
- (c) What is the probability that the lamp works between 1000 and 2000 hours?
37. The national weather service has issued a severe weather advisory for a particular county that indicates that severe thunderstorms will occur between 9 p.m. and 10 p.m. When the rain starts, the county places a call to the maintenance supervisor who opens the sluice gate to avoid flooding. Assuming the rain's start time is uniformly distributed between 9 p.m. and 10 p.m.,
- (a) at what time, on the average, will the county maintenance supervisor open the sluice gate?
- (b) What is the probability that the sluice gate will be opened before 9:30 p.m.?
- Note: Solve this problem both by hand and using R.
38. Assume the distribution of grades for a particular group of students has a bivariate normal distribution with parameters $\mu_X = 3.2$, $\mu_Y = 2.4$, $\sigma_X = 0.4$, $\sigma_Y = 0.6$, and $\rho = 0.6$, where X and Y represent the grade point averages in high school and the first year of college, respectively.
- (a) Set the seed equal to 194 (`set.seed(194)`), and use the function `mvrnorm()` from the MASS package to simulate the population, assuming the population of interest consists of 200 students. (Hint: Use `empirical = TRUE`.)
- (b) Compute the means of X and Y . Are they equal to 3.2 and 2.4, respectively?
- (c) Compute the variance of X and Y as well as the covariance between X and Y . Are the values 0.16, 0.36, and 0.144, respectively?
- (d) Create a scatterplot of Y versus X . If a different seed value is used, how do the simulated numbers differ?
39. Show that if X_1, X_2, \dots, X_n are independent random variables with means $\mu_1, \mu_2, \dots, \mu_n$ and variances $\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$, respectively, then the mean and variance of $Y = \sum_{i=1}^n c_i X_i$, where the c_i s are real-valued constants, are $\mu_Y = \sum_{i=1}^n c_i \mu_i$ and $\sigma_Y^2 = \sum_{i=1}^n c_i^2 \sigma_i^2$. (Hint: Use moment generating functions.)

Chapter 6

Sampling and Sampling Distributions

6.1 Sampling

The objective of statistical analysis is to gain knowledge about certain properties in a population that are of interest to the researcher. When the population is small, the best way to study the population of interest is to study all of the elements in the population one by one. This process of collecting information on the entire population of interest is called a **census**; however, it is usually quite challenging to collect information on an entire population of interest. Not only do monetary and time constraints prevent a census from being taken easily, but also the challenges of finding all the members of a population can make gathering an accurate census all but impossible. Under certain conditions, a random selection of certain elements actually returns more reliable information than can be obtained by using a census. Standard methods used to learn about the characteristics of a population of interest include simulation, designed experiments, and sampling.

Simulation studies typically generate numbers according to a researcher-specified model. For a simulation study to be successful, the chosen simulation model must closely follow the real life process the researcher is attempting to simulate. For example, the effects of natural disasters, such as earthquakes, on buildings and highways are often modeled with simulation.

When the researcher has the ability to control the research environment, or at least certain variables of interest in the study, **designed experiments** are typically employed. The objective of designed experiments is to gain an understanding about the influence that various levels of a factor have on the response of a given experiment. For example, an agricultural researcher may be interested in determining the optimal level of nitrogen when his company's fertilizer is used to grow wheat in a particular type of soil. The designed experiment might consist of applying the company's fertilizer to similar plots using three different concentrations of nitrogen in the fertilizer.

Sampling is the most frequently used form of collecting information about a population of interest. Many forms of sampling exist, such as random sampling, simple random sampling, systematic sampling, and cluster sampling. It will be assumed that the population from which one is sampling has size N and that the sample is of size $n < N$.

Random sampling is the process of selecting n elements from a population where each of the n elements has the same probability of being selected, namely, $\frac{1}{N}$. More precisely, the random variables X_1, X_2, \dots, X_n form a random sample of size n from a population with a **pdf** $f(x)$ if X_1, X_2, \dots, X_n are mutually independent random variables such that the marginal **pdf** of each X_i is $f(x)$. The statement " X_1, X_2, \dots, X_n are independent and identically distributed, i.i.d., random variables with **pdf** $f(x)$ " is often used to denote a random sample. The objective of random sampling is to obtain a representative sample of the population that can be used to make generalizations about the population.

This process of making generalizations about the population from sampled information

is called **inferential statistics**. For the generalizations to be valid, the sample must meet certain requirements. The key requirement for a random sample is that it be representative of the parent population from which it was taken.

The typical method of obtaining a random sample starts with using either a calculator's or a computer's random number generator to decide which elements of a population to sample. The numbers returned from random number generating functions are not, in the strictest sense, random. That is, because an algorithm is used to generate the numbers, they are not completely random. Depending on the quality or lack thereof for a given random number generator, the same numbers may begin to cycle after a number of iterations. This problem is encountered much less with the random number generating functions written for computers than it is with those for calculators. In general, random number generators return pseudo-random numbers from a $Unif(0,1)$ distribution. Since people tend to favor certain numbers, it is best not to allow humans to pick random numbers unless the process is one of selecting numbers from an urn or another similar process. To avoid possible biases, it is best to let a function written to generate random numbers pick a sample.

When the population is finite, it is possible to list all of the possible arrangements of samples of size n using the R command `expand.grid()`. For example, suppose all of the arrangements of size $n = 3$ from a population consisting of $N = 4$ items are to be listed. Clearly, there are $4 \times 4 \times 4 = 64$ possible arrangements. To enumerate the possible arrangements with R, type `expand.grid(1:4, 1:4, 1:4)`. In a similar fashion, if all of the possible arrangements from rolling two fair dice or all possible arrangements of size $n = 2$ from the population $X_1 = 2$, $X_2 = 5$, and $X_3 = 8$ are to be enumerated, type `expand.grid(1:6, 1:6)` or `expand.grid(c(2,5,8), c(2,5,8))`, respectively.

6.1.1 Simple Random Sampling

Simple random sampling is the most elementary form of sampling. In a simple random sample, each particular sample of size n has the same probability of occurring. In finite populations, each of the $\binom{N}{n}$ samples of size n is taken without replacement and has the same probability of occurring. If the population being sampled is infinite, the distinction between sampling with replacement and sampling without replacement becomes moot. That is, in an infinite population, the probability of selecting a given element is the same whether sampling is done with or without replacement. Conceptually, the population can be thought of as balls in an urn, a fixed number of which are randomly selected without replacement for the sample. Most sampling is done without replacement due to its ease and increased efficiency in terms of variability compared to sampling with replacement.

To list all of the possible combinations of size n when sampling without replacement from a finite population of size N , that is, the $\binom{N}{n}$ combinations, use the function `combn()`.

Example 6.1 Given a population of size $N = 5$, list all of the possible samples of size $n = 3$ with R. That is, list the $\binom{5}{3} = 10$ possible combinations.

Solution: Use the function `combn()` where the argument `x` is a vector of N elements and the argument `m` is the number of elements to choose from the population of size N .

```
> combn(x = 1:5, m = 3)

 [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    1    1    1    1    1    2    2    2    3
[2,]    2    2    2    3    3    4    3    3    4    4
[3,]    3    4    5    4    5    5    4    5    5    5
```

The 10 possible combinations are $(1, 2, 3)$, $(1, 2, 4)$, \dots , $(3, 4, 5)$, listed vertically in the output.

Example 6.1 on the preceding page assumed all of the values in the population of interest are sequential starting with the number one. It is not unusual to have non-sequential values for the population where the user desires to enumerate all possible combinations when sampling without replacement. To that end, one may use the function `srs()` from the `PASWR2` package that works in conjunction with `combn()` to list all of the possible combinations when using simple random sampling from a finite population.

Example 6.2 Given a population of size $N = 5$, where $X_1 = 2$, $X_2 = 5$, $X_3 = 8$, $X_4 = 12$, and $X_5 = 13$, use R to list all of the possible samples of size $n = 3$. That is, list the $\binom{5}{3} = 10$ possible combinations.

Solution: First, make sure the `PASWR2` package is loaded. Then, use the function `srs()` as shown in R Code 6.1.

R Code 6.1

```
> t(srs(popvalues = c(2, 5, 8, 12, 13), n = 3))

 [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    2    2    2    2    2    2    5    5    5     8
[2,]    5    5    5    8    8   12    8    8   12    12
[3,]    8   12   13   12   13   13   12   13   13    13
```

The 10 possible combinations are $(2, 5, 8)$, $(2, 5, 12)$, \dots , $(8, 12, 13)$, listed vertically in the output. The R function `t()` was used to transpose the data to conserve space. It is not obligatory to transpose the output; it is just as valid to type `srs(c(2, 5, 8, 12, 13), 3)` so that the samples are listed across the rows instead of down the columns.

Example 6.3 A teacher wants an algorithm that will randomly select 5 students from a large lecture section of 180 students to present their work at the board.

Solution: Assume the students in the class are numbered from 1 to 180 according to the class roll and that the students know their numbers. Then, a procedure for selecting 5 students is given in R Code 6.2.

R Code 6.2

```
> set.seed(13)                                     # done for reproducibility
> sample(x = 1:180, size = 5, replace=FALSE, prob = rep(1/180, 180))

[1] 129  46  72  18 175
```

Example 6.4 Randomly select 5 people from a group of 20 where the individuals are labeled from 1 to 20, where the individuals labeled 19 and 20 are four times more likely to be selected than the individuals labeled 1 through 18.

Solution: An unbiased procedure to select 5 people is given in R Code 6.3 on the following page.

R Code 6.3

```
> set.seed(13) # done for reproducibility
> sample(x = 1:20, size = 5, prob = c(rep(1/26, 18), rep(4/26, 2)),
+         replace = FALSE)
[1] 14 20  9 19  1
```



6.1.2 Stratified Sampling

Simple random sampling gives samples that closely follow the population of interest provided the individual elements of the population of interest are relatively homogeneous with respect to the characteristics of interest in the study. When the population of interest is not homogeneous with respect to the characteristics under study, a possible solution might be to use **stratified sampling**.

Stratified sampling is most commonly used when the population of interest can be easily partitioned into subpopulations or strata. The strata are chosen to divide the population into non-overlapping, homogeneous regions. Then, the researcher takes simple random samples from each region or group. When using stratified sampling, it is crucial to select strata that are as homogeneous as possible within strata and as heterogeneous as possible between strata. For example, when agricultural researchers study crop yields, they tend to classify regions as arid and watered. It stands to reason that crop yields within arid regions will be poor and quite different from the yields from watered regions. Additional examples where stratified sampling can be used include:

1. In a study of the eating habits of a certain species, geographical areas often form natural strata.
2. In a study of political affiliation, gender often forms natural strata.
3. The Internal Revenue Service (IRS) might audit tax returns based on the reported taxable income by creating three groups: returns with reported taxable income less than \$50,000; returns with reported income less than \$75,000 but more than \$50,000; and returns with reported taxable income of more than \$75,000.

In addition to taking random samples within the strata, stratified samples are typically proportional to the size of their strata or proportional to the variability of the strata.

Example 6.5 A botanist wants to study the characteristics of a common weed and its adaptation to various geographical regions on a remote island. The island has well-defined strata that can be classified as desert, forest, mountains, and swamp. If 5000 acres of the island are desert, 1000 acres are forest, 500 acres are mountains, and 3500 acres are swamp, and the botanist wants to sample 5% of the population using a stratified sampling scheme that is proportional to the strata, how many acres of each of the four regions will he have to sample?

Solution: Since the size of the island is 10,000 acres, the botanist will need to sample a total of $10,000 \times 0.05 = 500$ acres. The breakdown of the 500 acres is as follows: $500 \times \frac{500}{10000} = 25$ desert acres; $500 \times \frac{1000}{10000} = 50$ forest acres; $500 \times \frac{500}{10000} = 25$ mountain acres; and $500 \times \frac{3500}{10000} = 175$ swamp acres.



6.1.3 Systematic Sampling

Systematic sampling is used when the researcher is in possession of a list that contains all N members of a given population and desires to select every k^{th} value in the master list. This type of sampling is often used to reduce costs since one only needs to select the initial starting point at random. That is, after the starting point is selected, the remaining values to be sampled are automatically specified.

To obtain a systematic sample, choose a sample size n and let k be the closest integer to $\frac{N}{n}$. Next, find a random integer i between 1 and k to be the starting point for sampling. Then, the sample is composed of the units numbered $i, i + k, i + 2k, \dots, i + (n - 1)k$. For example, suppose a systematic sample of size 10 is desired from a list containing 1000 members. Then, $k = 1000/10 = 100$ and every 100th member of the list is to be sampled. To pick the initial starting point, select a number at random between 1 and 100. If the random number generated is 53, then the researcher simply samples the values numbered 53, 153, 253, ..., 953 from the master list. R Code 6.4 generates the locations to be sampled using a 1 in 100 systematic sampling strategy.

R Code 6.4

```
> set.seed(13) # done for reproducibility
> seq(sample(1:100, 1), 1000, 100)

[1] 72 172 272 372 472 572 672 772 872 972
```

Example 6.6 Produce a list of locations to sample for a systematic sample if $N = 1000$ and $n = 20$.

Solution: To take a systematic sample, every $k = \frac{1000}{20} = 50^{\text{th}}$ item will be observed. To start the process, select a random number between 1 and 50 using a random number generator. R Code 6.5 can be used to select a 1 in 50 systematic sample when $N = 1000$ and $k = 50$.

R Code 6.5

```
> set.seed(13) # done for reproducibility
> seq(sample(1:50, 1), 1000, 50)

[1] 36 86 136 186 236 286 336 386 436 486 536 586 636 686 736 786 836
[18] 886 936 986
```

6.1.4 Cluster Sampling

Cluster sampling does not require a list of all of the units in the population like systematic sampling does. Rather, it takes units and groups them together to form clusters of several units. In contrast to stratified sampling, clusters should be as heterogeneous as possible within clusters and as homogeneous as possible between clusters. The main difference between cluster sampling and stratified sampling is that in cluster sampling, the cluster is treated as the sampling unit and analysis is done on a population of clusters. In one-step cluster sampling, all elements are selected in the chosen clusters. In stratified sampling, the analysis is done on elements within each strata. The main objective of cluster sampling is to reduce costs by increasing sampling efficiency. Examples of cluster sampling include:

1. Houses on a block,
 2. Students in school, and
 3. Farmers in counties.
-

6.2 Parameters

Once a sample is taken, the primary objective becomes to extract the maximum and most precise information possible about the population from the sample. Specifically, the researcher is interested in learning as much as possible about the population's **parameters**. Parameters are what characterize probability distributions. More to the point, parameters are inherent in all probability models, and it is impossible to compute a probability without prior knowledge of the distribution's parameters. Parameters are treated as constants in classical statistics and as random variables in Bayesian statistics. In everything that follows, parameters are treated as constants. A parameter, θ , is a function of the probability distribution, F . That is, $\theta = t(F)$, where $t(\cdot)$ denotes the function applied to F . Each θ is obtained by applying some numerical procedure $t(\cdot)$ to the probability distribution function F . Although F has been used to denote the **cdf** exclusively until now, a more general definition of F is any description of \mathbf{X} 's probabilities. Note that the **cdf**, $\mathbb{P}(X \leq x)$, is included in this more general definition.

Example 6.7 Suppose F is the exponential distribution, $F = \text{Exp}(\lambda)$, and $t(F) = E_F(\mathbf{X}) = \theta$. Express θ in terms of λ .

Solution: Here, $t(\cdot)$ is the expected value of \mathbf{X} , so $\theta = 1/\lambda$. ■

6.2.1 Infinite Populations' Parameters

The most commonly estimated parameters are the mean (μ), the variance (σ^2), and the proportion of successes (π). What follows is a brief review of their definitions.

Population mean — The **population mean** is defined as the expected value of the random variable X .

- If X is a discrete random variable,

$$\mu_X = E[X] = \sum_{i=1}^{\infty} x_i \cdot \mathbb{P}(X = x_i), \text{ where } \mathbb{P}(X = x_i) \text{ is the } \mathbf{pdf} \text{ of } X.$$

- If X is a continuous random variable,

$$\mu_X = E[X] = \int_{-\infty}^{\infty} x \cdot f(x) dx, \text{ where } f(x) \text{ is the } \mathbf{pdf} \text{ of } X.$$

Population variance — The population variance is defined as $\text{Var}[X] = E[(X - \mu)^2]$.

- For the discrete case,

$$\sigma_X^2 = \text{Var}[X] = \sum_{i=1}^{\infty} (x_i - \mu)^2 \cdot \mathbb{P}(X = x_i) = \sum_{i=1}^{\infty} x_i^2 \cdot \mathbb{P}(X = x_i) - \mu^2.$$

- For the continuous case,

$$\sigma_X^2 = \text{Var}[X] = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx = \int_{-\infty}^{\infty} x^2 f(x) dx - \mu^2.$$

Population proportion — The population proportion π is the ratio

$$\pi = \frac{N_1}{N},$$

where N_1 is the number of values that fulfill a particular condition and N is the size of the population.

6.2.2 Finite Populations' Parameters

Suppose a finite population that consists of N elements, X_1, \dots, X_N , is defined. The most commonly defined parameters are in Table 6.1 on the next page. The parameters used in Table 6.1 are commonly used in sampling contexts.

6.3 Estimators

Population parameters are generally unknown. Consequently, one of the first tasks of estimation is to evaluate the unknown parameters using sample data. Estimates of the unknown parameters are computed with **estimators** or **statistics**. An estimator is a function of the sample, while an estimate (a number) is the realized value of an estimator that is obtained when a sample is actually taken. Given a random sample, $\{X_1, X_2, \dots, X_n\} = \mathbf{X}$ from a probability distribution F , a statistic, which is any function of the sample, is denoted as $T = t(\mathbf{X})$. Note that the estimator T of θ will at times also be denoted $\hat{\theta}$. Since a statistic is a function of the random variables \mathbf{X} , it follows that statistics are also random variables. The specific value of a statistic can only be known after a sample has been taken. The resulting number, computed from a statistic, is called an **estimate**. For example, the arithmetic mean of a sample

$$T = t(\mathbf{X}) = \bar{X} = \frac{\sum_{i=1}^n X_i}{n}, \quad (6.1)$$

is a statistic (estimator) constructed from a random sample $\{X_1, \dots, X_n\}$ used to estimate $E[X] = \mu_X$.

Until a sample is taken, the value of the statistic (the estimate) is unknown. Suppose a random sample has been taken that contains the following values: $\mathbf{x} = \{3, 5, 6, 1, 2, 7\}$. It follows that the value of the statistic $T = t(\mathbf{X})$, where $t(\mathbf{X})$ is defined in (6.1) as $t = t(\mathbf{x}) = \frac{3+5+6+1+2+7}{6} = 4$. The quantity $t(\mathbf{X}) = \frac{X_1 \times X_2}{6}$ is also a statistic; however, it does not have the same properties as the arithmetic mean defined in (6.1).

The essential distinction between parameters and estimators is that a parameter is a constant in classical statistics while an estimator is a random variable, since its value changes from sample to sample. Parameters are typically designated with lowercase Greek letters, while estimators are typically denoted with uppercase Latin letters; however, when working with finite populations, it is standard notation to use different uppercase Latin

Table 6.1: Finite populations' parameters

Population Parameter	Formula	Explanation
Mean	$\mu_f = \frac{\sum_{i=1}^N X_i}{N}$	
Total	$\tau = \sum_{i=1}^N X_i = N\mu_f$	
Proportion	$\pi_f = \frac{Y}{N}$	Where Y is the number of elements of the population that fulfill a certain characteristic.
Proportion (alternate)	$\pi_f = \frac{\sum_i Y_i}{N}$	The Y_i 's take on a value of 1 if they represent a certain characteristic and 0 if they do not possess the characteristic.
Variance(N)	$\begin{aligned}\sigma_{f;N}^2 &= \frac{\sum_{i=1}^N (X_i - \mu_f)^2}{N} \\ &= \frac{1}{N} \sum_{i=1}^N X_i^2 - (\mu_f)^2\end{aligned}$	
Variance ($N - 1$)	$\sigma_{f;N-1}^2 = \frac{\sum_{i=1}^N (X_i - \mu_f)^2}{N - 1}$	
Variance (dichotomous)	$\sigma_f^2 = \pi_f(1 - \pi_f)$	π_f represents the proportion of elements in the population with a common characteristic.
Standard Deviation	$\sigma_f = \sqrt{\frac{\sum_{i=1}^N (X_i - \mu_f)^2}{N}}$	

letters to denote both parameters and estimators. At times, it is also common to denote an estimator by placing a hat over a parameter such as $\hat{\beta}_1$. Some common parameters and their corresponding estimators are provided in Table 6.2.

Table 6.2: Parameters and their corresponding estimators

Parameter	Name	Estimator (Latin notation)	Estimator (Hat notation)
μ	population mean	\bar{X} sample mean	$\hat{\mu}$
σ^2	population variance	S^2 sample variance	$\hat{\sigma}^2$

Some of the statistics used to estimate parameters when sampling from a finite population are given in Table 6.3 while the more common statistics used when working with a random sample of size n are given in Table 6.4 on the following page.

Table 6.3: Finite population parameter estimators and the estimators of their standard deviations

Parameter	Estimator	$\hat{\sigma}_{\text{estimator}}$
Population Mean	$\bar{X}_f = \frac{\sum_{i=1}^n X_i}{n}$	$\frac{S}{\sqrt{n}} \sqrt{\frac{N-n}{N}}$
Population Total	$T_f = N\bar{X}_f$	$\frac{S}{\sqrt{n}} \cdot N \cdot \sqrt{\frac{N-n}{N}}$
Population Proportion	$P = \frac{Y}{n}$	$\sqrt{\frac{P(1-P)}{n-1} \left(\frac{N-n}{N}\right)}$

6.3.1 Plug-In Principle

The **plug-in principle** is an intuitive method of estimating parameters from samples. The **plug-in estimator** of a parameter $\theta = t(F)$ is defined to be $\hat{\theta} = t(\hat{F})$. Simply put, the estimate is the result of applying the function $t(\cdot)$ to the empirical probability distribution \hat{F} .

Example 6.8 What are the plug-in estimators of (a) the expected value and (b) the variance of a discrete distribution F ?

Solution: The answers are as follows:

- (a) When the expected value is $\theta = E_F(\mathbf{X})$, the plug-in estimator of the expected value is $\hat{\theta} = E_{\hat{F}}(\mathbf{X}) = \sum_{i=1}^n X_i \cdot \frac{1}{n} = \bar{X}$.
- (b) When the variance is $\theta = \text{Var}_F(\mathbf{X}) = E_F(\mathbf{X} - \mu)^2$, the plug-in estimator of the variance of \mathbf{X} is $\hat{\theta} = E_{\hat{F}}(X - \bar{X})^2 = \sum_{i=1}^n (X_i - \bar{X})^2 \cdot \frac{1}{n}$. ■

6.4 Sampling Distribution of \bar{X}

Suppose 10 college students are randomly selected from the population of college students in the state of Colorado, and compute the mean age of the sampled students. If this process were repeated three times, it is unlikely any of the computed sample means would be identical. Likewise, it is not likely that any of the three computed sample means would

Table 6.4: Statistics for samples of size n

Statistic	Formula	Explanation
Mean	$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$	
Total	$T = n\bar{X}$	
Proportion	$P = \frac{Y}{n}$	Where Y is the number of elements with a certain characteristic
Variance (uncorrected)	$\begin{aligned} S_u^2 &= \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n} \\ &= \frac{\sum_{i=1}^n X_i^2}{n} - \bar{X}^2 \end{aligned}$	
Variance	$S_{ud}^2 = P(1 - P)$	Uncorrected and dichotomous
Variance	$\begin{aligned} S^2 &= \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1} \\ &= \frac{n}{n - 1} S_u^2 \end{aligned}$	
Variance (dichotomous)	$S_d^2 = \frac{nP(1 - P)}{n - 1}$	If $n \geq 20$, S_d^2 can be approximated with the quantity $P(1 - P)$.
Standard Deviation	$S = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}}$	

be exactly equal to the population mean; however, these sample means are typically used to estimate the unknown population mean. How can the accuracy of the sampled value be assessed?

To assess the accuracy of a value (estimate) returned from a statistic, the probability distribution of the statistic of interest is used to place probabilistic bounds on the sampling error. The probability distribution associated with all of the possible values a statistic can assume is called the **sampling distribution** of the statistic. This section presents the sampling distribution of the sample mean. Before discussing the sampling distribution of \bar{X} , the mean and variance of \bar{X} for any random variable X are highlighted.

If X is a random variable with mean μ and variance σ^2 , and if a random sample X_1, \dots, X_n is taken, the expected value and variance of \bar{X} are written

$$E[\bar{X}] = \mu_{\bar{X}} = \mu, \quad (6.2)$$

$$\text{Var} [\bar{X}] = \sigma_{\bar{X}}^2 = \frac{\sigma^2}{n}. \quad (6.3)$$

The computations of the answers for (6.2) and (6.3) are the same as those for Example 5.14 on page 330, which are reproduced for the reader's benefit:

$$\begin{aligned} E [\bar{X}] &= \sum_{i=1}^n \frac{E[X_i]}{n} = \sum_{i=1}^n \frac{1}{n} \mu = \mu, \\ \text{Var} [\bar{X}] &= \text{Var} \left[\sum_{i=1}^n \frac{X_i}{n} \right] = \frac{1}{n^2} \sum_{i=1}^n \text{Var}[X_i] = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n}. \end{aligned}$$

Clearly, as the sample size increases, the variance of the sampling distribution of \bar{X} decreases.

Example 6.9 \triangleright **Sampling: Balls in an Urn** \triangleleft Consider an experiment where two balls are randomly selected from an urn containing six numbered balls. First, the sampling is done with replacement (Case 1), and then the sampling is done without replacement (Case 2). List the exact sampling distributions of \bar{X} and S^2 for both cases. Finally, create graphs that compare these four distributions.

Solution: Case 1 When the sampling is performed with replacement, the outcomes can be viewed as a random sample of size 2 drawn from a discrete uniform distribution. The mean and variance of the uniform distribution are

$$\mu = \frac{1 + 2 + \cdots + 6}{6} = 3.5$$

and

$$\sigma^2 = E(X^2) - \mu^2 = \frac{1^2 + 2^2 + \cdots + 6^2}{6} - (3.5)^2 = 2.9166.$$

Note that these values could also be computed using the formulas $\mu = (N + 1)/2$ and $\sigma^2 = (N^2 - 1)/12$ listed at the end of section 4.2.1 on page 249.

There are 36 possible samples of size 2 from this distribution listed in Table 6.5 on the next page. Using the fact that each of the samples listed in Table 6.5 is equally likely ($\frac{1}{36}$), construct both the sampling distribution of \bar{X} given in Table 6.6 on the following page and the sampling distribution of S^2 given in Table 6.7 on the next page.

R Code 6.6 can be used to verify the values in Table 6.5 on the next page.

R Code 6.6

```
> N <- 6
> n <- 2
> pop <- 1:N
> rs <- expand.grid(Draw1 = pop, Draw2 = pop) # Possible random samples
> xbarN <- apply(rs, 1, mean) # Means of all rs values
> s2N <- apply(rs, 1, var) # Variance of all rs values
> RSV <- cbind(rs, xbarN = xbarN, s2N = s2N)
> head(RSV, n = 1) # First 1 row of values for Case 1 (random sampling)
```

	Draw1	Draw2	xbarN	s2N
1	1	1	1	0

Table 6.5: Possible samples of size 2 with \bar{x} and s^2 for each sample — random sampling

(x_1, x_2)	\bar{x}	s^2	(x_1, x_2)	\bar{x}	s^2
(1 , 1)	1.0	0.0	(4 , 1)	2.5	4.5
(1 , 2)	1.5	0.5	(4 , 2)	3.0	2.0
(1 , 3)	2.0	2.0	(4 , 3)	3.5	0.5
(1 , 4)	2.5	4.5	(4 , 4)	4.0	0.0
(1 , 5)	3.0	8.0	(4 , 5)	4.5	0.5
(1 , 6)	3.5	12.5	(4 , 6)	5.0	2.0
(2 , 1)	1.5	0.5	(5 , 1)	3.0	8.0
(2 , 2)	2.0	0.0	(5 , 2)	3.5	4.5
(2 , 3)	2.5	0.5	(5 , 3)	4.0	2.0
(2 , 4)	3.0	2.0	(5 , 4)	4.5	0.5
(2 , 5)	3.5	4.5	(5 , 5)	5.0	0.0
(2 , 6)	4.0	8.0	(5 , 6)	5.5	0.5
(3 , 1)	2.0	2.0	(6 , 1)	3.5	12.5
(3 , 2)	2.5	0.5	(6 , 2)	4.0	8.0
(3 , 3)	3.0	0.0	(6 , 3)	4.5	4.5
(3 , 4)	3.5	0.5	(6 , 4)	5.0	2.0
(3 , 5)	4.0	2.0	(6 , 5)	5.5	0.5
(3 , 6)	4.5	4.5	(6 , 6)	6.0	0.0

Table 6.6: Sampling distribution of \bar{X} — random sampling

\bar{x}	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6
$f(\bar{x})$	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

Table 6.7: Sampling distribution of S^2 — random sampling

s^2	0	0.5	2	4.5	8	12.5
$f(s^2)$	6/36	10/36	8/36	6/36	4/36	2/36

R Code 6.7 can be used to verify the values in Tables 6.6 and 6.7.

R Code 6.7

```
> library(MASS)                      # used for function fractions()
> fractions(xtabs(~xbarN)/36)        # Sampling dist of xbar (random sampling)

xbarN
```

```

 1  1.5    2  2.5    3  3.5    4  4.5    5  5.5    6
1/36 1/18 1/12 1/9 5/36 1/6 5/36 1/9 1/12 1/18 1/36

> fractions(xtabs(~s2N)/36)      # Sampling dist of S2 (random sampling)

s2N
 0  0.5    2  4.5    8 12.5
1/6 5/18 2/9 1/6 1/9 1/18

```

The mean of the sampling distribution, $\mu_{\bar{X}} = E[\bar{X}]$, and the variance of the sampling distribution, $\sigma_{\bar{X}}^2 = E[\bar{X} - \mu_{\bar{X}}]^2$, are

$$\mu_{\bar{X}} = E[\bar{X}] = 1 \times \frac{1}{36} + 1.5 \times \frac{2}{36} + \cdots + 6 \times \frac{1}{36} = 3.5$$

and

$$\begin{aligned}\sigma_{\bar{X}}^2 &= E[(\bar{X} - \mu_{\bar{X}})^2] = (1 - 3.5)^2 \times \frac{1}{36} + (1.5 - 3.5)^2 \times \frac{2}{36} + \\ &\quad \cdots + (6 - 3.5)^2 \times \frac{1}{36} = 1.4583.\end{aligned}$$

Note that the computed values of $E[\bar{X}]$ and $\sigma_{\bar{X}}^2$ are in agreement with the formulas $E[\bar{X}] = \mu$ and $\sigma_{\bar{X}}^2 = \frac{\sigma^2}{n}$ given in (6.2) and (6.3). Also note that $E[S^2] = \sigma^2$. Specifically,

$$E[S^2] = 0 \times \frac{6}{36} + 0.5 \times \frac{10}{36} + \cdots + 12.5 \times \frac{2}{36} = \frac{35}{12} = 2.9167.$$

The $\mu_{\bar{X}}$ and $\sigma_{\bar{X}}^2$ when sampling with replacement are computed in R Code 6.8.

R Code 6.8

```

> T1 <- fractions(xtabs(~xbarN)/36)
> T2 <- fractions(xtabs(~s2N)/36)
> XBAR <- as.numeric(names(T1))           # Unique values of xbar
> S2 <- as.numeric(names(T2))             # Unique values of s2
> MU_xbarN <- sum(XBAR*T1)               # Expected value of xbarN
> MU_xbarN

[1] 7/2

> VAR_xbarN <- sum((XBAR - MU_xbarN)^2*T1) # Var xbarN
> VAR_xbarN

[1] 35/24

```

```
> MU_s2N <- sum(S2*T2)                                # Expected value of s2N
> MU_s2N

[1] 35/12
```

Case 2 When the sampling is performed without replacement, the outcomes can be viewed as a **simple random sample** of size 2 drawn from a discrete uniform distribution. Note that fewer samples exist when sampling without replacement ($\binom{6}{2} = 15$), but that each sample is equally likely to be drawn. The 15 possible samples of size 2 from this distribution are listed in Table 6.8 on the next page. Using the fact that each of the samples listed in Table 6.8 is equally likely (1/15), construct the sampling distribution of \bar{X} given in Table 6.9, and the sampling distribution of S^2 given in Table 6.10 both on the facing page. R Code 6.9 can be used to verify the values in Table 6.8 on the facing page.

R Code 6.9

```
> draw <- c("Draw1", "Draw2")
> SRS <- srs(1:N, n)    # possible simple random samples
> dimnames(SRS) <- list(NULL, draw)
> xbar <- apply(SRS, 1, mean)                      # Means of all SRS values
> s2n <- apply(SRS, 1, var)                         # Variance of all SRS values
> SRSV <- cbind(SRS, xbar = xbar, s2n = s2n)
> head(SRSV, n = 3)    # First 3 rows of values for Case 2 (SRS)

      Draw1 Draw2 xbar s2n
[1,]     1     2   1.5 0.5
[2,]     1     3   2.0 2.0
[3,]     1     4   2.5 4.5
```

R Code 6.10 can be used to verify the values in Tables 6.9 and 6.10 on the facing page.

R Code 6.10

```
> library(MASS)                                     # used for function fractions()
> fractions(xtabs(~xbar)/15)    # Sampling dist of xbar (SRS)

xbar
1.5     2   2.5     3   3.5     4   4.5     5   5.5
1/15  1/15  2/15  2/15  1/5   2/15  2/15  1/15  1/15

> fractions(xtabs(~s2n)/15)    # Sampling dist of S2 (SRS)

s2n
0.5     2   4.5     8  12.5
1/3   4/15  1/5   2/15  1/15
```

Table 6.8: Possible samples of size 2 with \bar{x} and s^2 — simple random sampling

(x_1, x_2)	\bar{x}	s^2
(1 , 2)	1.5	0.5
(1 , 3)	2	2.0
(1 , 4)	2.5	4.5
(1 , 5)	3	8.0
(1 , 6)	3.5	12.5
(2 , 3)	2.5	0.5
(2 , 4)	3	2.0
(2 , 5)	3.5	4.5
(2 , 6)	4	8.0
(3 , 4)	3.5	0.5
(3 , 5)	4	2.0
(3 , 6)	4.5	4.5
(4 , 5)	4.5	0.5
(4 , 6)	5	2.0
(5 , 6)	5.5	0.5

Table 6.9: Sampling distribution of \bar{X} — simple random sampling

\bar{x}	1.5	2	2.5	3	3.5	4	4.5	5	5.5
$f(\bar{x})$	1/15	1/15	2/15	2/15	3/15	2/15	2/15	1/15	1/15

Table 6.10: Sampling distribution of S^2 — simple random sampling

s^2	0.5	2	4.5	8	12.5
$f(s^2)$	5/15	4/15	3/15	2/15	1/15

The mean of the sampling distribution, $\mu_{\bar{X}} = E[\bar{X}]$, the variance of the sampling distribution, $\sigma_{\bar{X}}^2 = E[\bar{X} - \mu_{\bar{X}}]^2$, and the expected value of S^2 , $E[S^2]$, are

$$\mu_{\bar{X}} = E[\bar{X}] = 1.5 \times \frac{1}{15} + 2 \times \frac{1}{15} + \cdots + 5.5 \times \frac{1}{15} = 3.5,$$

$$\sigma_{\bar{X}}^2 = E[(\bar{X} - \mu_{\bar{X}})^2] = (1.5 - 3.5)^2 \times \frac{1}{15} + (2 - 3.5)^2 \times \frac{1}{15} + \cdots + (5.5 - 3.5)^2 \times \frac{1}{15} = 1.1667,$$

$$\text{and } E[S^2] = 0.5 \times \frac{5}{15} + 2 \times \frac{4}{15} + \cdots + 12.5 \times \frac{1}{15} = 3.5.$$

Remarkably, the sample mean is identical when sampling with and without replacement. In fact, the expected value of the sample mean is μ whether sampling with or without replacement. The variance of the sample mean and the expected value of the sample variance have changed, however. These changes are due to the fact that sampling is from a finite population without replacement. A summary of the formulas used to compute these results is found in Table 6.11.

Table 6.11: Summary results for sampling without replacement (finite population)

$\mu_{\bar{X}} = \mu_f$
$\sigma_{\bar{X}}^2 = \frac{\sigma^2}{n} \cdot \frac{N-n}{N-1}$
$E[S^2] = \frac{N}{N-1} \cdot \sigma^2$
$E[S_u^2] = \frac{N}{N-1} \cdot \frac{n-1}{n} \cdot \sigma^2$

The $\mu_{\bar{X}}$, $\sigma_{\bar{X}}^2$, and $E[S^2]$ when sampling without replacement (SRS) are computed in R Code 6.11.

R Code 6.11

```
> T3 <- fractions(xtabs(~xbarn)/15)
> T4 <- fractions(xtabs(~s2n)/15)
> XBAR <- as.numeric(names(T3))                      # Unique values of xbar
> S2 <- as.numeric(names(T4))                        # Unique values of s2
> MU_xbarn <- sum(XBAR*T3)                          # Expected value of xbarn
> MU_xbarn

[1] 7/2

> VAR_xbarn <- sum((XBAR - MU_xbarn)^2*T3)    # Var xbarn
> VAR_xbarn
```

```
[1] 7/6
```

```
> MU_s2n <- sum(S2*T4) # Expected value of s2n
> MU_s2n
```

```
[1] 7/2
```

Note that the computed values of $E[\bar{X}] = \mu_f = 3.5$, $\sigma_{\bar{X}}^2 = \frac{\sigma^2}{n} \cdot \frac{N-n}{N-1} = \frac{2.9166}{2} \cdot \frac{6-2}{6-1} = 1.1667$, and $E[S^2] = \frac{N}{N-1}\sigma^2 = \frac{6}{5}(2.9167) = 3.5$ for this example are in agreement with the formulas for sampling without replacement given in Table 6.11 on the preceding page. A comparison of the results from Case 1 and Case 2 can be found in Table 6.12.

Table 6.12: Computed values for random sampling (Case 1) and simple random sampling (Case 2)

	μ	$E[\bar{X}]$	σ^2	$E[S^2]$	$\sigma_{\bar{X}}^2$
Case 1	3.5	3.5	2.9167	2.9167	1.4583
Case 2	3.5	3.5	2.9167	3.5	1.1667

Graphs of the sampling distributions of \bar{X} and S^2 under random sampling (RS) and simple random sampling (SRS) for Example 6.9 on page 361 are given in Figure 6.1 on the following page. Note that the dispersion for the sampling distribution of \bar{X} is smaller under Case 2 (SRS) than it is with Case 1 (RS).

6.5 Sampling Distribution for a Statistic from an Infinite Population

Consider a population from which k random samples, each of size n , are taken. In general, if given k samples, k different values for the sample mean will result. If k is very large, theoretically infinite, the values of the means from each of the samples, denoted \bar{X}_i for each sample i , will be random variables with a resulting distribution referred to as the sampling distribution of the sample mean. The sampling distribution of a statistic, $t(X)$, is the resulting probability distribution for $t(X)$ calculated by taking an infinite number of random samples of size n . The resulting sampling distribution will typically not coincide with the distribution of the parent population.

6.5.1 Sampling Distribution for the Sample Mean

The sample mean is the average of a sample and the best measure of center when a distribution is bell shaped. The sampling distribution of the sample mean has some rather

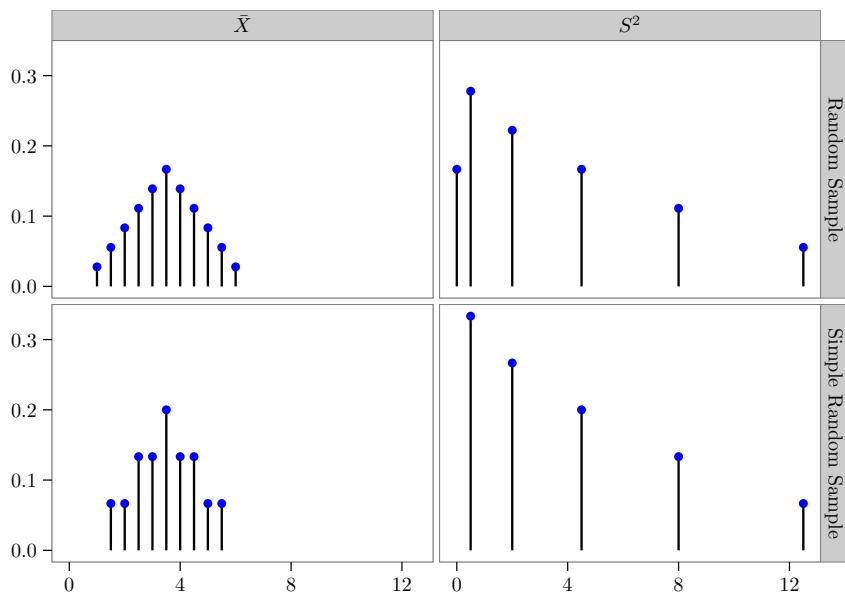


FIGURE 6.1: Sampling distributions of \bar{X} and S^2 under random sampling (RS) and simple random sampling (SRS) for Example 6.9

amazing properties. If one is sampling from a normal population, the sampling distribution of the sample mean is also a normal distribution. Even more incredible, as long as the variance of the population is finite, the sampling distribution of the sample mean is also normal when sampling from any other distribution as long as the sample size is sufficiently large. Both of these cases are described in this section.

6.5.1.1 First Case: Sampling Distribution of \bar{X} When Sampling from a Normal Distribution

When sampling from a normal distribution, the resulting sampling distribution for the sample mean is also a normal distribution. This is an immediate result of Theorem 5.1 on page 320. That is, \bar{X} is a linear combination of the X_i s where $a_i = \frac{1}{n}$. As observed earlier, the mean and the variance of the sampling distribution of \bar{X} are μ and σ^2/n regardless of the underlying population. So, the mean and variance of the sampling distribution of \bar{X} are always known. However, it is not always true that the resulting sampling distribution of \bar{X} is known. If $X \sim N(\mu, \sigma)$, then $\bar{X} \sim N(\mu, \frac{\sigma}{\sqrt{n}})$.

Example 6.10 It is well-known that the measurement errors committed by employees when they measure the length of a zipper in a particular assembly process follow a normal distribution with a mean of 0 and standard deviation of 2 millimeters.

- 95% of errors are expected to be less than what value?
- If a zipper is measured 10 times, the average of the measurements will be within how many millimeters of the truth with 95% confidence?
- How many measurements must be taken for the average measurement error to be less than 1 millimeter with 95% confidence?

Solution: The solutions are as follows:

(a) Let the random variable X represent the measurement error committed by employees when measuring zippers. Since $X \sim N(0, 2)$, $Z = \frac{X-0}{\sqrt{2}} \sim N(0, 1)$. It is known that

$$\mathbb{P}(-1.96 < Z < 1.96) = 0.95,$$

and $Z = \frac{X}{2}$, so

$$\mathbb{P}\left(-1.96 < \frac{X}{2} < 1.96\right) = 0.95.$$

Basic algebra then gives

$$|X| < 2(1.96) = 3.92.$$

One can be 95% confident that single errors will be less than 3.92 millimeters.

(b) In this question, the distribution of X is no longer the focus, but rather the distribution of \bar{X} is. Since $\bar{X} \sim N(0, 2/\sqrt{10})$, it follows that the average when measuring a zipper 10 times is within

$$|\bar{X}| = \frac{2}{\sqrt{10}}(1.96) = 1.24 \text{ millimeters of the truth with 95\% confidence.}$$

(c) Since $\bar{X} \sim N(0, 2/\sqrt{n})$, it follows that

$$|\bar{X}| = \frac{2}{\sqrt{n}}(1.96) \leq 1$$

must be solved for n . The solution is $n \geq (3.92)^2 = 15.37$. At least 16 measurements must be taken for the average error to be less than 1 millimeter with 0.95% confidence. ■

Example 6.11 If $X \sim N(\mu, 12)$, find the required sample size to guarantee $|\bar{X} - \mu| < 3$ with a probability of 0.95.

Solution: Changing the prose into a mathematical statement,

$$\mathbb{P}(|\bar{X} - \mu| < 3) = 0.95 \quad (6.4)$$

needs to be solved.

Since $X \sim N(\mu, \sigma^2 = 12)$, it follows that

$$\bar{X} \sim N\left(\mu, \sigma_{\bar{X}} = \frac{\sigma}{\sqrt{n}} = \frac{12}{\sqrt{n}}\right).$$

Consequently,

$$\mathbb{P}\left(\frac{|\bar{X} - \mu|}{\sigma/\sqrt{n}} < z_{0.95}\right) = 0.95.$$

Multiplying both sides by $\frac{\sigma}{\sqrt{n}}$ and substituting 12 for σ and 1.96 for $z_{0.95}$ gives

$$\mathbb{P}\left(|\bar{X} - \mu| < (1.96)\frac{12}{\sqrt{n}}\right) = 0.95. \quad (6.5)$$

Because (6.4) and (6.5) have a parallel structure, one can write

$$\begin{aligned}\frac{1.96 \cdot (12)}{\sqrt{n}} &= 3 \\ 1.96 \cdot (12) &= 3\sqrt{n} \\ \frac{1.96 \cdot (12)}{3} &= \sqrt{n} \\ n = \left(\frac{1.96 \cdot 12}{3}\right)^2 &= 61.4633.\end{aligned}$$

Consequently, a sample size of at least 62 is needed to guarantee $|\bar{X} - \mu| < 3$ with a probability of 0.95. ■

6.5.1.2 Second Case: Sampling Distribution of \bar{X} When X Is Not a Normal Random Variable

When the underlying population of X is not normal, provided the sample size is sufficiently large, the sampling distribution of \bar{X} is still normal. Specifically, the **Central Limit Theorem** states that if $X \sim (\mu, \sigma)$, then the limiting distribution of

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

as $n \rightarrow \infty$ is the standard normal distribution. Expressed in lay terms, the sampling distribution of \bar{X} , regardless of the underlying population, is approximately $N(\mu, \sigma/\sqrt{n})$ provided n is sufficiently large. Populations that are asymmetric require larger values of n compared to symmetric populations before the sampling distribution of \bar{X} appears normal.

Consider the left graph of Figure 6.2 on the facing page, which depicts a $Unif(0, 10)$ population, while the center graph of Figure 6.2 depicts the theoretical sampling distribution of \bar{X} for samples of size $n = 2$ when sampling is from a $Unif(0, 10)$ population. Finally, the far right graph of Figure 6.2 superimposes the theoretical sampling distribution of \bar{X} for samples of size $n = 2$ when sampling is from a $Unif(0, 10)$ population over a normal distribution with a mean and standard deviation corresponding to the mean and standard deviation of the sampling distribution of \bar{X} for samples of size $n = 2$ when sampling from the $Unif(0, 10)$ population. It is interesting to note in the far right graph in Figure 6.2 how closely the triangular distribution resembles the normal distribution.

The sampling distributions of \bar{X} associated with infinite populations are obviously impossible to enumerate; however, simulation can be used to gain insight into the sampling distribution of \bar{X} when sampling from known populations. That is, a large number of samples from a known population can be drawn and the distribution of \bar{X} can be studied.

In what follows, the various graphs depicted in Figure 6.3 on page 373 and Figure 6.4 on page 373 are examined to gain insight into how large the sample size, n , needs to be when working with both symmetric distributions and skewed distributions such as the uniform distribution and the exponential distribution, respectively. R Code 6.12 simulates $m = 100,000$ samples of sizes $n = 2, 16, 36$, and 100 from a $Unif(-3.66025, 13.66025)$ distribution and an $Exp(\frac{1}{5})$ distribution and stores the results in a data frame named BDF. Note that both the means and standard deviations are 5 and 5 for these distributions.

R Code 6.12

```
> uv <- (10 + sqrt(300))/2 # b of X ~ Unif(a, b)
> lv <- (10 - sqrt(300))/2 # a of X ~ Unif(a, b)
> c(lv, uv)
```

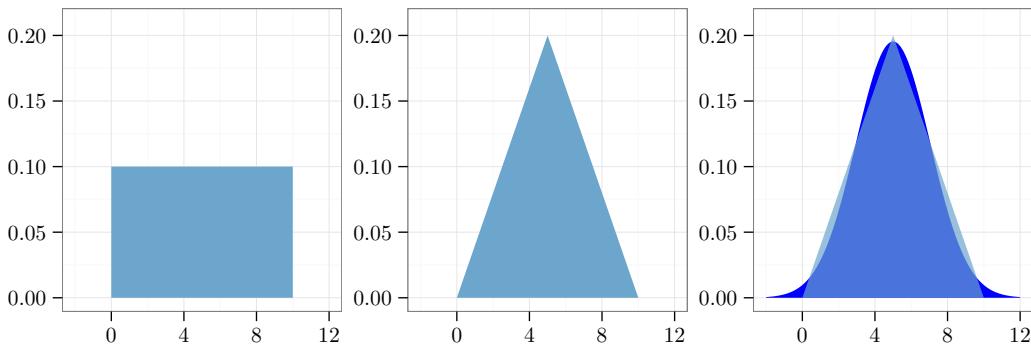


FIGURE 6.2: The far left graph depicts a $Unif(0, 10)$ distribution. The middle graph depicts the theoretical sampling distribution of \bar{X} for samples of size $n = 2$ when the samples are drawn from a $Unif(0, 10)$ distribution. The far left graph depicts a $N(5, 2.0412)$ distribution overlayed with the theoretical distribution of \bar{X} for samples of size $n = 2$ when the samples are drawn from a $Unif(0, 10)$ distribution.

```
[1] -3.660254 13.660254

> m <- 100000
> set.seed(15)
> MEANSunif2 <- numeric(m) # storage space for means
> for (i in 1:m) {MEANSunif2[i] <- mean(runif(2, lv, uv))}
> MEANsexp2 <- numeric(m) # storage space for means
> for (i in 1:m) {MEANsexp2[i] <- mean(rexp(2, 1/5))}
> MEANSunif16 <- numeric(m) # storage space for means
> for (i in 1:m) {MEANSunif16[i] <- mean(runif(16, lv, uv))}
> MEANsexp16 <- numeric(m) # storage space for means
> for (i in 1:m) {MEANsexp16[i] <- mean(rexp(16, 1/5))}
> MEANSunif36 <- numeric(m) # storage space for means
> for (i in 1:m) {MEANSunif36[i] <- mean(runif(36, lv, uv))}
> MEANsexp36 <- numeric(m) # storage space for means
> for (i in 1:m) {MEANsexp36[i] <- mean(rexp(36, 1/5))}
> MEANSunif100 <- numeric(m) # storage space for means
> for (i in 1:m) {MEANSunif100[i] <- mean(runif(100, lv, uv))}
> MEANsexp100 <- numeric(m) # storage space for means
> for (i in 1:m) {MEANsexp100[i] <- mean(rexp(100, 1/5))}
> BDF <- data.frame(Means = c(MEANSunif2, MEANsexp2, MEANSunif16,
+                                MEANsexp16, MEANSunif36, MEANsexp36,
+                                MEANSunif100, MEANsexp100),
+                     Type = rep(rep(c("Uniform", "Exponential"),
+                                    each = m), 4),
+                     Size = c(rep("n = 2", 2*m), rep("n = 16", 2*m),
+                             rep("n = 36", 2*m), rep("n = 100", 2*m)))
> BDF$Size <- factor(BDF$Size,
+                      levels = c("n = 2", "n = 16", "n = 36", "n = 100"))
> BDF>Type <- factor(BDF>Type, levels = c("Uniform", "Exponential"))
```

R Code 6.13 on the following page can be used to create graphs similar to those in Figure 6.3 on page 373, which depict the simulated sampling distribution of \bar{X} for samples

of sizes $n = 2$ and 16 when one samples from a $\text{Unif}(-3.66025, 13.66025)$ distribution and an $\text{Exp}\left(\frac{1}{5}\right)$ distribution, respectively. Recall that the uniform and the exponential distribution in this case both have means and standard deviations of 5 . Normal curves are superimposed over the simulated sampling distributions of \bar{X} that have a mean of 5 and a standard deviation, $\sigma_{\bar{x}} = 5/\sqrt{n}$, adjusted according to the sample size. By looking across the first row of Figure 6.3 on the facing page, one notices that the sampling distribution of \bar{X} when sampling from a uniform distribution for samples of size $n = 2$ is a crude approximation of the superimposed normal distribution. The sampling distribution of \bar{X} when sampling from a uniform distribution with samples of size $n = 16$ is very similar to the superimposed normal distribution. When looking across the second row of Figure 6.3 on the next page, one notes how skewed the sampling distribution of \bar{X} is for samples of size $n = 2$. When the sample size is increased to $n = 16$, the resulting sampling distribution of \bar{X} is still skewed to the right.

R Code 6.13

```
> p <- ggplot(data = subset(BDF, Size == "n = 2"), aes(x = Means)) +
+   geom_density(fill = "skyblue") +
+   facet_grid(Type ~ Size) +
+   coord_cartesian(xlim = c(lv, uv)) +
+   labs(x = expression(bar(x))) +
+   theme_bw()
> p + stat_function(fun = dnorm, args = list(5, 5/sqrt(2)), col = "blue",
+                     size = 1, fill = "blue", alpha = 0.2, geom = "area")
> p <- ggplot(data = subset(BDF, Size == "n = 16"), aes(x = Means)) +
+   geom_density(fill = "skyblue") +
+   facet_grid(Type ~ Size) +
+   coord_cartesian(xlim = c(0, 10)) +
+   labs(x = expression(bar(x))) +
+   theme_bw()
> p + stat_function(fun = dnorm, args = list(5, 5/sqrt(16)), col = "blue",
+                     size = 1, fill = "blue", alpha = 0.2, geom = "area")
```

Figure 6.4 on the facing page depicts the simulated sampling distribution of \bar{X} for samples of sizes $n = 36$ and 100 when sampling from a $\text{Unif}(-3.66025, 13.66025)$ distribution and an $\text{Exp}\left(\frac{1}{5}\right)$ distribution, respectively. What should become evident from looking at Figures 6.3 and 6.4 is that the sampling distribution of \bar{X} when sampling from a uniform distribution becomes approximately normal much sooner than does the sampling distribution of \bar{X} when sampling from an exponential distribution.

In addition to assessing the simulated sampling distributions of \bar{X} graphically by superimposing a normal density with mean and standard deviation equal to the mean and standard deviation of the sampling distribution of \bar{X} as shown in Figures 6.3 and 6.4, Table 6.13 on page 374 is provided, which contains the percent of the simulated sampling distribution of \bar{X} that falls within $(-\infty, \mu_{\bar{X}} - 2\sigma_{\bar{X}}]$, $(\mu_{\bar{X}} - 2\sigma_{\bar{X}}, \mu_{\bar{X}} - \sigma_{\bar{X}}]$, $(\mu_{\bar{X}} - \sigma_{\bar{X}}, \mu_{\bar{X}}]$, $(\mu_{\bar{X}}, \mu_{\bar{X}} + \sigma_{\bar{X}}]$, $(\mu_{\bar{X}} + \sigma_{\bar{X}}, \mu_{\bar{X}} + 2\sigma_{\bar{X}}]$, and $(\mu_{\bar{X}} + 2\sigma_{\bar{X}}, \infty]$ for sample sizes $n = 2, 16, 36$, and 100 when sampling from a $\text{Unif}(-3.66025, 13.66025)$ distribution and an $\text{Exp}\left(\frac{1}{5}\right)$ distribution. By studying the percentages from the simulations in Table 6.13 on page 374, one can see that the simulated sampling distribution of \bar{X} when sampling from an exponential distribution is still slightly skewed even for sample sizes as large as $n = 100$.

Example 6.12 Suppose that the shelf life, the number of days a product is on a store's shelf, for one-gallon cartons of milk is a random variable with a $\text{Unif}[1, 7]$ distribution. If

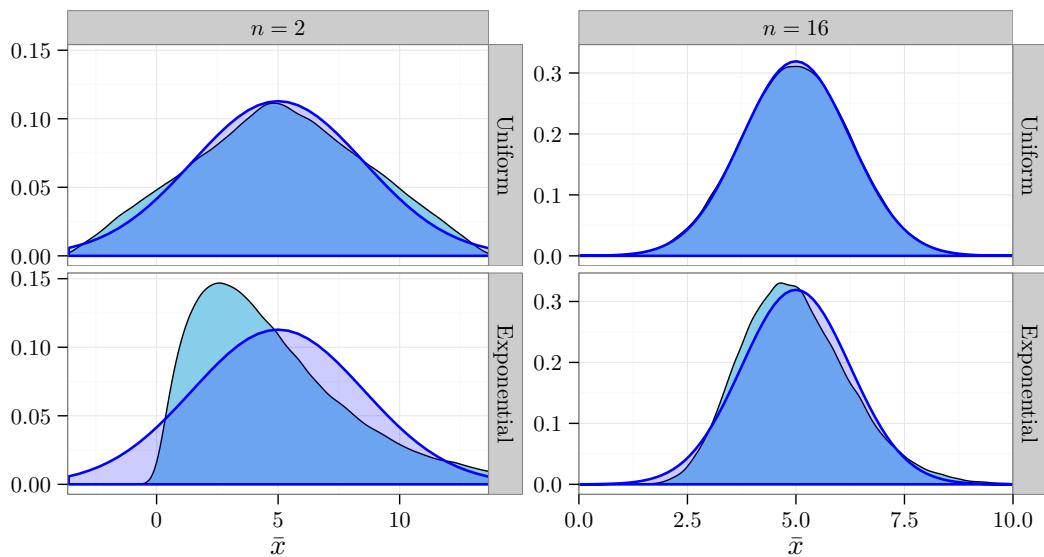


FIGURE 6.3: Simulation depicts the simulated sampling distribution of \bar{X} for samples of size $n = 2$ and $n = 16$ that are selected from a $\text{Unif}(-3.66025, 13.66025)$ and an $\text{Exp}\left(\frac{1}{5}\right)$ distribution with superimposed normal distributions for comparison purposes

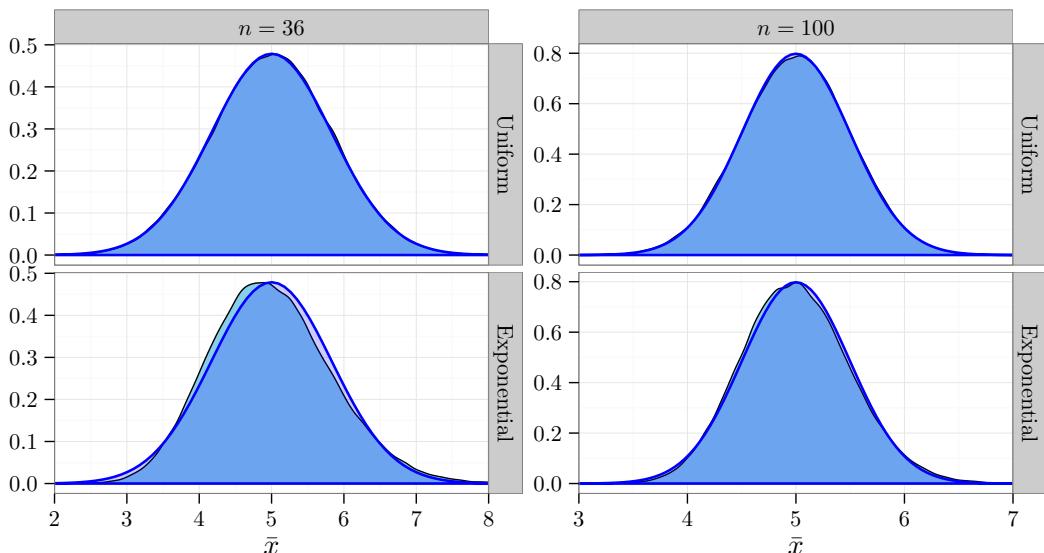


FIGURE 6.4: Simulation depicts the simulated sampling distribution of \bar{X} for samples of size $n = 36$ and $n = 100$ that are selected from a $\text{Unif}(-3.66025, 13.66025)$ and an $\text{Exp}\left(\frac{1}{5}\right)$ distribution with superimposed normal distributions for comparison purposes

a store puts out 100 cartons of such milk for sale, find the probability that the average number of days the cartons remain on the shelf exceeds 4.5 days.

Solution: Let the random variable X represent the number of days a one-gallon carton of milk is on a store's shelf. Since $X \sim \text{Unif}[1, 7]$, using the equations from (4.9), the pdf

Table 6.13: Comparison of simulated uniform and exponential distributions to the normal distribution, $Int1 = (-\infty, \mu_{\bar{X}} - 2\sigma_{\bar{X}}]$, $Int2 = (\mu_{\bar{X}} - 2\sigma_{\bar{X}}, \mu_{\bar{X}} - \sigma_{\bar{X}}]$, $Int3 = (\mu_{\bar{X}} - \sigma_{\bar{X}}, \mu_{\bar{X}}]$, $Int4 = (\mu_{\bar{X}}, \mu_{\bar{X}} + \sigma_{\bar{X}}]$, $Int5 = (\mu_{\bar{X}} + \sigma_{\bar{X}}, \mu_{\bar{X}} + 2\sigma_{\bar{X}}]$, $Int6 = (\mu_{\bar{X}} + 2\sigma_{\bar{X}}, \infty]$

	<i>Int1</i>	<i>Int2</i>	<i>Int3</i>	<i>Int4</i>	<i>Int5</i>	<i>Int6</i>
$N(0, 1)$	0.0228	0.1359	0.3413	0.3413	0.1359	0.0228
$n = 2$	<i>Unif</i>	0.0171	0.1570	0.3268	0.3243	0.1585
	<i>Exp</i>	0.0000	0.1152	0.4807	0.2595	0.0977
$n = 16$	<i>Unif</i>	0.0228	0.1379	0.3383	0.3409	0.1380
	<i>Exp</i>	0.0085	0.1479	0.3753	0.3115	0.1227
$n = 36$	<i>Unif</i>	0.0229	0.1366	0.3389	0.3418	0.1372
	<i>Exp</i>	0.0130	0.1449	0.3637	0.3207	0.1271
$n = 100$	<i>Unif</i>	0.0233	0.1369	0.3370	0.3438	0.1374
	<i>Exp</i>	0.0170	0.1418	0.3525	0.3313	0.1297
						0.0278

of X can be written as

$$f(x) = \frac{1}{6} \quad \text{if } x \in [1, 7],$$

and the mean and variance of X are

$$E[X] = \frac{a+b}{2} = \frac{1+7}{2} = 4, \quad \text{and} \quad Var[X] = \frac{(b-a)^2}{12} = \frac{(7-1)^2}{12} = 3.$$

Let X_i , $i = 1, \dots, 100$, represent the actual times cartons of milk remain on the store's shelf. Since

$$E[X_i] = 4 \quad \text{and} \quad Var[X_i] = 3,$$

the average time is computed as

$$\bar{X} = \frac{1}{100}(X_1 + \dots + X_{100}).$$

Consequently, the mean and variance of this sample mean are

$$E[\bar{X}] = 4, \quad Var[\bar{X}] = \frac{\sigma^2}{n} = \frac{3}{100} = 0.03.$$

Appealing to the Central Limit Theorem, write

$$\frac{\bar{X} - E[\bar{X}]}{\sqrt{Var[\bar{X}]}} = \frac{\bar{X} - 4}{\sqrt{0.03}} \sim N(0, 1),$$

which is equivalent to writing $\bar{X} \sim N(4, \sqrt{0.03})$. Consequently,

$$\mathbb{P}(\bar{X} > 4.5) = \mathbb{P}\left(Z > \frac{4.5 - 4}{\sqrt{0.03}}\right) = \mathbb{P}(Z > 2.89) = 0.002.$$

The following code computes the answer with R:

```
> round(1 - pnorm(4.5, 4, sqrt(0.03)), 3)
```

```
[1] 0.002
```

The probability that the average number of days the cartons remain on the shelf exceeds 4.5 days is 0.002. ■

Example 6.13 A building contractor provides a detailed estimate of his charges by listing the price of all of his material and labor charges to the nearest dollar. Suppose the rounding charge errors can be treated as independent random variables following $\text{Unif}[-10, 10]$ distributions. If a recent estimate from the building contractor listed 100 charges, find the maximum error for the contractor's estimate with probability of 0.95.

Solution: Using the equations from (4.9), if $e_i, i = 1, \dots, 100$ are the estimate errors, then $E[e_i] = \frac{b+a}{2} = 0$ and $\text{Var}[e_i] = \frac{(b-a)^2}{12} = \frac{400}{12}$. It follows then that $\mu_{\bar{e}} = 0$ and $\sigma_{\bar{e}}^2 = \frac{\frac{400}{12}}{100} = \frac{1}{3}$. Because of the relatively large ($n = 100$) sample size, the Central Limit Theorem tells us that the distribution of \bar{e} is approximately normal with mean 0 and standard deviation $\sqrt{\frac{1}{3}}$. Since the absolute error of the sum of the 100 charges is the sum of each one of the rounded errors, $e = e_1 + \dots + e_{100}$, $e = \bar{e} \cdot n$. Written mathematically,

$$\mathbb{P}\left(-1.96 < \frac{\bar{e} - 0}{\sqrt{\frac{1}{3}}} < 1.96\right) = 0.95.$$

Multiplying by $n = 100$ and $\sqrt{\frac{1}{3}}$ gives a probability expression for e :

$$\mathbb{P}\left(\sqrt{\frac{1}{3}} \cdot 100 \cdot (-1.96) < e < \sqrt{\frac{1}{3}} \cdot 100 \cdot (1.96)\right) = 0.95.$$

From the last expression, note that the maximum error for the estimate e_{Max} , is $\sqrt{\frac{1}{3}} \cdot 100 \cdot (1.96) = 113.1607$. In other words, the final job will not deviate more than 113 dollars from the original estimate with 95% confidence. ■

6.5.2 Sampling Distribution for $\bar{X} - \bar{Y}$ When Sampling from Two Independent Normal Populations

The sampling distribution for $\bar{X} - \bar{Y}$ is normal with mean $\mu_X - \mu_Y$ and standard deviation $\sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}$, where n_X and n_Y are the respective sample sizes. That is,

$$\bar{X} - \bar{Y} \sim N\left(\mu_X - \mu_Y, \sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}\right)$$

provided X and Y are independent random variables where $X \sim N(\mu_X, \sigma_X)$ and $Y \sim N(\mu_Y, \sigma_Y)$. Since X and Y are independent normal random variables, the distributions of

their means are known. Specifically,

$$\bar{X} \sim N\left(\mu_X, \frac{\sigma_X}{\sqrt{n_X}}\right) \quad \text{and} \quad \bar{Y} \sim N\left(\mu_Y, \frac{\sigma_Y}{\sqrt{n_Y}}\right).$$

Proof: Using the results from Theorem 5.1 on page 320 and letting $X_1 = \bar{X}$, $X_2 = \bar{Y}$, $a_1 = 1$, and $a_2 = -1$, obtain

$$\bar{X} - \bar{Y} \sim N\left(\mu_X - \mu_Y, \sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}\right). \quad (6.6)$$

Example 6.14  **Simulating $\bar{X} - \bar{Y}$**  Use simulation to verify empirically that if $X \sim N(\mu_X, \sigma_X)$ and $Y \sim N(\mu_Y, \sigma_Y)$, the resulting sampling distribution of $\bar{X} - \bar{Y}$ is as given in (6.6). Specifically, generate and store in a vector named `meansX` the means of 20,000 samples of size $n_X = 100$ from a normal distribution with $\mu_X = 100$ and $\sigma_X = 10$. Generate and store in a vector named `meansY` the means of 20,000 samples of size $n_Y = 81$ from a normal distribution with $\mu_Y = 50$ and $\sigma_Y = 9$. Compute the mean and standard deviation for the difference between `meansX` and `meansY`. Compute the empirical probability $\mathbb{P}(\bar{X} - \bar{Y} < 52)$ based on the simulated data as well as the theoretical probability $\mathbb{P}(\bar{X} - \bar{Y} < 52)$. Finally, produce density histogram of the differences between `meansX` and `meansY`, and superimpose the density histogram with a normal density having mean and standard deviation equal to the theoretical mean and standard deviation for $(\bar{X} - \bar{Y})$ in this problem.

Solution: In R Code 6.14, `m` represents the number of samples, and `nx`, `mux`, `sigx`, `ny`, `muy`, `sigy`, `muxy`, `meansX`, `meansY`, and `XY` represent n_X , μ_X , σ_X , n_Y , μ_Y , σ_Y , $\mu_X - \mu_Y$, \bar{X} , \bar{Y} , and $\bar{X} - \bar{Y}$, respectively. The `set.seed()` function is used so the same values can be generated at a later date. Before running the simulation, note that the theoretical distribution $(\bar{X} - \bar{Y}) \sim N(100 - 50 = 50, \sqrt{10^2/100 + 9^2/81} = \sqrt{2})$.

R Code 6.14

```
> set.seed(6)
> m <- 20000; nx <- 100; ny <- 81; mux <- 100; sigx <- 10
> muy <- 50; sigy <- 9; muxy <- mux - muy
> sigxy <- sqrt((sigx^2/nx) + (sigy^2/ny))
> meansX <- numeric(m) # vector of zeros
> meansY <- numeric(m) # vector of zeros
> for(i in 1:m){meansX[i] <- mean(rnorm(nx, mux, sigx))}
> for(i in 1:m){meansY[i] <- mean(rnorm(ny, muy, sigy))}
> XY <- meansX - meansY
> c(mean(XY), sd(XY))

[1] 49.979489 1.414943

> mean(XY < 52)           # Simulation answer to P(XY < 52)
[1] 0.92335

> pnorm(52, 50, sqrt(2))   # Theoretical answer to P(XY < 52)
[1] 0.9213504
```

R Code 6.15

```
> p <- ggplot(data = data.frame(x = XY), aes(x = x))
> p + geom_histogram(aes(y = ..density..), fill = "pink",
+                      color = "black", binwidth = 0.5) +
+   stat_function(fun = dnorm, args = list(50, sqrt(2)), size = 1) +
+   labs(x = expression(bar(X) - bar(Y)), y = "") +
+   theme_bw()
```

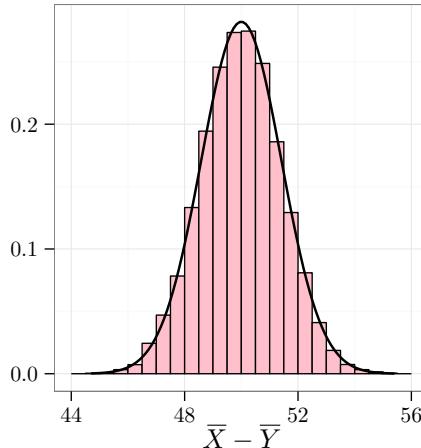


FIGURE 6.5: Density histogram for simulated distribution of $(\bar{X} - \bar{Y})$ with superimposed normal density with $\mu = 50$ and $\sigma = \sqrt{2}$

Note that the empirical mean and standard deviation for $(\bar{X} - \bar{Y})$ are 49.98 and 1.41, respectively, which are very close to the theoretical values of 50 and $\sqrt{2} \approx 1.41$. The empirical probability $\mathbb{P}(\bar{X} - \bar{Y} < 52)$ is computed by determining the proportion of $(\bar{X} - \bar{Y})$ values that are less than 52. Note that the empirical answer for $\mathbb{P}(\bar{X} - \bar{Y} < 52)$ is 0.9234, which is close to the theoretical answer of 0.9214. The density histogram for the empirical distribution of $(\bar{X} - \bar{Y})$ is shown in Figure 6.5. ■

6.5.3 Sampling Distribution for the Sample Proportion

When Y is a binomial random variable, $Y \sim \text{Bin}(n, \pi)$, which represents the number of successes obtained in n trials where the probability of success is π , the sample proportion of successes is typically computed as

$$P = \frac{Y}{n}. \quad (6.7)$$

The mean and variance, respectively, of the sample proportion of successes are

$$E[P] = \mu_P = \pi \quad (6.8)$$

and

$$\text{Var}[P] = \sigma_P^2 = \frac{\pi(1 - \pi)}{n}. \quad (6.9)$$

Equations (6.8) and (6.9) are easily derivable using the mean and variance of Y . Since

$$E[Y] = n\pi \quad \text{and} \quad \text{Var}[Y] = n\pi(1 - \pi),$$

it follows that

$$E[P] = E\left[\frac{Y}{n}\right] = \frac{1}{n}E[Y] = \pi,$$

and

$$\sigma_P^2 = \text{Var}[P] = \text{Var}\left[\frac{Y}{n}\right] = \frac{1}{n^2} \text{Var}[Y] = \frac{\pi(1 - \pi)}{n}.$$

The Central Limit Theorem tells us that the proportion of successes is asymptotically normal for sufficiently large values of n . So that the distribution of P is not overly skewed, both $n\pi$ and $n(1 - \pi)$ must be greater than or equal to 10. The larger $n\pi$ and $n(1 - \pi)$ are, the closer the distribution of P comes to resembling a normal distribution. The rationale for applying the Central Limit Theorem to the proportion of successes rests on the fact that the sample proportion can also be thought of as a sample mean. Specifically,

$$P = \frac{Y_1 + \cdots + Y_n}{n},$$

where each Y_i value takes on a value of 1 if the element possesses the particular attribute being studied and a 0 if it does not. That is, P is the sample mean for the Bernoulli random variable Y_i . Viewed in this fashion, write

$$Z = \frac{P - \pi}{\sqrt{\frac{\pi(1 - \pi)}{n}}} \rightsquigarrow N(0, 1). \quad (6.10)$$

It is also fairly common to approximate the sampling distribution of Y with a normal distribution using the relationship

$$Z = \frac{Y - n\pi}{\sqrt{n\pi(1 - \pi)}} \rightsquigarrow N(0, 1). \quad (6.11)$$

Example 6.15 In plain variety M&M candies, the percentage of green candies is 10%. Suppose a large bag of M&M candies contains 500 candies. What is the probability there will be

- (a) at least 11% green M&Ms?
- (b) no more than 12% green M&Ms?

Solution: First, note that the population proportion of green M&Ms is $\pi = 0.10$. Since neither $n \times \pi = 500 \times 0.10 = 50$ nor $n \times (1 - \pi) = 500 \times 0.90 = 450$ is less than 10, it seems reasonable to appeal to the Central Limit Theorem for the approximate distribution of P . Consequently,

$$P \rightsquigarrow N\left(\pi, \sqrt{\frac{\pi(1 - \pi)}{n}}\right),$$

which, when using the numbers from the problem, becomes

$$P \rightsquigarrow N\left(0.10, \sqrt{\frac{(0.10)(0.90)}{500}} = 0.0134\right).$$

If the random variable Y is equal to the number of green M&Ms, then the distribution of Y can be approximated by

$$Y \sim N\left(n\pi, \sqrt{n\pi(1 - \pi)}\right),$$

which, when using the numbers from the problem, becomes

$$Y \sim N\left(500 \cdot 0.10, \sqrt{500 \cdot 0.10 \cdot (1 - 0.10)} = 6.7082\right).$$

It is also possible to give the exact distribution of Y , which is $Y \sim Bin(n = 500, \pi = 0.10)$.

(a) The probabilities that at least 11% of the candies will be green M&Ms using the approximate distribution of P , the approximate distribution of Y , and finally using the exact distribution of Y are as follows:

$$\begin{aligned} \mathbb{P}(P \geq 0.11) &= \mathbb{P}\left(\frac{P - \pi}{\sigma_P} \geq \frac{0.11 - \pi}{\sigma_P}\right) \approx \mathbb{P}\left(Z \geq \frac{0.11 - 0.10}{0.0134}\right) \\ &= \mathbb{P}(Z \geq 0.7454) = 0.228, \\ \mathbb{P}(Y \geq 55) &= \mathbb{P}\left(\frac{Y - n\pi}{\sqrt{n\pi(1 - \pi)}} > \frac{55 - n\pi}{\sqrt{n\pi(1 - \pi)}}\right) \approx \mathbb{P}\left(Z \geq \frac{55 - 50}{6.7082}\right) \\ &= \mathbb{P}(Z \geq 0.7454) = 0.228, \text{ and} \\ \mathbb{P}(Y \geq 55) &= \sum_{i=55}^{500} \binom{500}{i} (0.10)^i (0.90)^{500-i} = 0.2477. \end{aligned}$$

(b) The probability that no more than 12% of the candies will be green M&Ms is

$$\begin{aligned} \mathbb{P}(P \leq 0.12) &= \mathbb{P}\left(\frac{P - \pi}{\sigma_P} \leq \frac{0.12 - \pi}{\sigma_P}\right) \approx \mathbb{P}\left(Z \leq \frac{0.12 - 0.10}{0.0134}\right) \\ &= \mathbb{P}(Z \leq 1.4907) = 0.932, \\ \mathbb{P}(Y \leq 60) &= \mathbb{P}\left(\frac{Y - n\pi}{\sqrt{n\pi(1 - \pi)}} < \frac{60 - n\pi}{\sqrt{n\pi(1 - \pi)}}\right) \approx \mathbb{P}\left(Z \leq \frac{60 - 50}{6.7082}\right) \\ &= \mathbb{P}(Z \leq 1.4907) = 0.932, \text{ and} \\ \mathbb{P}(Y \leq 60) &= \sum_{i=0}^{60} \binom{500}{i} (0.10)^i (0.90)^{500-i} = 0.9382. \end{aligned}$$

The following R commands compute the answers for (a) and (b):

```
> 1 - pnorm(0.11, 0.1, sqrt(0.1 * 0.9/500))
[1] 0.2280283

> 1 - pnorm(55, 500 * 0.1, sqrt(500 * 0.1 * 0.9))
[1] 0.2280283

> 1 - pbinom(54, 500, 0.1)
[1] 0.2476933

> pnorm(0.12, 0.1, sqrt(0.1 * 0.9/500))
```

```
[1] 0.9319814
> pnorm(60, 500 * 0.1, sqrt(500 * 0.1 * 0.9))
[1] 0.9319814
> pbinom(60, 500, 0.1)
[1] 0.9381745
```

The astute observer will notice that the approximations are not equal to the exact answers. This is due to the fact that a continuous distribution has been used to approximate a discrete distribution. The accuracy of the answers can be improved by applying what is called a **continuity correction**. Using the continuity correction, (6.10) and (6.11) become

$$Z = \frac{P \pm \frac{0.5}{n} - \pi}{\sqrt{\frac{\pi(1-\pi)}{n}}} \rightsquigarrow N(0, 1) \quad (6.12)$$

and

$$Z = \frac{Y \pm 0.5 - n\pi}{\sqrt{n\pi(1 - \pi)}} \rightsquigarrow N(0, 1). \quad (6.13)$$

When solving less than or equal type inequalities, add the continuity correction; and when solving greater than or equal type inequalities, subtract the continuity correction. Notice how much closer the approximations are to the exact answers when the appropriate continuity corrections are applied:

$$\begin{aligned} \mathbb{P}(P \geq 0.11) &= \mathbb{P}\left(\frac{P - \frac{0.5}{500} - \pi}{\sigma_P} \geq \frac{0.11 - \frac{0.5}{500} - \pi}{\sigma_P}\right) \\ &\approx \mathbb{P}\left(Z \geq \frac{0.11 - \frac{0.5}{500} - 0.10}{0.0134}\right) \\ &= \mathbb{P}(Z \geq 0.6708) = 0.2512, \\ \mathbb{P}(Y \geq 55) &= \mathbb{P}\left(\frac{Y - 0.5 - n\pi}{\sqrt{n\pi(1 - \pi)}} > \frac{55 - 0.5 - n\pi}{\sqrt{n\pi(1 - \pi)}}\right) \\ &\approx \mathbb{P}\left(Z \geq \frac{55 - 0.5 - 50}{6.7082}\right) \\ &= \mathbb{P}(Z \geq 0.6708) = 0.2512, \text{ and} \\ \mathbb{P}(Y \geq 55) &= \sum_{i=55}^{500} \binom{500}{i} (0.10)^i (0.90)^{500-i} = 0.2477. \end{aligned}$$

$$\begin{aligned}
 \mathbb{P}(P \leq 0.12) &= \mathbb{P}\left(\frac{P + \frac{0.5}{500} - \pi}{\sigma_P} \leq \frac{0.12 + \frac{0.5}{500} - \pi}{\sigma_P}\right) \\
 &\approx \mathbb{P}\left(Z \leq \frac{0.12 + \frac{0.5}{500} - 0.10}{0.0134}\right) \\
 &= \mathbb{P}(Z \leq 1.5652) = 0.9412, \\
 \mathbb{P}(Y \leq 60) &= \mathbb{P}\left(\frac{Y + 0.5 - n\pi}{\sqrt{n\pi(1-\pi)}} > \frac{60 + 0.5 - n\pi}{\sqrt{n\pi(1-\pi)}}\right) \\
 &\approx \mathbb{P}\left(Z \leq \frac{60 + 0.5 - 50}{6.7082}\right) \\
 &= \mathbb{P}(Z \leq 1.5652) = 0.9412, \text{ and} \\
 \mathbb{P}(Y \leq 60) &= \sum_{i=0}^{60} \binom{500}{i} (0.10)^i (0.90)^{500-i} = 0.9382.
 \end{aligned}$$



Example 6.16 The 1999 North Carolina Department of Public Instruction NC Youth Tobacco Use Survey reported that 38.3% of all North Carolina high school students used tobacco products. If a random sample of 250 North Carolina high school students is taken, find the probability that the sample proportion that use tobacco products will be between 0.36 and 0.40 inclusive.

Solution: Since neither $n \times \pi = 250 \times 0.383 = 95.75$ nor $n \times (1-\pi) = 250 \times 0.617 = 154.25$ is less than 10, it seems reasonable to appeal to the Central Limit Theorem for the approximate distribution of P . Consequently,

$$P \sim N\left(\pi, \sqrt{\frac{\pi(1-\pi)}{n}}\right),$$

which, when using the numbers from the problem, becomes

$$P \sim N\left(0.383, \sqrt{\frac{(0.383)(0.617)}{250}} = 0.0019\right).$$

Due to the discrete nature of the problem, appropriate continuity corrections should be used:

$$\mathbb{P}\left(0.36 - \frac{0.5}{250} \leq P \leq 0.40 + \frac{0.5}{250}\right) = \mathbb{P}(0.358 \leq P \leq 0.402) = 0.5236.$$

To calculate $\mathbb{P}(0.358 \leq P \leq 0.402)$ with R, use `pnorm()`:

```
> sig <- sqrt((0.383 * 0.617)/250)
> pnorm(0.402, 0.383, sig) - pnorm(0.358, 0.383, sig)
[1] 0.5236417
```

The exact answer to the problem can be solved using the binomial distribution as follows:

```
> pbinom(100, 250, 0.383) - pbinom(89, 250, 0.383)
[1] 0.5241166
```



6.5.4 Expected Value and Variance of the Uncorrected Sample Variance and the Sample Variance

Given a random sample X_1, X_2, \dots, X_n taken from a population with mean μ and variance σ^2 , the expected value of the uncorrected variance, S_u^2 , is

$$E [S_u^2] = \frac{1}{n} \sum_{i=1}^n E [(X_i - \bar{X})^2]. \quad (6.14)$$

Expanding the right-hand side of (6.14) gives

$$\begin{aligned} \sum_{i=1}^n (X_i - \bar{X})^2 &= \sum_{i=1}^n [(X_i - \mu) + (\mu - \bar{X})]^2 \\ &= \sum_{i=1}^n [(X_i - \mu)^2 + 2(\mu - \bar{X})(X_i - \mu) + (\mu - \bar{X})^2] \\ &= \sum_{i=1}^n (X_i - \mu)^2 + 2(\mu - \bar{X}) \sum_{i=1}^n (X_i - \mu) + n(\mu - \bar{X})^2 \\ &= \sum_{i=1}^n (X_i - \mu)^2 + 2(\mu - \bar{X})(n\bar{X} - n\mu) + n(\mu - \bar{X})^2 \\ &= \sum_{i=1}^n (X_i - \mu)^2 - 2n(\mu - \bar{X})^2 + n(\mu - \bar{X})^2 \\ &= \sum_{i=1}^n (X_i - \mu)^2 - n(\mu - \bar{X})^2. \end{aligned} \quad (6.15)$$

Substituting the expression $\sum_{i=1}^n (X_i - \mu)^2 - n(\mu - \bar{X})^2$ for $\sum_{i=1}^n (X_i - \bar{X})^2$ in (6.14) gives

$$\begin{aligned} E [S_u^2] &= \frac{1}{n} E \left[\sum_{i=1}^n (X_i - \mu)^2 - n(\mu - \bar{X})^2 \right] \\ E [S_u^2] &= \frac{1}{n} \left(n\sigma^2 - n \frac{\sigma^2}{n} \right) \\ E [S_u^2] &= \sigma^2 - \frac{\sigma^2}{n} \\ &= \sigma^2 \left(\frac{n-1}{n} \right). \end{aligned} \quad (6.16)$$

As (6.16) shows, the expected value of S_u^2 , $\sigma^2 \left(\frac{n-1}{n} \right)$, is less than σ^2 ; however, as n increases, this difference diminishes. The variance for the uncorrected variance S_u^2 , is given by

$$Var [S_u^2] = \frac{\mu_4 - \mu_2^2}{n} - \frac{2(\mu_4 - 2\mu_2^2)}{n^2} + \frac{\mu_4 - 3\mu_2^2}{n^3}, \quad (6.17)$$

where $\mu_k = E[(X - \mu)^k]$ is the k^{th} central moment. Using the definition for the sample variance from (6.4), the expected value of S^2 is readily verified to be σ^2 .

The probability distributions for S_u^2 and S^2 are typically skewed to the right. The skewness diminishes as n increases. Of course, the Central Limit Theorem indicates that the distributions of both are asymptotically normal; however, the convergence to a normal distribution is very slow and requires a very large n . The distributions of S_u^2 and S^2 are extremely important in statistical inference. Two special cases, examined next, are the sampling distributions of S_u^2 and S^2 when sampling from normal populations.

6.6 Sampling Distributions Associated with the Normal Distribution

When a normal distribution is the underlying population from which one is sampling, the sampling distributions of several statistics will naturally occur in various problems. If the sample variance is being examined, the chi-square distribution emerges as the sum of independent, squared, standard normal random variables. If the sample mean is under consideration and the population standard deviation is unknown, a t -distribution can be used to solve problems. A t is the ratio of a standard normal and the square root of a chi-square distribution divided by its degrees of freedom. To compare unknown variances, the ratio of two chi-square random variables divided by their degrees of freedom results in an F distribution. Depending on the statistic of interest, an underlying normal population will provide multiple avenues of analysis through the resulting sampling distributions.

6.6.1 Chi-Square Distribution (χ^2)

The chi-square distribution is a special case of the gamma distribution covered in Section 4.3.3 on page 282. In a paper published in 1900, Karl Pearson popularized the use of the chi-square distribution to measure goodness-of-fit. The **pdf**, $E[X]$, $Var[X]$, and the **mgf** for a chi-square random variable are given in (6.18), where $\Gamma\left(\frac{n}{2}\right)$ is defined in (4.15) on page 283.

Chi-Square Distribution
 $X \sim \chi_n^2$

$$f(x) = \begin{cases} \frac{1}{\Gamma\left(\frac{n}{2}\right) 2^{\frac{n}{2}}} \cdot x^{\frac{n}{2}-1} e^{-\frac{x}{2}} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (6.18)$$

$$E[X] = n$$

$$Var[X] = 2n$$

$$M_X(t) = (1 - 2t)^{-\frac{n}{2}} \text{ for } t < \frac{1}{2}$$

The chi-square distribution is strictly dependent on the parameter n , called the **degrees of freedom**. In general, the chi-square distribution is unimodal and skewed to the right. Three different chi-square distributions are represented in Figure 6.6 on the next page. The

notation used with the chi-square distribution to indicate α of the distribution is in the left tail when the distribution has n degrees of freedom is $\chi_{\alpha;n}^2$. For example, $\chi_{0.95;10}^2$ denotes the value such that 95% of the area is to the left of said value in a χ_{10}^2 distribution.

To find the value corresponding to $\chi_{0.95;10}^2$, use the R command `qchisq(p, df)`, where p is the area to the left (probability) and df is the degrees of freedom. The command gives

```
> qchisq(0.95, 10)
```

```
[1] 18.30704
```

which says that $\mathbb{P}(\chi_{10}^2 < 18.307) = 0.95$.

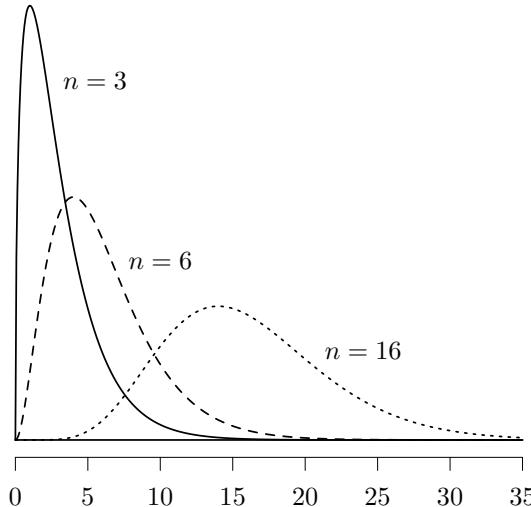


FIGURE 6.6: Illustrations of the **pdfs** of χ_3^2 , χ_6^2 , and χ_{16}^2 random variables

Asymptotic properties. For large values of n ($n > 100$), the distribution of $\sqrt{2\chi_n^2}$ has an approximate normal distribution with a mean of $\sqrt{2n-1}$ and a standard deviation of 1. In other words, because $\sqrt{2\chi_n^2} \rightsquigarrow N(\sqrt{2n-1}, 1)$, $Y = \sqrt{2\chi_n^2} - \sqrt{2n-1} \rightsquigarrow N(0, 1)$. For very large values of n , the approximation

$$Y = \frac{\chi_n^2 - n}{\sqrt{2n}} \rightsquigarrow N(0, 1)$$

may also be used.

Example 6.17 Compute the indicated quantities:

- (a) $\mathbb{P}(\chi_{150}^2 \geq 126)$
- (b) $\mathbb{P}(40 \leq \chi_{65}^2 \leq 50)$
- (c) $\mathbb{P}(\chi_{220}^2 \geq 260)$
- (d) The value a such that $\mathbb{P}(\chi_{100}^2 \leq a) = 0.6$

Solution: The answers are computed first by hand using the approximation $\sqrt{2\chi_n^2} \sim N(\sqrt{2n-1}, 1)$. Then, the exact probabilities are calculated with R.

$$(a) \mathbb{P}(\chi_{150}^2 \geq 126) = \mathbb{P}(\sqrt{2\chi_{150}^2} - \sqrt{299} \geq \sqrt{2(126)} - \sqrt{299}) \approx \mathbb{P}(Z \geq -1.42) = 0.9218.$$

```
> pnorm(sqrt(2 * 126) - sqrt(299), lower = FALSE)
[1] 0.9217744
> 1 - pchisq(126, 150)
[1] 0.9233931
```

The probability a χ_{150}^2 is greater than 126 is approximately 0.9218 and exactly 0.9234.

(b)

$$\begin{aligned} \mathbb{P}(40 \leq \chi_{65}^2 \leq 50) &= \mathbb{P}(\sqrt{2(40)} \leq \sqrt{2\chi_{65}^2} \leq \sqrt{2(50)}) \\ &= \mathbb{P}(\sqrt{80} - \sqrt{129} \leq \sqrt{2\chi_{65}^2} - \sqrt{129} \leq \sqrt{100} - \sqrt{129}) \\ &\approx \mathbb{P}(-2.41 \leq Z \leq -1.36) = 0.0794. \end{aligned}$$

```
> pnorm(sqrt(100) - sqrt(129)) - pnorm(sqrt(80) - sqrt(129))
[1] 0.07936184
> pchisq(50, 65) - pchisq(40, 65)
[1] 0.07861696
```

The probability a χ_{65}^2 is between 40 and 50 is approximately 0.0794 and exactly 0.0786.

(c)

$$\begin{aligned} \mathbb{P}(\chi_{220}^2 \geq 260) &= \mathbb{P}(\sqrt{2\chi_{220}^2} \geq \sqrt{2 \cdot 260}) \\ &= \mathbb{P}(\sqrt{2\chi_{220}^2} - \sqrt{2(220)-1} \geq \sqrt{2 \cdot 260} - \sqrt{2(220)-1}) \\ &\approx \mathbb{P}(Z \geq 1.85) = 0.0321. \end{aligned}$$

```
> pnorm(sqrt(2 * 260) - sqrt(2 * 220 - 1), lower = FALSE)
[1] 0.03207171
> pchisq(260, 220, lower = FALSE)
[1] 0.03335803
```

The probability a χ_{220}^2 is greater than 260 is approximately 0.0321 and exactly 0.0334.

(d)

$$\begin{aligned}\mathbb{P}(\chi_{100}^2 \leq a) &= 0.6 \\ \mathbb{P}\left(\sqrt{2\chi_{100}^2} - \sqrt{2(100)-1} \leq \sqrt{2a} - \sqrt{2(100)-1}\right) &= 0.6 \\ \mathbb{P}\left(Z \leq \sqrt{2a} - \sqrt{2(100)-1}\right) &= 0.6 \\ 0.2533 &= \sqrt{2a} - \sqrt{199} \\ \Rightarrow a &= 103.106.\end{aligned}$$

```
> (qnorm(0.6) + sqrt(199))^2/2
[1] 103.106
> qchisq(0.6, 100)
[1] 102.9459
```

The 0.60 quantile of a χ_{100}^2 is approximately 103.106 and exactly 102.9459.

Note that the approximations are close to the answers from R, but they are not exactly equal. ■

6.6.1.1 The Relationship between the χ^2 Distribution and the Normal Distribution

In addition to describing the χ^2 distribution as a special case of the gamma distribution, the χ^2 distribution can be defined as the sum of independent, squared, standard normal random variables. If n is the number of summed independent, squared, standard normal random variables, then the resulting distribution is a χ^2 distribution with n degrees of freedom, written χ_n^2 . That is,

$$\chi_n^2 = \sum_{i=1}^n Z_i^2, \quad Z_i \sim N(0, 1). \quad (6.19)$$

Theorem 6.1 If $Z \sim N(0, 1)$, then the random variable $Y = Z^2 \sim \chi_1^2$.

Proof: In this proof, it is shown that the distribution of Y is a χ_1^2 :

$$\begin{aligned}F_Y(y) &= \mathbb{P}(Z^2 \leq y) = \mathbb{P}(-\sqrt{y} \leq Z \leq \sqrt{y}) \\ &= 2\mathbb{P}(0 \leq Z \leq \sqrt{y}) = \frac{2}{\sqrt{2\pi}} \int_0^{\sqrt{y}} e^{-x^2/2} dx.\end{aligned}$$

To complete the proof of Theorem 6.1, recall that the derivative inside the integral when certain characteristics are satisfied is

$$\frac{d}{d\theta} \int_{a(\theta)}^{b(\theta)} f(x, \theta) dx = f(b(\theta), \theta) \frac{d}{d\theta} b(\theta) - f(a(\theta), \theta) \frac{d}{d\theta} a(\theta) + \int_{a(\theta)}^{b(\theta)} \frac{\partial f(x, \theta)}{\partial \theta} dx.$$

For the proof, the integral needed is

$$\begin{aligned} \frac{d}{d\theta} \int_{a(\theta)}^{b(\theta)} f(x; \theta) dx &= \frac{d}{dy} \int_0^{\sqrt{y}} e^{-x^2/2} dx \\ &= f(\sqrt{y}) \frac{d}{dy}(\sqrt{y}) - f(0) \frac{d}{dy}(0) + \int_0^{\sqrt{y}} \frac{\partial e^{-x^2/2}}{\partial y} dx \\ &= e^{-y/2} \frac{1}{2\sqrt{y}}. \end{aligned}$$

Taking the derivative of $F_Y(y)$ yields

$$f(y) = \frac{dF_Y(y)}{dy} = \frac{2}{\sqrt{2\pi}} \frac{1}{2\sqrt{y}} e^{-y/2} = \frac{1}{\sqrt{2\pi}\Gamma(1/2)} y^{(1/2)-1} e^{-y/2}, \quad 0 \leq y < \infty,$$

which is the **pdf** for a χ_1^2 . Recall that $\sqrt{\pi} = \Gamma(\frac{1}{2})$ and refer to (6.18).

Corollary 6.1 If $X \sim N(\mu, \sigma^2)$, then $Z = \frac{X-\mu}{\sigma} \sim N(0, 1)$, and $Z^2 \sim \chi_1^2$.

Theorem 6.2 If X_1, \dots, X_r are independent random variables with chi-square distributions $\chi_{n_1}^2, \dots, \chi_{n_r}^2$, respectively, then

$$Y = \sum_{i=1}^r X_i \sim \chi_s^2, \quad \text{where } s = \sum_{i=1}^r n_i.$$

Proof:

$$\begin{aligned} M_Y(t) &= E[e^{tY}] = \prod_{i=1}^r E[e^{tX_i}] = \prod_{i=1}^r M_{X_i}(t) \\ &= \prod_{i=1}^r (1-2t)^{-\frac{n_i}{2}} = (1-2t)^{-\frac{1}{2} \sum_{i=1}^r n_i} \end{aligned}$$

which is the **mgf** for a χ_s^2 distribution.

One of the properties of χ^2 distributions is that of reproducibility. In other words, the sum of independent χ^2 random variables is also a χ^2 distribution with degrees of freedom equal to the sum of the degrees of freedom of each of the independent χ^2 random variables. Corollaries 6.2 and 6.3 are direct consequences of Theorem 6.2.

Corollary 6.2 If X_1, \dots, X_n are independent random variables following a $N(0, 1)$ distribution, then

$$Y = \sum_{i=1}^n X_i^2 \sim \chi_n^2.$$

Corollary 6.3 If X_1, \dots, X_n are independent random variables with $N(\mu_i, \sigma_i^2)$ distributions, respectively, then

$$Y = \sum_{i=1}^n \frac{(X_i - \mu_i)^2}{\sigma_i^2} \sim \chi_n^2.$$

Example 6.18 Given 10 independent and identically distributed (i.i.d.) random variables Y_i , where $Y_i \sim N(0, \sigma = 5)$ for $i = 1, \dots, 10$, compute

$$(a) \mathbb{P}\left(\sum_{i=1}^{10} Y_i^2 \leq 600\right)$$

$$(b) \mathbb{P}\left(\frac{1}{10} \sum_{i=1}^{10} Y_i^2 \geq 12.175\right)$$

$$(c) \text{The number } a \text{ such that } \mathbb{P}\left(\sqrt{\frac{1}{10} \sum_{i=1}^{10} Y_i^2} \geq a\right) = 0.5$$

Solution: The answers are computed using R. Be sure to note that $Z = \frac{Y_i - 0}{5} = \frac{Y_i}{5}$.

(a)

$$\begin{aligned} \mathbb{P}\left(\sum_{i=1}^{10} Y_i^2 \leq 600\right) &= \mathbb{P}\left(\sum_{i=1}^{10} \left(\frac{Y_i}{5}\right)^2 \leq \frac{600}{25}\right) \\ &= \mathbb{P}(\chi_{10}^2 \leq 24) = 0.9924. \end{aligned}$$

Using the R command `pchisq(24, 10)` gives $\mathbb{P}(\chi_{10}^2 \leq 24) = 0.9924$.

```
> pchisq(24, 10)
```

```
[1] 0.9923996
```

(b)

$$\begin{aligned} \mathbb{P}\left(\frac{1}{10} \sum_{i=1}^{10} Y_i^2 \geq 12.175\right) &= \mathbb{P}\left(\sum_{i=1}^{10} \left(\frac{Y_i}{5}\right)^2 \geq \frac{12.175(10)}{25}\right) \\ &= \mathbb{P}(\chi_{10}^2 \geq 4.87) = 0.8997. \end{aligned}$$

```
> pchisq(12.175 * 10/25, 10, lower = FALSE)
```

```
[1] 0.8996911
```

(c)

$$\begin{aligned} \mathbb{P}\left(\frac{1}{10} \sum_{i=1}^{10} Y_i^2 \geq a^2\right) &= \mathbb{P}\left(\sum_{i=1}^{10} \left(\frac{Y_i}{5}\right)^2 \geq \frac{10a^2}{25}\right) \\ &= \mathbb{P}\left(\chi_{10}^2 \geq \frac{10a^2}{25}\right) = 0.5 \end{aligned}$$

Using the R command `qchisq()`, the value $\chi_{10,0.50}^2 = 9.3418$ is calculated:

```
> qchisq(0.5, 10)
```

```
[1] 9.341818
```

Consequently, $\frac{10a^2}{25} = 9.3418$, which yields $a = 4.8327$.

```
> a <- sqrt(qchisq(0.5, 10) * 25/10)
> a
[1] 4.832654
```



6.6.1.2 Sampling Distribution for S_u^2 and S^2 When Sampling from Normal Populations

In this section, the resulting sampling distributions for S_u^2 and S^2 given in Table 6.4 on page 360 when sampling from a normal distribution are considered. Note that $\sum_{i=1}^n (X_i - \bar{X})^2 = nS_u^2 = (n-1)S^2$ and that dividing this by σ^2 yields

$$\sum_{i=1}^n \frac{(X_i - \bar{X})^2}{\sigma^2} = \frac{nS_u^2}{\sigma^2} = \frac{(n-1)S^2}{\sigma^2}. \quad (6.20)$$

The first term in (6.20) appears to be some type of standardized normal random variable. However, it is not, since the sample mean of a random variable is itself a random variable and not a constant. So, what is the distribution then of nS_u^2/σ^2 ? Theorem 6.3 tells us that the distribution of nS_u^2/σ^2 is χ_{n-1}^2 .

Theorem 6.3 Let X_1, \dots, X_n be a random sample from a $N(\mu, \sigma^2)$ distribution. Then,

- (1) \bar{X} and S^2 are independent random variables. Likewise, \bar{X} and S_u^2 are independent random variables.
- (2) The random variable

$$\frac{nS_u^2}{\sigma^2} = \frac{(n-1)S^2}{\sigma^2} = \sum_{i=1}^n \frac{(X_i - \bar{X})^2}{\sigma^2} \sim \chi_{n-1}^2.$$

Proof: A detailed proof of part (1) in Theorem 6.3 is beyond the scope of the text, and the statement will simply be assumed to be true. The independence between \bar{X} and S^2 is a result of normal distributions. Almost without exception, the estimators \bar{X} and S^2 are dependent in all other distributions.

To prove part (2) of Theorem 6.3, use Corollary 6.3 to say that $\sum_{i=1}^n \frac{(X_i - \mu)^2}{\sigma^2} \sim \chi_n^2$. Then, rearrange the terms to find an expression for $\sum_{i=1}^n \frac{(X_i - \bar{X})^2}{\sigma^2}$ for which the distribution is recognizable. Start by rearranging the numerator of the χ_n^2 distribution:

$$\begin{aligned} \sum_{i=1}^n (X_i - \mu)^2 &= \sum_{i=1}^n [(X_i - \bar{X}) + (\bar{X} - \mu)]^2 \\ &= \sum_{i=1}^n (X_i - \bar{X})^2 + \sum_{i=1}^n (\bar{X} - \mu)^2 + 2 \sum_{i=1}^n (X_i - \bar{X})(\bar{X} - \mu). \end{aligned}$$

Since

$$\sum_{i=1}^n (X_i - \bar{X})(\bar{X} - \mu) = (\bar{X} - \mu) \sum_{i=1}^n (X_i - \bar{X}) = 0,$$

it follows that

$$\sum_{i=1}^n (X_i - \mu)^2 = \sum_{i=1}^n (X_i - \bar{X})^2 + n(\bar{X} - \mu)^2. \quad (6.21)$$

Dividing (6.21) by σ^2 gives

$$\frac{\sum_{i=1}^n (X_i - \mu)^2}{\sigma^2} = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{\sigma^2} + \frac{n(\bar{X} - \mu)^2}{\sigma^2},$$

which is the same as

$$\frac{\sum_{i=1}^n (X_i - \mu)^2}{\sigma^2} = \frac{(n-1)S^2}{\sigma^2} + \frac{n(\bar{X} - \mu)^2}{\sigma^2}. \quad (6.22)$$

Since $\bar{X} \sim N(\mu, \frac{\sigma}{\sqrt{n}})$, it follows that $\frac{n(\bar{X} - \mu)^2}{\sigma^2} \sim \chi_1^2$ by Corollary 6.1 on page 387. To simplify notation, let Y , Y_1 , and Y_2 represent $\frac{\sum_{i=1}^n (X_i - \mu)^2}{\sigma^2}$, $\frac{(n-1)S^2}{\sigma^2}$, and $\frac{n(\bar{X} - \mu)^2}{\sigma^2}$ in (6.22), respectively. By part (1) of Theorem 6.3 on the preceding page, Y_1 and Y_2 are independent. Therefore,

$$\begin{aligned} E[e^{tY}] &= E[e^{t(Y_1+Y_2)}] = E[e^{tY_1}] \cdot E[e^{tY_2}] \\ (1-2t)^{-\frac{n}{2}} &= E[e^{tY_1}] \cdot (1-2t)^{-\frac{1}{2}} \\ (1-2t)^{-\frac{(n-1)}{2}} &= E[e^{tY_1}] = M_{Y_1}(t) \Rightarrow Y_1 \sim \chi_{n-1}^2. \end{aligned}$$

Note that $Y_1 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{\sigma^2} \sim \chi_{n-1}^2$ is based on the n quantities $X_1 - \bar{X}, X_2 - \bar{X}, \dots, X_n - \bar{X}$, which sum to zero. Consequently, specifying the values of any $n-1$ of the quantities determines the remaining value. That is, only $n-1$ of the quantities are free to vary. In contrast, $Y = \frac{\sum_{i=1}^n (X_i - \mu)^2}{\sigma^2} \sim \chi_n^2$ has n degrees of freedom since there are no restrictions on the quantities $X_1 - \mu, X_2 - \mu, \dots, X_n - \mu$. In general, when statistics are used to estimate parameters, one degree of freedom is lost for each estimated parameter.

Example 6.19 Show that $E(S_u^2)$, $E(S^2)$, $Var(S_u^2)$, and $Var(S^2)$ are equal to $\frac{(n-1)\sigma^2}{n}$, σ^2 , $\frac{2(n-1)\sigma^4}{n^2}$, and $\frac{2\sigma^4}{n-1}$, respectively, when sampling from a normal distribution.

Solution: It is known that $\frac{nS_u^2}{\sigma^2} = \frac{(n-1)S^2}{\sigma^2} \sim \chi_{n-1}^2$ according to Theorem 6.3 on the previous page. Therefore,

(a)

$$\begin{aligned} E\left[\frac{nS_u^2}{\sigma^2}\right] &= E[\chi_{n-1}^2] = n-1, \text{ so} \\ \frac{n}{\sigma^2}E[S_u^2] &= n-1 \Rightarrow E[S_u^2] = \frac{(n-1)\sigma^2}{n}, \end{aligned}$$

(b)

$$\begin{aligned} E\left[\frac{(n-1)S^2}{\sigma^2}\right] &= E[\chi_{n-1}^2] = n-1 \\ \frac{(n-1)}{\sigma^2}E[S^2] &= n-1 \Rightarrow E[S^2] = \sigma^2, \end{aligned}$$

(c)

$$\begin{aligned} Var\left[\frac{nS_u^2}{\sigma^2}\right] &= Var[\chi_{n-1}^2] = 2(n-1) \\ \frac{n^2}{\sigma^4}Var[S_u^2] &= 2(n-1) \Rightarrow Var[S_u^2] = \frac{2(n-1)\sigma^4}{n^2}, \text{ and} \end{aligned}$$

(d)

$$\begin{aligned} \text{Var}\left[\frac{(n-1)S^2}{\sigma^2}\right] &= \text{Var}[\chi_{n-1}^2] = 2(n-1) \\ \frac{(n-1)^2}{\sigma^4} \text{Var}[S^2] &= 2(n-1) \Rightarrow \text{Var}[S^2] = \frac{2\sigma^4}{(n-1)}. \end{aligned}$$



Example 6.20 A random sample of size 11 is taken from a $N(\mu, \sigma)$ distribution where both the mean and the standard deviation are unknown and the sample variance S^2 is computed. Determine the $\mathbb{P}(0.4865182 < \frac{S^2}{\sigma^2} < 1.598718)$.

Solution: According to Theorem 6.3 on page 389, $\frac{(n-1)S^2}{\sigma^2} \sim \chi_{n-1}^2$, which implies $\frac{10S^2}{\sigma^2} \sim \chi_{10}^2$:

$$\begin{aligned} \mathbb{P}\left(0.4865182 < \frac{S^2}{\sigma^2} < 1.598718\right) &= \mathbb{P}\left(0.4865182(10) < \frac{10S^2}{\sigma^2} < 1.598718(10)\right) \\ &= \mathbb{P}(4.865182 < \chi_{10}^2 < 15.98718) \\ &= \mathbb{P}(\chi_{10}^2 < 15.98718) - \mathbb{P}(\chi_{10}^2 < 4.865182) \\ &= 0.9 - 0.1 = 0.8. \end{aligned}$$

To find $\mathbb{P}(\chi_{10}^2 < 15.98712)$ and $\mathbb{P}(\chi_{10}^2 < 4.865182)$, one can use the R command `pchisq()`:

```
> pchisq(15.98712, 10) - pchisq(4.865182, 10)
[1] 0.7999983
```



Example 6.21 A custom door manufacturer knows that the measurement error in the height of his final products (the door height minus the order height) follows a normal distribution with a variance of $\sigma^2 = 225 \text{ mm}^2$. A local contractor building custom bungalows orders 31 doors. What is the $\mathbb{P}(S > 18.11898 \text{ mm})$ for the 31 doors, and what is the expected value of S^2 ?

Solution:

$$\mathbb{P}(S > 18.11898) = \mathbb{P}\left(\frac{n-1}{\sigma^2} S^2 > \frac{30}{225} 18.11898^2\right) = \mathbb{P}(\chi_{30}^2 > 43.773) = 0.05.$$

The following computes $\mathbb{P}(\chi_{30}^2 > 43.773)$ with R:

```
> pchisq(30 * (18.11898^2)/225, 30, lower = FALSE)
[1] 0.0499998
```



Since the expected value of S^2 is the population variance, $E[S^2] = 225$.

Example 6.22 \triangleright **Probability Distribution of $(n-1)S^2/\sigma^2$** \triangleleft Use simulation to generate $m = 20,000$ samples of size $n = 15$ from both a $N(0, 1)$ distribution and an $Exp(1)$ distribution. Compute the statistic $(n-1)S^2/\sigma^2$ for both the normally and exponentially generated values, labeling the first NC14 and the second EC14. Produce density estimates of NC14 and EC14 and superimpose the theoretical distribution for a χ_{14}^2 distribution on

both. Repeat the entire process with samples of size $n = 100$. That is, use simulation to generate $m = 20,000$ samples of size $n = 100$ from both a $N(0, 1)$ distribution and an $Exp(1)$ distribution. Compute the statistic $(n - 1)S^2/\sigma^2$ for both the normally and exponentially generated values, labeling the first NC99 and the latter EC99. Produce density estimates for NC99 and EC99, and superimpose the theoretical distribution for a χ_{99}^2 distribution on both. What can be concluded about the probability distribution of $(n - 1)S^2/\sigma^2$ when sampling from a normal distribution and when sampling from an exponential distribution based on the density estimates?

Solution: R Code 6.16 generates the required values. To obtain reproducible values, use `set.seed()`. In this solution, `set.seed(302)` is used.

R Code 6.16

```
> m <- 20000
> n <- 15
> set.seed(302)
> varNC14 <- numeric(m) # allocate storage space
> varEC14 <- numeric(m) # allocate storage space
> for (i in 1:m) {
+   varNC14[i] <- var(rnorm(n))
+ }
> for (i in 1:m) {
+   varEC14[i] <- var(rexp(n))
+ }
> NC14 <- (n - 1) * varNC14/1^2
> EC14 <- (n - 1) * varEC14/1^2
> n <- 100
> varNC99 <- numeric(m) # allocate storage space
> varEC99 <- numeric(m) # allocate storage space
> for (i in 1:m) {
+   varNC99[i] <- var(rnorm(n))
+ }
> for (i in 1:m) {
+   varEC99[i] <- var(rexp(n))
+ }
> NC99 <- (n - 1) * varNC99/1^2
> EC99 <- (n - 1) * varEC99/1^2
```

R Code 6.17 stores the results in a data frame named `BDF` and uses `ggplot2` graphics to create the requested density estimates.

R Code 6.17

```
> BDF <- data.frame(RVs = c(NC14, EC14, NC99, EC99),
+                     Type = rep(rep(c("Normal", "Exponential"), each=m), 2),
+                     Size = c(rep("n = 15", 2*m), rep("n = 100", 2*m)) )
> BDF$Size <- factor(BDF$Size, levels = c("n = 15", "n = 100"))
> BDF>Type <- factor(BDF>Type, levels = c("Normal", "Exponential"))
> p <- ggplot(data = subset(BDF, Size == "n = 15"), aes(x = RVs)) +
+   geom_density(fill = "skyblue") +
+   facet_grid(Type ~ Size) +
+   coord_cartesian(xlim = c(0, 60)) +
```

```

+   labs(x = expression((n - 1)*s^2/sigma^2), y = "") +
+   theme_bw()
> p + stat_function(fun = dchisq, args = list(14), col ="blue", n = 500,
+                     size = 1, fill = "blue", alpha = 0.2, geom = "area")
> p <- ggplot(data = subset(BDF, Size == "n = 100"), aes(x = RVs)) +
+   geom_density(fill = "skyblue") +
+   facet_grid(Type ~ Size) +
+   coord_cartesian(xlim = c(0, 200)) +
+   labs(x = expression((n - 1)*s^2/sigma^2), y = "") +
+   theme_bw()
> p + stat_function(fun = dchisq, args = list(99), col ="blue", n = 500,
+                     size = 1, fill = "blue", alpha = 0.2, geom = "area")

```

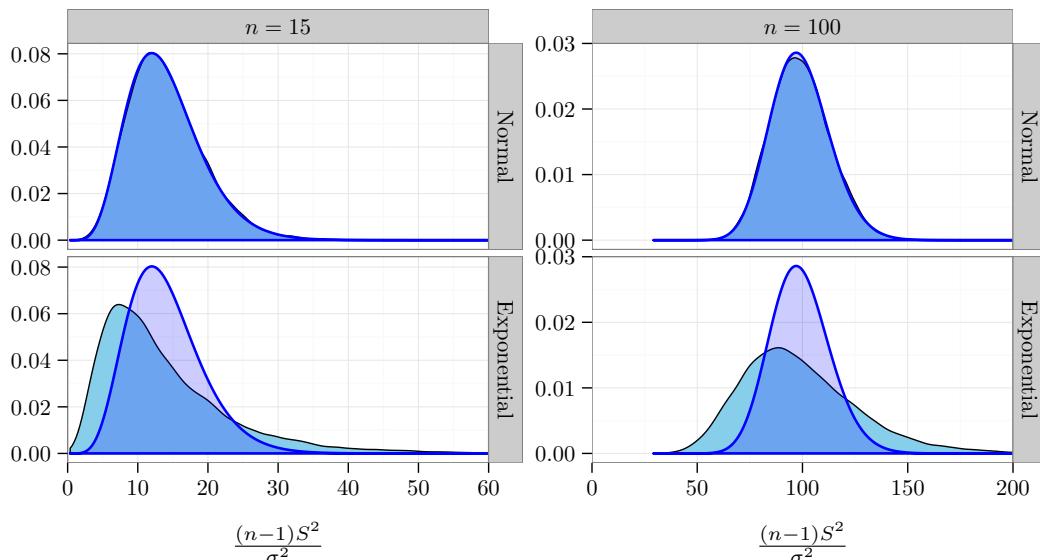


FIGURE 6.7: Probability histograms for simulated distributions of $\frac{(n-1)S^2}{\sigma^2}$ when sampling from normal and exponential distributions. The superimposed density on all curves is a χ_{n-1}^2 .

R Code 6.18 is used to create the values in Table 6.14 on the following page.

R Code 6.18

```

> nc14 <- c(mean(varNC14), var(varNC14), mean(NC14), var(NC14))
> ec14 <- c(mean(varEC14), var(varEC14), mean(EC14), var(EC14))
> nc99 <- c(mean(varNC99), var(varNC99), mean(NC99), var(NC99))
> ec99 <- c(mean(varEC99), var(varEC99), mean(EC99), var(EC99))
> MAT <- rbind(nc14, ec14, nc99, ec99)
> colName <- c("E(S^2)", "Var(S^2)", "E(X^2)", "Var(X^2)")
> rowName <- c("NC14", "EC14", "NC99", "EC99")
> dimnames(MAT) <- list(rowName, colName)
> print(MAT)

```

	$E(S^2)$	$Var(S^2)$	$E(X^2)$	$Var(X^2)$
NC14	1.0036703	0.14328429	14.05138	28.08372
EC14	0.9992937	0.52914289	13.99011	103.71201
NC99	0.9996078	0.02022434	98.96117	198.21877
EC99	0.9983292	0.07962425	98.83459	780.39725

Table 6.14: Output for Example 6.22

	$E[S^2]$	$Var[S^2]$	$E\left[\frac{(n-1)S^2}{\sigma^2}\right]$	$Var\left[\frac{(n-1)S^2}{\sigma^2}\right]$
NC14	1.0037	0.14328	14.05	28.08
EC14	0.9993	0.52914	13.99	103.71
NC99	0.9996	0.02022	98.96	198.22
EC99	0.9983	0.07962	98.83	780.40

Examine Table 6.14, and note that the means for the simulated S^2 values ($E(S^2)$) for NC14, EC14, NC99, and EC99 are all close to the theoretical variance ($\sigma^2 = 1$). However, only when sampling from a normal distribution does the variance of S^2 equal $2\sigma^4/(n - 1)$. That is, the simulated $Var(S^2)$ values for NC14 and NC99 are 0.14328 and 0.02022, which are close to the theoretical values of $\frac{2}{14} = 0.1429$ and $\frac{2}{99} = 0.0202$. The means and variances for the simulated $(n - 1)S^2/\sigma^2$ values are approximately $(n - 1)$ and $2(n - 1)$, respectively, for NC14 and NC99. However, the variances of $(n - 1)S^2/\sigma^2$ when sampling from an exponential are not close to the values returned with NC14 and NC99, nor is the simulated sampling distribution for $(n - 1)S^2/\sigma^2$ approximated very well with a χ_{n-1}^2 distribution when sampling from an exponential distribution, as evidenced by the graphs on the second row of Figure 6.7 on the preceding page. In other words, the sampling distribution for $(n - 1)S^2/\sigma^2$ can only be guaranteed to follow a χ_{n-1}^2 distribution when sampling is from a normal distribution.

6.6.2 t -Distribution

Given a random sample X_1, \dots, X_n that is drawn from a $N(\mu, \sigma)$ distribution, $\bar{X} \sim N(\mu, \sigma/\sqrt{n})$, which implies

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1). \quad (6.23)$$

The quantity (6.23) is used primarily for inference regarding μ . However, this inference assumes σ is known. The assumption of a known σ is generally not reasonable. That is, if μ is unknown, it almost certainly follows that σ will be unknown as well. Fortunately, inference regarding μ can still be performed if σ is replaced by S in (6.23). Specifically, the quantity

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} \quad (6.24)$$

follows a well-known distribution, described next.

DEFINITION 6.1: Given two independent random variables Z and U , where $Z \sim N(0, 1)$ and $U \sim \chi_\nu^2$, we define the t -distribution with ν degrees of freedom as the ratio of Z divided

by the square root of U divided by its degrees of freedom. That is,

$$T = \frac{Z}{\sqrt{\frac{U_\nu}{\nu}}} \sim t_\nu. \quad (6.25)$$

Using definition 6.1, one can readily see why (6.24) follows a t -distribution with $n-1$ degrees of freedom since

$$\frac{\bar{X} - \mu}{S/\sqrt{n}} = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}} \sqrt{\frac{(n-1)S^2}{(n-1)\sigma^2}}} = \frac{\frac{\bar{X} - \mu}{\sigma/\sqrt{n}}}{\sqrt{\frac{\chi_{n-1}^2}{n-1}}} = \frac{Z}{\sqrt{\frac{U_{n-1}}{n-1}}} \sim t_{n-1}.$$

The t -distribution, also called Student's t -distribution, was first described in a paper published by William Sealy Gosset under the pseudonym "Student." Gosset was employed by Guinness Breweries when his research relating to the t -distribution was published. Since Guinness Breweries had a policy preventing research publications by its staff, Gosset published his findings under the pseudonym "Student." Consequently, the t -distribution is often called Student's t -distribution in his honor. The **pdf**, expectation, and variance of a t -distribution with ν degrees of freedom are given in (6.26).

t -Distribution

$$X \sim t_\nu$$

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\Gamma(\frac{\nu}{2})} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \quad \text{for } -\infty < x < \infty \quad (6.26)$$

$$E[X] = 0$$

$$Var[X] = \frac{\nu}{\nu-2} \quad \text{for } \nu > 2$$

The shape of the t -distribution is similar to that of the normal distribution; but for small sample sizes, it has heavier tails than the $N(0, 1)$. Figure 6.8 illustrates the densities for t -distributions with 1, 3, and ∞ degrees of freedom, respectively. Note that $t_{\alpha;\infty} = z_\alpha$. To find the quantity $t_{\alpha;\nu}$, the R command `qt(alpha, nu)` can be used. In particular, suppose $t_{0.80;1}$, depicted in Figure 6.8, is desired. Using the R command `qt(0.80, 1)` gives 1.3764 for the answer.

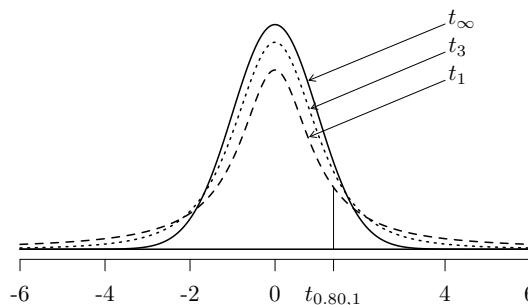


FIGURE 6.8: Illustrations of the **pdfs** of t_1 (dashed line), t_3 (dotted line), and t_∞ (solid line) random variables

Example 6.23 The tensile strength for a type of wire is normally distributed with an unknown mean μ and an unknown variance σ^2 . Five pieces of wire are randomly selected from a large roll, and the strength of each segment of wire is measured. Find the probability that \bar{Y} will be within $\frac{2S}{\sqrt{n}}$ of the true population mean, μ .

Solution: The solution is

$$\begin{aligned}\mathbb{P}\left(\mu - \frac{2S}{\sqrt{n}} \leq \bar{Y} \leq \mu + \frac{2S}{\sqrt{n}}\right) &= \mathbb{P}\left(-\frac{2S}{\sqrt{n}} \leq \bar{Y} - \mu \leq \frac{2S}{\sqrt{n}}\right) \\ &= \mathbb{P}\left(-2 \leq \frac{\bar{Y} - \mu}{S/\sqrt{n}} \leq 2\right) \\ &= \mathbb{P}(-2 \leq t_4 \leq 2) = 0.8839.\end{aligned}$$

```
> pt(2, 4) - pt(-2, 4)
```

```
[1] 0.8838835
```

Note that if σ were known, $\mathbb{P}(-2 \leq Z \leq 2) = 0.9545$.

```
> pnorm(2) - pnorm(-2)
```

```
[1] 0.9544997
```



The Sampling Distribution for $\bar{X} - \bar{Y}$ When σ_X and σ_Y Are Unknown but Assumed Equal

Theorem 6.4 Given two random samples X_1, \dots, X_{n_X} and Y_1, \dots, Y_{n_Y} that are taken from independent normal populations where $X \sim N(\mu_X, \sigma_X)$, $Y \sim N(\mu_Y, \sigma_Y)$, and $\sigma_X = \sigma_Y$, the random variable

$$\frac{[(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)]}{\sqrt{\frac{(n_X-1)S_X^2 + (n_Y-1)S_Y^2}{n_X+n_Y-2} \left(\frac{1}{n_X} + \frac{1}{n_Y}\right)}} \sim t_{n_X+n_Y-2}. \quad (6.27)$$

Proof: Since $\bar{X} - \bar{Y} \sim N\left(\mu_X - \mu_Y, \sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}\right)$, according to Theorem 5.1 on page 320,

$$Z = \frac{(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)}{\sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}} \sim N(0, 1).$$

By Theorem 6.3 on page 389, $\frac{(n_X-1)S_X^2}{\sigma_X^2} \sim \chi_{n_X-1}^2$ and $\frac{(n_Y-1)S_Y^2}{\sigma_Y^2} \sim \chi_{n_Y-1}^2$. Since X and Y are independent, it follows that

$$W = \frac{(n_X-1)S_X^2}{\sigma_X^2} + \frac{(n_Y-1)S_Y^2}{\sigma_Y^2} \sim \chi_{n_X+n_Y-2}^2$$

from Theorem 6.2 on page 387. Using the definition of the t -distribution, given in definition 6.1 on page 394, $\frac{Z}{\sqrt{W}} \sim t_\nu$. In this particular case, $\nu = n_X + n_Y - 2$ and, since

$\sigma_X = \sigma_Y = \sigma$ is assumed,

$$\begin{aligned} \frac{Z}{\sqrt{\frac{W}{\nu}}} &= \frac{\frac{(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)}{\sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}}}{\sqrt{\frac{(n_X - 1)S_X^2}{\sigma_X^2} + \frac{(n_Y - 1)S_Y^2}{\sigma_Y^2}}} \\ &= \frac{(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)}{\sigma \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}} \cdot \frac{1}{\frac{1}{\sigma} \sqrt{\frac{(n_X - 1)S_X^2 + (n_Y - 1)S_Y^2}{n_X + n_Y - 2}}} \\ &= \frac{[(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)]}{\sqrt{\frac{(n_X - 1)S_X^2 + (n_Y - 1)S_Y^2}{n_X + n_Y - 2}} \left(\frac{1}{n_X} + \frac{1}{n_Y} \right)} \sim t_{n_X + n_Y - 2}. \end{aligned}$$

6.6.3 The F Distribution

In Section 6.6.2, it was seen how the t -distribution can be used to make statements about an unknown mean μ when σ is also unknown. Another common problem statisticians face is that of comparing unknown variances, for example, in manufacturing processes, in mixtures, or in quality from different suppliers of goods. The distribution that allows us to make these comparisons is the F distribution.

DEFINITION 6.2: If U and V are independent random variables, each with a χ^2 distribution with ν_1 and ν_2 degrees of freedom, respectively, then

$$\frac{\frac{U}{\nu_1}}{\frac{V}{\nu_2}} \sim F_{\nu_1, \nu_2}.$$

The **pdf**, expected value, and variance of an F distribution are given in (6.28).

F Distribution

$$X \sim F_{\nu_1, \nu_2}$$

$$f(x) = \frac{\Gamma\left(\frac{\nu_1 + \nu_2}{2}\right)}{\Gamma\left(\frac{\nu_1}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)} \left(\frac{\nu_1}{\nu_2}\right)^{\frac{\nu_1}{2}} x^{\frac{\nu_1}{2}-1} \left(1 + \frac{\nu_1}{\nu_2}x\right)^{-\frac{1}{2}(\nu_1 + \nu_2)}, \quad x > 0$$

$$E[X] = \frac{\nu_2}{\nu_2 - 2}$$

$$Var[X] = \frac{2\nu_2^2(\nu_1 + \nu_2 - 2)}{\nu_1(\nu_2 - 2)^2(\nu_2 - 4)} \text{ provided } \nu_2 > 4$$

(6.28)

The F distribution depends on its degrees of freedom and is characterized by a positive skew. Figure 6.9 on the next page illustrates three different F density curves.

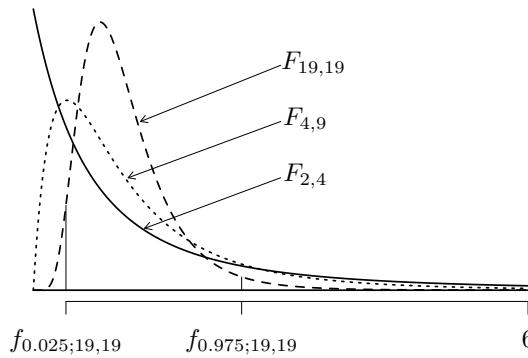


FIGURE 6.9: Illustrations of the **pdfs** of $F_{2,4}$ (solid line), $F_{4,9}$ (dotted line), and $F_{19,19}$ (dashed line) random variables

Theorem 6.5 If there are two random samples X_1, \dots, X_{n_X} and Y_1, \dots, Y_{n_Y} that are taken from independent normal populations where $X \sim N(\mu_X, \sigma_X)$ and $Y \sim N(\mu_Y, \sigma_Y)$, then the random variable

$$\frac{\frac{S_X^2}{\sigma_X^2}}{\frac{S_Y^2}{\sigma_Y^2}} \sim F_{n_X-1, n_Y-1}. \quad (6.29)$$

Proof: Since $\frac{S_X^2}{\sigma_X^2} \sim \frac{\chi_{n_X-1}^2}{n_X-1}$ and $\frac{S_Y^2}{\sigma_Y^2} \sim \frac{\chi_{n_Y-1}^2}{n_Y-1}$, by Theorem 6.3 on page 389, it follows that

$$\frac{\frac{S_X^2}{\sigma_X^2}}{\frac{S_Y^2}{\sigma_Y^2}} \sim F_{n_X-1, n_Y-1}.$$

To find the value $f_{\alpha; \nu_1, \nu_2}$, where $\mathbb{P}(F_{\nu_1, \nu_2} < f_{\alpha; \nu_1, \nu_2}) = \alpha$, with R, use the command `qf(p, df1, df2)`, where p is the area to the left (probability) in an F distribution with $\nu_1 = \text{df1}$ and $\nu_2 = \text{df2}$.

Example 6.24 Find the constants c and d such that $\mathbb{P}(F_{5,10} < c) = 0.95$ and $\mathbb{P}(F_{5,10} < d) = 0.05$.

Solution: Using the R commands `qf(0.95, 5, 10)` and `qf(0.05, 5, 10)` returns the values 3.3258 and 0.2112, respectively.

```
> qf(0.95, 5, 10) # f_{0.95; 5, 10}
[1] 3.325835

> qf(0.05, 5, 10) # f_{0.05; 5, 10}
[1] 0.2111904
```



Example 6.25 Use R to find $\mathbb{P}(F_{19,19} \leq 2.53)$ and $\mathbb{P}(F_{19,19} \leq 0.40)$.

Solution: The answers using R are

```
> pf(2.53, 19)    # P(X <= 2.53)
[1] 0.9751673
> pf(0.40, 19)    # P(X <= 0.40)
[1] 0.02628577
```



Note that a relationship exists between the t - and F distributions. Namely, $t_{\nu}^2 = F_{1,\nu}$, and the relationship between the values in both distributions is

$$t_{1-\alpha/2; \nu}^2 = f_{1-\alpha; 1, \nu}. \quad (6.30)$$

For example, $t_{0.975; 5}^2 = 2.5706^2 = 6.6079 = F_{0.95; 1, 5}$. R Code 6.19 shows the computation with R.

R Code 6.19

```
> qt(0.975, 5)      # t_{0.975; 5}
[1] 2.570582
> qt(0.975, 5)^2    # t_{0.975; 5}^2
[1] 6.607891
> qf(0.95, 1, 5)    # f_{0.95; 1, 5}
[1] 6.607891
```

6.7 Problems

1. How many ways can a host randomly choose 8 people out of 90 in the audience to participate in a TV game show?
2. Let X be a t_5 .
 - (a) Find $\mathbb{P}(X < 3)$.
 - (b) Calculate $\mathbb{P}(2 < X < 3)$.
 - (c) Find a so that $\mathbb{P}(X < a) = 0.05$.
3. If $(1 - 2t)^{-5}$, $t < \frac{1}{2}$, is the **mgf** of a random variable X , find $\mathbb{P}(X < 15.99)$.
4. If $X \sim \chi^2_{10}$, find the constants a and b so that $\mathbb{P}(a < X < b) = 0.90$ and $\mathbb{P}(X < a) = 0.05$.
5. Let X be a χ^2_{10} . Calculate $\mathbb{P}(X < 8)$ and $\mathbb{P}(X > 6)$. Calculate a so that $\mathbb{P}(X < a) = 0.15$. What are the population mean and population variance of X ?
6. Let X be distributed as an $F_{2,5}$. Calculate $\mathbb{P}(X < 1)$ and the median of X . Calculate a so that $\mathbb{P}(X < a) = 0.10$. What are the population mean and population variance of X ?
7. Assume a population with 5 elements:

$$X_1 = 0, \quad X_2 = 1, \quad X_3 = 2, \quad X_4 = 3, \quad X_5 = 4.$$
 - (a) Calculate μ and σ^2 .
 - (b) Calculate the sampling distribution of the mean for random samples of size 3 taken without replacement. Verify that the mean of \bar{X} is 2 and that the variance of \bar{X} is $\frac{\sigma^2}{6}$.
 - (c) Calculate the sampling distribution of \bar{X} for random samples of size 3 taken with replacement. Verify that the mean of \bar{X} is 2 and that the variance of \bar{X} is $\frac{\sigma^2}{n}$.
8. A population has the following elements: 2, 5, 8, 12, 13.
 - (a) Enumerate all the samples of size 2 that can be drawn with and without replacement.
 - (b) Calculate the mean of the population.
 - (c) Calculate the variance of the population.
 - (d) Calculate the standard deviation of the population.
 - (e) Calculate the mean of the sample mean, $E[\bar{X}]$.
 - (f) Calculate the variance of the sampled mean, $Var(\bar{X})$.
 - (g) Calculate the standard deviation of the sample mean.
 - (h) Calculate the mean of the sample variance, $E[S^2]$.
 - (i) Is the variance of \bar{X} larger when sampling with or without replacement? Explain your answer.

9. Determine whether the following expressions are statistics or not:

- (a) $\sum_{i=1}^n X_i$
- (b) $\sum_{i=1}^n X_i - \bar{X}$
- (c) $\bar{X} - \sigma$
- (d) $X_1 + X_2/6$

10. Use the data frame **WHEATUSA2004** from the **PASWR2** package; draw all samples of sizes 2, 3, and 4; and calculate the mean of the means. What size provides the best approximation to the population mean? What is the variance of these means? [—PROB]

11. Given a random sample of size 6 from $N(0, \sigma)$, calculate

- (a) $\mathbb{P}\left(\frac{\bar{X}}{S} > 2\right)$ and
- (b) $\mathbb{P}\left(|\frac{\bar{X}}{S_u}| \leq 4\right)$.

12. Constant velocity joints (CV joints) allow a rotating shaft to transmit power through a variable angle, at constant rotational speed, without an appreciable increase in friction or play. An after-market company produces CV joints. To optimize energy transfer, the drive shaft must be very precise. The company has two different branches that produce CV joints where the variability of the drive shaft is known to be 2 mm. A sample of $n_1 = 10$ is drawn from the first branch, and a sample of $n_2 = 15$ is drawn from the second branch. Suppose that the diameter follows a normal distribution. What is the probability that the drive shafts coming from the first branch will have greater variability than those of the second branch?

13. Given a population $N(\mu, \sigma)$ with unknown mean and variance, a sample of size 11 is drawn and the sample variance S^2 is calculated. Calculate the probability $\mathbb{P}(0.5 < \frac{S^2}{\sigma^2} < 1.2)$.

14. The vendor in charge of servicing coffee dispensers is adjusting the one located in the department of statistics. To maximize profit, adjustments are made so that the average quantity of liquid dispensed per serving is 200 milliliters per cup. Suppose the amount of liquid per cup follows a normal distribution and 5.5% of the cups contain more than 224 milliliters.

- (a) Find the probability that a given cup contains between 176 and 224 milliliters.
- (b) If the machine can hold 20 liters of liquid, find the probability that the machine must be replenished before dispensing 99 cups.
- (c) If 6 random samples of 5 cups are drawn, what is the probability that the sample mean is greater than 210 milliliters in at least 2 of them?

15. The pill weight for a particular type of vitamin follows a normal distribution with a mean of 0.6 grams and a standard deviation of 0.015 grams. It is known that a particular therapy consisting of a box of vitamins with 125 pills is not effective if more than 20% of the pills are under 0.58 grams.

- (a) Find the probability that the therapy with a box of vitamins is not effective.
- (b) A supplement manufacturer sells vitamin bottles containing 125 vitamins per bottle with 50 bottles per box with a guarantee that at least 47 bottles per box weigh more than 74.7 grams. Find the probability that a randomly chosen box does not meet the guaranteed weight.
16. Verify empirically that $(\bar{X} - \mu)/(S/\sqrt{n})$ follows a t_{n-1} distribution when sampling from a normal distribution. Specifically, set the seed to 78 and generate 10,000 samples of size $n = 15$ from a $N(\mu = 3, \sigma = 2)$. Perform the appropriate arithmetic to arrive at the simulated sampling distribution. Create a density histogram of the results and superimpose a theoretical t_{14} density.
17. Simulate 20,000 random samples of sizes 30, 100, 300, and 500 from an exponential distribution with a mean of $\frac{1}{5}$. Estimate the density of the sampling distribution of the sample mean with the function `density()`. Superimpose a theoretical normal density with appropriate mean and standard deviation. What sample size is needed to get an estimated density close to a normal density?
18. The plastic tubes produced by company X for the irrigation system used in golf courses have a mean length of 1.5 meters and a standard deviation of 0.1 meter. The plastic tubes produced by company Y have a mean length of 1 meter and a standard deviation of 0.09 meter. Suppose that both tube lengths follow normal distributions.
- (a) Calculate the probability that a random sample of 15 tubes from company X has a mean length at least 0.45 meter greater than the mean length of a random sample of size 20 from company Y .
- (b) Suppose that the population variances are unknown but equal, $S_x = 0.1$, and $S_y = 0.09$. Calculate the probability that a random sample of 15 plastic tubes from company X has a mean length at least 0.45 meter greater than the mean length of a random sample of 20 plastic tubes from company Y .
19. Plot the density function of an $F_{4,6}$ random variable. Find the area to the left of $x = 3$ and shade this region in the original plot.
20. Let X_1, X_2, X_3, X_4 be a random sample from a $N(0, \sigma)$. Calculate the distribution of $\frac{(X_1 - X_2)^2}{(X_3 + X_4)^2}$.
21. Let $X_1, X_2, X_3, X_4, X_5, X_6$ be a random sample drawn from a $N(0, \sigma)$ population. Find the values of c so that the statistic $\frac{cX_1 + X_2 + X_3}{\sqrt{X_4^2 + X_5^2 + X_6^2}}$ follows a t_3 -distribution.
22. A company manufactures cast iron sewer pipes that have a length with a $N(1.5 \text{ m}, 0.01 \text{ m})$ distribution.
- (a) If the quality control standard for the company is to scrap any pipe with length less than 1.49 m, what percent of the company's cast iron sewer pipes are scrapped?
- (b) Pipes are transported in hard plastic crates with 50 pipes per crate after passing quality control. The plastic crate will close only if all pipes are less than 1.54 m. Find the probability a randomly selected plastic crate cannot be closed.

- (c) A building company orders 100 cast iron pipes they need to connect a local sewer line 149.8 m from the main sewer line. If the cast iron pipes are fit together with solder that does not add to the length of fitted pipes, what is the probability 100 pipes reach the main sewer line?
23. Consider a random sample of size n from an exponential distribution with parameter λ . Use moment generating functions to show that the sample mean follows $a\Gamma(n, \lambda n)$. Graph the theoretical sampling distribution of \bar{X} when sampling from an $Exp(\lambda = 1)$ for $n = 30, 100, 300$, and 500 . Superimpose an appropriate normal density for each $\Gamma(n, \lambda n)$. At what sample size do the sampling distribution and superimposed density virtually coincide?
24. Set the seed equal to 10, and simulate 20,000 random samples of size $n_x = 65$ from a $N(4, \sigma_x = \sqrt{2})$, 20,000 random samples of size $n_y = 90$ from a $N(5, \sigma_y = \sqrt{3})$, and verify that the simulated statistic $\frac{S_x^2/\sigma_x^2}{S_y^2/\sigma_y^2}$ follows an $F_{64,89}$ distribution.
25. Set the seed equal to 95, and simulate $m = 20,000$ random samples of size $n = 1000$ from a $Bernoulli(\pi = 0.4)$. Verify that the sample proportion follows an approximate normal distribution with a mean approximately equal to 0.4 and a standard deviation approximately equal to 0.01549.
26. Given $X \sim N(0, \sigma = 1)$, $Y \sim N(2, \sigma = 2)$, and $Z \sim N(4, \sigma = 3)$, what is the distribution of $W = X + Y + Z$? Set the seed equal to 368 and simulate 10,000 samples, each of size 1 for X , Y , and Z . Add the values in the three vectors to obtain W 's empirical distribution. Create a density histogram of the simulated values of W and superimpose the theoretical density of W .
27. Set the seed equal to 48, and simulate a χ_3^2 distribution by summing the squares of three simulated standard normal random variables, each having length 20,000. Create a density histogram of the simulated χ_3^2 random variable. Superimpose the theoretical χ_3^2 density over the histogram.
28. If X_1, X_2, X_3, X_4, X_5 , and X_6 are independent normal random variables with $\mu = 0$ and $\sigma = 5$,
- Verify empirically that
- $$T = \frac{X_1 + X_2 + X_3}{\sqrt{(X_4^2 + X_5^2 + X_6^2)}} \sim t_3.$$
- Specifically, set the seed to 28 and generate 10,000 values from a $N(\mu = 0, \sigma = 5)$ distribution for each X_i , $i = 1, \dots, 6$. Create a density histogram of the results and superimpose a theoretical t_3 density. If the value of σ is changed, does the random variable T still have a t_3 distribution?
- Compute the empirical probability that $\mathbb{P}(T > 1)$. Compute the percent difference between the empirical answer to $\mathbb{P}(T > 1)$, and the theoretical probability $\mathbb{P}(T = t_3 > 1)$.
29. Verify empirically that

$$F = \frac{\frac{Z^2}{\chi_{20}^2}}{\frac{20}{20}} \sim F_{1,20}$$

by setting the seed equal to 37, and generating 20,000 values from a $N(0, 1)$, and another 20,000 values from a χ^2_{20} to simulate the random variable of interest. Create a density histogram of the results and superimpose a theoretical $F_{1,20}$ density.

- (a) Compute the empirical probability that $\mathbb{P}(1.5 < F < 2)$ and compare the answer to the theoretical probability $\mathbb{P}(1.5 < F_{1,20} < 2)$.
- (b) Compute the empirical probability that $\mathbb{P}(F < 2 | F > 1.5)$ and compare the answer to the theoretical probability $\mathbb{P}(F_{1,20} < 2 | F_{1,20} > 1.5)$.

30. Verify empirically that

$$\frac{N(0, 1)}{\left(\frac{1}{5}\chi^2_5\right)^{\frac{1}{2}}} \sim t_5$$

by setting the seed equal to 36 and generating a sample of size 20,000 from a $N(0, 1)$ distribution. Generate another sample of size 20,000 from a χ^2_5 distribution. Perform the appropriate arithmetic to arrive at the simulated sampling distribution. Create a density histogram of the results and superimpose a theoretical t_5 density.

31. A farmer is interested in knowing the mean weight of his chickens when they leave the farm. Suppose that the standard deviation of the chickens' weight is 500 grams.

- (a) What is the minimum number of chickens needed to ensure that the standard deviation of the mean is no more than 100 grams?
- (b) If the farm has three coops and the mean chicken weight in each coop is 1.8, 1.9, and 2 kg, respectively, calculate the probability that a random sample of 50 chickens with an average weight larger than 1.975 kg comes from the first coop. Assume the weight of the chickens follows a normal distribution.

32. 15.3% of the Spanish Internet domain names are “.org.” If a sample of 2000 Spanish domain names is taken,

- (a) Calculate the exact probability that at least 300 domain names will be “.org.”
- (b) Compute an approximate answer that at least 300 domain names will be “.org.” with a normal approximation.

33. Set the seed equal to 86, and simulate $m_1 = 20,000$ samples of size $n_1 = 1000$ from a $Bin(n_1, \pi = 0.3)$ and $m_2 = 20,000$ samples of size $n_2 = 1100$ from a $Bin(n_2, \pi = 0.7)$. Verify that the difference of sampling proportions follows a normal distribution.

34. Given a random sample of size n from an exponential distribution with parameter λ , prove that the sample mean follows $a\Gamma(n, \lambda n)$. Set the seed equal to 679, and simulate $m = 20,000$ random samples of size $n = 100$ from an $Exp(\lambda = 1)$, and check that the normal approximation of the mean is appropriate. Repeat this exercise with random samples of size $n = 3$, and verify that, in this case, $\Gamma(3, 3)$ is more appropriate to use than the normal distribution.

35. Suppose $X_1, X_2, \dots, X_n \stackrel{i.i.d.}{\sim} Unif(0, 1)$. Use the central limit theorem to find n such that $\mathbb{P}(|\bar{X} - 1/2| < 0.10) \geq 0.90$.

Chapter 7

Point Estimation

7.1 Introduction

Throughout this chapter, random samples drawn from known distributions where the unknown parameters that characterize these distributions will be of interest. To specify completely a probability distribution, whether it be discrete or continuous, the distribution's parameters must be specified. For example, a random variable may follow a normal distribution; however, if both the mean and the standard deviation of the normal distribution are not known, the distribution at hand cannot be completely specified. In a similar fashion, a Poisson random variable requires knowledge of the parameter λ to specify that distribution completely. In general, the **pdf** of a random variable X is $f(x|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the vector of parameters that characterize the **pdf**. The vector of parameters $\boldsymbol{\theta}$ is defined over a parameter space denoted Θ . For each value of $\boldsymbol{\theta} \in \Theta$, there is a different **pdf**. To obtain possible values for the vector of parameters, a random sample from the population of interest is taken, and statistics called **estimators** are constructed. The values of the estimators are called **point estimates**. For example, \bar{X} may be used as a point estimator for μ ; in which case, \bar{x} is a point estimate of μ .

7.2 Properties of Point Estimators

Since estimators are statistics or functions of random variables, they themselves are random variables. Studying the sampling distributions of estimators as well as their statistical properties such as mean squared error, bias, efficiency, consistency, and robustness, all of which will be defined in this chapter, will give guidelines about which estimators to employ.

7.2.1 Mean Square Error

The desirability of an estimator is related to how close its estimates are to the true parameter. The difference between an estimator T for an unknown parameter θ and the parameter θ itself is called the error. Since this quantity can be either positive or negative, it is common to square the error so that various estimators T_1, T_2, \dots , can be compared using a non-negative measure of error. To that end, the **mean square error** of an estimator, denoted $MSE[T]$, is defined as $MSE[T] = E[(T - \theta)^2]$. Estimators with small MSE s will have a distribution such that the values in the distribution will be close to the true parameter. In fact, the MSE consists of two non-negative components, the variance of the estimator T , defined as $Var[T] = E[(T - E[T])^2]$, and the squared bias of the estimator T ,

where bias is defined as $E[T] - \theta$ since

$$\begin{aligned}
 MSE[T] &= E[(T - E[T])^2] \\
 &= E[(T - E[T))^2] + E[(E[T] - \theta)^2] + 2E[(T - E[T])(E[T] - \theta)] \\
 &= Var[T] + (E[T] - \theta)^2 + 2(E[T] - E[T])(E[T] - \theta) \\
 &= Var[T] + (E[T] - \theta)^2 \\
 &= Var[T] + (Bias[T])^2.
 \end{aligned} \tag{7.1}$$

The concepts of variance and bias are illustrated in Figure 7.1, which depicts the shot patterns for four marksmen on their respective targets. When the marksman's weapon is properly sighted, the center of the target represents θ .

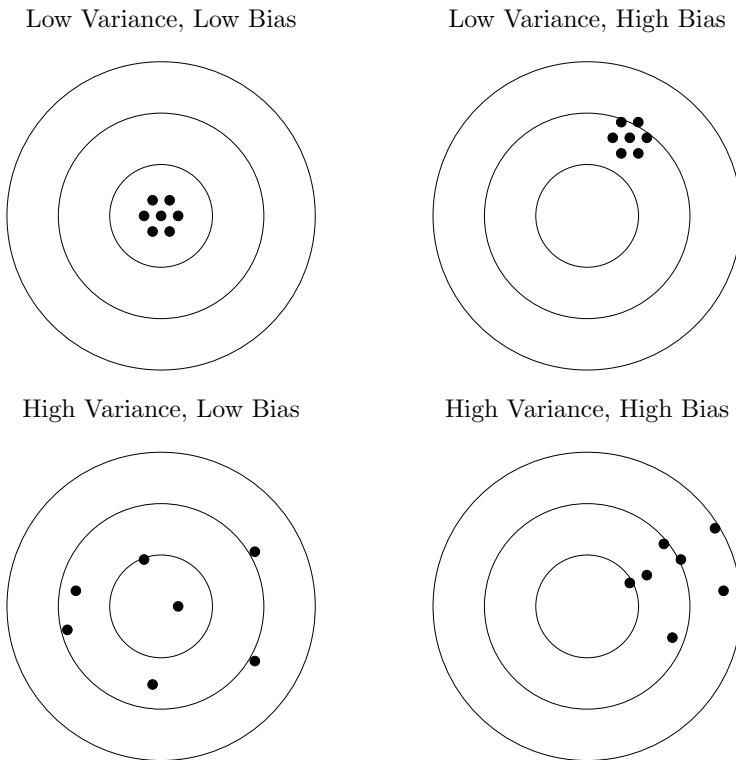


FIGURE 7.1: Visual representations of variance and bias

It seems logical to think that the most desirable estimators are those that minimize the MSE . However, estimators that minimize the MSE for all possible values of θ do not always exist. In other words, an estimator may have the minimum MSE for some values of θ and not others.

7.2.2 Unbiased Estimators

Since estimators are random variables, the point estimates they return will vary from sample to sample; however, one would like some assurance that the chosen estimator is returning a value close to the unknown parameter. Estimators whose expected values are

equal to the parameters they are estimating are **unbiased**. That is, when $E[T] = \theta$, T is an **unbiased** estimator of θ . When an estimator is unbiased, its MSE is equal to its variance, that is, $MSE[T] = Var[T]$. On the other hand, when $E[T] \neq \theta$, the estimator is biased.

Example 7.1 Show that the sample mean and the sample variance are unbiased estimators of the population mean and the population variance, respectively.

Solution: To show that S^2 is an unbiased estimator of σ^2 , use the fact that

$$\sum_{i=1}^n (X_i - \bar{X})^2 = \sum_{i=1}^n (X_i - \mu)^2 - n(\mu - \bar{X})^2.$$

From (6.15) on page 382:

$$\begin{aligned} E[\bar{X}] &= E\left[\sum_{i=1}^n \frac{X_i}{n}\right] = \frac{n\mu}{n} = \mu \text{ and} \\ E[S^2] &= E\left[\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}\right] = E\left[\frac{\sum_{i=1}^n (X_i - \mu)^2 - n(\mu - \bar{X})^2}{n-1}\right] \\ &= \frac{1}{n-1} \left[\sum_{i=1}^n E[(X_i - \mu)^2] - nE[(\bar{X} - \mu)^2] \right]. \end{aligned}$$

Using the definitions for the variances of X and \bar{X} as given in (3.9) and (6.3), respectively,

$$E[S^2] = \frac{1}{n-1} \left[n\sigma^2 - n\frac{\sigma^2}{n} \right] = \sigma^2.$$



Example 7.2 Suppose $X \sim Pois(\lambda)$, where λ is unknown. Show

- (a) \bar{X} is an unbiased estimator of λ .
- (b) $2\bar{X}$ is an unbiased estimator of 2λ .
- (c) \bar{X}^2 is a biased estimator of λ^2 .

Solution: To solve the problems, keep in mind that if $X \sim Pois(\lambda)$, $E[X] = \lambda$ and $Var[X] = \lambda$.

(a) Since $E[\bar{X}] = E\left[\sum_{i=1}^n \frac{X_i}{n}\right] = \sum_{i=1}^n \frac{E[X_i]}{n} = \frac{n\lambda}{n} = \lambda$, it follows that \bar{X} is an unbiased estimator of λ .

(b) Since $E[2\bar{X}] = 2E[\bar{X}] = 2\lambda$, it follows that $2\bar{X}$ is an unbiased estimator of 2λ .

(c) Since $E[\bar{X}^2] = Var[\bar{X}] + \mu_{\bar{X}}^2 = \frac{\lambda}{n} + \lambda^2$, it follows that \bar{X}^2 is a biased estimator of λ^2 . However, \bar{X}^2 is an asymptotically unbiased estimator of λ^2 . That is, as n tends to infinity, the estimator becomes unbiased.



Example 7.3 Suppose $\{X_1, X_2, \dots, X_n\}$ is a random sample from a $N(\mu, \sigma^2)$ distribution.

- (a) Find the expected value of S .

(b) Use graphical techniques to show that S is a biased estimator of σ .

Solution: Recall that $\frac{(n-1)S^2}{\sigma^2} \sim \chi_{n-1}^2$.

(a) Let $X = \frac{(n-1)S^2}{\sigma^2}$ and take the square root and the expected value of both sides:

$$E[\sqrt{X}] = E\left[\frac{\sqrt{n-1}}{\sigma} \cdot S\right].$$

Since $X \sim \chi_{n-1}^2$, the expected value of \sqrt{X} is $\int_{-\infty}^{\infty} \sqrt{x} f(x) dx$, where $f(x)$ is the pdf of a chi-square random variable:

$$\begin{aligned} E[\sqrt{X}] &= \int_0^{\infty} \sqrt{x} \frac{1}{\Gamma(\frac{n-1}{2}) 2^{\frac{n-1}{2}}} x^{\frac{n-1}{2}-1} e^{-\frac{x}{2}} dx \\ &= \frac{1}{\Gamma(\frac{n-1}{2}) 2^{\frac{n-1}{2}}} \int_0^{\infty} x^{\left(\frac{n-1}{2}-1+\frac{1}{2}\right)} e^{-\frac{x}{2}} dx \\ &= \frac{1}{\Gamma(\frac{n-1}{2}) 2^{\frac{n-1}{2}}} \int_0^{\infty} x^{\frac{n}{2}-1} e^{-\frac{x}{2}} dx. \end{aligned} \quad (7.2)$$

Next, use the change of variable $x/2 = t$ where $dx = 2dt$ in an attempt to force the right-hand side of (7.2) to look like a gamma function. Specifically, recall that $\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx$ for $\alpha > 0$:

$$\begin{aligned} E[\sqrt{X}] &= \frac{1}{\Gamma(\frac{n-1}{2}) 2^{\frac{n-1}{2}}} \int_0^{\infty} (2t)^{\frac{n}{2}-1} e^{-t} 2 dt \\ &= \frac{2^{\frac{n}{2}}}{\Gamma(\frac{n-1}{2}) 2^{\frac{n-1}{2}}} \int_0^{\infty} t^{\frac{n}{2}-1} e^{-t} dt = \frac{\sqrt{2} \Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})}. \end{aligned}$$

Since

$$E[\sqrt{X}] = E\left[\frac{\sqrt{n-1}}{\sigma} S\right] = \frac{\sqrt{2} \Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})},$$

it follows that

$$E[S] = \sigma \frac{\sqrt{2} \Gamma(\frac{n}{2})}{\sqrt{n-1} \Gamma(\frac{n-1}{2})}. \quad (7.3)$$

(b) Numerically evaluate and graph the coefficient $\frac{\sqrt{2} \Gamma(\frac{n}{2})}{\sqrt{n-1} \Gamma(\frac{n-1}{2})}$ that multiplies σ on the right-hand side of (7.3) for values of n from 2 to 50. R Code 7.1 on the next page creates a graph similar to the one depicted in Figure 7.2 on the facing page. Note that the coefficient $\frac{\sqrt{2} \Gamma(\frac{n}{2})}{\sqrt{n-1} \Gamma(\frac{n-1}{2})}$ is virtually 1 for values of $n \geq 20$, so that S is a reasonable, though biased, estimator of σ for $n \geq 20$. The graph in Figure 7.2 on the next page is created with ggplot2. One could create a similar graph with base graphics using the call `curve(f, ...)`; however, one would have to tinker with the margins to ensure the mathematical symbols are included in the final output.

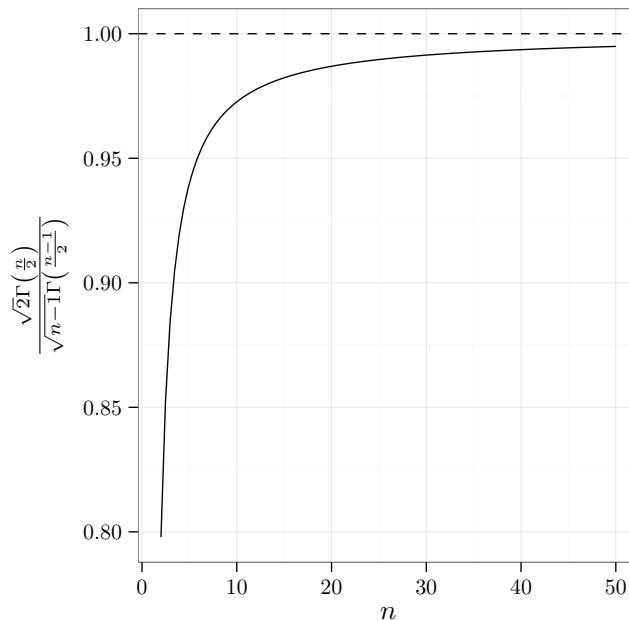


FIGURE 7.2: Graph representing the bias of S when it is used to estimate σ when sampling from a normal distribution

R Code 7.1

```
> f <- function(x){sqrt(2/(x - 1))*gamma(x/2)/gamma((x - 1)/2)}
> p <- ggplot(data.frame(x = c(2, 50)), aes(x = x))
> p + stat_function(fun = f) +
+   labs(x = "n", y = expression(frac(sqrt(2)*phantom(0)*Gamma*
+     bgroup(",frac(n, 2),")", sqrt(n - 1)*phantom(0)*Gamma*
+     bgroup(",frac(n - 1, 2),")")))) +
+   geom_hline(yintercept = 1, lty = "dashed") +
+   theme_bw()
```

Because Figure 7.2 shows that the coefficient of σ in the $E[S]$ is something other than one, S is a biased estimator of σ . ■

7.2.3 Efficiency

Another desirable property of a good estimator is not only to be unbiased, but also to have a small variance, which translates into a small MSE for estimators, regardless of whether they are biased or unbiased. The efficiency of an estimator T is the inverse of its mean squared error, written as

$$eff[T] = \frac{1}{MSE[T]}. \quad (7.4)$$

An estimator T_1 is said to be more precise than the estimator T_2 if $eff[T_1] \geq eff[T_2]$ or if $MSE[T_1] \leq MSE[T_2]$.

7.2.3.1 Relative Efficiency

One way to compare the *MSEs* of two estimators is by using **relative efficiency**. Given two estimators T_1 and T_2 , the efficiency of T_1 relative to T_2 , written $eff(T_1, T_2)$, is

$$eff(T_1, T_2) = \frac{eff(T_1)}{eff(T_2)} = \frac{MSE[T_2]}{MSE[T_1]}. \quad (7.5)$$

When the estimators in (7.5) are unbiased, the efficiency of T_1 relative to T_2 is simply the ratio of estimators variances, written

$$eff(T_1, T_2) = \frac{Var[T_2]}{Var[T_1]}.$$

The estimator T_1 is more efficient than the estimator T_2 if, for any sample size, $MSE[T_1] \leq MSE[T_2]$, which then implies that $eff(T_1, T_2) \geq 1$. When the estimators are unbiased, the estimator T_1 is more efficient than the estimator T_2 if, for any sample size, $Var[T_1] \leq Var[T_2]$, which also implies that $eff(T_1, T_2) \geq 1$. If a choice is to be made among a small number of unbiased estimators, simply compute the variance of all of the estimators, and select the estimator with minimum variance. If the estimator that has the smallest variance among all possible unbiased estimators must be chosen, an infinite number of variances would need to be calculated. Clearly, this is not a viable solution.

Example 7.4 Suppose a sample of size $n = 5$ is taken from an $Exp(\lambda = 1/\theta)$ distribution, and estimators T_1 and T_2 of the mean are defined as

$$T_1 = \frac{X_1 + X_2 + X_3 + X_4 + X_5}{5} \text{ and } T_2 = \frac{2X_1 + X_2 + X_3 - X_4 + 3X_5}{6}.$$

Find the relative efficiency of T_1 to T_2 .

Solution: Both T_1 and T_2 are unbiased estimators of λ since

$$E[T_1] = E[\bar{X}] = \theta, \text{ and}$$

$$E[T_2] = E\left[\frac{2X_1 + X_2 + X_3 - X_4 + 3X_5}{6}\right] = \frac{2\theta + \theta + \theta - \theta + 3\theta}{6} = \theta.$$

The variance of T_1 and T_2 are

$$Var[T_1] = Var[\bar{X}] = \frac{\theta^2}{5}, \text{ and}$$

$$\begin{aligned} Var[T_2] &= Var\left[\frac{2X_1 + X_2 + X_3 - X_4 + 3X_5}{6}\right] \\ &= \frac{1}{36}(4Var[X_1] + Var[X_2] + Var[X_3] + Var[X_4] + 9Var[X_5]) = \frac{16\theta^2}{36} = \frac{4\theta^2}{9}. \end{aligned}$$

The relative efficiency of T_1 to T_2 is

$$eff(T_1, T_2) = \frac{MSE(T_2)}{MSE(T_1)} = \frac{Var(T_2)}{Var(T_1)} = \frac{\frac{4\theta^2}{9}}{\frac{\theta^2}{5}} = \frac{20}{9} > 1,$$

which implies that T_1 is more efficient than T_2 .



Example 7.5 ▷ Comparing Estimators: Blue Jean Length ◁ Suppose the true manufactured length of new 32L blue jeans follows a normal distribution with unknown μ and $\sigma = 0.5$ inch. It is known that 32L blue jeans sold in stores have a length of at least 31 inches. If a random sample of size $n = 3$ of 32L blue jeans is taken to estimate μ , which of the estimators $\hat{\mu}_1$ or $\hat{\mu}_2$ is better in terms of bias, variance, and relative efficiency where $\hat{\mu}_1 = 0.33 \cdot (X_1 + X_2 + X_3)$ and $\hat{\mu}_2 = 0.50 \cdot (X_1 + X_2)$?

Solution: Since

$$\begin{aligned} E[\hat{\mu}_1] &= 0.33 \cdot E[X_1 + X_2 + X_3] = 0.33 \cdot (E[X_1] + E[X_2] + E[X_3]) \\ &= 0.33(\mu + \mu + \mu) = 0.99\mu, \end{aligned}$$

it follows that $\hat{\mu}_1$ is a biased estimator of μ with bias $0.99\mu - \mu = -0.01\mu$. On the other hand,

$$E[\hat{\mu}_2] = 0.50 \cdot E[X_1 + X_2] = 0.50 \cdot (E[X_1] + E[X_2]) = 0.50 \cdot (\mu + \mu) = \mu,$$

which makes $\hat{\mu}_2$ an unbiased estimator of μ . The variances of $\hat{\mu}_1$ and $\hat{\mu}_2$ are

$$\begin{aligned} \text{Var}[\hat{\mu}_1] &= \text{Var}[0.33 \cdot (X_1 + X_2 + X_3)] \\ &= 0.33^2 \cdot (\text{Var}[X_1] + \text{Var}[X_2] + \text{Var}[X_3]) \\ &= 0.33^2 \cdot (0.25 + 0.25 + 0.25) = 0.081675, \text{ and} \\ \text{Var}[\hat{\mu}_2] &= \text{Var}[0.50 \cdot (X_1 + X_2)] = 0.50^2 \cdot (\text{Var}[X_1] + \text{Var}[X_2]) \\ &= 0.25 \cdot (0.25 + 0.25) = 0.125, \text{ respectively.} \end{aligned}$$

Before looking at the relative efficiency of $\hat{\mu}_1$ to $\hat{\mu}_2$, compute the *MSE* for each estimator using the fact that $\text{MSE} = \text{Variance} + \text{Bias}^2$:

$$\begin{aligned} \text{MSE}[\hat{\mu}_1] &= 0.081675 + (0.01\mu)^2 = 0.081675 + 0.0001\mu^2 \text{ and} \\ \text{MSE}[\hat{\mu}_2] &= 0.125 + 0^2 = 0.125. \end{aligned}$$

If

$$\text{eff}(\hat{\mu}_1, \hat{\mu}_2) = \frac{\text{MSE}(\hat{\mu}_2)}{\text{MSE}(\hat{\mu}_1)} = \frac{0.125}{0.081675 + 0.0001\mu^2} < 1,$$

one can conclude that $\hat{\mu}_2$ is both more efficient and has a smaller *MSE* than does $\hat{\mu}_1$.

```
> Mu <- sqrt((0.125 - 0.081675)/1e-04)
```

```
> Mu
```

```
[1] 20.81466
```

Values for μ that satisfy the equation are $|\mu| > 20.81$, and since it is known that $\mu \geq 31$ inches according to the problem, $\hat{\mu}_2$ is both more efficient and has a smaller *MSE* than does $\hat{\mu}_1$. See Figure 7.3 on the next page for a graphical representation of the distributions of $\hat{\mu}_1$ and $\hat{\mu}_2$. ■

Cramér-Rao Lower Bound It can be shown that if $T = \hat{\theta}$ is an unbiased estimator of θ and a random sample of size n , X_1, X_2, \dots, X_n , has pdf $f(x|\theta)$, then the variance of the unbiased estimator, $\hat{\theta}$, must satisfy the inequality

$$\text{Var}[\hat{\theta}] \geq \frac{1}{n \cdot E \left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta} \right)^2 \right]}, \quad (7.6)$$

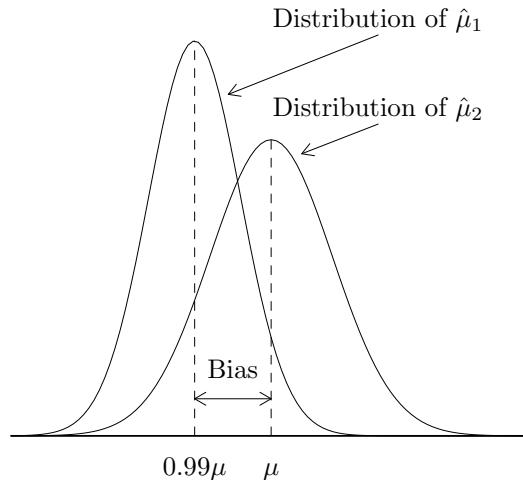


FIGURE 7.3: Graphical representations for the sampling distributions of $\hat{\mu}_1$ and $\hat{\mu}_2$

where $f(X|\theta)$ is the density function of the distribution of interest evaluated at the random variable X (Casella and Berger, 2002, p. 335). In the discrete case, $p(X|\theta)$ is used instead of $f(X|\theta)$. In general, the probability distributions of both discrete and continuous distributions are referred to using the notation $f(x)$. The inequality in (7.6) is known as the **Cramér-Rao inequality**, and the quantity on the right-hand side of the equation is known as the Cramér-Rao lower bound (CRLB).

DEFINITION 7.1: If $\hat{\theta}$ is an unbiased estimator of θ and

$$\text{Var}[\hat{\theta}] = \frac{1}{n \cdot E \left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta} \right)^2 \right]} \quad (7.7)$$

then $\hat{\theta}$ is a **minimum variance unbiased** estimator of θ .

Not all parameters have unbiased estimators whose variance equals the CRLB; however, when the variance of an unbiased estimator equals the CRLB, the estimator is **efficient** or **minimum variance**. The quantity in the denominator of (7.7) is known as the **Fisher information** about θ that is supplied by the sample. That is, the smaller the variance of the estimator, the greater the information.

Example 7.6 Show that \bar{X} is a minimum variance unbiased estimator of the mean λ of a Poisson population.

Solution: If $X \sim \text{Pois}(\lambda)$, then, according to (4.5), $E[X] = \lambda$, $\text{Var}[X] = \lambda$, and the **pdf** of X is

$$\mathbb{P}(X = x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}. \quad (7.8)$$

Since $E[\bar{X}] = \sum_{i=1}^n \frac{E[X_i]}{n} = \frac{n\lambda}{n} = \lambda$, it follows that \bar{X} is an unbiased estimator of λ , with variance $\frac{\lambda}{n}$ because the $\text{Var}[\bar{X}] = \text{Var}\left[\sum_{i=1}^n \frac{X_i}{n}\right] = \frac{1}{n^2} \sum_{i=1}^n \text{Var}[X_i] = \frac{n\lambda}{n^2} = \frac{\lambda}{n}$. Consequently, if the CRLB equals $\frac{\lambda}{n}$, \bar{X} is a minimum variance unbiased estimator of λ according to Definition 7.1. By taking the natural logarithm of (7.8),

$$\ln \mathbb{P}(x|\lambda) = x \ln(\lambda) - \lambda - \ln(x!). \quad (7.9)$$

Taking the derivative of (7.9) with respect to λ gives

$$\frac{\partial \ln \mathbb{P}(x|\lambda)}{\partial \lambda} = \frac{x}{\lambda} - 1 = \frac{x - \lambda}{\lambda}.$$

Hence

$$E \left[\left(\frac{\partial \ln \mathbb{P}(X|\lambda)}{\partial \lambda} \right)^2 \right] = E \left[\left(\frac{X - \lambda}{\lambda} \right)^2 \right] = \frac{E[(X - \lambda)^2]}{\lambda^2} = \frac{Var[X]}{\lambda^2}.$$

Therefore,

$$E \left[\left(\frac{\partial \ln \mathbb{P}(X|\lambda)}{\partial \lambda} \right)^2 \right] = \frac{Var[X]}{\lambda^2} = \frac{\lambda}{\lambda^2} = \frac{1}{\lambda},$$

and the CRLB is

$$\frac{1}{n \cdot E \left[\left(\frac{\partial \ln f(X|\lambda)}{\partial \lambda} \right)^2 \right]} = \frac{\lambda}{n}.$$

Consequently, since \bar{X} is unbiased and $Var[\bar{X}] = \frac{\lambda}{n}$, it follows that \bar{X} is a minimum variance unbiased estimator of λ . ■

Example 7.7 Show that \bar{X} is a minimum variance unbiased estimator of the mean θ of an exponential population.

Solution: If $X \sim Exp(\frac{1}{\theta})$, then, according to (4.12), when using the substitution $\theta = \frac{1}{\lambda}$, $E[X] = \theta$, $Var[X] = \theta^2$, and the pdf of X is

$$f(x) = \begin{cases} \frac{1}{\theta} e^{-x/\theta} & \text{if } x \geq 0 \text{ and} \\ 0 & \text{if } x < 0. \end{cases} \quad (7.10)$$

Since $E[\bar{X}] = \sum_{i=1}^n \frac{E[X_i]}{n} = \frac{n\theta}{n} = \theta$, it follows that \bar{X} is an unbiased estimator of θ , with variance $\frac{\theta^2}{n}$ since $Var[\bar{X}] = Var[\sum_{i=1}^n \frac{X_i}{n}] = \frac{1}{n^2} \sum_{i=1}^n Var[X_i] = \frac{n\theta^2}{n^2} = \frac{\theta^2}{n}$. Consequently, if the CRLB equals $\frac{\theta^2}{n}$, \bar{X} is a minimum variance unbiased estimator of θ according to Definition 7.1 on the preceding page. By taking the natural logarithm of (7.10),

$$\ln f(x|\theta) = -\ln(\theta) - \frac{x}{\theta}. \quad (7.11)$$

Taking the derivative of (7.11) with respect to θ gives

$$\frac{\partial \ln f(x|\theta)}{\partial \theta} = -\frac{1}{\theta} + \frac{x}{\theta^2} = \frac{x - \theta}{\theta^2}.$$

Hence

$$E \left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta} \right)^2 \right] = E \left[\left(\frac{X - \theta}{\theta^2} \right)^2 \right] = \frac{E[(X - \theta)^2]}{\theta^4} = \frac{Var[X]}{\theta^4}.$$

Therefore,

$$E \left[\left(\frac{\partial \ln f(X|\lambda)}{\partial \theta} \right)^2 \right] = \frac{Var[X]}{\theta^4} = \frac{\theta^2}{\theta^4} = \frac{1}{\theta^2},$$

and the CRLB is

$$\frac{1}{n \cdot E \left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta} \right)^2 \right]} = \frac{\theta^2}{n}.$$

Consequently, since \bar{X} is unbiased and $Var[\bar{X}] = \frac{\theta^2}{n}$, it follows that \bar{X} is a minimum variance unbiased estimator of θ . ■

7.2.4 Consistent Estimators

The next property of estimators that is considered is **consistency**. Consistency is a property of a sequence of estimators rather than a single estimator; however, it is rather common to refer to an estimator as being consistent. A sequence of estimators means that the same estimation procedure is carried out for each sample of size n . If T is an estimator of θ and X_1, X_2, \dots are observed according to a distribution $f(x|\theta)$, a sequence of estimators T_1, T_2, \dots, T_n can be constructed by performing the same estimation procedure for samples of sizes $1, 2, \dots, n$, respectively. In other words, the sequence is

$$T_1 = t(X_1), T_2 = t(X_1, X_2), \dots, T_n = t(X_1, X_2, \dots, X_n).$$

A sequence of estimators T_n (defined for all n) is a **consistent** estimator of the parameter θ for every $\theta \in \Theta$ if

$$\lim_{n \rightarrow \infty} \mathbb{P}(|T_n - \theta| \geq \epsilon) = 0, \text{ for all } \epsilon > 0. \quad (7.12)$$

An equivalent statement of (7.12) is that a sequence of estimators T_n (defined for all n) is a **consistent** estimator of the parameter θ for every $\theta \in \Theta$ if

$$\lim_{n \rightarrow \infty} \mathbb{P}(|T_n - \theta| < \epsilon) = 1, \text{ for all } \epsilon > 0. \quad (7.13)$$

Both definitions (7.12) and (7.13) state that a consistent sequence of estimators **converges in probability** to the parameter θ , where θ is the parameter the consistent sequence of estimators is estimating. In practical terms, this implies that the variance of a consistent estimator decreases as n increases and that the expected value of T_n tends to θ as n increases. Further, given a consistent sequence of estimators, say T_n , Chebyshev's inequality (3.20) guarantees that

$$\mathbb{P}(|T_n - \theta| \geq \epsilon) = \mathbb{P}(|T_n - \theta|^2 \geq \epsilon^2) \leq \frac{E[(T_n - \theta)^2]}{\epsilon^2},$$

for every $\theta \in \Theta$. Since $E[(T_n - \theta)^2]$ can be expressed as

$$E[(T_n - \theta)^2] = \text{Var}[T_n] + (\text{Bias}[T_n])^2,$$

if

$$\lim_{n \rightarrow \infty} \text{Var}[T_n] = 0 \quad \text{and} \quad \lim_{n \rightarrow \infty} (\text{Bias}[T_n])^2 = 0, \quad (7.14)$$

then T_n is a consistent sequence of estimators of θ . Whenever the conditions in (7.14) are true, T_n converges in *MSE* to the true value of θ . The conditions in (7.14) are sufficient but not necessary conditions for a sequence of estimators to be consistent.

Example 7.8 Let $\{X_1, X_2, \dots, X_n\}$ be a random sample of size n from a distribution with mean μ and variance σ^2 . Show that \bar{X}_n is a consistent estimator of μ .

Solution: For \bar{X}_n to be a consistent estimator of μ , it must be shown that

$$\lim_{n \rightarrow \infty} \mathbb{P}(|\bar{X}_n - \mu| \geq \epsilon) = 0 \text{ for all } \epsilon > 0.$$

Using version (c) on page 232 of Chebyshev's inequality and the fact that $E[\bar{X}_n] = \mu$ and $\text{Var}[\bar{X}_n] = \sigma^2/n$,

$$\mathbb{P}(|\bar{X}_n - \mu| \geq k\sigma/\sqrt{n}) \leq \frac{1}{k^2}.$$

By setting $\epsilon = k\sigma/\sqrt{n}$, $k = \sqrt{n}\epsilon/\sigma$, so that

$$\frac{1}{k^2} = \frac{\sigma^2}{n\epsilon^2},$$

from which it follows that

$$\mathbb{P}(|\bar{X}_n - \mu| \geq \epsilon) \leq \frac{\sigma^2}{n\epsilon^2}. \quad (7.15)$$

Given that $\sigma^2 < \infty$ (finite), taking the limit as $n \rightarrow \infty$ on both sides of the \leq sign of (7.15) gives

$$\lim_{n \rightarrow \infty} \mathbb{P}(|\bar{X}_n - \mu| \geq \epsilon) = 0 \text{ for all } \epsilon.$$

Consequently, \bar{X}_n is a consistent estimator of μ . This is essentially the **weak law of large numbers** given in (3.21) of Section 3.4.4. ■

7.2.5 Robust Estimators

The idea of statistical **robustness** has received considerable attention in recent years, but there is not a consensus on what defines a robust estimator. The essence of a **robust** estimator is an estimator whose sampling distribution is not seriously affected by violations of underlying assumptions. For example, when estimating the average useful life of an electronic component, one may think that an exponential distribution is being sampled when in fact a gamma or Weibull distribution is being sampled. If the estimation of the unknown parameter is not seriously affected by the fact that an incorrect distribution is being assumed, the estimator is robust. The concept of robustness has also been used to refer to the ability of a particular estimator to provide reasonable estimates when atypical observations are encountered in the sample. For example, if the largest value in a sample is made 1000 times larger, the sample median remains the same in both the sample with the original value and in the sample where the value is 1000 times larger than the largest value in the original sample. In this sense, the median is a robust estimator.

In particular, the median provides a robust measure of center whenever the underlying distribution is skewed. In a similar fashion, a robust measure of variability is the **median absolute deviation (MAD)**. The *MAD* is defined as

$$MAD = \text{median}|x_i - \text{sample median}|. \quad (7.16)$$

When working with normal distributions, a robust estimator of σ is *MAD1*, where $MAD1 = \frac{1}{0.6745} MAD$. The value 0.6745 corresponds to the 75th percentile of a $N(0, 1)$ distribution ($z_{0.75} = 0.6745$). When working with R, the default value returned when working with the function `mad()` corresponds to the definition of *MAD1*. To compute the *MAD* as defined in (7.16), use the argument `constant = 1` inside the `mad()` function.

Example 7.9 A botanist interested in studying the effects of a new herbicide on *trifolium repens* (white clover) measures and records the stem lengths in centimeters of ten specimens as 5.3, 2.8, 3.4, 7.2, 8.3, 1.7, 6.2, 9.3, 3.2, and 5.9. Compute the mean, median, standard deviation, and *MAD*. Suppose the botanist makes a field error and records an 83 instead of an 8.3. What effect will the recording error have on the computed quantities?

Solution: The stem measurements are entered without the recording error in the vector `stem1` (in increasing order) and the stem measurements with the recording error in the vector `stem2`. That is, `stem2` has an 83 rather than an 8.3.

```

> stem1 <- c(1.7, 2.8, 3.2, 3.4, 5.3, 5.9, 6.2, 7.2, 8.3, 9.3)
> stem2 <- c(1.7, 2.8, 3.2, 3.4, 5.3, 5.9, 6.2, 7.2, 83, 9.3)
> c(mean(stem1), sqrt(var(stem1)))

[1] 5.330000 2.516634

> c(mean(stem2), sqrt(var(stem2)))

[1] 12.80000 24.77185

> c(median(stem1), mad(stem1, constant = 1))

[1] 5.6 2.3

> c(median(stem2), mad(stem2, constant = 1))

[1] 5.6 2.3

> median(abs(stem1 - median(stem1)))

[1] 2.3

> median(abs(stem2 - median(stem2)))

[1] 2.3

```

Note that the mean and standard deviation of `stem1` (5.33, 2.5166) are dramatically different from the mean and standard deviation of `stem2` (5.33, 24.7718); however, the median and *MAD* (5.6, 2.3) are the same for the values in both `stem1` and `stem2`. What has been demonstrated is the robustness of the median and the *MAD* to outliers. ■

7.3 Point Estimation Techniques

Section 7.2 discussed several ways to measure the “goodness” of an estimator. In what follows, the framework for deriving estimators is provided. In general, these topics are intertwined. Specifically, two methods are considered: the method of moments and the method of maximum likelihood. Before proceeding further, some notation is emphasized. Recall that capital letters are used to denote random variables. In particular, the information in a random sample X_1, X_2, \dots, X_n is used to make inferences about the unknown θ . The observed values of the random sample are denoted x_1, x_2, \dots, x_n . Further, a random sample X_1, X_2, \dots, X_n is referred to with the boldface \mathbf{X} and the observed values in a random sample x_1, x_2, \dots, x_n with the boldface \mathbf{x} . The joint **pdf** of X_1, X_2, \dots, X_n is given by

$$\begin{aligned}
f(\mathbf{x}|\theta) &= f(x_1, x_2, \dots, x_n|\theta) \\
&= f(x_1|\theta) \times f(x_2|\theta) \times \cdots \times f(x_n|\theta) = \prod_{i=1}^n f(x_i|\theta).
\end{aligned} \tag{7.17}$$

7.3.1 Method of Moments Estimators

The idea behind the **method of moments** is to equate population moments about the origin to their corresponding sample moments, where the r^{th} **sample moment about the origin**, denoted m_r , is defined as

$$m_r = \frac{1}{n} \sum_{i=1}^n X_i^r, \quad (7.18)$$

and subsequently to solve for estimators of the unknown parameters. Recall that the r^{th} population moment about the origin of a random variable X , denoted α_r , was defined in (3.8) as $E[X^r]$. It follows that $\alpha_r = E[X^r] = \sum_{i=1}^{\infty} x_i^r \mathbb{P}(X = x_i)$ for discrete X , and that $\alpha_r = E[X^r] = \int_{-\infty}^{\infty} x^r f(x) dx$ for continuous X . Specifically, given a random sample X_1, X_2, \dots, X_n from a population with pdf $f(x|\theta_1, \theta_2, \dots, \theta_k)$, the method of moments estimators, denoted $\hat{\theta}_i$ for $i = 1, \dots, k$, are found by equating the first k population moments about the origin to their corresponding sample moments and solving the resulting system of simultaneous equations:

$$\left\{ \begin{array}{l} \alpha_1(\theta_1, \dots, \theta_k) = m_1 \\ \alpha_2(\theta_1, \dots, \theta_k) = m_2 \\ \vdots \qquad \vdots \\ \alpha_k(\theta_1, \dots, \theta_k) = m_k \end{array} \right. \quad (7.19)$$

The method of moments is an appealing technique for deriving estimators due to its simplicity and to the fact that method of moments estimators are consistent. In fact, the theoretical justification for equating the sample moments to the population moments is that, under certain conditions, it can be shown that the sample moments converge in probability to the population moments and that the sample moments about the origin are unbiased estimators of their corresponding population moments.

Example 7.10 Given a random sample of size n from a $\text{Bin}(1, \pi)$ population, find the method of moments estimator of π .

Solution: The first sample moment m_1 is \bar{X} , and the first population moment about zero for the binomial random variable is $\alpha_1 = E[X^1] = 1 \cdot \pi$. By equating the first population moment to the first sample moment,

$$\alpha_1(\pi) = \pi \stackrel{\text{set}}{=} \bar{X} = m_1,$$

which implies that the method of moments estimator for π is $\tilde{\pi} = \bar{X}$. ■

Example 7.11 Given a random sample of size m from a $\text{Bin}(n, \pi)$ population, find the method of moments estimator of π .

Solution: The first sample moment m_1 is \bar{X} , and the first population moment about zero for the binomial random variable is $\alpha_1 = E[X^1] = n \cdot \pi$. By equating the first population moment to the first sample moment,

$$\alpha_1(\pi) = n\pi \stackrel{\text{set}}{=} \bar{X} = m_1,$$

which implies that the method of moments estimator for π is $\tilde{\pi} = \frac{\bar{X}}{n}$. ■

Example 7.12 Given a random sample of size n from a $Pois(\lambda)$ population, find the method of moments estimator of λ .

Solution: The first sample moment m_1 is \bar{X} , and the first population moment about zero for a Poisson random variable is $\alpha_1 = E[X^1] = \lambda$. By equating the first population moment to the first sample moment,

$$\alpha_1(\pi) = \lambda \stackrel{\text{set}}{=} \bar{X} = m_1,$$

which implies that the method of moments estimator for λ is $\tilde{\lambda} = \bar{X}$. ■

Example 7.13 Given a random sample of size n from a $N(\mu, \sigma^2)$ population, find the method of moments estimators of μ and σ^2 .

Solution: The first and second sample moments m_1 and m_2 are \bar{X} and $\frac{1}{n} \sum_{i=1}^n X_i^2$, respectively. The first and second population moments about zero for a normal random variable are $\alpha_1 = E[X^1] = \mu$ and $\alpha_2 = E[X^2] = \sigma^2 + \mu^2$. By equating the first two population moments to the first two sample moments,

$$\begin{cases} \alpha_1(\mu, \sigma^2) = \mu \stackrel{\text{set}}{=} \bar{X} = m_1 \\ \alpha_2(\mu, \sigma^2) = \sigma^2 + \mu^2 \stackrel{\text{set}}{=} \frac{1}{n} \sum_{i=1}^n X_i^2 = m_2. \end{cases} \quad (7.20)$$

Solving the system of equations in (7.20) yields $\tilde{\mu} = \bar{X}$ and $\tilde{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n X_i^2 - \bar{X}^2 = S_u^2$ as the method of moments estimators for μ and σ^2 , respectively. ■

Example 7.14 Given a random sample of size n from a $Gamma(\alpha, \lambda)$ population, find the method of moments estimators of α and λ .

Solution: According to (4.16), $E[X] = \frac{\alpha}{\lambda}$, and $Var[X] = \frac{\alpha}{\lambda^2}$ for a random variable X that follows a gamma distribution. The first and second sample moments m_1 and m_2 are \bar{X} and $\frac{1}{n} \sum_{i=1}^n X_i^2$, respectively. The first and second population moments for a gamma random variable are

$$\alpha_1 = E[X^1] = \frac{\alpha}{\lambda},$$

and

$$\alpha_2 = E[X^2] = \sigma^2 + E[X]^2 = \frac{\alpha}{\lambda^2} + \frac{\alpha^2}{\lambda^2} = \frac{\alpha(1 + \alpha)}{\lambda^2},$$

respectively. By equating the first two population moments to the first two sample moments,

$$\begin{cases} \alpha_1(\alpha, \lambda) = \frac{\alpha}{\lambda} \stackrel{\text{set}}{=} \bar{X} = m_1 \\ \alpha_2(\alpha, \lambda) = \frac{\alpha(1 + \alpha)}{\lambda^2} \stackrel{\text{set}}{=} \frac{1}{n} \sum_{i=1}^n X_i^2 = m_2. \end{cases} \quad (7.21)$$

When it is recalled that $S_u^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}$, the system of equations in (7.21) can be solved to obtain $\tilde{\alpha} = \frac{\bar{X}^2}{S_u^2}$ and $\tilde{\lambda} = \frac{\bar{X}}{S_u^2}$ as the method of moments estimators for α and λ , respectively. ■

7.3.2 Likelihood and Maximum Likelihood Estimators

When sampling from a population described by a **pdf** $f(x|\theta)$, knowledge of θ provides knowledge of the entire population. The idea behind maximum likelihood is to select the value for θ that makes the observed data most likely under the assumed probability model. When x_1, x_2, \dots, x_n are the observed values of a random variable X from a population with parameter θ , the notation $L(\theta|\mathbf{x}) = f(\mathbf{x}|\theta)$ will be used to indicate that the distribution depends on the parameter θ , and \mathbf{x} to indicate the distribution is dependent on the observed values from the sample. Once the sample values are observed, $L(\theta|\mathbf{x})$ can still be evaluated in a formal sense, although it no longer has a probability interpretation (in the discrete case) as does (7.17). $L(\theta|\mathbf{x})$ is the **likelihood function** of θ for \mathbf{x} and is denoted by

$$L(\theta|\mathbf{x}) = f(\mathbf{x}|\theta) = \prod_{i=1}^n f(x_i|\theta) = f(x_1|\theta) \times f(x_2|\theta) \times \cdots \times f(x_n|\theta). \quad (7.22)$$

The key difference between (7.17) and (7.22) is that the joint **pdf** given in (7.17) is a function of \mathbf{x} for a given θ and the likelihood function given in (7.22) is a function of θ for given \mathbf{x} . The value of θ that maximizes $L(\theta|\mathbf{x})$ is called the **maximum likelihood estimate** (mle) of θ .

Example 7.15  A box contains five pieces of candy. Some of the candies are alcoholic, and some are not. In an attempt to estimate the proportion of alcoholic candies, a sample of size $n = 3$ is taken with replacement that results in (a, a, n) (two alcoholic candies and one non-alcoholic candy). Write out the maximum likelihood function and use it to select the maximum likelihood estimate of π , the true proportion of alcoholic candies.

Solution: The possible values for π are $\frac{0}{5}, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}$, and $\frac{5}{5}$. Since there is at least one alcoholic candy and there is at least one non-alcoholic candy, the values $\pi = 0$ and $\pi = 1$ must be ruled out. In this case, the observed sample values are $\mathbf{x}=(a, a, n)$. The likelihood function is

$$\begin{aligned} L(\pi|\mathbf{x}) &= f(\mathbf{x}|\pi) \\ &= f(a|\pi) \times f(a|\pi) \times f(n|\pi). \end{aligned}$$

Box	π	$L(\pi a, a, n)$
aaaan	$\frac{4}{5}$	$\frac{4}{5} \cdot \frac{4}{5} \cdot \frac{1}{5} = \frac{16}{125}$
aaann	$\frac{3}{5}$	$\frac{3}{5} \cdot \frac{3}{5} \cdot \frac{2}{5} = \frac{18}{125}$
aannn	$\frac{2}{5}$	$\frac{2}{5} \cdot \frac{2}{5} \cdot \frac{3}{5} = \frac{12}{125}$
annnn	$\frac{1}{5}$	$\frac{1}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} = \frac{4}{125}$

Since the value $\pi = \frac{3}{5}$ maximizes the likelihood function when $\mathbf{x}=(a, a, n)$, $\hat{\pi}(\mathbf{x}) = \frac{3}{5}$ is the maximum likelihood estimate for the proportion of candies that are alcoholic. 

Another way to think of the mle is the mode of the likelihood function. The maximum likelihood estimate is denoted as $\hat{\theta}(\mathbf{x})$, and the maximum likelihood estimator (MLE), a statistic, as $\hat{\theta}(\mathbf{X})$. In general, the likelihood function may be difficult to manipulate, and it is usually more convenient to work with the natural logarithm of $L(\theta|\mathbf{x})$, called the **log-likelihood function**, since it converts products into sums. Finding the value θ that

maximizes the log-likelihood function ($\ln L(\theta|\mathbf{x})$) is equivalent to finding the value of θ that maximizes $L(\theta|\mathbf{x})$ since the natural logarithm is a monotonically increasing function. If $L(\theta|\mathbf{x})$ is differentiable with respect to θ , a possible mle is the solution to

$$\frac{\partial(\ln L(\theta|\mathbf{x}))}{\partial\theta} = 0. \quad (7.23)$$

Note that a possible mle is the solution to (7.23). A possible solution is used since a solution to (7.23) is a necessary but not sufficient condition for the solution to be a maximum, since the solution to (7.23) could be a local or global minimum, a local or global maximum, or a point of inflection. Recall that stationary points where

$$\frac{\partial^2(\ln L(\theta|\mathbf{x}))}{\partial\theta^2} \Big|_{\theta=\hat{\theta}(\mathbf{x})} < 0 \quad (7.24)$$

indicate some type of maximum, either local or global. Further, the solution to (7.23) does not include points on the boundaries of the parameter space. Consequently, when evaluating the maximum of $L(\theta|\mathbf{x})$, the boundaries of the parameter space Θ as well as solutions to (7.23) must be evaluated.

Example 7.16 \triangleright **General MLE** \triangleleft The random variable X can take on the values 0, 1, 2, and 3 with probabilities $\mathbb{P}(X = 0) = p^3$, $\mathbb{P}(X = 1) = (1 - p)p^2$, $\mathbb{P}(X = 2) = (1 - p)^2$, and $\mathbb{P}(X = 3) = 2p(1 - p)$, where $0 < p < 1$.

- (a) Do the given probabilities for the random variable X satisfy the conditions for a probability distribution of X ?
- (b) Find the maximum likelihood estimate for p if a random sample of size $n = 150$ resulted in a 0 twenty-four times, a 1 fifty-four times, a 2 thirty-two times, and a 3 forty times.
- (c) Graph the log-likelihood function and determine its maximum using the function `nlm()`.

Solution: The answers are as follows:

- (a) For the distribution of X to be a valid **pdf**, it must satisfy the following two conditions:
 - (1) $p(x) \geq 0$ for all x .
 - (2) $\sum_x p(x) = 1$.

Condition (1) is satisfied since $0 < p < 1$. Condition (2) is also satisfied since

$$\begin{aligned} \sum_x p(x) &= p^3 + (1 - p)p^2 + (1 - p)^2 + 2p(1 - p) \\ &= p^3 + p^2 - p^3 + 1 + p^2 - 2p + 2p - 2p^2 = 1. \end{aligned}$$

- (b) The likelihood function is

$$\begin{aligned} L(p|\mathbf{x}) &= [(p^3)]^{24} [(1 - p)p^2]^{54} [(1 - p)^2]^{32} [2p(1 - p)]^{40} \\ &= 2^{40} p^{220} (1 - p)^{158}, \end{aligned}$$

and the log-likelihood function is

$$\ln [L(p|\mathbf{x})] = 40 \ln 2 + 220 \ln p + 158 \ln(1 - p). \quad (7.25)$$

Next, look for the value that maximizes the log-likelihood function by taking the first-order derivative of (7.25) with respect to p and setting the answer equal to zero:

$$\frac{\partial \ln [L(p|\mathbf{x})]}{\partial p} = \frac{220}{p} - \frac{158}{1-p} \stackrel{\text{set}}{=} 0. \quad (7.26)$$

The solution to (7.26) is $p = 0.58$. In order for $p = 0.58$ to be a maximum, the second-order derivative of (7.25) with respect to p must be negative. Since

$$\frac{\partial^2 \ln [L(\pi|\mathbf{x})]}{\partial p^2} = -\frac{220}{p^2} - \frac{158}{(1-p)^2} < 0 \text{ for all } p,$$

this value is a global maximum. Therefore, the maximum likelihood estimate of p , $\hat{p}(\mathbf{x}) = 0.58$.

(c) R Code 7.2 can be used to create a graph of the log-likelihood function similar to the one depicted in Figure 7.4.

R Code 7.2

```
> loglike <- function(p){40*log(2) + 220*log(p) + 158*log(1 - p)}
> negloglike <- function(p){(-1)*(40*log(2) + 220*log(p) + 158*log(1 - p))}
> p1 <- ggplot(data = data.frame(x = c(0, 1)), aes(x = x))
> p1 + stat_function(fun = loglike, n = 200) +
+   labs(x = "p",
+         y = expression(textstyle(ln ~ ~ L(p*| *bold(x)))) ) +
+   geom_vline(xintercept = 0.58, lty = "dashed")
```

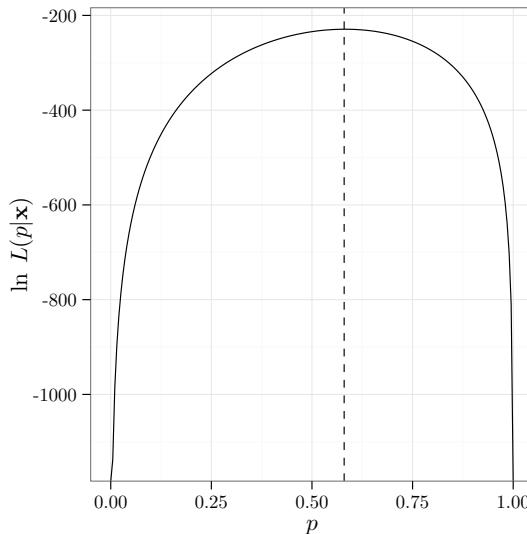


FIGURE 7.4: Illustration of the $\ln L(p|\mathbf{x})$ function for Example 7.16

R Code 7.3 on the following page computes the maximum of the log-likelihood function using both `nlm()` and `optimize()`. The minimization function, `nlm()`, has two required arguments, `f=` and `p=`, where the value passed to the argument `p=` are starting parameter

values for the minimization. Since `nlm()` searches for a minimum of a function, and finding a maximum likelihood estimate requires finding a maximum, `nlm()` will be used on the negative of the log-likelihood function.

R Code 7.3

```
> nlm(f = negloglike, p = 0.01)$estimate
[1] 0.5820101
> optimize(f = loglike, interval = c(0, 1), maximum = TRUE)$maximum
[1] 0.5820066
```



Example 7.17 Given a random sample of size n taken from a $Bernoulli(\pi)$ distribution, compute the maximum likelihood estimate and maximum likelihood estimator of the parameter π .

Solution: According to (4.2), the `pdf` for $X \sim Bernoulli(\pi)$ is

$$P(X = x|\pi) = \pi^x(1 - \pi)^{1-x},$$

where x takes on the value 1 with probability π and 0 with probability $1 - \pi$. The likelihood function for the n observed values is

$$L(\pi|\mathbf{x}) = \prod_{i=1}^n \pi^{x_i}(1 - \pi)^{1-x_i}.$$

Taking the natural logarithm of the likelihood function gives

$$\begin{aligned} \ln L(\pi|\mathbf{x}) &= \ln \left[\prod_{i=1}^n \pi^{x_i}(1 - \pi)^{1-x_i} \right] = \sum_{i=1}^n \ln [\pi^{x_i}(1 - \pi)^{1-x_i}] \\ &= \sum_{i=1}^n [x_i \ln \pi + (1 - x_i) \ln(1 - \pi)]. \end{aligned} \tag{7.27}$$

To find the value that maximizes (7.27), take the first-order derivative of $\ln L(\pi|\mathbf{x})$ with respect to π and set the answer equal to zero:

$$\frac{\partial \ln L(\pi|\mathbf{x})}{\partial \pi} = \frac{\sum_{i=1}^n x_i}{\pi} - \frac{n - \sum_{i=1}^n x_i}{1 - \pi} \stackrel{\text{set}}{=} 0. \tag{7.28}$$

The solution to (7.28) is $\pi = \frac{\sum_{i=1}^n x_i}{n} = \bar{x}$. For $\pi = \bar{x}$ to be a maximum, the second-order derivative of the log-likelihood function must be negative at $\pi = \bar{x}$. The second-order derivative is

$$\frac{\partial^2 \ln L(\pi|\mathbf{x})}{\partial \pi^2} = \frac{-\sum_{i=1}^n x_i}{\pi^2} - \frac{n - \sum_{i=1}^n x_i}{(1 - \pi)^2}.$$

Evaluating the second-order derivative at $\pi = \bar{x}$ yields

$$\frac{\partial^2 \ln L(\pi|\mathbf{x})}{\partial \pi^2} = \frac{-n\bar{x}}{\bar{x}^2} - \frac{(n - n\bar{x})}{(1 - \bar{x})^2} = -\frac{n}{\bar{x}} - \frac{n}{1 - \bar{x}} = \frac{-n}{\bar{x}(1 - \bar{x})},$$

which is less than zero since $0 \leq \bar{x} \leq 1$ and $n > 0$. Finally, since the values of the likelihood function at the boundaries of the parameter space, $\pi = 0$ and $\pi = 1$, are 0, it follows that $\pi = \bar{x}$ is the value that maximizes the likelihood function. The maximum likelihood estimate $\hat{\pi}(\mathbf{x}) = \bar{x}$ and the maximum likelihood estimator $\hat{\pi}(\mathbf{X}) = \bar{X}$. ■

Example 7.18 ▷ MLEs with R: Oriental Cockroaches ◁ A laboratory is interested in testing a new child-friendly pesticide on *Blatta orientalis* (oriental cockroaches). The scientists from the lab apply the new pesticide to 81 randomly selected *Blatta orientalis oothecae* (eggs). The results from the experiment are stored in the data frame ROACHEGGS in the variable eggs. A zero in the variable eggs indicates that nothing hatched from the egg while a 1 indicates the birth of a cockroach. Assuming the selected *Blatta orientalis* eggs are representative of the population of *Blatta orientalis* eggs, estimate the proportion of *Blatta orientalis* eggs that result in a birth after being sprayed with the child-friendly pesticide. Use the nlm() function to solve the problem iteratively. Produce a graph of the log-likelihood function and indicate with a dashed line where it achieves its maximum value.

Solution: Note that whether or not a *Blatta orientalis* egg hatches is a Bernoulli trial with unknown parameter π . Using the maximum likelihood estimate from Example 7.17 on the preceding page, $\hat{\pi}(\mathbf{x}) = \bar{x} = 0.2099$.

```
> eggs <- ROACHEGGS$eggs
> mean(eggs)

[1] 0.2098765

> loglike <- function(PI){(sum(eggs)*log(PI)+ sum(1 - eggs)*log(1 - PI))}
> negloglike <- function(PI){(-1)*(sum(eggs)*log(PI) +
+                               sum(1 - eggs)*log(1 - PI))}

> nlm(f = negloglike, p = 0.2)

$minimum
[1] 41.61724

$estimate
[1] 0.209876

$gradient
[1] 1.421085e-08

$code
[1] 1

$iterations
[1] 4
```

R Code 7.4 can be used to represent graphically the log-likelihood function in a fashion similar to Figure 7.5 on the next page.

R Code 7.4

```
> p <- ggplot(data.frame(x = c(0, 1)), aes(x = x))
> p + stat_function(fun = loglike, n = 200) +
+   labs(x = expression(pi),
```

```
+     y = expression(textstyle(ln)~~L(pi*"|"~*bold(x))) ) +
+ geom_vline(xintercept = mean(eggs), lty = "dashed")
```

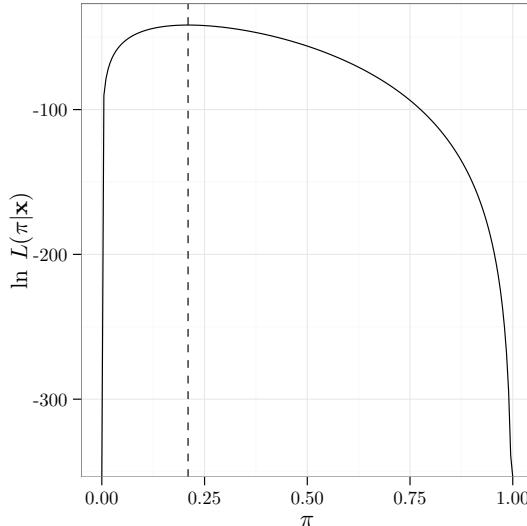


FIGURE 7.5: Illustration of the $\ln L(\pi|x)$ function for Example 7.18



The function `optimize()` approximates a local optimum of a continuous univariate function within a given interval. The function searches the user-provided interval for either a minimum (default) or maximum of the function specified in the `f=` argument. R Code 7.5 solves Example 7.18 with the function `optimize()`.

R Code 7.5

```
> optimize(f = loglike, interval = c(0, 1), maximum = TRUE)
$maximum
[1] 0.2098906

$objective
[1] -41.61724
```

Example 7.19 Let X_1, X_2, \dots, X_m be a random sample from a $Bin(n, \pi)$ population. Compute the maximum likelihood estimator and the maximum likelihood estimate for the parameter π . Verify your answer with simulation by generating 1000 random values from a $Bin(n = 3, \pi = 0.5)$ population.

Solution: The likelihood function is

$$\begin{aligned} L(\pi|\mathbf{x}) &= \prod_{i=1}^m \binom{n}{x_i} \pi^{x_i} (1-\pi)^{n-x_i} \\ &= \binom{n}{x_1} \pi^{x_1} (1-\pi)^{n-x_1} \times \cdots \times \binom{n}{x_m} \pi^{x_m} (1-\pi)^{n-x_m}, \end{aligned} \quad (7.29)$$

and the log-likelihood function is

$$\begin{aligned} \ln L(\pi|\mathbf{x}) &= \ln \left[\prod_{i=1}^m \binom{n}{x_i} \pi^{x_i} (1-\pi)^{n-x_i} \right] = \sum_{i=1}^m \ln \left[\binom{n}{x_i} \pi^{x_i} (1-\pi)^{n-x_i} \right] \\ &= \sum_{i=1}^m \left[\ln \binom{n}{x_i} + x_i \ln \pi + (n-x_i) \ln(1-\pi) \right]. \end{aligned} \quad (7.30)$$

Next, look for the value that maximizes the log-likelihood function by taking the first-order derivative with respect to π of (7.30) and setting the answer to zero:

$$\frac{\partial \ln L(\pi|\mathbf{x})}{\partial \pi} = \frac{\sum_{i=1}^m x_i}{\pi} - \frac{mn - \sum_{i=1}^m x_i}{1-\pi} \stackrel{\text{set}}{=} 0. \quad (7.31)$$

The solution to (7.31) is $\pi = \frac{\sum_{i=1}^m x_i}{mn} = \frac{\bar{x}}{n}$. For $\pi = \frac{\bar{x}}{n}$ to be a maximum, the second-order derivative of the log-likelihood function must be negative at $\pi = \frac{\bar{x}}{n}$. The second-order derivative is

$$\frac{\partial^2 \ln L(\pi|\mathbf{x})}{\partial \pi^2} = \frac{-\sum_{i=1}^m x_i}{\pi^2} - \frac{mn - \sum_{i=1}^m x_i}{(1-\pi)^2}.$$

Evaluating the second-order derivative at $\pi = \frac{\bar{x}}{n}$ and using the substitution $\sum_{i=1}^m x_i = m\bar{x}$ yields

$$\begin{aligned} \frac{\partial^2 \ln L(\pi|\mathbf{x})}{\partial \pi^2} &= -\frac{m\bar{x}}{\left(\frac{\bar{x}}{n}\right)^2} - \frac{mn - m\bar{x}}{(1-\frac{\bar{x}}{n})} \\ &= -\frac{mn^2}{\bar{x}} - \frac{m(n-\bar{x})}{\frac{(n-\bar{x})^2}{n^2}} = -\frac{mn^2}{\bar{x}} - \frac{mn^2}{n-\bar{x}} \\ &= \frac{-mn^3}{\bar{x}(n-\bar{x})} < 0. \end{aligned}$$

Finally, since the values of the likelihood function at the boundaries of the parameter space, $\pi = 0$ and $\pi = 1$, are 0, it follows that $\pi = \frac{\bar{x}}{n}$ is the value that maximizes the likelihood function. The maximum likelihood estimate $\hat{\pi}(\mathbf{x}) = \frac{\bar{x}}{n}$ and the maximum likelihood estimator $\hat{\pi}(\mathbf{X}) = \frac{\bar{X}}{n}$.

To simulate $\hat{\pi} = \frac{\sum_{i=1}^m x_i}{mn} = \frac{\bar{x}}{n}$, generate 1000 random values from a $\text{Bin}(n = 3, \pi = 0.5)$ population. Pay particular attention to the fact that $n = 3$ and $m = 1000$.

Simulation of $\hat{\pi} = \frac{\sum_{i=1}^m x_i}{mn}$

```
> set.seed(23)
> pihat1 <- sum(rbinom(1000, 3, 0.5))/(1000 * 3)
> pihat1
[1] 0.5063333
```

Simulation of $\hat{\pi} = \frac{\bar{x}}{n}$

```
> set.seed(23)
> pihat2 <- mean(rbinom(1000, 3, 0.5))/3
> pihat2
[1] 0.5063333
```

The simulated value of $\hat{\pi}$ is 0.5063. ■

Example 7.20 Let X_1, X_2, \dots, X_n be a random sample from a $Pois(\lambda)$ population. Compute the maximum likelihood estimator and the maximum likelihood estimate for the parameter λ . Verify your answer with simulation by generating 20,000 random values from a $Pois(\lambda = 5)$ population.

Solution: The likelihood function is

$$L(\lambda|\mathbf{x}) = \prod_{i=1}^n \frac{e^{-\lambda} \lambda^{x_i}}{x_i!} = e^{-n\lambda} \prod_{i=1}^n \frac{\lambda^{x_i}}{x_i!}, \quad (7.32)$$

and the log-likelihood function is

$$\ln L(\lambda|\mathbf{x}) = \ln \left[e^{-n\lambda} \prod_{i=1}^n \frac{\lambda^{x_i}}{x_i!} \right] = -n\lambda + \sum_{i=1}^n x_i \ln \lambda - \sum_{i=1}^n \ln(x_i!). \quad (7.33)$$

Next, look for the value that maximizes the log-likelihood function by taking the first-order derivative of (7.33) and setting the answer to zero:

$$\frac{\partial \ln L(\lambda|\mathbf{x})}{\partial \lambda} = -n + \frac{\sum_{i=1}^n x_i}{\lambda} \stackrel{\text{set}}{=} 0. \quad (7.34)$$

The solution to (7.34) is $\lambda = \frac{\sum_{i=1}^n x_i}{n} = \bar{x}$. For $\lambda = \bar{x}$ to be a maximum, the second-order derivative of the log-likelihood function must be negative at $\lambda = \bar{x}$. The second-order derivative is

$$\frac{\partial^2 \ln L(\lambda|\mathbf{x})}{\partial \lambda^2} = -\frac{\sum_{i=1}^n x_i}{\lambda^2}.$$

Evaluating the second-order derivative at $\lambda = \bar{x}$ yields

$$\frac{\partial^2 \ln L(\lambda|\mathbf{x})}{\partial \lambda^2} = -\frac{n\bar{x}}{\bar{x}^2} = -\frac{n}{\bar{x}} < 0.$$

Finally, since the values of the likelihood function at the boundaries of the parameter space, $\lambda = 0$ and $\lambda = \infty$, are 0, it follows that $\lambda = \bar{x}$ is the value that maximizes the likelihood function. The maximum likelihood estimate $\hat{\lambda}(\mathbf{x}) = \bar{x}$ and the maximum likelihood estimator $\hat{\lambda}(\mathbf{X}) = \bar{X}$.

R Code 7.6 simulates $\hat{\lambda}(\mathbf{x}) = \bar{x}$ by generating 20,000 random values from a $Pois(\lambda = 5)$ population.

R Code 7.6

```
> set.seed(99)
> lambdahat <- mean(rpois(20000, 5))
> lambdahat
[1] 4.99415
```

The simulated value of $\hat{\lambda}$ is 4.9942. ■

Example 7.21 A farmer cans and sells mild and hot peppers at the local market. The farmer recently hired an assistant to label his products. The assistant is new to working with peppers and has mislabeled some of the hot peppers as mild peppers. The farmer performs a random check of 100 of the mild pepper cans labeled by the assistant to assess his work. Out of the 100 cans labeled mild peppers, it turns out that 8 are actually hot peppers.

- Which of the following proportions, 0.05, 0.08, or 0.10, maximizes the likelihood function?
- What is the maximum likelihood estimate for the proportion of cans the assistant has mislabeled?

Solution: The answers are as follows:

(a) First define the random variable X as the number of mislabeled cans. In this definition of the random variable X , it follows that $n = 100$ and $m = 1$ since $X \sim \text{Bin}(100, \theta)$. The likelihood function for a random sample of size m from a $\text{Bin}(n, \pi)$ population was computed in (7.29) as

$$L(\pi | \mathbf{x}) = \prod_{i=1}^m \binom{n}{x_i} \pi^{x_i} (1 - \pi)^{n-x_i}.$$

Since $m = 1$ here, it follows that the likelihood function is

$$L(\pi | \mathbf{x}) = \binom{n}{x} \pi^x (1 - \pi)^{n-x}.$$

Consequently, the value for π that maximizes

$$\mathbb{P}(X = 8 | \pi) = \binom{100}{8} \pi^8 (1 - \pi)^{92}$$

is the solution to the problem. The likelihoods for the three values of π are

$$\mathbb{P}(X = 8 | 0.05) = \binom{100}{8} 0.05^8 (1 - 0.05)^{92} = 0.0649,$$

$$\mathbb{P}(X = 8 | 0.08) = \binom{100}{8} 0.08^8 (1 - 0.08)^{92} = 0.1455,$$

and

$$\mathbb{P}(X = 8 | 0.10) = \binom{100}{8} 0.10^8 (1 - 0.10)^{92} = 0.1148.$$

Conclude that the value $\pi = 0.08$ is the value that maximizes the likelihood function among the three values of π provided.

- (b) Recall that the maximum likelihood estimator for a binomial distribution was computed in Example 7.19 on page 424 as $\hat{\pi}(\mathbf{X}) = \frac{\sum_{i=1}^m x_i}{mn}$. Therefore, the maximum likelihood estimate for the proportion of mislabeled cans is $\hat{\pi}(\mathbf{x}) = \frac{8}{100} = 0.08$.

Example 7.22 ▷ **I.I.D. Uniform Random Variables** ◃ Suppose $\{X_1, X_2, \dots, X_n\}$ is a random sample from a $\text{Unif}(0, \theta)$ distribution. Find the maximum likelihood estimator of θ . Find the maximum likelihood estimate for a randomly generated sample of 1000 $\text{Unif}(0, 2)$ random variables.

Solution: According to (4.9), the **pdf** of a random variable $X \sim \text{Unif}(0, \theta)$ is

$$f(x|\theta) = \frac{1}{\theta}, \quad 0 \leq x \leq \theta.$$

The likelihood function is

$$L(\theta|\mathbf{x}) = \begin{cases} \frac{1}{\theta^n} & \text{for } 0 \leq x_1 \leq \theta, 0 \leq x_2 \leq \theta, \dots, 0 \leq x_n \leq \theta \\ 0 & \text{otherwise.} \end{cases}$$

In this problem, the standard calculus approach fails since the maximum of the likelihood function occurs at a point of discontinuity. Consider the graph in Figure 7.6. Clearly $\frac{1}{\theta^n}$ is maximized for small values of θ ; however, the likelihood function is only defined as a value greater than zero if $\theta \geq \max(x_i)$. Specifically, if $\theta < \max(x_i)$, $L(\theta|\mathbf{x}) = 0$. It follows then that the maximum likelihood estimator is $\hat{\theta}(\mathbf{X}) = \max(X_i)$. R Code 7.7 finds the maximum likelihood estimate of 1000 randomly generated $\text{Unif}(0, 2)$ random variables.

R Code 7.7

```
> set.seed(6)
> mle <- max(runif(n = 1000, min = 0, max = 2))
> mle
[1] 1.999284
```

Thus, even though a standard calculus approach could not be used, the mle 1.9993 is close to the true $\theta = 2$.

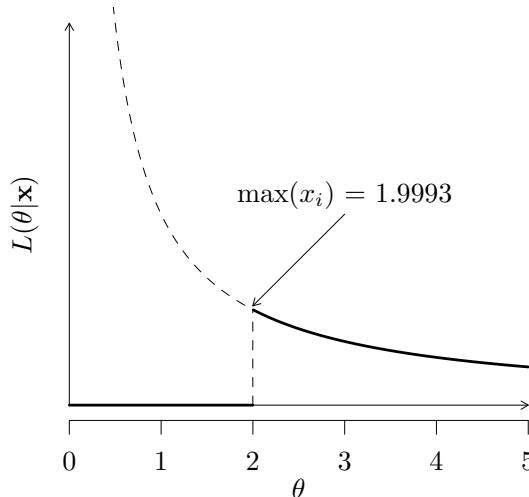


FIGURE 7.6: Illustration of the likelihood function in Example 7.22

Example 7.23 Suppose $\{X_1, X_2, \dots, X_n\}$ is a random sample from a $N(\mu, \sigma)$ distribution, where σ is assumed known. Find the maximum likelihood estimator of μ .

Solution: According to (4.23), the **pdf** of a random variable $X \sim N(\mu, \sigma^2)$ is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty.$$

The likelihood function is

$$L(\mu|\mathbf{x}) = \prod_{i=1}^n f(x_i) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}, \quad (7.35)$$

and the log-likelihood function is

$$\ln L(\mu|\mathbf{x}) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{\sum_{i=1}^n (x_i - \mu)^2}{2\sigma^2}. \quad (7.36)$$

To find the value of μ that maximizes $\ln L(\mu|\mathbf{x})$, take the first-order derivative of (7.36) with respect to μ , set the answer equal to zero, and solve. The first-order derivative of $\ln L(\mu|\mathbf{x})$ with respect to μ is

$$\frac{\partial \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \mu} = \frac{\sum_{i=1}^n (x_i - \mu)}{\sigma^2} \stackrel{\text{set}}{=} 0. \quad (7.37)$$

The solution to (7.37) is $\mu = \frac{\sum_{i=1}^n x_i}{n} = \bar{x}$. For $\mu = \bar{x}$ to be a maximum, the second-order derivative of the log-likelihood function with respect to μ must be negative at $\mu = \bar{x}$. The second-order derivative of (7.36) is

$$\frac{\partial^2 \ln L(\mu|\mathbf{x})}{\partial \mu^2} = -\frac{n}{\sigma^2} < 0. \quad (7.38)$$

Since (7.35) goes to zero at $\pm\infty$, the boundary values, it follows that $\mu = \bar{x}$ is a global maximum. Consequently, the maximum likelihood estimator of μ is $\hat{\mu}(\mathbf{X}) = \bar{X}$, and the maximum likelihood estimate of μ is $\hat{\mu}(\mathbf{x}) = \bar{x}$. ■

Example 7.24 Suppose $\{X_1, X_2, \dots, X_n\}$ is a random sample from a $N(\mu, \sigma^2)$ distribution, where μ is assumed known. Find the maximum likelihood estimator of σ^2 .

Solution: According to (4.23), the **pdf** of a random variable $X \sim N(\mu, \sigma^2)$ is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty.$$

The likelihood function is

$$L(\sigma^2|\mathbf{x}) = \prod_{i=1}^n f(x_i) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}, \quad (7.39)$$

and the log-likelihood function is

$$\ln L(\sigma^2|\mathbf{x}) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{\sum_{i=1}^n (x_i - \mu)^2}{2\sigma^2}. \quad (7.40)$$

To find the value of σ^2 that maximizes $\ln L(\sigma^2|\mathbf{x})$, take the first-order derivative of (7.40) with respect to σ^2 , set the answer equal to zero, and solve. The first-order derivative of $\ln L(\sigma^2|\mathbf{x})$ with respect to σ^2 is

$$\frac{\partial \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{\sum_{i=1}^n (x_i - \mu)^2}{2\sigma^4} \stackrel{\text{set}}{=} 0. \quad (7.41)$$

The solution to (7.41) is $\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$. For $\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$ to be a maximum, the second-order derivative of the log-likelihood function with respect to σ^2 must be negative at σ^2 . For notational ease, let $r = \sigma^2$ in (7.40) so that

$$\ln L(r|\mathbf{x}) = \ln L(\sigma^2|\mathbf{x}) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(r) - \frac{\sum_{i=1}^n (x_i - \mu)^2}{2r}. \quad (7.42)$$

The second-order derivative of (7.42) is

$$\frac{\partial^2 \ln L(r|\mathbf{x})}{\partial r^2} = \frac{n}{2}r^{-2} - \sum_{i=1}^n (x_i - \mu)^2 r^{-3} \stackrel{?}{<} 0. \quad (7.43)$$

Multiplying the left-hand side of (7.43) by r^3 gives

$$\frac{n}{2}r - \sum_{i=1}^n(x_i - \mu)^2 \stackrel{?}{<} 0. \quad (7.44)$$

By substituting the value for the mle, $r = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$, the ? above the < can be removed since

$$\frac{r}{2} < \frac{\sum_{i=1}^n (x_i - \mu)^2}{n} = \sigma^2 = r.$$

Since (7.39) goes to zero at $\pm\infty$, the boundary values, it follows that $\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$ is a global maximum. Consequently, the maximum likelihood estimator of σ^2 is $\widehat{\sigma^2}(\mathbf{X}) = \frac{\sum_{i=1}^n (X_i - \mu)^2}{n}$, and the maximum likelihood estimate of σ^2 is $\widehat{\sigma^2}(\mathbf{x}) = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$. ■

Example 7.25 Use `random.seed(33)` to generate 1000 $N(4, 1)$ random variables. Write log-likelihood functions for the simulated random variables and verify that the simulated maximum likelihood estimates for μ when $\sigma^2 = 1$ and σ^2 when $\mu = 4$ are reasonably close to the true parameters. Produce graphs of $\ln L(\mu|\mathbf{x}, \sigma = 1)$ versus μ and $\ln L(\sigma^2|\mathbf{x}, \mu = 4)$ versus σ^2 indicating where the simulated maximum occurs in each graph.

Solution: R Code 7.8 generates the requested data and stores the results in \mathbf{x} . The log likelihood functions ($\ln L(\mu|\mathbf{x}, \sigma = 1)$ and $\ln L(\sigma^2|\mathbf{x}, \mu = 4)$) and negative log likelihood functions ($-\ln L(\mu|\mathbf{x}, \sigma = 1)$ and $-\ln L(\sigma^2|\mathbf{x}, \mu = 4)$) are defined and stored in the objects `loglikemu`, `loglikesig2`, `negloglikemu`, and `negloglikesig2`, respectively. Both functions `nls()` and `optimize()` are used to find maximum likelihood estimates for μ and σ^2 .

R Code 7.8

```

> negloglikesig2 <- function(sig2){(-1)*(-n/2*log(2*pi) - n/2*log(sig2) -
+                                         (sum((x - mu1)^2))/(2*sig2))} 
> EM1 <- nlm(f = negloglikemu, p = 2)$estimate
> EM1
[1] 4.019708

> EM2 <- optimize(loglikemu, interval = c(2, 6), maximum = TRUE)$maximum
> EM2
[1] 4.019708

> ES1 <- nlm(f = negloglikesig2, p = 0.5)$estimate
> ES1
[1] 1.000426

> ES2 <- optimize(f = loglikesig2, interval = c(0.5, 2.5),
+                   maximum = TRUE)$maximum
> ES2
[1] 1.000417

```

Note that the maximum likelihood estimates for μ and σ^2 from the simulation are 4.0197 and 1.0004, respectively, which are reasonably close to the parameters $\mu = 4$ and $\sigma^2 = 1$.

R Code 7.9 can be used to create graphs of $\ln L(\mu|\mathbf{x}, \sigma = 1)$ versus μ and $\ln L(\sigma^2|\mathbf{x}, \mu = 4)$ versus σ^2 , similar to Figure 7.7 on the next page.

R Code 7.9

```

> p <- ggplot(data.frame(x = c(2, 6.05)), aes(x = x))
> p + stat_function(fun = loglikemu, n = 200) +
+   labs(x = expression(mu),
+         y = expression(textstyle(ln)~~L(mu*"\\"*bold(x), sigma==1))) +
+   geom_vline(xintercept = EM1, lty = "dashed")
> p <- ggplot(data.frame(x = c(0.5, 2.45)), aes(x = x))
> p + stat_function(fun = loglikesig2, n = 200) +
+   labs(x = expression(sigma^2),
+         y = expression(textstyle(ln)~~L(sigma^2*"\\"*bold(x), mu==4))) +
+   geom_vline(xintercept = ES1, lty = "dashed")

```

7.3.2.1 Fisher Information

Now that proficiency has been gained at calculating point estimates and estimators with maximum likelihood procedures, some measure of the variance of these estimators is desired. Investigating a quantity known as the **Fisher information** or simply the **information number** will give this measure. The Fisher information is the amount of information that an observable random variable X carries about an unknown parameter θ , upon which the likelihood function of X , $L(\theta|\mathbf{x})$, depends. This is the quantity

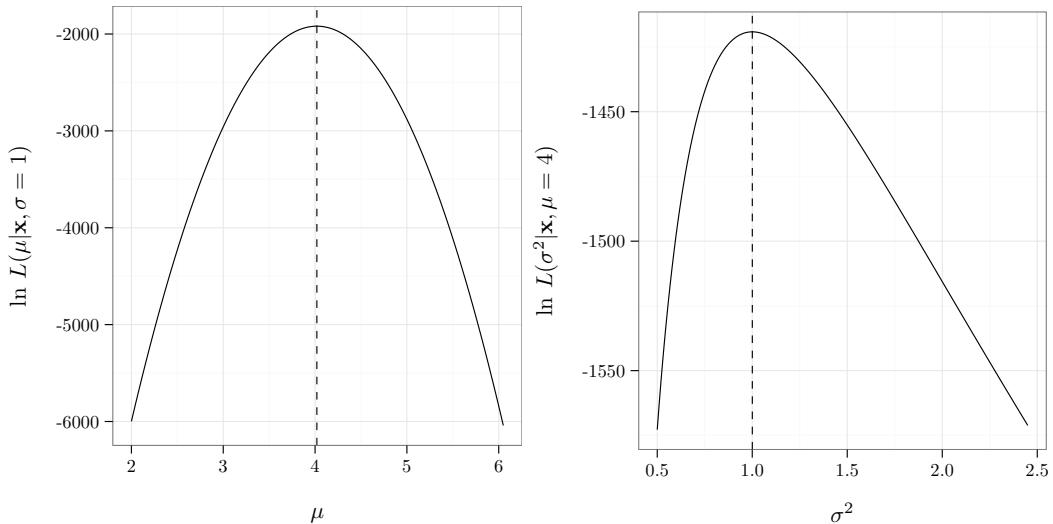


FIGURE 7.7: Illustration of $\ln L(\mu|\mathbf{x}, \sigma = 1)$ versus μ and $\ln L(\sigma^2|\mathbf{x}, \mu = 4)$ versus σ^2

$$E \left[\left(\frac{\partial \ln f(\mathbf{X}|\theta)}{\partial \theta} \right)^2 \right]. \quad (7.45)$$

This expression was briefly mentioned as the denominator of (7.7), the CRLB. However, the denominator of (7.7) used the form

$$n \cdot E \left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta} \right)^2 \right], \quad (7.46)$$

which is equivalent to (7.45) for random samples. Assume that X is a continuous random variable with **pdf** $f(x|\theta)$ (discrete random variables are handled in a similar fashion by exchanging integration for summation), where the following regularity conditions for $f(x|\theta)$ are satisfied:

1. The limits of support of $f(x|\theta)$ do not depend on θ .
2. The first two derivatives of $f(x|\theta)$ exist.
3. The order of integration and differentiation can be exchanged.

The inverse of the information number provides a bound for the variance of the best unbiased estimator of θ . Consequently, it makes sense to say the information number for a random sample of size n denoted $I_n(\theta)$ is the variance of the first-order partial derivative of the log-likelihood function. That is,

$$I_n(\theta) = \text{Var} \left[\left(\frac{\partial \ln f(\mathbf{X}|\theta)}{\partial \theta} \right) \right]. \quad (7.47)$$

When a random sample X_1, X_2, \dots, X_n is taken from a **pdf** $f(x|\theta)$, recall that $f(\mathbf{x}|\theta) = \prod_{i=1}^n f(x_i|\theta)$ so that $\ln f(\mathbf{x}|\theta) = \sum_{i=1}^n \ln f(x_i|\theta)$. When the random sample is of size $n = 1$, the Fisher information is denoted as simply $I(\theta)$, which is defined as

$$I(\theta) = \text{Var} \left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta} \right) \right]. \quad (7.48)$$

Since the random variables are independent, it should be clear that $I_n(\theta) = nI(\theta)$. The two common forms of expressing the information number for a random sample of size n are

$$I_n(\theta) = E \left[\left(\frac{\partial \ln f(\mathbf{X}|\theta)}{\partial \theta} \right)^2 \right] = nI(\theta) = nE \left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta} \right)^2 \right], \quad (7.49)$$

and

$$I_n(\theta) = -E \left[\left(\frac{\partial^2 \ln f(\mathbf{X}|\theta)}{\partial \theta^2} \right) \right] = nI(\theta) = -nE \left[\left(\frac{\partial^2 \ln f(X|\theta)}{\partial \theta^2} \right) \right]. \quad (7.50)$$

The form of the problem will often dictate which expression is easier to compute, as will be seen in the examples. The astute reader will have noticed that the equivalence of (7.49) and (7.50) was not shown, nor was the equivalence of (7.47) to (7.49) and (7.50).

Example 7.26 Given the **pdf** of a normal distribution with unknown mean μ and known variance σ^2 , find the Fisher information for μ using both (7.49) and (7.50) given a random sample of size n from said normal distribution.

Solution: According to (4.23), the **pdf** of a random variable $X \sim N(\mu, \sigma^2)$ is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty.$$

Note that

$$\frac{\partial \ln f(x|\mu)}{\partial \mu} = \frac{(x-\mu)}{\sigma^2},$$

and

$$\frac{\partial^2 \ln f(x|\mu)}{\partial \mu^2} = -\frac{1}{\sigma^2}.$$

Using (7.49), write

$$\begin{aligned} I_n(\mu) &= nE \left[\left(\frac{\partial \ln f(X|\mu)}{\partial \mu} \right)^2 \right] \\ &= nE \left[\left(\frac{X-\mu}{\sigma^2} \right)^2 \right] = n \frac{E[(X-\mu)^2]}{\sigma^4} = \frac{n\sigma^2}{\sigma^4} = \frac{n}{\sigma^2}. \end{aligned}$$

Using (7.50), write

$$\begin{aligned} I_n(\mu) &= -nE \left[\left(\frac{\partial^2 \ln f(X|\theta)}{\partial \theta^2} \right) \right] \\ &= -nE \left[-\frac{1}{\sigma^2} \right] = \frac{n}{\sigma^2}. \end{aligned}$$

Consequently, the smaller the variance σ^2 , the more information there is in a random sample of size n about μ . ■

7.3.2.2 Fisher Information for Several Parameters

Given a random variable X with **pdf** $f(x|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_k)$ is a k -dimensional vector of parameters, denote the information matrix of $\boldsymbol{\theta}$ as $\mathbf{I}(\boldsymbol{\theta})$. The $(i, j)^{\text{th}}$ element of the information matrix is defined as

$$\mathbf{I}_{i,j}(\boldsymbol{\theta}) = E \left[\left(\frac{\partial \ln f(X|\boldsymbol{\theta})}{\partial \theta_i} \right) \left(\frac{\partial \ln f(X|\boldsymbol{\theta})}{\partial \theta_j} \right) \right] = E \left[\left(\frac{\partial^2 \ln f(X|\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} \right) \right], \quad (7.51)$$

which is a generalization of (7.48). Likewise, when working with random samples,

$$\mathbf{I}_n(\boldsymbol{\theta}) = E \left[\left(\frac{\partial \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_i} \right) \left(\frac{\partial \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_j} \right) \right] = n \mathbf{I}_{i,j}(\boldsymbol{\theta}), \quad (7.52)$$

and

$$\mathbf{I}_n(\boldsymbol{\theta}) = -E \left[\frac{\partial^2 \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} \right] = n \mathbf{I}_{i,j}(\boldsymbol{\theta}), \quad (7.53)$$

are the generalizations of (7.49) and (7.50), respectively.

Example 7.27 Given a random sample of size n from a $N(\mu, \sigma^2)$ population, where $\boldsymbol{\theta} = (\mu, \sigma^2)$, find $\mathbf{I}_n(\boldsymbol{\theta})$.

Solution: According to (4.23), the pdf of a random variable $X \sim N(\mu, \sigma^2)$ is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty.$$

It follows then that

$$f(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^n f(x_i|\boldsymbol{\theta}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}},$$

and that

$$\ln f(\mathbf{x}|\boldsymbol{\theta}) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{\sum_{i=1}^n (x_i - \mu)^2}{2\sigma^2}.$$

Taking partial derivatives of $\ln f(\mathbf{x}|\boldsymbol{\theta})$ with respect to $\theta_1 = \mu$, and $\theta_2 = \sigma^2$ gives

$$\begin{aligned} \frac{\partial \ln f(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_1} &= \frac{\sum_{i=1}^n (x_i - \mu)}{\sigma^2}, \\ \frac{\partial^2 \ln f(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_1} &= -\frac{n}{\sigma^2}, \\ \frac{\partial \ln f(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_2} &= -\frac{n}{2\sigma^2} + \frac{\sum_{i=1}^n (x_i - \mu)^2}{2(\sigma^2)^2} \\ \frac{\partial^2 \ln f(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_2} &= \frac{n}{2(\sigma^2)^2} - \frac{\sum_{i=1}^n (x_i - \mu)^2}{(\sigma^2)^3}, \text{ and} \\ \frac{\partial^2 \ln f(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_2} &= \frac{\partial^2 \ln f(\mathbf{x}|\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_1} = -\frac{\sum_{i=1}^n (x_i - \mu)}{(\sigma^2)^2}. \end{aligned}$$

Using (7.53) gives

$$\mathbf{I}_n(\boldsymbol{\theta}) = -E \left[\frac{\partial^2 \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} \right] = \begin{pmatrix} -E \left[\frac{\partial^2 \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_1} \right] & -E \left[\frac{\partial^2 \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_2} \right] \\ -E \left[\frac{\partial^2 \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_1} \right] & -E \left[\frac{\partial^2 \ln f(\mathbf{X}|\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_2} \right] \end{pmatrix},$$

or

$$\mathbf{I}_n(\boldsymbol{\theta}) = \begin{pmatrix} -E \left[-\frac{n}{\sigma^2} \right] & -E \left[-\frac{\sum_{i=1}^n (X_i - \mu)}{(\sigma^2)^2} \right] \\ -E \left[-\frac{\sum_{i=1}^n (X_i - \mu)}{(\sigma^2)^2} \right] & -E \left[\frac{n}{2(\sigma^2)^2} - \frac{\sum_{i=1}^n (X_i - \mu)^2}{(\sigma^2)^3} \right] \end{pmatrix},$$

which, upon taking expected values, becomes

$$\mathbf{I}_n(\boldsymbol{\theta}) = \begin{pmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{pmatrix}.$$

■

7.3.2.3 Properties of Maximum Likelihood Estimators

Now that the Fisher information has been examined and several problems have been worked with maximum likelihood estimation, the properties of maximum likelihood estimators are formally enumerated:

1. MLEs are not necessarily unbiased. For example, when sampling from a $N(\mu, \sigma^2)$ population, the MLE of σ^2 is $\widehat{\sigma^2}(\mathbf{X}) = \sum_{i=1}^n \frac{(X_i - \bar{X})^2}{n}$, which is a biased estimator of σ^2 . Although some MLEs may be biased, all MLEs are consistent, which makes them asymptotically unbiased. Symbolically, MLEs $\not\Rightarrow$ unbiased estimators; however, MLEs \Rightarrow asymptotically unbiased estimators since MLEs \Rightarrow consistent estimators.
2. If T is a MLE of θ and g is any function, then $g(T)$ is the MLE of $g(\theta)$. This is known as the **invariance** property of MLEs. For example, if \bar{X} is the MLE of θ , then \bar{X}^2 is the MLE of θ^2 .
3. When certain regularity conditions on $f(x|\theta)$ are satisfied and an efficient estimator exists for the estimated parameter, the efficient estimator is the MLE of the estimated parameter. Be careful, not all MLEs are efficient! If an efficient estimator exists, the efficient estimator is also the MLE. That is, efficiency \Rightarrow MLE, but MLE $\not\Rightarrow$ efficiency.
4. Under certain regularity conditions on $f(x|\theta)$, the MLE $\hat{\theta}(\mathbf{X})$ of θ based on a sample of size n from $f(x|\theta)$ is asymptotically normally distributed with mean θ and variance $I_n(\theta)^{-1}$. That is, as $n \rightarrow \infty$,

$$\hat{\theta}(\mathbf{X}) \sim N\left(\theta, \sqrt{I_n(\theta)^{-1}}\right). \quad (7.54)$$

The statement in (7.54) is the basis for large sample hypothesis tests (covered in Chapter 9) and confidence intervals (covered in Chapter 8).

Note that the asymptotic variance of MLEs equals the Cramér-Rao lower bound since they are asymptotically efficient. That is, MLEs \Rightarrow asymptotic efficiency. Consequently, a reasonable approximation to the distribution of $\hat{\theta}(\mathbf{X})$ for large sample sizes can be obtained; however, a normal distribution for $\hat{\theta}(\mathbf{X})$ cannot be guaranteed when the sample size is small.

Example 7.28 In Example 7.19 on page 424, it was found that the sample proportion of successes for a random sample of size m from a $Bin(n, \pi)$ distribution had $\hat{\pi} = \frac{\sum_{i=1}^m X_i}{mn}$ for its mle. That is, the MLE for the binomial proportion π is $\hat{\pi}(\mathbf{X}) = \frac{\sum_{i=1}^m X_i}{mn}$. What is the MLE for the variance of the sample proportion of successes where the random variable $\hat{\pi}$ is defined as $\frac{\sum_{i=1}^m X_i}{mn}$?

Solution: Given that $X \sim \text{Bin}(n, \pi)$, the variance of X is $n\pi(1 - \pi)$. Therefore,

$$\text{Var}[\hat{\pi}] = \text{Var}\left[\frac{\sum_{i=1}^m X_i}{mn}\right] = \frac{\sum_{i=1}^m \text{Var}[X_i]}{m^2 n^2} = \frac{mn\pi(1 - \pi)}{m^2 n^2} = \frac{\pi(1 - \pi)}{mn}.$$

Since $\text{Var}[\hat{\pi}]$ is a function of the MLE $\hat{\pi}(\mathbf{X})$, it follows using the invariance property of MLEs that the MLE of the variance of $\hat{\pi}$ is

$$\widehat{\text{Var}}[\hat{\pi}(\mathbf{X})] = \frac{\hat{\pi}(1 - \hat{\pi})}{mn}.$$

Note: Many texts will list the MLE of the variance of the sample proportion of successes in a binomial distribution as $\frac{\hat{\pi}(1 - \hat{\pi})}{n}$ because they use $m = 1$ in their definition of $\hat{\pi}$. ■

Example 7.29 ▷ **MOM and MLE for a Gamma** ◃ Given a random sample of size n from a population with pdf

$$f(x|\theta) = \frac{x}{\theta^2} e^{-\frac{x}{\theta}}, \quad x \geq 0, \quad \theta > 0,$$

- (a) Find an estimator of θ using the method of moments.
- (b) Find an estimator of θ using the method of maximum likelihood.
- (c) Are the method of moments and maximum likelihood estimators of θ unbiased?
- (d) Compute the variance of the MLE of θ .
- (e) Is the MLE of θ efficient?

Solution: Since $X \sim \text{Gamma}(\alpha = 2, \lambda = \frac{1}{\theta})$, according to (4.16), $E[X] = \frac{\alpha}{\lambda} = 2\theta$ and $\text{Var}[X] = \frac{\alpha}{\lambda^2} = 2\theta^2$.

(a) Equating the first population moment about the origin to the first sample moment about the origin gives

$$\alpha_1(\theta) = 2\theta \stackrel{\text{set}}{=} \bar{X} = m_1,$$

which implies that the method of moments estimator for θ is $\tilde{\theta} = \frac{\bar{X}}{2}$.

(b) The likelihood equation is given as

$$L(\theta|\mathbf{x}) = \prod_{i=1}^n f(x_i) = \frac{\prod_{i=1}^n x_i}{\theta^{2n}} e^{-\frac{\sum_{i=1}^n x_i}{\theta}}, \quad (7.55)$$

and the log-likelihood function is

$$\ln L(\theta|\mathbf{x}) = -2n \ln(\theta) + \sum_{i=1}^n \ln(x_i) - \frac{\sum_{i=1}^n x_i}{\theta}. \quad (7.56)$$

To find the value of θ that maximizes $\ln L(\theta|\mathbf{x})$, take the first-order derivative of (7.56) with respect to θ , set the answer equal to zero, and solve. The first-order derivative of $\ln L(\theta|\mathbf{x})$ with respect to θ is

$$\frac{\partial \ln L(\theta|\mathbf{x})}{\partial \theta} = -\frac{2n}{\theta} + \frac{\sum_{i=1}^n x_i}{\theta^2} \stackrel{\text{set}}{=} 0. \quad (7.57)$$

The solution to (7.57) is $\theta = \frac{\bar{X}}{2}$, which agrees with the method of moments estimator; however, to ensure that $\theta = \frac{\bar{X}}{2}$ is a maximum, the second-order derivative with respect to θ must be negative. The second-order derivative of (7.56) is

$$\frac{\partial^2 \ln L(\theta|\mathbf{x})}{\partial \theta^2} = \frac{2n}{\theta^2} - \frac{2 \sum_{i=1}^n x_i}{\theta^3} < 0. \quad (7.58)$$

By using $\theta = \frac{\bar{X}}{2}$ in (7.58), arrive at the expression

$$-\frac{8n}{\bar{X}^2} < 0. \quad (7.59)$$

The ? above the $<$ in (7.59) can be removed provided $\bar{X} \neq 0$. Since $\mathbb{P}(X = 0) = \int_0^0 f(x) dx = 0 \Rightarrow \exists x_i \neq 0 \Rightarrow \bar{X} \neq 0 \Rightarrow \bar{X}^2 > 0$. Finally, since (7.55) goes to zero as $\theta \rightarrow \infty$, it can be concluded that $\theta = \frac{\bar{X}}{2}$ is a global maximum. Consequently, the maximum likelihood estimator of θ is $\hat{\theta}(\mathbf{X}) = \frac{\bar{X}}{2}$.

(c) Since both the method of moments and the method of maximum likelihood returned the same estimator for θ , that is, $\hat{\theta}(\mathbf{X}) = \tilde{\theta} = \frac{\bar{X}}{2}$, the question is

$$E[\hat{\theta}(\mathbf{X})] = E[\tilde{\theta}] = \theta.$$

Both $\tilde{\theta}$ and $\hat{\theta}(\mathbf{X})$ are therefore unbiased estimators since

$$E[\hat{\theta}(\mathbf{X})] = E[\tilde{\theta}] = E\left[\frac{\bar{X}}{2}\right] = \frac{\sum_{i=1}^n E[X_i]}{2n} = \frac{n \cdot 2\theta}{2n} = \theta.$$

(d) The variance of the MLE of θ is

$$Var[\hat{\theta}(\mathbf{X})] = Var\left[\frac{\bar{X}}{2}\right] = Var\left[\frac{\sum_{i=1}^n X_i}{2n}\right] = \frac{n Var[X]}{4n^2} = \frac{n2\theta^2}{4n^2} = \frac{\theta^2}{2n}.$$

(e) For $\hat{\theta}(\mathbf{X}) = \frac{\bar{X}}{2}$ to be considered an efficient or minimum variance estimator of θ , the variance of $\frac{\bar{X}}{2}$ must equal the CRLB. That is, does

$$Var[\hat{\theta}(\mathbf{X})] = \frac{\theta^2}{2n} \stackrel{?}{=} \frac{1}{n \cdot E\left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta}\right)^2\right]}?$$

Since $f(x|\theta) = \frac{x}{\theta^2} e^{-\frac{x}{\theta}}$ for $x \geq 0$ and $\theta > 0$, it follows that $\ln f(x|\theta) = \ln x - 2 \ln \theta - \frac{x}{\theta}$, and that $\frac{\partial \ln f(x|\theta)}{\partial \theta} = \frac{x-2\theta}{\theta^2}$. Consequently,

$$\frac{1}{n \cdot E\left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta}\right)^2\right]} = \frac{1}{n \cdot E\left[\left(\frac{X-2\theta}{\theta^2}\right)^2\right]} = \frac{1}{\frac{n \cdot Var[X]}{\theta^4}} = \frac{1}{\frac{n \cdot 2\theta^2}{\theta^4}} = \frac{\theta^2}{2n},$$

therefore $\frac{\bar{X}}{2}$ is an efficient estimator of θ . ■

Example 7.30 ▷ MLEs for Exponentials ◁ Given a random sample of size n from an exponential distribution with pdf

$$f(x) = \frac{1}{\theta} e^{-\frac{x}{\theta}} \quad x \geq 0, \quad \theta > 0, \quad (7.60)$$

- (a) find the MLE of θ^2 ,
- (b) show that the MLE of θ^2 is a biased estimator of θ^2 ,
- (c) provide an unbiased estimator of θ^2 ,
- (d) find the variance of your MLE of θ^2 ,
- (e) find the variance of your unbiased estimator of θ^2 , and
- (f) show that the variance for the MLE of θ^2 converges to $I_n(\theta)^{-1}$ as $n \rightarrow \infty$ according to property 4 of the Properties of MLEs on page 435.

Solution: To find the MLE of θ^2 , there are two possibilities. First, the MLE of θ could be found and the invariance property could be used to say that this estimate squared is the MLE of θ^2 . Second, and this is the current approach, the MLE of θ^2 can be found directly.

- (a) For notational ease, use the change of variable $\theta^2 = p$ and $\theta = \sqrt{p}$ in (7.60). The resulting **pdf** using the change of variable is

$$f(x) = \frac{1}{\sqrt{p}} e^{-\frac{x}{\sqrt{p}}} \quad x \geq 0, \quad p > 0.$$

The likelihood function is

$$L(p|\mathbf{x}) = \prod_{i=1}^n f(x_i) = \prod_{i=1}^n \frac{1}{\sqrt{p}} e^{-\frac{x_i}{\sqrt{p}}} = \frac{1}{(\sqrt{p})^n} e^{-\frac{\sum_{i=1}^n x_i}{\sqrt{p}}}, \quad (7.61)$$

and the log-likelihood function is

$$\ln L(p|\mathbf{x}) = -\frac{n}{2} \ln p - \frac{\sum_{i=1}^n x_i}{\sqrt{p}}. \quad (7.62)$$

To find the value of p that maximizes $\ln L(p|\mathbf{x})$, take the first-order derivative of (7.62) with respect to p , set the answer equal to zero, and solve. The first-order derivative of $\ln L(p|\mathbf{x})$ with respect to p is

$$\frac{\partial \ln L(p|\mathbf{x})}{\partial p} = -\frac{n}{2p} + \frac{\sum_{i=1}^n x_i}{2p^{\frac{3}{2}}} \stackrel{\text{set}}{=} 0. \quad (7.63)$$

The solution to (7.63) is $p = \bar{x}^2$. For $p = \bar{x}^2$ to be a maximum, the second-order derivative of the log-likelihood function with respect to p must be negative at $p = \bar{x}^2$. The second-order derivative of (7.62) is

$$\frac{\partial^2 \ln L(p|\mathbf{x})}{\partial p^2} = \frac{n}{2p^2} - \frac{3 \sum_{i=1}^n x_i}{4p^{\frac{5}{2}}} \stackrel{?}{<} 0. \quad (7.64)$$

By substituting $p = \bar{x}^2$ in the right-hand side of (7.64), the ? above the $<$ can be removed since $\bar{x} < \frac{3\bar{x}}{2}$ because $\bar{x} > 0$ for any sample due to the fact that $\mathbb{P}(X = 0) = 0$ for any continuous distribution. Finally, since as $p \rightarrow \infty$, $L(p|\mathbf{x}) \rightarrow 0$, it can be concluded that the MLE of $p = \theta^2$ is $\hat{p}(\mathbf{X}) = \hat{\theta}^2(\mathbf{X}) = \bar{X}^2$.

- (b) Next, show that \bar{X}^2 is a biased estimator of θ^2 . The easiest way to determine the mean and variance of \bar{X}^2 is with moment-generating functions. It is known that the moment-generating function of an exponential random variable, X , is $M_X(t) = (1 - \theta t)^{-1}$. Furthermore, if $Y = \sum_{i=1}^n c_i X_i$ and each X_i has a moment-generating function $M_{X_i}(t)$, then the moment-generating function of Y is $M_Y(t) = \prod_{i=1}^n M_{X_i}(c_i t)$. In the case where $Y =$

$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$, each $c_i = \frac{1}{n}$. For the special case of the exponential, the moment-generating function for \bar{X} is

$$M_{\bar{X}}(t) = M_Y(t) = \prod_{i=1}^n \left(1 - \theta \cdot \frac{t}{n}\right)^{-1} = \left(1 - \frac{\theta t}{n}\right)^{-n}.$$

Thus, to calculate the mean and variance of \bar{X}^2 , take the first through fourth derivatives of $M_{\bar{X}}(t)$ and evaluate them when $t = 0$ to find $E[\bar{X}^i]$ for $i = 1, 2, 3$, and 4. The first, second, third, and fourth derivatives of $M_{\bar{X}}(t)$, respectively, are

$$\begin{aligned} M'_{\bar{X}}(t) &= -n \left(1 - \frac{\theta}{n}t\right)^{-n-1} \left(-\frac{\theta}{n}\right) = \theta \left(1 - \frac{\theta}{n}t\right)^{-n-1}, \\ M''_{\bar{X}}(t) &= \theta(-n-1) \left(1 - \frac{\theta}{n}t\right)^{-n-2} \left(-\frac{\theta}{n}\right), \\ M'''_{\bar{X}}(t) &= \frac{\theta^2(n+1)}{n}(-n-2) \left(1 - \frac{\theta}{n}t\right)^{-n-3} \left(-\frac{\theta}{n}\right), \text{ and} \\ M^{(4)}_{\bar{X}}(t) &= \frac{\theta^3(n+1)(n+2)}{n^2}(-n-3) \left(1 - \frac{\theta}{n}t\right)^{-n-4} \left(-\frac{\theta}{n}\right). \end{aligned}$$

Evaluating these derivatives at $t = 0$ gives the expected values of \bar{X} to the first, second, third, and fourth powers:

$$\begin{aligned} M'_{\bar{X}}(0) &= \theta = E[\bar{X}], \\ M''_{\bar{X}}(0) &= \frac{\theta^2(n+1)}{n} = E[\bar{X}^2], \\ M'''_{\bar{X}}(0) &= \frac{\theta^3(n+1)(n+2)}{n^2} = E[\bar{X}^3], \text{ and} \\ M^{(4)}_{\bar{X}}(0) &= \frac{\theta^4(n+1)(n+2)(n+3)}{n^3} = E[\bar{X}^4]. \end{aligned}$$

Since $E[\bar{X}^2] = \frac{\theta^2(n+1)}{n} \neq \theta^2$, \bar{X}^2 is a biased estimator of θ^2 .

(c) An unbiased estimator of θ^2 would be to use the quantity $\frac{n\bar{X}^2}{n+1}$.

(d) The variance of \bar{X}^2 can be computed as $E[\bar{X}^4] - (E[\bar{X}^2])^2$:

$$\begin{aligned} \text{Var}[\bar{X}^2] &= \frac{\theta^4(n+1)(n+2)(n+3)}{n^3} - \left(\frac{\theta^2(n+1)}{n}\right)^2 \\ &= \frac{2\theta^4(2n^2 + 5n + 3)}{n^3} \\ &= \frac{2\theta^4((2n+3)(n+1))}{n^3}. \end{aligned} \tag{7.65}$$

(e) The variance of the unbiased estimator of θ^2 is

$$\begin{aligned} \text{Var}\left[\frac{n\bar{X}^2}{n+1}\right] &= \frac{n^2}{(n+1)^2} \text{Var}[\bar{X}^2] \\ &= \frac{n^2}{(n+1)^2} \cdot \frac{2\theta^4((2n+3)(n+1))}{n^3} \\ &= \frac{2\theta^4(2n+3)}{n(n+1)}. \end{aligned}$$

(f) The Fisher information is computed as

$$\begin{aligned} I_n(p) &= -E\left[\left(\frac{\partial^2 \ln f(\mathbf{X}|p)}{\partial p^2}\right)\right] = -E\left[\frac{n}{2p^2} - \frac{3\sum_{i=1}^n X_i}{4p^{\frac{5}{2}}}\right] \\ &= -\left[\frac{n}{2p^2} - \frac{3n\sqrt{p}}{4p^{\frac{5}{2}}}\right] = \frac{n}{2p^2} \left[-1 + \frac{3}{2}\right] = \frac{n}{4p^2}. \end{aligned}$$

Since $p = \theta^2$, it follows that $I_n(p) = I_n(\theta^2) = \frac{n}{4\theta^4}$, and that $I_n(\theta^2)^{-1} = \frac{4\theta^4}{n}$. Note that the variance of the MLE estimator \bar{X}^2 given in (7.65) converges to $I_n(\theta^2)^{-1} = 0$ as $n \rightarrow \infty$. ■

7.3.2.4 Finding Maximum Likelihood Estimators for Multiple Parameters

When the **pdf** contains more than one parameter, the procedure for finding the MLEs for several parameters proceeds in a fashion analogous to the one-parameter case. Given a vector $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_k\}$, the likelihood function is represented as

$$\begin{aligned} L(\boldsymbol{\theta}|\mathbf{x}) &= L(\theta_1, \dots, \theta_k | x_1, \dots, x_n) \\ &= f(x_1 | \theta_1, \dots, \theta_k) \times \dots \times f(x_n | \theta_1, \dots, \theta_k). \end{aligned} \tag{7.66}$$

The value of $\boldsymbol{\theta}$ that maximizes $L(\boldsymbol{\theta}|\mathbf{x})$ is the mle of $\boldsymbol{\theta}$. In the multiple parameter case, denote the mle of $\boldsymbol{\theta}$ as $\hat{\boldsymbol{\theta}}(\mathbf{x})$ and the MLE of $\boldsymbol{\theta}$ as $\hat{\boldsymbol{\theta}}(\mathbf{X})$. As with the univariate case, one will typically work with the log-likelihood function ($\ln L(\boldsymbol{\theta}|\mathbf{x})$) instead of the likelihood function. If $L(\boldsymbol{\theta}|\mathbf{x})$ is differentiable with respect to $\boldsymbol{\theta}$, a possible mle for $\boldsymbol{\theta}$ are the θ_i s, $i = 1, \dots, k$, that solve

$$\frac{\partial \ln L(\boldsymbol{\theta}|\mathbf{x})}{\partial \theta_i} \stackrel{\text{set}}{=} 0 \Leftrightarrow \begin{cases} \frac{\partial \ln L(\boldsymbol{\theta}|\mathbf{x})}{\partial \theta_1} = \sum_{i=1}^n \frac{\partial \ln f(x_i | \boldsymbol{\theta})}{\partial \theta_1} \stackrel{\text{set}}{=} 0 \\ \vdots \\ \frac{\partial \ln L(\boldsymbol{\theta}|\mathbf{x})}{\partial \theta_k} = \sum_{i=1}^n \frac{\partial \ln f(x_i | \boldsymbol{\theta})}{\partial \theta_k} \stackrel{\text{set}}{=} 0. \end{cases} \tag{7.67}$$

Just as with the univariate case, possible mles for $\boldsymbol{\theta}$ are the solutions to (7.67). Solutions to the k equations in (7.67) are a necessary but not sufficient condition for the solutions to be maxima; however, a sufficient condition to guarantee the solutions to (7.67) are maxima is for the Hessian matrix (matrix whose elements are the second-order partial derivatives with respect to the parameters being estimated) to be negative definite when evaluated at the maximum likelihood estimators. Any symmetric $p \times p$ matrix is **negative definite** provided

the leading principal minors (the determinants of the upper left square submatrices) have alternating signs where the top left element in the matrix is negative. These principal minors are denoted by D_i for $i = 1, \dots, p$ and satisfy the following conditions: $D_1 < 0$, $D_2 > 0, \dots$, ending with $D_p > 0$ if p is even and $D_p < 0$ if p is odd (Casella and Berger, 1990). Furthermore, the solutions to (7.67) will yield minima when the determinants of the leading principal minors are all positive.

Example 7.31 Given a random sample of size n from a normal distribution with unknown mean μ and variance σ^2 , find the MLEs for μ and σ^2 .

Solution: The pdf for a random variable $X \sim N(\mu, \sigma^2)$ according to (4.23) is

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

The likelihood function is

$$L(\mu, \sigma^2 | \mathbf{x}) = \prod_{i=1}^n f(x_i) = \frac{1}{(2\pi)^{\frac{n}{2}} (\sigma^2)^{\frac{n}{2}}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2},$$

and the log-likelihood function is

$$\ln L(\mu, \sigma^2 | \mathbf{x}) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2. \quad (7.68)$$

To find the θ that maximizes (7.68), take the first-order partial derivatives with respect to $\theta = (\mu, \sigma^2)$, set those first-order partial derivatives equal to zero, and solve the simultaneous equations:

$$\frac{\partial \ln L(\theta | \mathbf{x})}{\partial \theta} \stackrel{\text{set}}{=} 0 \Leftrightarrow \begin{cases} \frac{\partial \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \mu} = \frac{\sum_{i=1}^n (x_i - \mu)}{\sigma^2} \stackrel{\text{set}}{=} 0 \\ \frac{\partial \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{\sum_{i=1}^n (x_i - \mu)^2}{2\sigma^4} \stackrel{\text{set}}{=} 0. \end{cases}$$

The solution to the system of equations is

$$\mu = \bar{x} \quad \text{and} \quad \sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}.$$

A sufficient condition for the values in θ to be maxima is for the Hessian matrix to be negative definite. In this case, the Hessian matrix is

$$H = \begin{pmatrix} \frac{\partial^2 \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \mu^2} & \frac{\partial^2 \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \mu \partial \sigma^2} \\ \frac{\partial^2 \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \sigma^2 \partial \mu} & \frac{\partial^2 \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial (\sigma^2)^2} \end{pmatrix}.$$

Specifically, the second-order partial derivatives are

$$\begin{aligned} \frac{\partial^2 \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \mu^2} &= -\frac{n}{\sigma^2}, \\ \frac{\partial^2 \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial (\sigma^2)^2} &= \frac{n}{2\sigma^4} - \frac{1}{\sigma^6} \sum_{i=1}^n (x_i - \mu)^2, \quad \text{and} \\ \frac{\partial^2 \ln L(\mu, \sigma^2 | \mathbf{x})}{\partial \mu \partial \sigma^2} &= -\frac{1}{\sigma^4} \sum_{i=1}^n (x_i - \mu). \end{aligned}$$

By substituting the values $\mu = \bar{x}$ and $\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} = s_u^2$ in the second-order derivatives, the Hessian matrix is expressed as

$$H = \begin{pmatrix} -\frac{n}{s_u^2} & 0 \\ 0 & -\frac{n}{2s_u^4} \end{pmatrix}.$$

Note that H is negative definite since $D_1 = -\frac{n}{s_u^2} < 0$ and $D_2 = \frac{n^2}{2s_u^6} > 0$, implying that the solutions, $\mu = \bar{x}$ and $\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$, are maxima. Finally, the solutions $\mu = \bar{x}$ and $\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$ can be considered global maxima since the likelihood function goes to zero for both $\mu = \pm\infty$ and $\sigma^2 = \infty$. Consequently, the MLE of θ is written as $\hat{\theta}(\mathbf{X}) = (\bar{X}, S_u^2)$, and the mle of θ as $\hat{\theta}(\mathbf{x}) = (\bar{x}, s_u^2)$.

Example 7.32 Use `set.seed(11)` to generate 500 values from a $N(2, 1)$ population, and treat the generated values as a random sample of size $n = 500$ from a normal distribution with unknown parameters. Find the maximum likelihood estimates for μ and σ^2 based on the generated sample.

Solution: According to the results of Example 7.31 on the preceding page, the MLE of θ when sampling from a normal distribution with unknown mean and variance is $\hat{\theta}(\mathbf{X}) = (\bar{X}, S_u^2)$. R Code 7.10 performs the requested simulation.

R Code 7.10

```
> set.seed(11)
> n <- 500
> x <- rnorm(n, 2, 1)
> Xbar <- mean(x)
> S2u <- sum((x - mean(x))^2/n)
> c(Xbar, S2u)

[1] 1.9973596 0.9764792
```

From this simulation, $\hat{\theta}(\mathbf{x}) = (1.9974, 0.9765)$. Another approach is to allow R to find the values that maximize the negative log-likelihood function analytically using `nlm()` as shown in R Code 7.11.

R Code 7.11

```
> negloglike <- function(p){ (n/2)*log(2*pi) + (n/2)*log(2*p[2]) +
+ (1/(2*p[2]))*sum((x - p[1])^2) }
> nlm(f = negloglike, p = c(3, 2))$estimate

[1] 1.9973587 0.9764787
```

7.3.2.5 Multi-Parameter Properties of MLEs

The four properties for a MLE $\hat{\theta}(\mathbf{X})$ of θ given in Section 7.3.2.3 on page 435 also apply to a k -dimensional vector $\theta = (\theta_1, \theta_2, \dots, \theta_k)$ of parameters. Of particular importance is the generalization of Property 4 on page 435. Specifically, Property 4 on page 435 can be

extended so that under certain regularity conditions on $f(x|\theta)$, the MLE $\hat{\theta}(\mathbf{X})$ of θ based on a sample of size n is asymptotically normally distributed with mean $\boldsymbol{\theta}$ and variance-covariance matrix $\mathbf{I}_n(\boldsymbol{\theta})^{-1}$. That is,

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta}, \mathbf{I}_n(\boldsymbol{\theta})^{-1}),$$

and the variance-covariance MLEs are

$$\widehat{\mathbf{I}_n(\boldsymbol{\theta})}^{-1} = \mathbf{I}_n(\boldsymbol{\theta})^{-1}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(\mathbf{X})}.$$

Example 7.33 Given a random sample of size n from a $N(\mu, \sigma^2)$ population, find the MLE of the variance of \bar{X} and the variance of S_u^2 .

Solution: In Example 7.31 on page 441, the MLE of $\boldsymbol{\theta}$ was $\hat{\boldsymbol{\theta}}(\mathbf{X}) = (\bar{X}, S_u^2)$, and in Example 7.27 on page 434, the Fisher information matrix was

$$\mathbf{I}_n(\boldsymbol{\theta}) = \begin{pmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{pmatrix}.$$

Consequently,

$$\begin{aligned} \widehat{\mathbf{I}_n(\boldsymbol{\theta})}^{-1} &= \mathbf{I}_n(\boldsymbol{\theta})^{-1}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(\mathbf{X})} = \left(\begin{pmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{pmatrix}^{-1} \right) \Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(\mathbf{X})} \\ &= \left(\begin{pmatrix} \frac{n}{S_u^2} & 0 \\ 0 & \frac{n}{2S_u^4} \end{pmatrix}^{-1} \right) = \left(\begin{pmatrix} \frac{S_u^2}{n} & 0 \\ 0 & \frac{2S_u^4}{n} \end{pmatrix} \right), \end{aligned}$$

from which it can be concluded that

$$\widehat{I_{11}(\boldsymbol{\theta})}^{-1} = \widehat{\text{Var}(\bar{X})} = \frac{S_u^2}{n},$$

and

$$\widehat{I_{22}(\boldsymbol{\theta})}^{-1} = \widehat{\text{Var}(S_u^2)} = \frac{2S_u^4}{n}. \quad \blacksquare$$

7.4 Problems

1. Use the data from the data frame **WHEATSPAIN** to answer the questions.
 - (a) Find the mean, median, *MAD*, standard deviation, and *IQR* of the surface area measured in hectares.
 - (b) Remove the **Castilla-Leon** community and again find the mean, median, *MAD*, standard deviation, and *IQR* of the same variable. Which statistics are preferred as measures for these data? Comment on the results.
2. Given the estimators of the mean $T_1 = (X_1 + 2X_2 + X_3)/4$ and $T_2 = (X_1 + X_2 + X_3)/3$, where X_1, X_2, X_3 is a random sample from a $N(\mu, \sigma^2)$ distribution, prove that T_2 is more efficient than T_1 .
3. Given a random sample of size $n + 1$ from a $N(\mu, \sigma^2)$ distribution, show that the median, m , is roughly 64% less efficient than the sample mean for estimating the population mean. (Hint: In large samples, $\text{Var}(m) = \pi\sigma^2/4n$.)
4. Let X be a $\text{Bin}(n, \pi)$ random variable.
 - (a) Find the mean squared error of the π parameter estimators $T_1 = X/n$ and $T_2 = (X + 1)/(n + 2)$.
 - (b) When $n = 100$ and $\pi = 0.4$, which estimator, T_1 or T_2 , has the smaller *MSE*?
 - (c) Plot the efficiency of T_2 relative to T_1 versus π values in $(0, 1)$ for n values from 1 to 10.
5. Let X be a $\text{Bin}(n, \pi)$ random variable.
 - (a) Find the mean squared error of the π parameter estimators $T_1 = X/n$ and $T_2 = (X + 2)/(n + 4)$.
 - (b) When $n = 20$ and $\pi = 0.4$, which estimator, T_1 or T_2 , has the smaller *MSE*?
 - (c) Plot the efficiency of T_2 relative to T_1 versus π values in $(0, 1)$ for n values from 1 to 10.
6. Consider a random sample of size n from a $\Gamma(2, \lambda)$ distribution and the following estimators for $1/\lambda$:

$$T_1 = \frac{\bar{X}}{2} \quad \text{and} \quad T_2 = \frac{\sum_{i=1}^n X_i}{2(n+1)}.$$
 - (a) Graph the relative efficiency of T_2 with regard to T_1 for values of λ from 0.01 to 100 with a sample size of 50.
 - (b) Interpret the graph in (a).
 - (c) Plot the relative efficiency of both estimators versus sample sizes from 1 to 30.
 - (d) Interpret the graph in (c).

- (e) Generalize your findings.

(Hint: $X \sim \Gamma(\alpha, \lambda)$, $E[X] = \alpha/\lambda$, $\text{Var}[X] = \alpha/\lambda^2$.)

7. Consider a random variable $X \sim \text{Exp}(\lambda)$ and two estimators of $\frac{1}{\lambda}$, the expected value of X :

$$T_1 = \bar{X} \quad \text{and} \quad T_2 = \frac{\sum_{i=1}^n X_i + 1}{n+2}.$$

- (a) Derive an expression for the relative efficiency of T_2 with respect to T_1 .
 (b) Plot $\text{eff}(T_2, T_1)$ versus n values of 1, 2, 3, 4, 20, 25, 30.
 (c) Generalize your findings.

8. A baseball pitching machine launches fast balls whose speed follows a $N(\mu, \sigma = 5 \text{ km/h})$ distribution. Given the independent random samples \mathbf{X} and \mathbf{Y} , where $n_X = n_Y = 6$,

- (a) Show that the estimators $T_1 = \bar{X}$ and $T_2 = \frac{\sum_{i=2}^6 Y_i}{5}$ are unbiased estimators of μ .
 (b) Given the estimator $T_3 = \theta T_1 + (1 - \theta)T_2$, find the value of θ so that the MSE is a minimum.
9. Verify that $\text{Var}\left[\frac{\partial \ln f(X|\theta)}{\partial \theta}\right] = E\left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta}\right)^2\right]$. (Hint: show that $E\left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta}\right)\right] = 0$.)
10. Verify that $E\left[\left(\frac{\partial \ln f(X|\theta)}{\partial \theta}\right)^2\right] = -E\left[\left(\frac{\partial^2 \ln f(X|\theta)}{\partial \theta^2}\right)\right]$. (Hint: differentiate with respect to θ the equation $\int_{-\infty}^{\infty} \frac{\partial \ln f(x|\theta)}{\partial \theta} f(x|\theta) dx = 0$.)

11. The probability of obtaining a tail when flipping a coin can be $\pi = \frac{1}{2}$, $\pi = \frac{1}{3}$, or $\pi = \frac{2}{3}$. To estimate π , the coin is flipped three times and one head is obtained on the first flip and tails on the second and third flips. Find the maximum likelihood estimator of π .

12. A manufacturer produces needles for a sewing machine in 5 units per parcel. The parcels are in boxes of 120 units. The manufacturer guarantees that only one out of 100 parcels is defective; however, the owner of a store thinks that at least 4 parcels out of 100 are defective. To solve the controversy, the manufacturer randomly chooses 18 boxes and checks the number of defective parcels. The results follow:

Number of defective parcels: 3, 1, 1, 2, 4, 2, 0, 1, 4, 1, 6, 2, 2, 3, 1, 4, 4, 2

Who is more likely to be right, the manufacturer or the store owner?

13. Consider a random sample of size n from a geometric distribution.
- (a) Find the method of moments estimator of π .
 (b) Find the maximum likelihood estimator of π .
 (c) Use the results from (a) and (b) to compute the method of moments and maximum likelihood estimates from the sample $\{8, 1, 2, 0, 0, 0, 2, 1, 3, 3\}$, which represents the number of Bernoulli trials that resulted in failure before the first success in 10 experiments.

14. The following random samples $\mathbf{X}=(x_1, \dots, x_7)$ and $\mathbf{Y}=(y_1, \dots, y_{10})$ are drawn from $Pois(\lambda)$ and $Pois(2\lambda)$, respectively:

$\mathbf{X} \sim Pois(\lambda)$	4	2	5	7	3	4	3
$\mathbf{Y} \sim Pois(2\lambda)$	6	10	1	6	3	5	5

- (a) Derive the maximum likelihood estimator of λ and calculate its variance.
 - (b) Compute the maximum likelihood estimate of λ and its variance using the two random samples given.
15. Find the maximum likelihood estimator for μ if samples of size n are taken from a $N(\mu, \sigma = \sqrt{\mu})$ distribution.
- (a) Use the maximum likelihood estimator to calculate the maximum likelihood estimate that results from the sample
4.37, 9.30, 1.67, 1.25, 4.30, 6.97, 2.68, 5.49, 4.36, 4.46.
 - (b) Plot the log-likelihood function versus μ for values between 4 and 5.2.
16. Consider a random sample of size n from a distribution with a density function given by
- $$f(x) = \theta \left(\frac{1}{x}\right)^{\theta+1}, \quad x \geq 1, \theta > 1.$$
- (a) Find the method of moments and the maximum likelihood estimators of θ .
 - (b) Find the method of moments and maximum likelihood estimates of θ for the sample $\{3, 4, 2, 1.5, 4, 2, 3, 2, 4, 2\}$.
 - (c) Set the seed equal to 11, and generate 20,000 values from $f(x)$ using $\theta = 32$. Compute the method of moments and maximum likelihood estimates of θ using the generated values.
17. Consider the density function
- $$f(x) = (\theta + 1)(1 - x)^\theta, \quad 0 \leq x \leq 1, \theta > 0.$$
- (a) Find the maximum likelihood estimator of θ for a random sample of size n .
 - (b) Set the seed equal to 3, and generate 20,000 values from $f(x)$ when $\theta = 5$. Calculate the maximum likelihood estimate of θ from the generated values.
 - (c) How close is the maximum likelihood estimate in (b) to $\theta = 5$?
18. Consider the density function
- $$f(x) = \frac{3}{\lambda} x^2 e^{-x^3/\lambda}, \quad x > 0, \quad \lambda > 0.$$

- (a) Find the maximum likelihood estimator of λ for a random sample of size n .

- (b) Verify that the maximum likelihood estimator is unbiased, consistent, and efficient.
- (c) Find the method of moments estimator of λ for a random sample of size n .
19. Consider an exponential distribution with mean θ and the following estimators of θ :
- $$\hat{\theta}_1 = X_1, \quad \hat{\theta}_2 = \frac{X_1 + X_2}{2}, \quad \hat{\theta}_3 = \bar{X}, \quad \hat{\theta}_4 = \min\{X_1, X_2, X_3\}.$$
- (a) Find the mean and variance of each estimator.
- (b) Are any of the estimators efficient?
- (c) Which estimator is the MLE?
- (d) Let X be an exponential random variable with mean $\theta + 2$. Which estimator is an unbiased estimator of θ ?
20. Consider a random sample of size n from a population of size N , where the items in the population are sequentially numbered from 1 to N .
- (a) Derive the method of moments estimator of N .
- (b) Derive the maximum likelihood estimator of N .
- (c) What are the method of moments and maximum likelihood estimates of N for this sample of size 7: $\{2, 5, 13, 6, 15, 9, 21\}$?
21. The lifetime of a particular resistor follows an exponential distribution with parameter λ . The manufacturer claims the mean life of the resistor is 6 years. A distributor of the resistor is suing the manufacturer for excess warranty claims, saying that the mean life of the resistor is a mere 4 years. To resolve the issue, an accelerated test of the predicted lifetimes of 20 resistors is undertaken, yielding the following values:
- | | | | | | | | | | |
|------|------|-------|-------|------|------|------|-------|------|------|
| 3.70 | 1.76 | 3.63 | 15.73 | 5.85 | 0.20 | 9.87 | 14.55 | 0.43 | 2.46 |
| 0.45 | 5.09 | 10.53 | 12.41 | 3.19 | 3.41 | 3.80 | 1.66 | 0.40 | 1.10 |
- (a) The judge calls you as an expert witness to determine the validity of the suit. What do you tell the judge?
- (b) What value of λ maximizes the probability for values reported from the experiment?
- (c) Graph the log-likelihood function versus λ values ranging from 0.05 to 0.51.
22. Data frame `birthwt` from the MASS package has 10 variables recorded for each of 189 babies born at a U.S. hospital. The variable `low` takes the value 1 when the baby weighs less than 2.5 kg and 0 otherwise.
- (a) What distribution would be appropriate to model the values in `low`?
- (b) How many babies had birth weights less than 2.5 kg?
- (c) Find the maximum likelihood estimate for the parameter of the distribution selected in (a).

(d) Interpret the mle found in part (c).

23. In 1876, Charles Darwin had his book *The Effect of Cross- and Self-Fertilization in the Vegetable Kingdom* published. Darwin planted two seeds, one obtained by cross-fertilization and the other by auto-fertilization, in two opposite but separate locations of a pot. Self-fertilization, also called autogamy or selfing, is the fertilization of a plant with its own pollen. Cross-fertilization, or allogamy, is the fertilization with pollen of another plant, usually of the same species. Darwin recorded the plants' heights in inches. The data frame **FERTILIZE** from the **PASWR2** package contains the data from this experiment.

Cross-fert	23.5	12.0	21.0	22.0	19.1	21.5	22.1	20.4
	18.3	21.6	23.3	21.0	22.1	23.0	12.0	
Self-fert	17.4	20.4	20.0	20.0	18.4	18.6	18.6	15.3
	16.5	18.0	16.3	18.0	12.8	15.5	18.0	

- (a) Create a variable **DD** defined as the difference between the variables **cross** and **self**.
- (b) Perform an exploratory analysis of **DD** to see if **DD** might follow a normal distribution.
- (c) Use the function **fitdistr()** found in the **MASS** package to obtain the maximum likelihood estimates of μ and σ if **DD** did follow a normal distribution.
- (d) Verify that the results from (c) are the sample mean and the uncorrected sample standard deviation of **DD**.

24. The lognormal distribution has the following density function:

$$g(w) = \frac{1}{w\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(\ln w - \mu)^2}, \quad w \geq 0, \quad -\infty < \mu < \infty, \quad \sigma \neq 0$$

where $\ln(W) \sim N(\mu, \sigma)$. The mean and variance of W are, respectively,

$$E[W] = e^{\mu + \frac{\sigma^2}{2}} \quad \text{and} \quad \text{Var}[W] = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1).$$

Find the maximum likelihood estimators for $E[W]$ and $\text{Var}[W]$.

- 25. Consider the variable **brain** from the **Animals** data frame in the **MASS** package.
 - (a) Estimate with maximum likelihood techniques the mean and variance of **brain**. Specifically, use the R function **fitdistr()** with a lognormal distribution.
 - (b) Suppose that **brain** is a lognormal variable; then the log of this variable is normal. To check this assertion, plot the cumulative distribution function of **brain** versus a lognormal cumulative distribution function. In another plot, represent the cumulative distribution function of **log-brain** versus a normal cumulative distribution function. Is it reasonable to assume that **brain** follows a lognormal distribution?
 - (c) Find the mean and standard deviation of **brain** assuming a lognormal distribution.
 - (d) Repeat this exercise without the dinosaurs. Comment on the changes in the mean and variance estimates.

26. The data in **GD** available in the **PASWR2** package are the times until failure in hours for a particular electronic component subjected to an accelerated stress test.

- (a) Find the method of moments estimates of α and λ if the data come from $a\Gamma(\alpha, \lambda)$ distribution.
- (b) Create a density histogram of times until failure. Superimpose a gamma distribution using the estimates from part (a) over the density histogram.
- (c) Find the maximum likelihood estimates of α and λ if the data come from $a\Gamma(\alpha, \lambda)$ distribution by using the function **fitdistr()** from the **MASS** package.
- (d) Create a density histogram of times until failure. Superimpose a gamma distribution using the estimates from part (c) over the density histogram.
- (e) Plot the cumulative distribution for time until failure. Superimpose the theoretical cumulative gamma distribution using both the method of moments and the maximum likelihood estimates of α and λ . Which estimates appear to model the data better?

27. The time a client waits to be served by the mortgage specialist at a bank has density function

$$f(x) = \frac{1}{2\theta^3} x^2 e^{-x/\theta} \quad x > 0, \theta > 0.$$

- (a) Derive the maximum likelihood estimator of θ for a random sample of size n .
- (b) Show that the estimator derived in (a) is unbiased and efficient.
- (c) Derive the method of moments estimator of θ .
- (d) If the waiting times of 15 clients are 6, 12, 15, 14, 12, 10, 8, 9, 10, 9, 8, 7, 10, 7, and 3 minutes, compute the maximum likelihood estimate of θ .

28. If the function

$$f(x|\theta) = kx^3 e^{-\frac{1}{\theta}x} \quad x \geq 0, \theta > 0$$

is a density function,

- (a) find k ;
- (b) derive the maximum likelihood estimator of θ for a random sample of size n ;
- (c) derive the method of moments estimator of θ for a random sample of size n ; and
- (d) show that the estimators from parts (b) and (c) are both unbiased and efficient.

29. Consider the function

$$f(x) = \frac{\theta}{x^2}, \quad x \geq \theta, \quad \theta > 0.$$

- (a) Verify that it is a density function.
- (b) Find the maximum likelihood estimators of θ and $\frac{1}{\theta}$ for random samples of size n .
- (c) Is the maximum likelihood estimator of θ unbiased?
- (d) Find the method of moments estimators of θ and $\frac{1}{\theta}$.

30. The lifetime (in days) of a new 100-watt fluorescent light bulb follows an exponential distribution with mean $\frac{1}{\lambda}$. The following data are the lifetimes of 109 light bulbs:

Time	Bubbles
[0, 50)	25
[50, 100)	19
[100, 150)	11
[150, 200)	8
[200, 250)	9
[250, 300)	7
[300, 450)	22
<u>[450, 1050)</u>	<u>8</u>

- (a) Find the maximum likelihood estimator of λ .
- (b) Graph the logarithm of the likelihood function versus the parameter λ and indicate the value of λ where the lifetime is maximized.

31. Consider the density function

$$f(x) = \frac{1}{\theta} x^{\frac{1-\theta}{\theta}}, \quad 0 < x < 1, \quad 0 < \theta < \infty.$$

- (a) Derive the maximum likelihood estimator of θ for a random sample of size n .
- (b) Derive the method of moments estimator of θ for a random sample of size n .
- (c) Show that the maximum likelihood estimator is unbiased.

32. Consider the density function

$$f(x) = \theta x^{\theta-1} \quad 0 \leq x \leq 1, \quad \theta > 0.$$

- (a) What distribution has this density function? Be sure to specify the parameter.
- (b) Find the maximum likelihood estimator of θ for random samples of size n .
- (c) Find the asymptotic variance of the maximum likelihood estimator.
- (d) Find the method of moments estimator of θ for a random sample of size n .
- (e) Calculate the maximum likelihood and method of moments estimates of θ for the sample $\{0.1, 0.7, 0.5, 0.85, 0.9\}$.

33. Consider the density function

$$f(x) = \theta \left(\frac{1}{x}\right)^{\theta+1}, \quad x \geq 1, \theta > 1.$$

- (a) Find the maximum likelihood estimator of θ for a random sample of size n .
- (b) Find the method of moments estimator of θ for a random sample of size n .
- (c) Calculate the maximum likelihood and method of moments estimates of θ using the sample values $\{2, 3, 2, 2.5, 1, 2, 2, 3, 1, 4, 6, 3, 4.4\}$.
- (d) Find the mean of the distribution.
- (e) Estimate the mean of the distribution using the maximum likelihood estimate of θ .

34. Consider the density function

$$f(x) = \begin{cases} 1 - \theta & \text{for } -\frac{1}{2} \leq x \leq 0 \text{ and} \\ 1 + \theta & \text{for } 0 < x \leq \frac{1}{2}. \end{cases}$$

- (a) Find the maximum likelihood estimator of θ for a random sample of size n .
 - (b) Show that the maximum likelihood estimator is unbiased and efficient.
- (Hint: Denote the number of observations as n_1 in the sample so that $0 < x_i \leq 1/2$.)

35. Consider the density function

$$f(x) = 3\pi\theta x^2 e^{-\theta\pi x^3}, \quad x \geq 0.$$

- (a) Set the seed equal to 102, and generate a random sample of size $n = 20,000$ with $\theta = 5$.
- (b) Find the sample mean and the sample variance of the random values generated in (a).
- (c) Create a density histogram of the simulated values from (a) and superimpose the density function over the density histogram.
- (d) Find the maximum likelihood estimate of θ .
- (e) Plot the logarithm of the likelihood function versus θ . Use values for θ from 0 to 15.

36. Consider the density function

$$f(x) = e^{-(x-\alpha)}, \quad -\infty < \alpha \leq x.$$

- (a) Find the maximum likelihood and method of moments estimators of α .
- (b) Are both estimators found in (a) asymptotically unbiased?

37. Set the seed equal to 8675, and generate 309 values from a $\beta(\alpha = 3, \beta = 2, A = 0, B = 1)$ distribution. Assume that these values are a random sample of size 309 from a β distribution with unknown parameters. Use maximum likelihood techniques to obtain estimates of α and β from this sample.

38. Consider a random sample of size n from an exponential distribution with **pdf**

$$f(x) = \frac{1}{\theta} e^{-\frac{x}{\theta}} \quad x \geq 0, \quad \theta > 0. \tag{7.69}$$

- (a) Find the MLE of θ .
- (b) Given the answer in part (a), what is the MLE of θ^2 ?

Chapter 8

Confidence Intervals

8.1 Introduction

In Chapter 7, techniques to find point estimators, such as the method of moments and maximum likelihood, were introduced as well as were criteria to evaluate the “goodness” of an estimator; however, even the most efficient unbiased estimator is not likely to estimate the population parameter exactly. Further, a point estimate provides no information about the precision or reliability of the estimate. Consequently, the construction of an **interval estimate** or **confidence interval** (*CI*), where the user can control the precision (width) of the interval as well as the reliability (confidence) that the true parameter will be found in the confidence interval, is desirable.

A $(1 - \alpha)$ confidence interval for a parameter θ , denoted $CI_{1-\alpha}(\theta)$, is constructed by first selecting a confidence level, denoted by $(1 - \alpha)$ and typically expressed as a percentage, $(1 - \alpha) \cdot 100\%$. The confidence level is simply a measure of the degree of reliability in the procedure used to construct the confidence interval. Typical confidence levels are 90%, 95%, or 99%. A confidence level of 99% implies that 99% of all samples would provide confidence intervals that would contain θ . Clearly, it is desirable to have a high degree of reliability. Unfortunately, with increased reliability, the width of the confidence interval increases. So, the goal is to construct a confidence interval with a width the practitioner finds useful while maintaining a degree of reliability that is as high as possible. The relationship between the width and confidence level in a confidence interval will become clearer once some actual confidence interval formulas are examined. The confidence interval has two limits, a lower limit denoted by $L(\mathbf{X})$ and an upper limit denoted by $U(\mathbf{X})$. The **confidence level** is defined as $\mathbb{P}(\theta \in [L(\mathbf{X}), U(\mathbf{X})])$. That is, an interval should be constructed such that

$$\mathbb{P}(L(\mathbf{X}) \leq \theta \leq U(\mathbf{X})) = 1 - \alpha. \quad (8.1)$$

It is important to note that the interval $[L(\mathbf{X}), U(\mathbf{X})]$ is a random interval since it depends on the random variables of \mathbf{X} . After a sample is obtained and values for $[L(\mathbf{X}), U(\mathbf{X})]$ are calculated, the probability that the parameter θ will be included in the interval $[L(\mathbf{x}), U(\mathbf{x})]$ is either 0 or 1, depending, of course, on whether θ is between the lower limit $L(\mathbf{x})$ and the upper limit $U(\mathbf{x})$. Note that \mathbf{X} changes to an \mathbf{x} once there are values, x_i , for the random variables, X_i . The probability the parameter θ is contained in the random interval $[L(\mathbf{X}), U(\mathbf{X})]$ from (8.1) is $(1 - \alpha)$. Once the values for the random variables are observed, (8.1) is written as

$$CI_{1-\alpha}(\theta) = [L(\mathbf{x}), U(\mathbf{x})], \quad (8.2)$$

which is called a $(1 - \alpha)$ confidence interval. Consequently, it makes no sense to talk about a $(1 - \alpha)$ probability interval. Frequently, the confidence level is expressed as a percentage, and the interval is often called a $(1 - \alpha) \cdot 100\%$ confidence interval. A $(1 - \alpha) \cdot 100\%$ confidence interval is typically interpreted as “One is $(1 - \alpha) \cdot 100\%$ confident θ is contained in the interval $[L(\mathbf{x}), U(\mathbf{x})]$.” The word *confidence* in such statements applies to the procedure

used to construct the interval, not the interval itself. That is, if there were an infinite number of samples, $(1 - \alpha) \cdot 100\%$ of them would contain θ .

Confidence intervals of the form $[L(\mathbf{x}), U(\mathbf{x})]$ are referred to as two-sided confidence intervals; however, some applications will only require a single bound. For example, only a lower confidence bound on the mean shear strength of an aluminum tube is required to ensure the minimum design specification for a top tube of a bicycle is met. Likewise, only an upper confidence bound on the mean level of NO_3 in potable water is required to ensure the maximum allowable limit is not exceeded. One-sided confidence intervals take the form

$$\mathbb{P}(L(\mathbf{X}) \leq \theta) = 1 - \alpha \quad \text{or} \quad \mathbb{P}(\theta \leq U(\mathbf{X})) = 1 - \alpha,$$

depending on whether the confidence interval is an upper confidence interval, $[L(\mathbf{x}), \infty)$, or a lower confidence interval, $(-\infty, U(\mathbf{x})]$, respectively. Unless otherwise specified, a confidence interval will refer to a two-sided confidence interval.

There are several techniques used to obtain both one-sided and two-sided confidence intervals. One of the more popular methods for constructing confidence intervals uses pivotal quantities. A random variable $Q(\mathbf{X}; \theta)$ is a **pivotal quantity** or **pivot** if the distribution of Q is independent of the parameter θ . A method of constructing confidence intervals using pivots is introduced in Section 8.2.1 and is used to derive most of the confidence interval formulas in this chapter.

8.2 Confidence Intervals for Population Means

There are many different averages that will be of interest to the statistician. A reasonable range where the true average ought to lie is invaluable in making good decisions. Several different contexts are examined in this section, all assuming that the underlying population is normal. Confidence intervals for a single mean when the population variance is both known and unknown are derived as are those for differences of means and the mean of the differences with both known and unknown population variances with various sample sizes and underlying assumptions. The most appropriate procedure depends on those assumptions' truth.

8.2.1 Confidence Interval for the Population Mean When Sampling from a Normal Distribution with Known Population Variance

A random sample of size n is taken from a normal distribution with mean μ and variance σ^2 . To obtain a confidence interval for μ , recall that the sampling distribution for the sample mean is $\bar{X} \sim N(\mu, \sigma/\sqrt{n})$. Using the sampling distribution of \bar{X} , create the pivotal quantity

$$Q(\mathbf{X}; \mu) = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1). \quad (8.3)$$

To obtain a confidence interval with a $(1 - \alpha)$ confidence level, construct a region such that the area between $z_{\alpha/2}$ and $z_{1-\alpha/2}$ is $(1 - \alpha)$, as shown in Figure 8.1 on the facing page. In other words,

$$\mathbb{P}\left(z_{\alpha/2} \leq \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \leq z_{1-\alpha/2}\right) = 1 - \alpha. \quad (8.4)$$

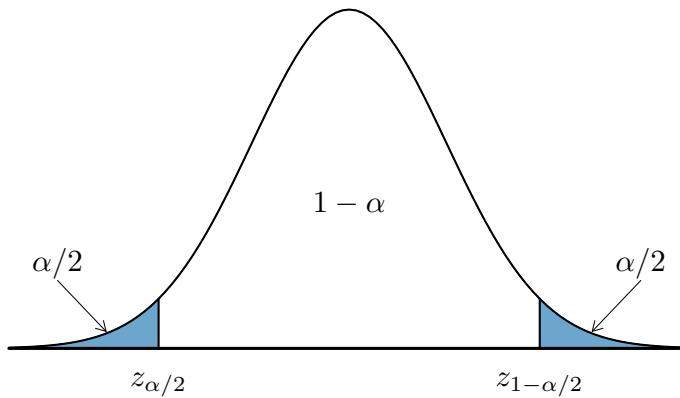


FIGURE 8.1: Standard normal distribution with an area of $\alpha/2$ in each tail

Multiply both sides of (8.4) by σ/\sqrt{n} , to obtain

$$\mathbb{P}\left(z_{\alpha/2}\frac{\sigma}{\sqrt{n}} \leq \bar{X} - \mu \leq z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha.$$

Subtract \bar{X} from both sides, multiply both sides by -1 , and rearrange the inequalities, to get

$$\mathbb{P}\left(\bar{X} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha.$$

Consequently, the $(1-\alpha)$ confidence interval for μ , when sampling from a normal distribution with known variance, is given by

$$\left[\bar{x} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}, \bar{x} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right],$$

or, equivalently, by recognizing that $z_{\alpha/2} = -z_{1-\alpha/2}$, write the standard form as

$$CI_{1-\alpha}(\mu) = \left[\bar{x} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}, \bar{x} + z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right]. \quad (8.5)$$

Note that \bar{X} in the probability statement changes to \bar{x} in the confidence interval formula.

To obtain a one-sided (either upper or lower) confidence interval in a symmetric distribution, proceed in a similar fashion. That is, write

$$\mathbb{P}\left(-z_{1-\alpha} \leq \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}\right) = 1 - \alpha \quad \text{or} \quad \mathbb{P}\left(\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \leq z_{1-\alpha}\right) = 1 - \alpha$$

and rearrange the quantities inside the probability statements to obtain

$$\mathbb{P}\left(\mu \leq \bar{X} + z_{1-\alpha}\frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha \quad \text{or} \quad \mathbb{P}\left(\bar{X} - z_{1-\alpha}\frac{\sigma}{\sqrt{n}} \leq \mu\right) = 1 - \alpha.$$

Thus,

$$LCI_{1-\alpha}(\mu) = \left(-\infty, \bar{x} + z_{1-\alpha}\frac{\sigma}{\sqrt{n}}\right] \quad \text{or} \quad UCI_{1-\alpha}(\mu) = \left[\bar{x} - z_{1-\alpha}\frac{\sigma}{\sqrt{n}}, \infty\right).$$

Note that a one-sided confidence interval can be obtained from a two-sided confidence interval by simply changing the $z_{1-\alpha/2}$ value to a $z_{1-\alpha}$ value and replacing the lower or upper bound with $-\infty$ or ∞ , respectively, depending on whether a lower or an upper confidence interval is desired.

Example 8.1 Write a function that will generate 100 samples, each of size 36, from a $N(100, 18)$ distribution. For each of the 100 samples of size 36, calculate a 95% confidence interval for the population mean. Graph the confidence intervals, highlighting those that do not contain $\mu = 100$. Finally, determine how many of the 100 intervals contain the population mean, $\mu = 100$. This number is the simulated confidence level.

Solution: R Code 8.1 creates the function `norsim()` which graphs the simulated confidence intervals and determines the simulated confidence level.

R Code 8.1

```
> norsim <- function(sims = 100, n = 36, mu = 100, sigma = 18,
+   conf.level = 0.95) {
+   alpha <- 1 - conf.level
+   CL <- conf.level * 100
+   ll <- numeric(sims)
+   ul <- numeric(sims)
+   for (i in 1:sims) {
+     xbar <- mean(rnorm(n, mu, sigma))
+     ll[i] <- xbar - qnorm(1 - alpha/2) * sigma/sqrt(n)
+     ul[i] <- xbar + qnorm(1 - alpha/2) * sigma/sqrt(n)
+   }
+   notin <- sum((ll > mu) + (ul < mu))
+   percentage <- round((notin/sims) * 100, 2)
+   SCL <- 100 - percentage
+   plot(ll, type = "n", ylim = c(min(ll), max(ul)), xlab = " ",
+     ylab = " ")
+   for (i in 1:sims) {
+     low <- ll[i]
+     high <- ul[i]
+     if (low < mu & high > mu) {
+       segments(i, low, i, high)
+     } else if (low > mu & high > mu) {
+       segments(i, low, i, high, col = "red", lwd = 5)
+     } else {
+       segments(i, low, i, high, col = "blue", lwd = 5)
+     }
+   }
+   abline(h = mu)
+   cat(SCL, "\b% of the random confidence intervals contain Mu =",
+     mu, "\b.", "\n")
+ }
```

The results from setting the seed to 10 are shown in R Code 8.2 on the facing page and the resulting graph is depicted in Figure 8.2 on the next page. Note that this is a simulation, and consequently, the number of confidence intervals that contain $\mu = 100$ will vary and will not always equal the expected 95. A more general function that can be used to generate

random data and subsequently to create confidence intervals is the function `cisim()` from the `PASWR2` package.

R Code 8.2

```
> set.seed(10)
> norsim(sims = 100, n = 36, mu = 100, sigma = 18, conf.level = 0.95)

95 % of the random confidence intervals contain Mu = 100 .
```

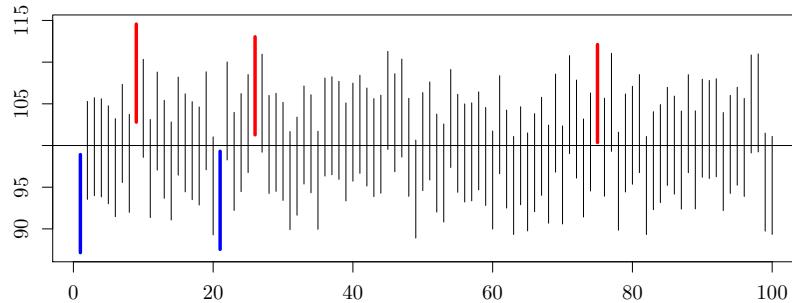


FIGURE 8.2: Simulated confidence intervals for the population mean when sampling from a normal distribution with known variance



Example 8.2 A random sample of size 30 is taken from a normal distribution with unknown mean μ and standard deviation $\sigma = 2.5$. Given that $\sum_{i=1}^{30} x_i = 77$, calculate a 95% confidence interval for the population mean.

Solution: First, determine \bar{x} :

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{77}{30} = 2.5667.$$

Since the sample was taken from a normal distribution with known variance, it is permissible to write

$$\mathbb{P}\left(\bar{X} - z_{0.975} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + z_{0.975} \frac{\sigma}{\sqrt{n}}\right) = 0.95.$$

A 95% confidence interval for the mean using (8.5) is written as

$$CI_{0.95}(\mu) = \left[2.5667 - (1.96) \frac{2.5}{\sqrt{30}}, 2.5667 + (1.96) \frac{2.5}{\sqrt{30}}\right].$$

In other words, one can be 95% confident that the mean, μ , will be found in the interval $CI_{0.95}(\mu) = [1.67, 3.46]$. It is important to note that the sample mean ($\bar{x} = 2.5667$) is the center point of this interval; however, this will only be the case in symmetric distributions.



Example 8.3 ▷ Confidence Interval for μ : Grocery Spending ◁ The consumer expenditure survey, created by the U.S. Department of Labor, was administered to 30 households in Watauga County, North Carolina, to see how the cost of living in Watauga

County with respect to total dollars spent on groceries compares with that of other counties. The amount of money each household spent per week on groceries is given in Table 8.1 and stored in the data frame **GROCERY**.

- Construct a 97% confidence interval for the true mean weekly grocery expenditure for Watauga County households. Historical records indicate that the variance for grocery expenditure per household in Watauga County is 900 dollars².
- A grocery chain is considering building a new grocery store in Watauga County. However, it will only do so if it is 99% confident the average amount spent on groceries each week is at least \$105. Does a $UCI_{0.99}(\mu)$ include \$105? If so, what does that imply?

Table 8.1: Weekly spending in dollars (**GROCERY**)

90.74	104.02	85.64	134.71	108.85	142.19	162.87	138.2	98.73	98.18
139.84	159.69	147.03	151.16	105.68	116.93	97.46	146.64	90.92	134.54
110.82	109.90	106.74	122.10	152.28	136.01	126.00	108.69	135.06	57.38

Solution: The answers are as follows:

- (a) Before using (8.5), the confidence interval formula for μ with known σ on page 455, it is necessary to verify that the assumption of normality is satisfied. To do this, create a normal quantile-quantile plot using the `qqnorm()` function as follows:

```
> with(data = GROCERY, qqnorm(amount))
> with(data = GROCERY, qqline(amount))
```

The resulting normal quantile-quantile plot is shown in Figure 8.3 on the next page. Note that the plotted values fall relatively close to the plotted line, indicating the assumption of normality is reasonable. Consequently, one decides the assumption for using (8.5) on page 455 is satisfied and continues by finding the sample mean:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{3619}{30} = 120.6333.$$

Using the historical value of 900 for σ^2 , the 97% confidence interval is given by

$$\begin{aligned} CI_{0.97}(\mu) &= \left[\bar{x} - z_{0.985} \frac{\sigma}{\sqrt{n}}, \bar{x} + z_{0.985} \frac{\sigma}{\sqrt{n}} \right] \\ &= \left[120.6333 - (2.1701) \frac{\sqrt{900}}{\sqrt{30}}, 120.6333 + (2.1701) \frac{\sqrt{900}}{\sqrt{30}} \right]. \end{aligned}$$

In other words, one can be 97% confident that the mean grocery spending will be found in the interval [108.75, 132.52] dollars.

To do this calculation with R, enter

Normal Q-Q Plot

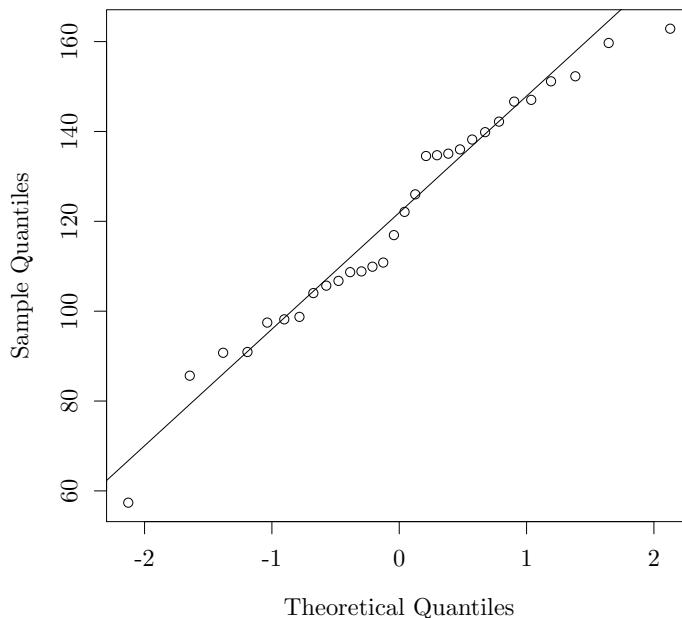


FIGURE 8.3: Quantile-quantile (normal distribution) plot of weekly monies spent on groceries for 30 randomly selected Watauga households

```

> xbar <- mean(GROCERY$amount)
> z <- qnorm(0.985)
> CI1 <- xbar + c(-1, 1) * z * sqrt(900)/sqrt(30)
> CI1
[1] 108.7473 132.5194

> # Or using the z.test function
> CI2 <- z.test(GROCERY$amount, sigma.x = sqrt(900), n.x = 30,
+                 conf.level = .97)$conf
> CI2
[1] 108.7473 132.5194
attr(", "conf.level")
[1] 0.97

```

(b) Part (a) already verified that the data follow a normal distribution, so one calculates the one-sided 99% confidence interval as

$$\begin{aligned}
UCI_{0.99}(\mu) &= \left[\bar{x} - z_{0.99} \frac{\sigma}{\sqrt{n}}, \infty \right) \\
&= \left[120.6333 - 2.3263 \frac{30}{\sqrt{30}}, \infty \right) \\
&= [107.89, \infty).
\end{aligned}$$

This interval does not include \$105, and its lower bound is above \$105, so the grocery

chain can be more than 99% confident the mean grocery spending is greater than \$105. To compute the confidence interval with `z.test()`, enter

```
> z.test(GROCERY$amount, sigma.x = sqrt(900), n.x = 30, conf.level = 0.99,
+         alternative = "greater")$conf
[1] 107.8914      Inf
attr(,"conf.level")
[1] 0.99
```



8.2.1.1 Determining Required Sample Size

Larger sample sizes generally result in narrower confidence intervals. Researchers will often desire a confidence interval not to exceed a specific width at some predetermined level of significance (one that usually has some practical significance to their research). The problem addressed in this section is how to determine the minimum required sample size to be within a given distance of μ when estimating the population mean with known variance, σ^2 . To start, recall the probability statement about μ for normal distributions with known variance in (8.6). Use this equation when working with normal populations ($N(\mu, \sigma)$), as well as with various other distributions, provided the sample size is sufficiently large:

$$\mathbb{P}\left(\bar{X} - z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha, \quad (8.6)$$

which implies

$$\mathbb{P}\left(|\bar{X} - \mu| \leq z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha,$$

where $|\bar{X} - \mu|$ is the **error of estimation**. In general, the error of estimation is a measure of the goodness of the estimate. Many texts refer to the error of estimation as the **margin of error** or the **bound on the error**. Denote this quantity by B . If one assumes the maximum error is

$$B = |\bar{x} - \mu| = z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}},$$

one can solve for n as shown in (8.7):

$$n = \frac{(z_{1-\alpha/2})^2 \sigma^2}{(B)^2} = \left(\frac{z_{1-\alpha/2} \sigma}{B}\right)^2. \quad (8.7)$$

Consequently, any time a confidence level is specified and the value of σ is known, one can determine the required sample size, n , to be within the maximum error, B , that is acceptable.

Example 8.4 Determine the required sample size to estimate the true value of μ within ± 0.02 with a confidence level of 95% when sampling from a normal distribution with $\sigma = 0.1$.

Solution: To determine the required sample size, use (8.7) as follows:

$$n = \frac{(z_{1-\alpha/2})^2 \sigma^2}{(B)^2} = \frac{(1.96)^2 (0.1)^2}{(0.02)^2} = 96.04.$$

In order to have a confidence of at least $1 - \alpha$, one always takes the ceiling of n ; therefore, the required sample size n to estimate the population mean with a 0.95 confidence level

so that the margin of error is no more than 0.02 is $n = 97$. R Code 8.3 uses the function `nsize()` from the `PASWR2` package to compute the required sample size.

R Code 8.3

```
> nsize(b = 0.02, sigma = 0.1, conf.level = 0.95, type = "mu")
```

The required sample size (n) to estimate the population mean with a 0.95 confidence interval so that the margin of error is no more than 0.02 is 97 .



Example 8.5 Suppose a random sample of size n from a normal distribution with unknown mean μ and standard deviation $\sigma = 5$ is taken. Calculate the minimum sample size so that one can be 95% confident the interval $[\bar{x} - 1, \bar{x} + 1]$ contains the true value of μ .

Solution: Given that the sample was taken from a normal distribution with known variance, one can write

$$\mathbb{P}\left(\bar{X} - (1.96)\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + (1.96)\frac{\sigma}{\sqrt{n}}\right) = 0.95.$$

Since one needs to be 95% confident the interval $[\bar{x} - 1, \bar{x} + 1]$ contains μ , write

$$\mathbb{P}(\bar{X} - 1 \leq \mu \leq \bar{X} + 1) = 0.95,$$

set $1.96\frac{\sigma}{\sqrt{n}} = 1$, and solve for n given that $\sigma = 5$:

$$n = (1.96)^2(5)^2 = 96.0365.$$

Since a sample of 96.0365 is impossible, take the ceiling of n to make sure the confidence is at or above the specified level. Consequently, the minimum sample size to be at least 95% confident the interval $[\bar{x} - 1, \bar{x} + 1]$ contains μ is $n = 97$ when $\sigma = 5$. R Code 8.4 uses the function `nsize()` from the `PASWR2` package to compute the required sample size.

R Code 8.4

```
> nsize(b = 1, sigma = 5, conf.level = 0.95, type = "mu")
```

The required sample size (n) to estimate the population mean with a 0.95 confidence interval so that the margin of error is no more than 1 is 97 .



Example 8.6 \triangleright **Sample Size: Defective Containers** \triangleleft In a company that produces glass containers, the probability of producing a defective container is $\pi = 0.03$, and the probability of obtaining a functional container is $(1 - \pi) = 0.97$. Determine how many containers need to be manufactured to guarantee that at least 100 containers are defective with a probability of at least 0.95.

Solution: Three solutions are presented for this problem. The first is the exact answer based on a negative binomial distribution and requires the use of a computer. The second is an approximation that can be used in the absence of a computer. The third is the exact answer from a ($\text{Bin}(n, 0.03)$), the distribution approximated in (b).

- (a) Let X be the number of failures prior to the $r = 100^{\text{th}}$ success (defective container). The distribution of X is $NB(100, 0.03)$. The problem requests $\mathbb{P}(X = x|r = 100, \pi = 0.03) \geq 0.95$. That is, one must find the number x of non-defective containers to guarantee the probability is at least 0.95 upon obtaining the 100^{th} defective container. R Code 8.5 indicates the total number of containers that must be manufactured to guarantee 100 are defective with a probability over 0.95 is 3891.

R Code 8.5

```
> f <- 0      # f = number of failures
> p <- 0      # p = probability
> s <- 100    # s = number of successes
> while(p < 0.95){
+   f <- f + 1
+   p <- pnbinom(f, s, 0.03)
+ }
> ans <- c(f + s, p)  # f + s = Containers
> names(ans) <- c("Containers", "Probability")
> ans

Containers  Probability
3891.0000000  0.9500444
```

- (b) Let the random variable X represent the number of defective containers. The distribution of X is $\text{Bin}(n, \pi = 0.03)$. Consequently,

$$E[X] = n\pi = 0.03n \quad \text{and} \quad \text{Var}[X] = n\pi(1 - \pi) = 0.0291n.$$

If it is assumed n is sufficiently large and the production of each container is independent, one can approximate the distribution of X using a normal distribution where

$$\mathbb{P}(X \geq 100) = 0.95.$$

Equivalently,

$$\mathbb{P}\left(\frac{X - n\pi}{\sqrt{n\pi(1 - \pi)}} \geq \frac{100 - n\pi}{\sqrt{n\pi(1 - \pi)}}\right) = \mathbb{P}\left(Z \geq \frac{100 - 0.03n}{\sqrt{0.0291n}}\right) = 0.95.$$

Note that $\mathbb{P}(Z \geq -z_{1-\alpha}) = 0.95 \Rightarrow -z_{1-\alpha} = -z_{0.95} = -1.6449$, and solve the equation

$$\frac{100 - 0.03n}{\sqrt{0.0291n}} = -1.6449, \tag{8.8}$$

which is equivalent to solving

$$0.0009n^2 - 6.078731n + 100^2 = 0. \tag{8.9}$$

The solutions to (8.9) are $n = 2835.3077$ or $n = 3918.8379$; however, the value 2835.3077 is not acceptable since it does not satisfy (8.8). Consequently, the number of containers the factory needs to manufacture to be 95% confident of getting at least 100 defective containers is 3919. R Code 8.6 uses the function `polyroot()` to find the roots of (8.9). Note that the values for the argument `z` is a vector of polynomial coefficients given in increasing order.

R Code 8.6

```
> roots <- polyroot(z = c(100^2, -6.078731, 9e-04))
> roots

[1] 2835.308-0i 3918.838+0i

> Re(roots)

[1] 2835.308 3918.838

> r1 <- Re(roots[1]) # Real root
> r2 <- Re(roots[2]) # Real root
> c(r1, r2)

[1] 2835.308 3918.838

> (100 - 0.03 * r1)/sqrt(0.0291 * r1) # This does not equal -1.645
[1] 1.64485

> (100 - 0.03 * r2)/sqrt(0.0291 * r2) # This does equal -1.645
[1] -1.64485
```

- (c) Let the random variable X again represent the number of defective containers. The distribution of X is $\text{Bin}(n, \pi = 0.03)$. The $\mathbb{P}(X \geq 100) \geq 0.95$ is solved in R Code 8.7. Remember that $\mathbb{P}(X \geq 100) = 1 - \mathbb{P}(X \leq 99)$. R Code 8.7 indicates the total number of containers that must be manufactured to guarantee 100 are defective with a probability over 0.95 is 3891 when using the binomial random variable. This agrees with the answer that was found when modeling the number of defective containers obtained with a negative binomial random variable.

R Code 8.7

```
> n <- 0      # Number of containers
> p <- 0      # Probability
> while(p < 0.95) {
+   n <- n + 1
+   p <- 1 - pbinom(99, n, 0.03)
+ }
> ans <- c(n, p)
```

```
> names(ans) <- c("Containers", "Probability")
> ans

Containers  Probability
3891.000000  0.9500444
```



The confidence intervals discussed in the remainder of this chapter are commonly used confidence intervals based, for the most part, on the normal distribution. When constructing confidence intervals, if historical evidence does not support normality or the text narrative does not explicitly specify the sample information was collected from a normal distribution, one should not blindly use techniques that require the normality assumption! Checking normality assumptions graphically with normal quantile-quantile plots as discussed in Section 4.3.7 on page 301 should become a habit.

8.2.2 Confidence Interval for the Population Mean When Sampling from a Normal Distribution with Unknown Population Variance

Suppose a random sample of size n is taken from a normal distribution with unknown mean μ and unknown variance σ^2 . To construct a confidence interval for μ , use the pivotal quantity

$$Q(\mathbf{X}; \mu) = \frac{\sqrt{n}(\bar{X} - \mu)}{S} \sim t_{n-1}.$$

Operating in a similar fashion to the derivation of the confidence interval for μ , using (8.3) from Section 8.2.1, one obtains the interval

$$CI_{1-\alpha}(\mu) = \left[\bar{x} - t_{1-\alpha/2;n-1} \frac{s}{\sqrt{n}}, \bar{x} + t_{1-\alpha/2;n-1} \frac{s}{\sqrt{n}} \right]. \quad (8.10)$$

Example 8.7 A random sample of size 12 is taken from a population that follows a $N(\mu, \sigma)$ distribution where the value for σ is unknown. Given:

$$\sum_{i=1}^{12} x_i = 61.9, \quad \text{and} \quad \sum_{i=1}^{12} x_i^2 = 450,$$

determine a 90% confidence interval for the population mean.

Solution: First find the sample mean and the sample variance.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{61.9}{12} = 5.1583, \text{ and}$$

$$s^2 = \frac{\sum_{i=1}^n x_i^2 - n\bar{x}^2}{n-1} = \frac{450 - (12)(5.1583)^2}{12-1} = 11.8817.$$

The sample standard deviation is $s = 3.447$ and $t_{0.95;11} = 1.7959$. Using (8.10),

$$CI_{0.9}(\mu) = \left[5.1583 - (1.7959) \frac{3.447}{\sqrt{12}}, 5.1583 + (1.7959) \frac{3.447}{\sqrt{12}} \right] = [3.3713, 6.9453].$$

One is 90% confident the population mean lies in [3.3713, 6.9453]. R Code 8.8 computes the answers using (8.10) as well as using the function `tsum.test()` from `PASWR2`.

R Code 8.8

```
> n <- 12
> xbar <- 61.9/n
> s <- ((450 - n * (xbar)^2)/(n - 1))^(1/2)
> ct <- qt(0.95, n - 1)
> c(n, xbar, s, ct)

[1] 12.000000 5.158333 3.446990 1.795885

> CI <- c(xbar + c(-1, 1) * ct * s/sqrt(n))
> CI

[1] 3.371319 6.945347

> # Or using tsum.test()
> tsum.test(mean.x = xbar, s.x = s, n.x = n, conf.level = 0.9)$conf

[1] 3.371319 6.945347
attr("conf.level")
[1] 0.9
```



Example 8.8 ▷ Confidence Interval for μ : House Prices ◁ Estimate the mean house price for three-bedroom/two-bath houses in Watauga County, North Carolina. A random sample of 14 three-bedroom/two-bath houses was taken from the Watauga County Multiple Listing Service real estate listings (2003), and the results are reported in Table 8.2 and stored in the data frame `HOUSE`. Calculate a 95% confidence interval for the average price of a three bedroom/two bath house in this county.

Table 8.2: House prices (in thousands of dollars) for three-bedroom/two-bath houses in Watauga County, North Carolina (`HOUSE`)

Neighborhood	Price	Neighborhood	Price
Valley Crucis	184.9	Blowing Rock	279.5
Valley Crucis	160.0	Valley Crucis	294.9
Valley Crucis	298.0	Blowing Rock	324.5
Blowing Rock	269.9	Blowing Rock	226.0
Parkway	189.9	Valley Crucis	329.9
Blowing Rock	229.9	Green Valley	199.9
Cove Creek	175.0	Park Valley	133.9

Solution: Before using the confidence interval formula in (8.10), one needs to verify the assumption of normality is satisfied. Consequently, a normal quantile-quantile plot for the values reported in Table 8.2 on the preceding page was constructed with the R functions `qqnorm()` and `qqline()` and is shown in Figure 8.4 on the next page. Since the points in Figure 8.4 fall relatively close to the straight line, it is decided that the normality assumption for using (8.10) is satisfied. Thus, continue by calculating the sample mean as

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{3296.2}{14} = 235.4429$$

and the sample variance as

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1} = \frac{\sum_{i=1}^{14} (x_i - 235.4429)^2}{13} = 4084.464.$$

The sample standard deviation is $s = 63.9098$, and a 95% confidence interval using (8.10) is calculated as

$$\begin{aligned} CI_{0.95}(\mu) &= \left[\bar{x} - t_{0.975; n-1} \frac{s}{\sqrt{n}}, \bar{x} + t_{0.975; n-1} \frac{s}{\sqrt{n}} \right] \\ &= \left[235.4429 - (2.1604) \frac{63.9098}{\sqrt{14}}, 235.4429 + (2.1604) \frac{63.9098}{\sqrt{14}} \right] \\ &= [198.5424, 272.3433]. \end{aligned}$$

Thus, one is 95% confident the mean house price falls in [198.5424, 272.3433] thousands of dollars.

R Code 8.9 constructs a confidence interval for the mean using the functions `mean()`, `sd()`, and `qt()`. A second solution is also provided that computes the confidence interval with the `t.test()` function directly.

R Code 8.9

```
> xbar <- mean(HOUSE$price)
> CT <- qt(0.975, 13) # critical t value
> ST <- sd(HOUSE$price) # standard deviation
> xbar + c(-1, 1) * CT * ST/sqrt(14)

[1] 198.5424 272.3433

> # Second approach
> t.test(HOUSE$price, conf.level = 0.95)$conf

[1] 198.5424 272.3433
attr("conf.level")
[1] 0.95
```

To change the confidence level, say to 90%, the argument `conf.level = 0.90` is specified inside the `t.test()` function as `t.test(object, conf.level= 0.90)$conf`.

8.2.3 Confidence Interval for the Difference in Population Means When Sampling from Independent Normal Distributions with Known Equal Variances

Consider two normal and independent populations, $N(\mu_X, \sigma_X)$ and $N(\mu_Y, \sigma_Y)$, where $\sigma_X = \sigma_Y = \sigma$ is known. If one takes random samples of sizes n_X and n_Y , respectively, a

Normal Q-Q Plot

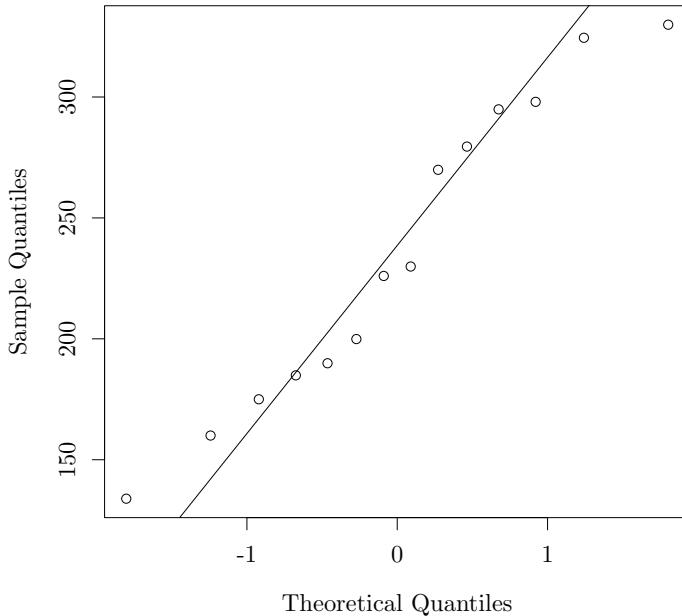


FIGURE 8.4: Quantile-quantile plot of the asking price for 14 randomly selected three-bedroom/two-bath houses in Watauga County, North Carolina

confidence interval for $\mu_X - \mu_Y$ is easily derived using the sampling distribution of

$$\bar{X} - \bar{Y} \sim N \left(\mu_X - \mu_Y, \sigma \sqrt{\left(\frac{1}{n_X} + \frac{1}{n_Y} \right)} \right),$$

which provides a pivotal quantity,

$$Q(\mathbf{X}, \mathbf{Y}; \mu_X - \mu_Y) = \frac{(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)}{\sigma \sqrt{\left(\frac{1}{n_X} + \frac{1}{n_Y} \right)}},$$

which has a standard normal distribution independent of the value of $\mu_X - \mu_Y$. The $(1 - \alpha) \cdot 100\%$ confidence interval for the difference in population means, $\mu_X - \mu_Y$, is given by

$$CI_{1-\alpha}(\mu_X - \mu_Y | \sigma_X = \sigma_Y \text{ is known}) = \left[(\bar{x} - \bar{y}) - z_{1-\alpha/2} \sigma \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}, (\bar{x} - \bar{y}) + z_{1-\alpha/2} \sigma \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}} \right]. \quad (8.11)$$

Example 8.9 Suppose independent random samples are taken from two normal distributions, $N(\mu_X, \sigma = 3)$ and $N(\mu_Y, \sigma = 3)$, respectively, such that $n_X = 15$, $\sum_{i=1}^{n_X} x_i = 60$, $n_Y = 22$, and $\sum_{i=1}^{n_Y} y_i = 97$. Calculate a 95% confidence interval for the difference in population means ($\mu_X - \mu_Y$).

Solution: Since

$$\bar{x} = \frac{\sum_{i=1}^{n_X} x_i}{n_X} = \frac{60}{15} = 4 \quad \text{and} \quad \bar{y} = \frac{\sum_{i=1}^{n_Y} y_i}{n_Y} = \frac{97}{22} = 4.41,$$

the 95% confidence interval for the difference in population means ($\mu_X - \mu_Y$) is calculated using (8.11) as

$$\begin{aligned} CI_{0.95}(\mu_X - \mu_Y) &= \left[(4 - 4.41) - (1.96)(3)\sqrt{\frac{1}{15} + \frac{1}{22}}, \right. \\ &\quad \left. (4 - 4.41) + (1.96)(3)\sqrt{\frac{1}{15} + \frac{1}{22}} \right] \\ &= [-2.3789, 1.5589]. \end{aligned}$$

To calculate the confidence interval with R, key in

```
> z <- qnorm(0.975)
> pe <- 4 - 4.41
> sigma <- 3
> nx <- 15
> ny <- 22
> pe + c(-1, 1) * z * sigma * sqrt(1/nx + 1/ny)

[1] -2.378853 1.558853

> # Second approach using zsum.test()
> CI <- zsum.test(mean.x = 4, mean.y = 4.41, sigma.x = 3,
+   sigma.y = 3, n.x = 15, n.y = 22, conf.level = 0.95)$conf
> CI

[1] -2.378853 1.558853
attr(,"conf.level")
[1] 0.95
```

So, one is 95% confident $\mu_X - \mu_Y$ lies in $[-2.3789, 1.5589]$. ■

Example 8.10 The hardness of a piece of fruit is a good indicator of the fruit's ripeness. An experiment was undertaken where 17 recently picked (fresh) apples were randomly selected and measured for hardness. Seventeen apples were also randomly selected from a warehouse where the apples had been stored for one week. Construct a 95% confidence interval for the mean difference between the hardness of fresh apples and the hardness for apples that were picked one week ago. Assume the distributions for both recently picked apples and for apples picked one week ago have known and equal variances of $2.25 \text{ (kg/meter}^2\text{)}^2$. The data are provided in Table 8.3 on the next page and can be found in the data frame **APPLE**.

Solution: Before the confidence interval formula in (8.11) can be used, one needs to make sure the assumption of normality is satisfied. Consequently, a normal quantile-quantile plot for the values reported in Table 8.3 on the facing page was constructed and is shown in Figure 8.5.

Since the points in Figure 8.5 fall relatively close to straight lines, it is decided that the normality assumptions for using (8.11) are satisfied. R Code 8.10 on the next page was used to produce Figure 8.5 on the facing page.

Table 8.3: Apple hardness measurements (APPLE)

Fresh			Warehouse		
7.27	8.38	9.20	7.79	9.17	10.05
6.65	5.83	7.89	7.11	6.31	8.58
5.76	7.70	7.77	6.27	8.39	8.42
6.53	5.86	6.48	7.22	6.19	7.07
8.09	5.53	8.28	8.83	6.31	8.83
9.56	6.54		10.5	7.17	

R Code 8.10

```
> ggpplot(data = APPLE, aes(sample = hardness, shape = location)) +
+   stat_qq() +
+   theme_bw()
```

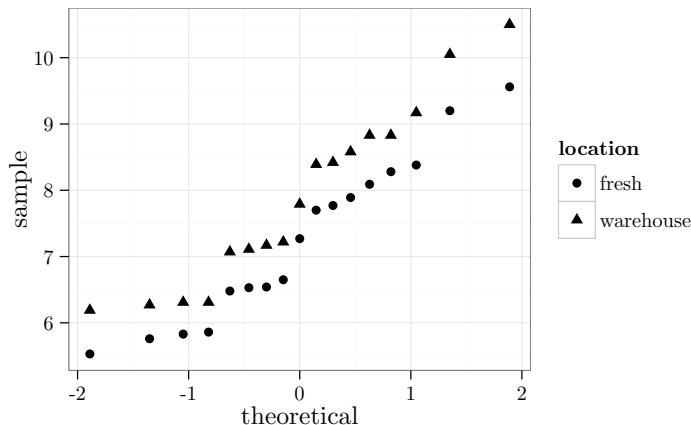


FIGURE 8.5: Normal quantile-quantile plots of the hardness values for fresh and warehoused apples

Thus, continue solving the problem by calculating the sample mean hardness for both the fresh and warehoused apples as

$$\bar{x} = \frac{\sum_{i=1}^{n_X} x_i}{n_X} = \frac{123.25}{17} = 7.25 \quad \text{and} \quad \bar{y} = \frac{\sum_{i=1}^{n_Y} y_i}{n_Y} = \frac{134.13}{17} = 7.89.$$

Using (8.11), the 95% confidence interval for $\mu_X - \mu_Y$ is

$$CI_{0.95}(\mu_X - \mu_Y) = \left[(7.2541 - 7.8947) - (1.96)(1.5)\sqrt{\frac{1}{17} + \frac{1}{17}}, (7.2541 - 7.8947) + (1.96)(1.5)\sqrt{\frac{1}{17} + \frac{1}{17}} \right] = [-1.6490, 0.3678].$$

R Code 8.11 computes the quantities needed for (8.11) and uses R to obtain the requested confidence interval. Next, the confidence interval is computed directly with `z.test()` after storing the hardness for each location in a separate vector using the function `subset()`.

R Code 8.11

```
> MEANS <- tapply(APPLE$hardness, APPLE$location, mean)
> MEANS

    fresh warehouse
7.254118  7.894706

> pe <- MEANS[1] - MEANS[2]
> pe + c(-1, 1)*qnorm(0.975)*1.5*sqrt(1/17 + 1/17)

[1] -1.6489814  0.3678049

> # Using z.test
> freshH <- subset(APPLE, select = hardness,
+                     subset = location == "fresh", drop = TRUE)
> wareH <- subset(APPLE, select = hardness,
+                     subset = location == "warehouse", drop = TRUE)
> CI <- z.test(x = freshH, y = wareH, sigma.x = 1.5, sigma.y = 1.5)$conf
> CI

[1] -1.6489814  0.3678049
attr(,"conf.level")
[1] 0.95
```

Thus, one is 95% confident that the difference in mean hardness for fresh and warehoused apples falls in the interval $[-1.649, 0.3678]$ kg/meter². Since this interval contains zero, one can say that there is essentially no difference between the hardnesses for fresh and warehoused apples. ■

8.2.4 Confidence Interval for the Difference in Population Means When Sampling from Independent Normal Distributions with Known but Unequal Variances

Consider two independent normal populations, $N(\mu_X, \sigma_X)$ and $N(\mu_Y, \sigma_Y)$, with known but unequal variances σ_X^2 and σ_Y^2 , respectively. If one takes random samples of size n_X and n_Y , respectively, the confidence interval for $\mu_X - \mu_Y$ can be constructed from knowledge of the sampling distribution of the statistic $\bar{X} - \bar{Y}$. Since

$$\bar{X} - \bar{Y} \sim N\left(\mu_X - \mu_Y, \sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}\right),$$

the $(1 - \alpha) \cdot 100\%$ confidence interval for $(\mu_X - \mu_Y)$ is

$$\boxed{CI_{1-\alpha}(\mu_X - \mu_Y | \sigma_X \neq \sigma_Y \text{ but known}) = \left[(\bar{x} - \bar{y}) - z_{1-\alpha/2} \sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}, (\bar{x} - \bar{y}) + z_{1-\alpha/2} \sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}} \right].} \quad (8.12)$$

Example 8.11 Suppose random samples of sizes $n_X = 50$ and $n_Y = 46$ are drawn from normal populations with standard deviations of 4.5 and 6, respectively, such that

$$\sum_{i=1}^{n_X} x_i = 420 \quad \text{and} \quad \sum_{i=1}^{n_Y} y_i = 405.$$

Construct a 97% confidence interval for $\mu_X - \mu_Y$.

Solution: Given that

$$\bar{x} = \frac{\sum_{i=1}^{50} x_i}{50} = \frac{420}{50} = 8.4 \quad \text{and} \quad \bar{y} = \frac{\sum_{i=1}^{46} y_i}{46} = \frac{405}{46} = 8.8043,$$

the 97% confidence interval for $\mu_X - \mu_Y$ is constructed using (8.12) as

$$\begin{aligned} CI_{0.97}(\mu_X - \mu_Y) &= \left[(8.4 - 8.8) - 2.17 \sqrt{\frac{(4.5)^2}{50} + \frac{6^2}{46}}, (8.4 - 8.8) + 2.17 \sqrt{\frac{(4.5)^2}{50} + \frac{6^2}{46}} \right] \\ &= [-2.7693, 1.9606]. \end{aligned}$$

Note that since zero is contained in the interval, one concludes μ_X is not significantly different from μ_Y . To construct the confidence interval with R, key in

```
> nx <- 50
> ny <- 46
> xbar <- 420/nx
> ybar <- 405/ny
> sigmax <- 4.5
> sigmay <- 6
> pe <- xbar - ybar # point estimate
> z <- qnorm(0.985) # z_0.985
> pe + c(-1, 1)*z*sqrt(sigmax^2/nx + sigmay^2/ny) # CI
[1] -2.769257  1.960562

> # Or using zsum.test
> CI <- zsum.test(mean.x = xbar, mean.y = ybar, sigma.x = sigmax,
+                   sigma.y = sigmay, n.x = nx, n.y = ny, conf.level = 0.97)$conf
> CI
[1] -2.769257  1.960562
attr(,"conf.level")
[1] 0.97
```

So, one is 97% confident that $\mu_X - \mu_Y$ lies in $[-2.7693, 1.9606]$. ■

Example 8.12 ▷ **Confidence Interval for $\mu_X - \mu_Y$: Calculus** ◁ Table 8.4 and data frame **CALCULUS** provide the mathematical assessment scores for 36 students enrolled in a biostatistics course according to whether or not the students had successfully completed a calculus course prior to enrolling in the biostatistics course. Construct a 95% confidence interval for the difference in the means of the mathematical assessment scores for students who had successfully completed a calculus course (X) and of those who had not (Y). Assume the distributions for X and Y have variances of 25 and 144, respectively. Determine if it is advantageous to take calculus prior to taking the biostatistics course.

Table 8.4: Mathematical assessment scores for students enrolled in a biostatistics course (**CALCULUS**)

Y						X					
No Calculus						Calculus					
73	39	55	72	88	64	82	90	85	87	86	79
57	58	75	44	76	68	85	92	89	82	92	82
64	55	62	61	76	40	85	87	92	85	95	90

Solution: Before using the confidence interval formula in (8.12), one needs to make sure the assumption of normality is satisfied. Consequently, a normal quantile-quantile plot for the values reported in Table 8.4 was constructed and is shown in Figure 8.6 on the facing page. Since the points in Figure 8.6 fall relatively close to straight lines, one decides the normality assumptions for using (8.12) are satisfied and continues by calculating the sample means for students who successfully completed calculus and those who have not yet successfully completed calculus as

$$\bar{x} = \frac{\sum_{i=1}^{18} x_i}{18} = \frac{1565}{18} = 86.9444 \quad \text{and} \quad \bar{y} = \frac{\sum_{i=1}^{18} y_i}{18} = \frac{1127}{18} = 62.6111.$$

The 95% confidence interval for $(\mu_X - \mu_Y)$ is constructed using (8.12) as

$$\begin{aligned} CI_{0.95}(\mu_X - \mu_Y) &= \\ &\left[(86.9444 - 62.6111) - (1.96) \sqrt{\frac{25}{18} + \frac{144}{18}}, (86.9444 - 62.6111) + (1.96) \sqrt{\frac{25}{18} + \frac{144}{18}} \right] \\ &= [18.3278, 30.3389]. \end{aligned}$$

R Code 8.12 on the next page computes the needed quantities to use (8.12) and then computes the requested confidence interval. The function `z.test()` is also used to compute the requested confidence interval after separating the assessment scores into separate vectors based on whether a student has taken calculus with the `subset()` function.

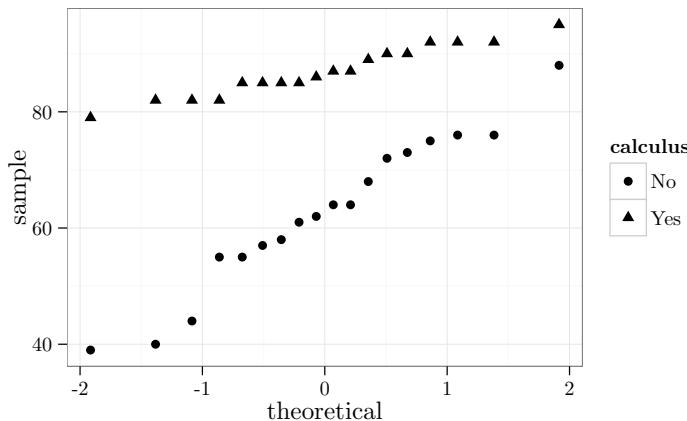


FIGURE 8.6: Normal quantile-quantile plots of the mathematical assessment scores for students enrolled in a biostatistics course who had successfully completed calculus and the mathematical assessment scores for students who had not successfully completed calculus

R Code 8.12

```
> MEANS <- tapply(CALCULUS$score, CALCULUS$calculus, mean)
> MEANS

      No          Yes
62.61111 86.94444

> pe <- MEANS[2] - MEANS[1]
> z <- qnorm(0.975)
> pe + c(-1, 1) * z * sqrt(25/18 + 144/18)

[1] 18.32775 30.33892

> # Using z.test
> ScoreYesCalc <- subset(CALCULUS, select = score, subset = calculus ==
+   "Yes", drop = TRUE)
> ScoreNoCalc <- subset(CALCULUS, select = score, subset = calculus ==
+   "No", drop = TRUE)
> CI <- z.test(x = ScoreYesCalc, y = ScoreNoCalc, sigma.x = sqrt(25),
+   sigma.y = sqrt(144), conf.level = 0.95)$conf
> CI

[1] 18.32775 30.33892
attr(,"conf.level")
[1] 0.95
```

Therefore, one is 95% confident that the difference in mean assessment scores for students who have successfully completed calculus prior to enrolling in biostatistics and those students who have not successfully completed calculus prior to enrolling in biostatistics lies in [18.3278, 30.3389]. It is advantageous to take calculus prior to taking biostatistics.

Note, once again, that the internal R function `t.test()` was not used to construct the confidence interval since `t.test()` assumes one is working with unknown variances; and in

Example 8.12, the variances are known. If σ is unknown, use (8.16) on page 476.

8.2.5 Confidence Interval for the Difference in Means When Sampling from Independent Normal Distributions with Variances That Are Unknown but Assumed Equal

Suppose random samples of size n_X and n_Y , respectively, are taken from two normal distributions $N(\mu_X, \sigma)$ and $N(\mu_Y, \sigma)$, where σ is unknown. To obtain a confidence interval for $\mu_X - \mu_Y$, take advantage of Theorem 6.4 on page 396, which allows the use of the pivot

$$Q(\mathbf{X}, \mathbf{Y}; \mu_X - \mu_Y) = \frac{[(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)]}{\sqrt{S_p^2 \left(\frac{1}{n_X} + \frac{1}{n_Y} \right)}} \sim t_{n_X + n_Y - 2}. \quad (8.13)$$

The denominator of the pivotal quantity in (8.13) is an estimator for the variance of $\bar{X} - \bar{Y}$, where

$$S_p^2 = \frac{(n_X - 1)S_X^2 + (n_Y - 1)S_Y^2}{n_X + n_Y - 2}. \quad (8.14)$$

Note that S_p^2 is a pooled estimator of the variance that weights the contributions of S_X^2 and S_Y^2 in proportion to the respective sample sizes n_X and n_Y . The degrees of freedom $n_X + n_Y - 2$ are denoted ν_p in the $(1 - \alpha) \cdot 100\%$ confidence interval for $\mu_X - \mu_Y$ given in (8.15).

$$CI_{1-\alpha}(\mu_X - \mu_Y | \text{Assuming } \sigma_X = \sigma_Y \text{ but unknown}) = \left[(\bar{x} - \bar{y}) - t_{1-\alpha/2; \nu_p} s_p \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}, (\bar{x} - \bar{y}) + t_{1-\alpha/2; \nu_p} s_p \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}} \right]. \quad (8.15)$$

Example 8.13 A random sample from a $N(\mu_X, \sigma)$ population is taken where

$$n_X = 15, \quad \sum_{i=1}^{15} x_i = 53, \quad \text{and} \quad \sum_{i=1}^{15} x_i^2 = 222.$$

Another random sample is taken from a $N(\mu_Y, \sigma)$ population independent of the first sample such that

$$n_Y = 11, \quad \sum_{i=1}^{11} y_i = 77, \quad \text{and} \quad \sum_{i=1}^{11} y_i^2 = 560.$$

Obtain a 95% confidence interval for $\mu_X - \mu_Y$ by assuming the true but unknown variances are equal.

Solution: The sample means and sample variances are calculated as

$$\bar{x} = \frac{\sum_{i=1}^{n_X} x_i}{n_X} = \frac{53}{15} = 3.5333, \quad S_X^2 = \frac{\sum_{i=1}^{n_X} x_i^2 - n_X \bar{x}^2}{n_X - 1} = \frac{222 - (15)(3.5333)^2}{15 - 1} = 2.481,$$

$$\bar{y} = \frac{\sum_{i=1}^{n_Y} y_i}{n_Y} = \frac{77}{11} = 7, \quad \text{and} \quad S_Y^2 = \frac{\sum_{i=1}^{n_Y} y_i^2 - n_Y \bar{y}^2}{n_Y - 1} = \frac{560 - (11)(7)^2}{11 - 1} = 2.1.$$

The pooled variance is given by

$$S_p^2 = \frac{(15 - 1)(2.481) + (11 - 1)(2.1)}{24} = 2.3222,$$

where $s_p = 1.5239$. Keeping in mind that $t_{0.975;24} = 2.0639$, the 95% confidence interval for $\mu_X - \mu_Y$ is constructed using (8.15) as

$$CI_{0.95}(\mu_X - \mu_Y) = \left[(3.5333 - 7) - (2.0639)(1.5239)\sqrt{\frac{1}{15} + \frac{1}{11}}, (3.5333 - 7) + (2.0639)(1.5239)\sqrt{\frac{1}{15} + \frac{1}{11}} \right] = [-4.7152, -2.2182].$$

To construct this confidence interval with R, type

```
> nx <- 15
> ny <- 11
> xbar <- 53/nx
> ybar <- 77/ny
> s2x <- (222 - nx * xbar^2)/(nx - 1)
> s2y <- (560 - ny * ybar^2)/(ny - 1)
> sp2 <- ((nx - 1) * s2x + (ny - 1) * s2y)/(nx + ny - 2)
> sp <- sqrt(sp2)
> ct <- qt(0.975, nx + ny - 2)
> pe <- xbar - ybar
> c(xbar, ybar, pe, s2x, s2y, sp2, sp, ct)

[1] 3.533333 7.000000 -3.466667 2.480952 2.100000 2.322222 1.523884
[8] 2.063899

> pe + c(-1, 1) * ct * sp * sqrt(1/nx + 1/ny)

[1] -4.715156 -2.218177

> # Or using tsum.test
> CI <- tsum.test(mean.x = xbar, mean.y = ybar, s.x = sqrt(s2x),
+   s.y = sqrt(s2y), n.x = nx, n.y = ny, var.equal = TRUE)$conf
> CI

[1] -4.715156 -2.218177
attr("conf.level")
[1] 0.95
```

That is, one is 95% confident that the difference of means lies in $[-4.7152, -2.2182]$. ■

Example 8.14 Given the information from Example 8.10 on page 468, construct a 95% confidence interval for the difference in hardness between fresh and warehoused apples. Assume the samples come from normal and independent distributions with unknown but equal variances.

Solution: According to the solution for Example 8.10, the sample means for fresh and warehoused apples are $\bar{x} = 7.254$ and $\bar{y} = 7.895$, respectively. Next, calculate the respective sample variances as

$$s_X^2 = \frac{\sum_{i=1}^{n_X} (x_i - \bar{x})^2}{n_X - 1} = \frac{\sum_{i=1}^{17} (x_i - 7.254)^2}{16} = 1.5104 \quad \text{and}$$

$$s_Y^2 = \frac{\sum_{i=1}^{n_Y} (y_i - \bar{y})^2}{n_Y - 1} = \frac{\sum_{i=1}^{17} (y_i - 7.895)^2}{16} = 1.7910.$$

Note that the t -distribution has $n_X + n_Y - 2 = 17 + 17 - 2 = 32$ degrees of freedom and s_p is calculated as

$$s_p = \sqrt{\frac{16(1.5104) + 16(1.7910)}{32}} = 1.28.$$

Finally, the 95% confidence interval for $\mu_X - \mu_Y$ is calculated as

$$CI_{0.95}(\mu_X - \mu_Y) = \left[(7.2541 - 7.8947) - (2.04)(1.28)\sqrt{\frac{1}{17} + \frac{1}{17}}, (7.2541 - 7.8947) + (2.04)(1.28)\sqrt{\frac{1}{17} + \frac{1}{17}} \right] = [-1.5382, 0.2570].$$

This confidence interval can be constructed with R by keying in

```
> CI <- t.test(hardness ~ location, data = APPLE, var.equal = TRUE)$conf
> CI

[1] -1.5382253 0.2570488
attr(,"conf.level")
[1] 0.95
```

So, one is 95% confident the difference in means for fresh and warehoused apple hardness falls in $[-1.5382, 0.257]$ kg/meter². ■

8.2.6 Confidence Interval for a Difference in Means When Sampling from Independent Normal Distributions with Variances That Are Unknown and Unequal

If random samples of size n_X and n_Y are drawn from two independent normal distributions, say $N(\mu_X, \sigma_X)$ and $N(\mu_Y, \sigma_Y)$, where σ_X and σ_Y are unknown and unequal, a $(1 - \alpha) \cdot 100\%$ confidence interval for $\mu_X - \mu_Y$ is given by

$$CI_{1-\alpha}(\mu_X - \mu_Y | \text{Unknown } \sigma_X \neq \sigma_Y) = \left[(\bar{x} - \bar{y}) - t_{1-\alpha/2; \nu} \sqrt{\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}}, (\bar{x} - \bar{y}) + t_{1-\alpha/2; \nu} \sqrt{\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}} \right]. \quad (8.16)$$

The degrees of freedom, ν , for (8.16) are determined by (8.17). When ν does not give an integer value, it is truncated to give a conservative approximation. “Conservative” means that the resulting confidence interval will have a confidence level of at least $1 - \alpha$.

$$\nu = \frac{\left(\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}\right)^2}{\frac{(s_X^2/n_X)^2}{n_X - 1} + \frac{(s_Y^2/n_Y)^2}{n_Y - 1}}. \quad (8.17)$$

The standardized test statistic in (8.18) is used to construct a confidence interval for $\mu_X - \mu_Y$. The sampling distribution of (8.18) is very complicated, but Welch's approximation of a t_ν -distribution provides adequate results and will be used in this text:

$$\frac{(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)}{\sqrt{\frac{S_X^2}{n_X} + \frac{S_Y^2}{n_Y}}} \sim t_\nu. \quad (8.18)$$

Example 8.15 Suppose a random sample is taken from a $N(\mu_X, \sigma_X)$ population where

$$n_X = 15, \quad \sum_{i=1}^{15} x_i = 63, \quad \text{and} \quad \sum_{i=1}^{15} x_i^2 = 338.$$

A second random sample is taken from a $N(\mu_Y, \sigma_Y)$ population independent from the first sample such that

$$n_Y = 11, \quad \sum_{i=1}^{11} y_i = 66.4, \quad \text{and} \quad \sum_{i=1}^{11} y_i^2 = 486.$$

Construct a 95% confidence interval for $\mu_X - \mu_Y$ assuming the variances for the two populations are unknown and unequal.

Solution: Start by calculating the sample means and sample variances for the respective samples as well as ν , the value for the degrees of freedom:

$$\begin{aligned} \bar{x} &= \frac{63}{15} = 4.2 & s_X^2 &= \frac{\sum_{i=1}^{n_X} x_i^2 - n_X \bar{x}^2}{n_X - 1} = \frac{338 - (15)(4.2)^2}{15 - 1} = 5.2429 \\ \bar{y} &= \frac{66.4}{11} = 6.0364 & s_Y^2 &= \frac{\sum_{i=1}^{n_Y} y_i^2 - n_Y \bar{y}^2}{n_Y - 1} = \frac{486 - (11)(6.0364)^2}{11 - 1} = 8.5185. \end{aligned}$$

Next, (8.17) is used with the sample variances and respective sample sizes to determine ν :

$$\nu = \frac{\left(\frac{5.2429}{15} + \frac{8.5185}{11} \right)^2}{\frac{(5.2294/15)^2}{14} + \frac{(8.5185/11)^2}{10}} = 18.3883.$$

The 95% confidence interval for $\mu_X - \mu_Y$ is constructed using (8.16) as

$$\begin{aligned} CI_{0.95}(\mu_X - \mu_Y) &= \left[(4.2 - 6.0364) - t_{0.975; 18.3883} \sqrt{\frac{5.2429}{15} + \frac{8.5185}{11}}, \right. \\ &\quad \left. (4.2 - 6.0364) + t_{0.975; 18.3883} \sqrt{\frac{5.2429}{15} + \frac{8.5185}{11}} \right] \\ &= [-1.8364 - (2.0977)(1.0602), -1.8364 + (2.0977)(1.0602)] \\ &= [-4.0603, 0.3876]. \end{aligned}$$

R Code 8.13 on the following page finds the confidence interval by computing the values for (8.16). Then, the confidence interval is computed using the function `tsum.test()`. Note that both approaches return the same confidence interval.

R Code 8.13

```

> nx <- 15
> ny <- 11
> xbar <- 63/nx
> ybar <- 66.4/ny
> pe <- xbar - ybar
> s2x <- (338 - nx*xbar^2)/(nx - 1)
> s2y <- (486 - ny*ybar^2)/(ny - 1)
> se <- sqrt(s2x/nx + s2y/ny)
> nu <- (s2x/nx + s2y/ny)^2 /
+   ((s2x/nx)^2/(nx - 1) + (s2y/ny)^2/(ny - 1))
> ct <- qt(0.975, nu)
> c(xbar, ybar, pe, s2x, s2y, se, nu, ct)

[1] 4.200000 6.036364 -1.836364 5.242857 8.518545 1.060159 18.388284
[8] 2.097747

> CI <- pe + c(-1, 1)*ct*se
> CI

[1] -4.0603086 0.3875813

> tsum.test(mean.x = xbar, mean.y = ybar, s.x = sqrt(s2x), s.y = sqrt(s2y),
+           n.x = nx, n.y = ny, conf.level =0.95)$conf

[1] -4.0603086 0.3875813
attr(,"conf.level")
[1] 0.95

```

One is 95% confident the difference of means lies in $[-4.0603, 0.3876]$. ■

Example 8.16 Using the information from Example 8.12, which provided the mathematical assessment scores for students enrolled in a biostatistics course according to whether they had completed a calculus course prior to enrolling in the biostatistics course, construct a 95% confidence interval for $\mu_X - \mu_Y$ assuming the samples are taken from distributions where the variances are unknown and unequal ($\sigma_X^2 \neq \sigma_Y^2$).

Solution: Recall from Example 8.12 that $\bar{x} = 86.9444$ and $\bar{y} = 62.6111$. Also recall that the assumption of normality for these data seemed plausible based on the normal quantile-quantile plot provided in Figure 8.6 on page 473. The respective sample variances are

$$s_X^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1} = \frac{\sum_{i=1}^{18} (x_i - 86.9444)^2}{17} = 18.6438 \quad \text{and}$$

$$s_Y^2 = \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1} = \frac{\sum_{i=1}^{18} (y_i - 62.6111)^2}{17} = 174.8399.$$

Next, (8.17) on page 476 is used with the sample variances and respective sample sizes to determine ν :

$$\nu = \frac{\left(\frac{18.6438}{18} + \frac{174.8399}{18}\right)^2}{\frac{(18.6438/18)^2}{17} + \frac{(174.8399/18)^2}{17}} = 20.5848.$$

The 95% confidence interval for $\mu_X - \mu_Y$ is constructed using (8.16) as

$$\begin{aligned}
 CI_{0.95}(\mu_X - \mu_Y) &= \left[(86.9444 - 62.6111) - t_{0.975;20} \sqrt{\frac{18.6438}{18} + \frac{174.8399}{18}}, \right. \\
 &\quad \left. (86.9444 - 62.6111) + t_{0.975;20} \sqrt{\frac{18.6438}{18} + \frac{174.8399}{18}} \right] \\
 &= [24.3333 - (2.0822)(3.2786), 24.3333 + (2.0822)(3.2786)] \\
 &= [17.5068, 31.1599].
 \end{aligned}$$

R Code 8.14 computes the confidence interval directly from the data frame **CALCULUS**. The first approach filters the scores for those who took calculus and those who did not as vectors to the function **t.test()**. The second approach passes a formula to the function **t.test()**; however, since the levels for the factor **calculus** are **No** and **Yes** (alphabetical), the levels are switched to **Yes** and **No** before using **t.test()**.

R Code 8.14

```

> t.test(CALCULUS$score[CALCULUS$calculus == "Yes"],
+         CALCULUS$score[CALCULUS$calculus == "No"])$conf
[1] 17.50677 31.15990
attr(,"conf.level")
[1] 0.95

> levels(CALCULUS$calculus)
[1] "No"   "Yes"

> CALCULUS$calculus <- factor(CALCULUS$calculus, levels = c("Yes", "No"))
> t.test(score ~ calculus, data = CALCULUS)$conf
[1] 17.50677 31.15990
attr(,"conf.level")
[1] 0.95

```

One is 95% confident that the difference of means lies in [17.5068, 31.1599]. Note that R can compute quantiles in the t -distribution with non-integer degrees of freedom. In particular, R uses the exact value for ν from (8.17) to find the critical value $t_{1-\alpha/2;\nu}$ in its confidence interval computation rather than truncating the value of ν . ■

When working with normal distributions that have unknown variances, not pooling the variances and using (8.16) is generally the better method when the sample sizes are the same. It is also better when the sample sizes are unequal and the larger variance is associated with the larger sample size. Pooling the variances and using (8.15) should only be done if one is relatively confident that the variances are equal or if the larger variance is associated with the smaller sample size. For a summary of these methods, see Table 8.5 on the following page.

Table 8.5: Methods for analyzing normal data

Condition	Method	Equation
Same Sample Sizes	Do Not Pool Variances	(8.16)
Larger Variance with Larger Sample	Do Not Pool Variances	(8.16)
Variances Equal	Pool Variances	(8.15)
Larger Variance with Smaller Sample	Pool Variances	(8.15)

8.2.7 Confidence Interval for the Mean Difference When the Differences Have a Normal Distribution

Information from two dependent distributions is often called **paired** or **dependent data**. Paired samples have some common intrinsic features such as members of the same family, animals from the same litter, etc. Data are also considered to be paired when the same sample is observed at different times. For example, suppose one is interested in evaluating the time undergraduate economic majors spend studying in the first month of the semester and how much time they spend studying in the last month of the semester. To help in the analysis, record the total time students spend studying in the first and last months of the semester. This information is considered paired data since there are two measurements for each student. Scores recorded from a pre-test and post-test on the same group of people are also considered to be a paired or a dependent sample. In general, when the researcher is presented with paired samples, the standard approach is to analyze the differences between the paired data. In other words, if the population of pairs is $((X_1, Y_1), (X_2, Y_2), \dots)$, analyze the paired differences $D = (X_1 - Y_1, X_2 - Y_2, \dots)$. When there is a paired sample of size n_D , denote the sample differences as $d = (x_1 - y_1, \dots, x_{n_D} - y_{n_D})$. Provided the distribution of population differences is

$$D \sim N(\mu_D = \mu_X - \mu_Y, \sigma_D), \quad (8.19)$$

a confidence interval formula for μ_D when σ_D is unknown can be constructed using the pivotal quantity

$$Q(\mathbf{X}; \mu_D) = \frac{\bar{D} - \mu_D}{S_D / \sqrt{n_D}} \sim t_{n_D - 1}, \quad (8.20)$$

where n_D represents the number of pairs in the sample and S_D is the standard deviation of the differences. Using (8.20) as a pivot, a $(1 - \alpha) \cdot 100\%$ confidence interval for the difference in two dependent population means, $\mu_X - \mu_Y$, is given as

$$CI_{1-\alpha}(\mu_X - \mu_Y) = CI_{1-\alpha}(\mu_D) = \left[\bar{d} - t_{1-\alpha/2; n_D - 1} \frac{s_D}{\sqrt{n_D}}, \bar{d} + t_{1-\alpha/2; n_D - 1} \frac{s_D}{\sqrt{n_D}} \right]. \quad (8.21)$$

Example 8.17 To compare the speed differences between two different brands of workstations (Sun and Digital), the times each brand took to complete complex simulations were recorded. Five complex simulations were selected, and the five selected simulations were

run on both workstations. The resulting times in minutes for the five simulations are given in Table 8.6 and stored in the data frame **SUNDIG**. Construct a 95% confidence interval for μ_D , the average time difference between Sun and Digital workstations. Is one of the workstations faster than the other?

Table 8.6: Time to complete a complex simulation in minutes (**SUNDIG**)

Simulation	Sun	Digital	Difference
1	110	102	8
2	125	120	5
3	141	135	6
4	113	114	-1
5	182	175	7

Solution: Since each one of the five selected complex simulations was run on both workstations, these samples are dependent. The differences between the dependent samples are $d = (8, 5, 6, -1, 7)$, $\bar{d} = 5$ minutes, and $s_D = 3.5355$ minutes. Before using (8.21), one needs to verify the distribution of differences is normal. To check the normality assumption, use the functions `qqnorm()` and `qqline()` on the sample differences, d . The resulting normal quantile-quantile plot is shown in Figure 8.7 on the next page. Based on Figure 8.7, it is not immediately clear that the distribution of differences is normal due to the outlier in the lower left of the plot. At this point, one should look at several normal quantile-quantile plots for samples of size five using the `ntester()` function. The results of using the function `ntester()` on the sample differences are shown in Figure 8.8 on page 483. After using the `ntester()` function on the differences and viewing the output in Figure 8.8, one can conclude that it is not unreasonable to assume the distribution of differences between Sun and Digital workstations follow a normal distribution and can use (8.21) to construct the 95% confidence interval for $\mu_D = \mu_{\text{SUN}} - \mu_{\text{DIG}}$ as follows:

$$\begin{aligned} CI_{0.95}(\mu_{\text{SUN}} - \mu_{\text{DIG}}) &= \left[\bar{d} - t_{0.975; n_D - 1} \frac{s_D}{\sqrt{n_D}}, \bar{d} + t_{0.975; n_D - 1} \frac{s_D}{\sqrt{n_D}} \right] \\ &= \left[5 - (2.7764) \frac{3.5355}{\sqrt{5}}, 5 + (2.7764) \frac{3.5355}{\sqrt{5}} \right] = [0.6101, 9.3899]. \end{aligned}$$

One is 95% confident μ_D lies in $[0.6101, 9.3899]$ minutes. Since the confidence interval does not contain zero, one can be 95% confident that $\mu_D = \mu_{\text{SUN}} - \mu_{\text{DIG}} > 0$. This implies that $\mu_{\text{SUN}} > \mu_{\text{DIG}}$, which means that the Digital workstation is faster than the Sun workstation. To verify the value $t_{0.975; 4}$ and to calculate a 95% confidence interval for μ_D with R, enter

```
> qt(0.025, 4)
[1] -2.776445
> t.test(SUNDIG$sun, SUNDIG$digital, paired = TRUE)$conf
```

Normal Q-Q Plot

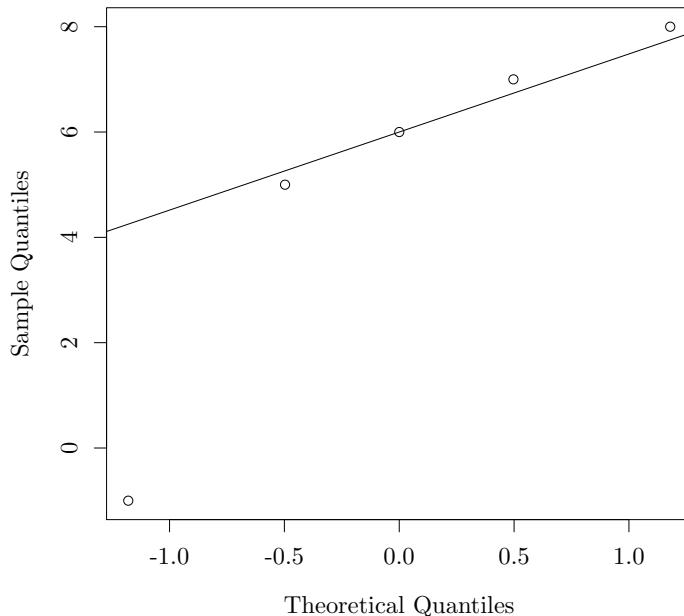


FIGURE 8.7: Normal quantile-quantile plot of the time differences between Sun and Digital workstations to complete complex simulations

```
[1] 0.6100548 9.3899452
attr(,"conf.level")
[1] 0.95
```

8.3 Confidence Intervals for Population Variances

The confidence intervals constructed for either a single population variance or a ratio of variances are quite sensitive to whether or not the underlying population is normal. Both are based on sampling distributions associated with the normal distribution. The chi-square distribution is helpful for constructing the confidence interval for a single population variance, and the F -distribution is used to construct a confidence interval for the ratio of population variances. Unless it is reasonable to assume the underlying distribution is normal, neither of these confidence intervals will be accurate. If the assumptions are violated, the procedures from Chapter 10, “Nonparametric Methods,” will be more appropriate.

8.3.1 Confidence Interval for the Population Variance When Sampling from a Normal Population

This section considers a normal population $N(\mu, \sigma^2)$ from which a random sample of size n is taken. The confidence interval for σ^2 is based on the pivot

Simulated Normal Data on Perimeter - Actual Data in Center

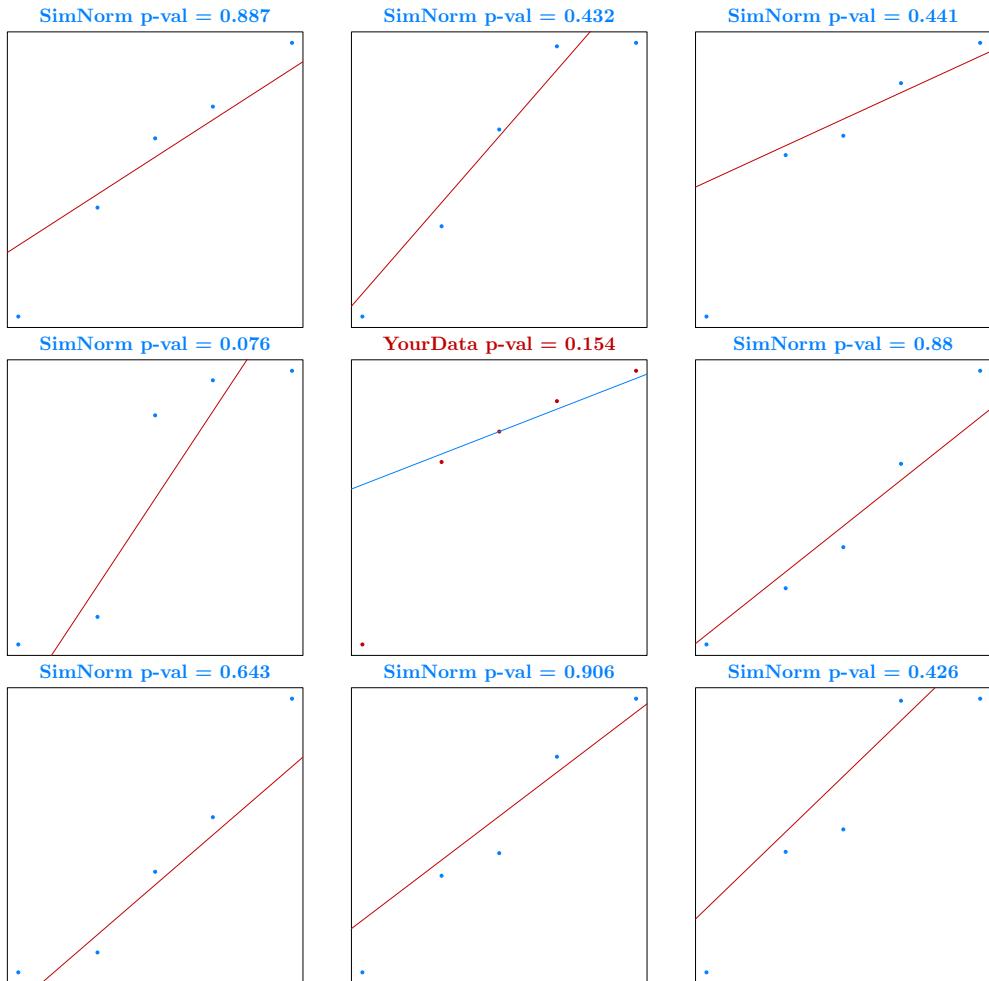


FIGURE 8.8: Quantile-quantile plot of the time differences between Sun and Digital workstations to complete complex simulations shown in the middle with normal quantile-quantile plots of random normal data depicted on the outside plots

$$Q(\mathbf{X}; \sigma^2) = \frac{(n-1)S^2}{\sigma^2} \sim \chi^2_{n-1} \quad (8.22)$$

The pivotal quantity (8.22) is not very robust with respect to the normality assumption. Consequently, before constructing a confidence interval for σ^2 , one should always check the sample for normality using a graphical procedure such as a normal quantile-quantile plot (`qqnorm()`). Although Pearson's χ^2 distribution is not symmetric (see Figure 8.9 on the next page), one can use the sampling distribution of the statistic $(n-1)S^2/\sigma^2$ and the definition of percentiles to obtain

$$\mathbb{P}\left(\chi^2_{\alpha/2;n-1} \leq \frac{(n-1)S^2}{\sigma^2} \leq \chi^2_{1-\alpha/2;n-1}\right) = 1 - \alpha. \quad (8.23)$$

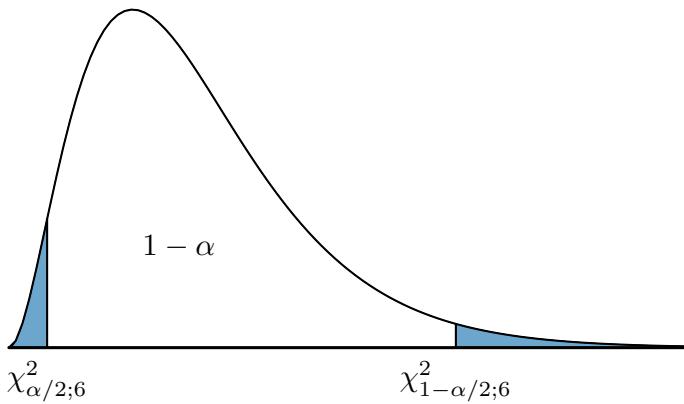


FIGURE 8.9: Chi-square distribution with six degrees of freedom depicting the points $\chi_{\alpha/2;6}^2$ and $\chi_{1-\alpha/2;6}^2$

To arrive at the standard confidence interval form for the variance, first take the reciprocal inside the probability statement of (8.23) as shown in (8.24). Then, multiply everything inside the probability statement of (8.24) by $(n-1)S^2$ to obtain the probability statement shown in (8.25):

$$\mathbb{P}\left(\frac{1}{\chi_{\alpha/2;n-1}^2} \geq \frac{\sigma^2}{(n-1)S^2} \geq \frac{1}{\chi_{1-\alpha/2;n-1}^2}\right) = 1 - \alpha, \quad (8.24)$$

$$\mathbb{P}\left(\frac{(n-1)S^2}{\chi_{1-\alpha/2;n-1}^2} \leq \sigma^2 \leq \frac{(n-1)S^2}{\chi_{\alpha/2;n-1}^2}\right) = 1 - \alpha. \quad (8.25)$$

Using the probability statement (8.25) at a fixed confidence level of $(1 - \alpha)$, the standard form for the confidence interval for σ^2 is illustrated in (8.26). Note that the confidence interval for the variance is not centered around the point estimate (the sample variance, s^2).

$$CI_{1-\alpha}(\sigma^2) = \left[\frac{(n-1)s^2}{\chi_{1-\alpha/2;n-1}^2}, \frac{(n-1)s^2}{\chi_{\alpha/2;n-1}^2} \right]. \quad (8.26)$$

Example 8.18 Construct an 80% confidence interval for σ_X^2 using the information from Example 8.15.

Solution: Recall that the underlying distribution in Example 8.15 was assumed to be $N(\mu_X, \sigma_X)$ and the sample information provided revealed that $n_X = 15$, $\bar{x} = 4.2$, and $s_X^2 = 5.2429$. Using (8.26), the 80% confidence interval for σ_X^2 is calculated as

$$\begin{aligned} CI_{0.8}(\sigma_X^2) &= \left[\frac{(n_X - 1)s_X^2}{\chi_{0.9;n-1}^2}, \frac{(n_X - 1)s_X^2}{\chi_{0.1;n-1}^2} \right] = \left[\frac{14(5.2429)}{\chi_{0.9;14}^2}, \frac{14(5.2429)}{\chi_{0.1;14}^2} \right] \\ &= \left[\frac{73.4006}{21.0641}, \frac{73.4006}{7.7895} \right] = [3.4860, 9.4266]. \end{aligned}$$

To construct this confidence interval with R, type

```
> lchi <- qchisq(0.1, 14)
> uchi <- qchisq(0.9, 14)
> lep <- (15 - 1) * 5.2449/uchi
> uep <- (15 - 1) * 5.2449/lchi
> c(lep, uep)

[1] 3.485952 9.426572
```

Therefore, one is 80% confident the variance falls in [3.486, 9.4266]. ■

Example 8.19 The data frame **barley** is in the **lattice** package and contains yield, variety, year, and site, giving barley yields (bushels/acre) in 1931 and 1932 for 10 varieties of barley grown at six sites.

- (a) Construct a 95% confidence interval for μ , the mean barley yield in 1932.
- (b) Construct a 95% confidence interval for σ^2 , the variance of barley yield in 1932.

Solution: Start by looking at the distribution of 1932 barley yield using the functions **qqnorm()** and **qqline()** to create the normal quantile-quantile plot shown in Figure 8.10. Since the values in Figure 8.10 are fairly linear, it is decided the assumptions to use both (8.10) and (8.26) are satisfied.

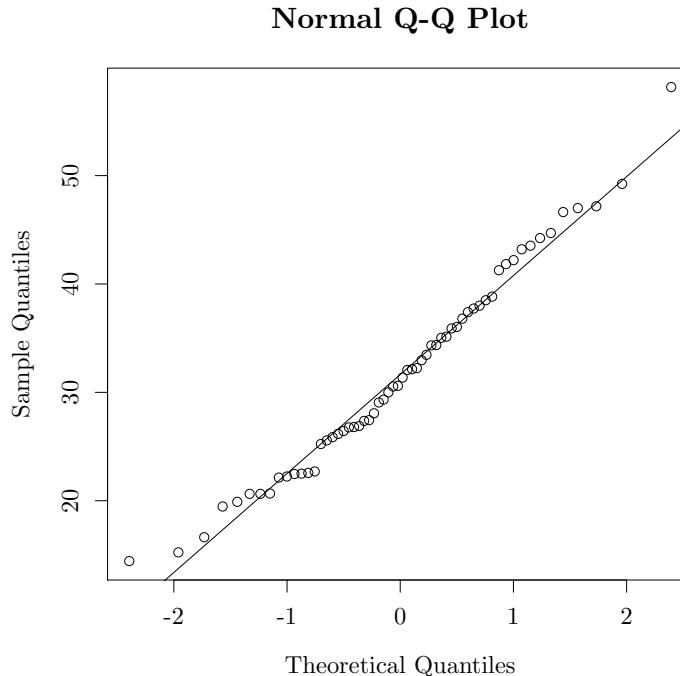


FIGURE 8.10: Quantile-quantile plot of 1932 barley yield in bushels/acre

- (a) The values (8.10) required to construct a 95% confidence interval for μ are found in R Code 8.15 on the following page.

$$\begin{aligned}
CI_{1-0.05}(\mu) &= \left[\bar{x} - t_{1-0.05/2;n-1} \frac{s}{\sqrt{n}}, \bar{x} + t_{1-0.05/2;n-1} \frac{s}{\sqrt{n}} \right] \\
&= \left[31.7633 - (2.001) \frac{9.3845}{\sqrt{60}}, 31.7633 + (2.001) \frac{9.3845}{\sqrt{60}} \right] \\
&= [29.3391, 34.1876].
\end{aligned} \tag{8.27}$$

R Code 8.15

```

> library(lattice)
> BarleyYield1932 <- subset(barley, select = yield, subset = year ==
+     "1932", drop = TRUE)
> n <- sum(!is.na(BarleyYield1932))
> n

[1] 60

> mean(BarleyYield1932)

[1] 31.76333

> var(BarleyYield1932)

[1] 88.06803

> sd(BarleyYield1932)

[1] 9.384457

> qt(0.975, n - 1)

[1] 2.000995

```

To construct the confidence interval directly with R, key in

```

> t.test(BarleyYield1932, conf.level = 0.95)$conf
[1] 29.33907 34.18759
attr(,"conf.level")
[1] 0.95

```

So, one is 95% confident that the mean barley yield (bushels/acre) lies in the interval [29.3391, 34.1876].

(b) To construct a 95% confidence interval for σ^2 , use (8.26):

$$\begin{aligned}
CI_{0.95}(\sigma^2) &= \left[\frac{(n-1)s^2}{\chi^2_{0.975;n-1}}, \frac{(n-1)s^2}{\chi^2_{0.025;n-1}} \right] = \left[\frac{59(88.068)}{82.1174}, \frac{59(88.068)}{39.6619} \right] \\
&= [63.2754, 131.0078].
\end{aligned}$$

To verify the previous values and construct this confidence interval with R, enter

```

> s2 <- var(BarleyYield1932)
> lchi <- qchisq(0.025, n - 1)
> uchi <- qchisq(0.975, n - 1)
> c(s2, lchi, uchi)

[1] 88.06803 39.66186 82.11741

> ll <- (n - 1) * s2/uchi
> ul <- (n - 1) * s2/lchi
> CI <- c(ll, ul)
> CI

[1] 63.27543 131.00782

```

One is 95% confident that the variance of barley yield lies in the interval [63.2754, 131.0078] (bushels/acre)². ■

8.3.2 Confidence Interval for the Ratio of Population Variances When Sampling from Independent Normal Distributions

Now consider the construction of confidence intervals for σ_X^2/σ_Y^2 when there are two normal and independent populations, $N(\mu_X, \sigma_X)$ and $N(\mu_Y, \sigma_Y)$, from which one takes random samples of sizes n_X and n_Y , respectively. The goal is to construct a confidence interval for the ratio of the variances, σ_X^2/σ_Y^2 . Generally, one is looking for 1 to be in the interval, indicating that the variances are equal. To construct a confidence interval for σ_X^2/σ_Y^2 , use Theorem 6.5 on page 398, which states that if one has two random samples X_1, \dots, X_{n_X} and Y_1, \dots, Y_{n_Y} that are taken from independent normal populations where $X \sim N(\mu_X, \sigma_X)$ and $Y \sim N(\mu_Y, \sigma_Y)$, then the random variable

$$\frac{S_Y^2/\sigma_Y^2}{S_X^2/\sigma_X^2} \sim F_{n_Y-1, n_X-1}. \quad (8.28)$$

By using (8.28), construct the $(1-\alpha)$ probability statement shown in (8.29) and graphically illustrated in Figure 8.11 on the following page for an F distribution with ten and ten degrees of freedom:

$$\mathbb{P}\left(f_{\alpha/2; n_Y-1, n_X-1} \leq \frac{S_Y^2/\sigma_Y^2}{S_X^2/\sigma_X^2} \leq f_{1-\alpha/2; n_Y-1, n_X-1}\right) = 1 - \alpha. \quad (8.29)$$

After multiplying everything inside the probability statement given in (8.29) by $\frac{\sigma_X^2}{S_Y^2}$, (8.30) is used to derive the final confidence interval statement given in (8.31):

$$\mathbb{P}\left(f_{\alpha/2; n_Y-1, n_X-1} \frac{S_X^2}{S_Y^2} \leq \frac{\sigma_X^2}{\sigma_Y^2} \leq f_{1-\alpha/2; n_Y-1, n_X-1} \frac{S_X^2}{S_Y^2}\right) = 1 - \alpha. \quad (8.30)$$

$$CI_{1-\alpha} \left(\frac{\sigma_X^2}{\sigma_Y^2} \right) = \left[f_{\alpha/2; n_Y-1, n_X-1} \frac{s_X^2}{s_Y^2}, f_{1-\alpha/2; n_Y-1, n_X-1} \frac{s_X^2}{s_Y^2} \right] \quad (8.31)$$

For sheer convenience, denote the larger sample variance as s_X^2 when constructing a confidence interval for the ratio of two population variances. Consequently, the numerator

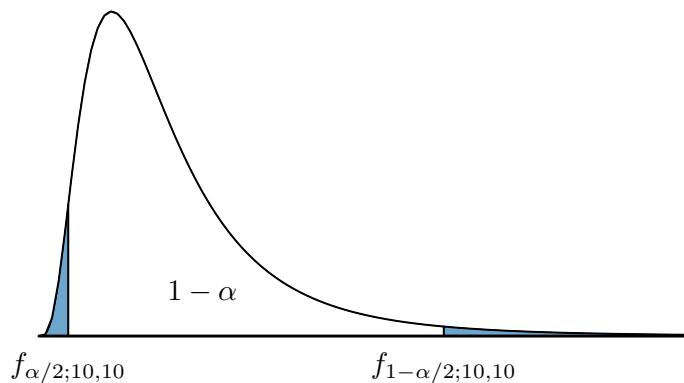


FIGURE 8.11: F distribution with ten and ten degrees of freedom depicting the points $f_{\alpha/2;10,10}$ and $f_{1-\alpha/2;10,10}$

for the ratio of the sample variances will always contain the larger of the two sample variances. Many tables involving the F distribution only provide values for percentiles in the right tail. However, this does not present a problem provided one remembers that values in the left tail of the F distribution can be found from the values in the right tail of an F distribution by using (8.32). Note that the order of the degrees of freedom changes in the reciprocal.

$$f_{\alpha/2;n_Y-1,n_X-1} = \frac{1}{f_{1-\alpha/2;n_X-1,n_Y-1}}. \quad (8.32)$$

Example 8.20 Using the information from Example 8.13 on page 474, construct a 90% confidence interval for the ratio of variances.

Solution: In Example 8.13, the larger sample variance, s_X^2 , was 2.481, $n_X = 15$, and the smaller sample variance, s_Y^2 , was 2.1, $n_Y = 11$. Consequently, the 90% confidence interval for the ratio of variances is constructed using (8.31) as shown in the following where $f_{0.05;10,14} = 0.3491$ and $f_{0.95;10,14} = 2.6022$:

$$CI_{0.9} \left(\frac{\sigma_X^2}{\sigma_Y^2} \right) = \left[f_{0.05;10,14} \frac{s_X^2}{s_Y^2}, f_{0.95;10,14} \frac{s_X^2}{s_Y^2} \right] \quad (8.33)$$

$$= \left[(0.3491) \frac{2.481}{2.1}, (2.6022) \frac{2.481}{2.1} \right] = [0.4124, 3.0743]. \quad (8.34)$$

To find $f_{0.05;10,14}$, $f_{0.95;10,14}$ and a 95% confidence interval for $\frac{\sigma_X^2}{\sigma_Y^2}$ with R, type

```
> lf <- qf(0.05, 10, 14) # lower f value
> uf <- qf(0.95, 10, 14) # upper f value
> c(lf, uf)

[1] 0.3490733 2.6021551

> lep <- lf*2.481/2.1 # lower CI point
> uep <- uf*2.481/2.1 # upper CI point
> CI <- c(lep, uep)
> CI

[1] 0.4124052 3.0742603
```

So, one is 90% confident the ratio of the variance lies in $[0.4124, 3.0743]$. Note that this interval includes 1, which indicates there is not evidence to suggest the variances are different. ■

Example 8.21 Given the information in Table 8.3 on page 469, construct a 95% confidence interval for the ratio of the variances.

Solution: According to Example 8.14, $s_X^2 = 1.5104$ and $s_Y^2 = 1.7910$. Also recall that in the solution to Example 8.10, a normal quantile-quantile plot was created and illustrated in Figure 8.5 on page 469 that justified the assumptions that both fresh and warehoused apples follow a normal distribution. Consequently, the appropriate confidence interval formula for the ratio of the variances is given in (8.31). However, since $s_Y^2 = 1.7910$ and $s_X^2 = 1.5104$, reverse s_X^2 for s_Y^2 in the confidence interval formula provided in (8.31) to construct a 95% confidence interval for the ratio of population variances:

$$\begin{aligned} CI_{0.95} \left(\frac{\sigma_Y^2}{\sigma_X^2} \right) &= \left[f_{0.025; 16, 16} \frac{s_Y^2}{s_X^2}, f_{0.975; 16, 16} \frac{s_Y^2}{s_X^2} \right] \\ &= [(0.3621)(1.1857), (2.7614)(1.1857)] = [0.4294, 3.2742]. \end{aligned} \quad (8.35)$$

To verify the previous values and to construct a 95% confidence interval for the ratio of variances with R, see R Code 8.16. The function `var.test()` will compute a confidence interval based on (8.31), that is a confidence interval for the ratio σ_X^2/σ_Y^2 . To compute a confidence interval for the ratio σ_Y^2/σ_X^2 , the levels of the factor `location` will need to be changed in the data frame `APPLE`.

R Code 8.16

```
> VAR <- tapply(APPLE$hardness, APPLE$location, var)
> VAR

      fresh  warehouse
1.510438 1.790951

> RVAR <- VAR[2]/VAR[1]
> names(RVAR) <- NULL
> lf <- qf(0.025, 16, 16) # lower f value
> uf <- qf(0.975, 16, 16) # upper f value
> c(lf, uf, RVAR)

[1] 0.3621405 2.7613591 1.1857165

> lep <- lf*RVAR # lower CI end point
> uep <- uf*RVAR # upper CI end point
> CI <- c(lep, uep)
> CI

[1] 0.429396 3.274189

> # using var.test()
> levels(APPLE$location) # show default levels of location

[1] "fresh"     "warehouse"
```

```

> APPLE$location <- factor(APPLE$location,
+                               levels = c("warehouse", "fresh"))
> levels(APPLE$location) # changed levels of location
[1] "warehouse" "fresh"

> var.test(hardness ~ location, data = APPLE)$conf
[1] 0.429396 3.274189
attr(,"conf.level")
[1] 0.95

```

One is 95% confident that the ratio of variances falls in [0.4294, 3.2742], which indicates that a pooled variance could be justified for confidence interval calculations regarding the means. ■

8.4 Confidence Intervals Based on Large Samples

Provided the sample size, n , is sufficiently large, one can take advantage of the asymptotic properties of maximum likelihood estimators to construct confidence intervals since, as $n \rightarrow \infty$,

$$\hat{\theta}(\mathbf{X}) \sim N\left(\theta, \sqrt{I_n(\theta)^{-1}}\right). \quad (8.36)$$

Using (8.36), one can construct asymptotic confidence intervals of the type given in (8.37). Note that $\sigma_{\hat{\theta}(\mathbf{X})}$ is the standard deviation of the estimator $\hat{\theta}(\mathbf{X})$. Specifically, in the multi-parameter case, $\sigma_{\hat{\theta}(\mathbf{X})}$ is the square root of the corresponding diagonal element of the inverse of the information matrix. When $\sigma_{\hat{\theta}(\mathbf{X})}$ is unknown, the estimate $\hat{\sigma}_{\hat{\theta}(\mathbf{x})}$ is used in place of $\sigma_{\hat{\theta}(\mathbf{X})}$. Be sure to see that $\hat{\sigma}_{\hat{\theta}(\mathbf{x})}$ is calculated from the data \mathbf{x} .

$$CI_{1-\alpha}(\theta) = \left[\hat{\theta}(\mathbf{x}) - z_{1-\alpha/2} \cdot \sigma_{\hat{\theta}(\mathbf{X})}, \hat{\theta}(\mathbf{x}) + z_{1-\alpha/2} \cdot \sigma_{\hat{\theta}(\mathbf{X})} \right] \quad (8.37)$$

Example 8.22 Given a random sample of size 200 from an exponential distribution, find a 90% confidence interval for θ if it is true that

$$\sum_{i=1}^{200} x_i = 400.$$

As a reminder, the exponential distribution is

$$f(x, \theta) = \frac{1}{\theta} e^{-\frac{1}{\theta}x}, \quad x \geq 0, \quad \theta > 0. \quad (8.38)$$

Solution: The reader should verify that the maximum likelihood estimator of θ is $\hat{\theta}(\mathbf{X}) = \bar{X}$ and the variance of \bar{X} is $\frac{\theta^2}{n}$. (Hint: See Example 7.7 on page 413.) Because \bar{X} is the

maximum likelihood estimator of θ , it follows that the maximum likelihood estimator of $\frac{\theta^2}{n}$ is $\frac{\bar{X}^2}{n}$ due to the invariance property of MLEs (property 2 on page 435). From the sample information, calculate

$$\hat{\theta}(\mathbf{x}) = \bar{x} = 2 \quad \text{and} \quad \hat{\sigma}_{\hat{\theta}(\mathbf{x})}^2 = \frac{\bar{x}^2}{n} = 0.02.$$

Given that the confidence level is 0.9, $z_{1-\alpha/2} = z_{0.95} = 1.6449$, the 90% confidence interval for θ is constructed using (8.37):

$$CI_{0.90}(\theta) = \left[2 - 1.6449\sqrt{0.02}, 2 + 1.6449\sqrt{0.02} \right] = [1.7674, 2.2326]. \quad (8.39)$$

So, one is 90% confident the exponential parameter θ falls in [1.7674, 2.2326]. ■

8.4.1 Confidence Interval for the Population Proportion

The maximum likelihood estimator of the population proportion π is P , the sample proportion. See Example 7.17 on page 422 for the derivation of the maximum likelihood estimator of π . To calculate the Fisher information $I_n(\pi)$, use (7.50) on page 433. Since

$$\frac{\partial^2 \ln L(\pi|\mathbf{X})}{\partial \pi^2} = \frac{-\sum_{i=1}^n x_i}{\pi^2} - \frac{n - \sum_{i=1}^n x_i}{(1-\pi)^2} \quad (8.40)$$

from Example 7.17 on page 422, multiplying (8.40) by -1 , and taking the expected value of the result, gives

$$\begin{aligned} -E\left[\frac{\partial^2 \ln L(\pi|\mathbf{X})}{\partial \pi^2}\right] &= E\left[\frac{\sum_{i=1}^n x_i}{\pi^2}\right] + E\left[\frac{n - \sum_{i=1}^n x_i}{(1-\pi)^2}\right] \\ &= \frac{n\pi}{\pi^2} + \frac{n-n\pi}{(1-\pi)^2} \\ &= \frac{n}{\pi} + \frac{n}{1-\pi} = \frac{n}{\pi(1-\pi)}. \end{aligned} \quad (8.41)$$

Consequently, the Fisher information, $I_n(\pi)^{-1}$, is given in (8.42):

$$I_n(\pi)^{-1} = \frac{\pi(1-\pi)}{n}. \quad (8.42)$$

Taking advantage of the asymptotic properties of MLE estimators allows one to write

$$\hat{\pi}(\mathbf{X}) = P \sim N\left(\pi, \sqrt{\frac{\pi(1-\pi)}{n}}\right) \text{ as } n \rightarrow \infty;$$

and using (8.37), one can construct a $(1-\alpha) \cdot 100\%$ asymptotic confidence interval for π as shown in (8.43) where

$$\hat{\sigma}_{\hat{\pi}(\mathbf{x})} = \sqrt{\frac{p(1-p)}{n}}.$$

The confidence interval in (8.43) can also be derived using the approximate sampling distribution of P from Section 6.5.3. The R package `binom` has the function `binom.confint()`, which computes the confidence interval for a binomial probability where the user can select

any or all of the eight available methods. To compute a confidence interval based on (8.43) one would specify `methods = "asymptotic"` in the `binom.confint()` function.

$$CI_{1-\alpha}(\pi) = \left[p - z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n}}, p + z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n}} \right] \quad (8.43)$$

Example 8.23 ▷ 90% Asymptotic Confidence Interval Consider a random variable $X \sim Bin(n = 25, \pi)$, and define $P = \frac{X}{n}$. If $X = 15$, construct a 90% asymptotic confidence interval for π using (8.43), and verify the confidence interval is the same as the one returned with the function `binom.confint()` from the `binom` package.

Solution: R Code 8.17 computes a 90% asymptotic confidence interval based on (8.43) and stores the result in `CI`. The function `binom.confint()` from the `binom` package is also used to find the requested confidence interval.

R Code 8.17

```
> n <- 25
> p <- 15/n                      # sample proportion
> alpha <- 0.10                   # alpha level
> z <- qnorm(1 - alpha/2)        # critical value
> me <- z*sqrt(p*(1 - p)/n)     # margin of error
> CI <- p + c(-1, 1)*me
> CI
[1] 0.4388379 0.7611621

> library(binom)
> binom.confint(x = 15, n = 25, conf.level = 0.90, methods = "asymptotic")
   method  x  n mean      lower      upper
1 asymptotic 15 25  0.6 0.4388379 0.7611621
```

Using (8.43), the results stored in `CI` are $CI_{0.90}(\pi) = [0.4388, 0.7612]$, which agree with the values returned from using `binom.confint()` with appropriate arguments.

Equation (8.43) is simple and is the confidence interval formula provided in many elementary statistics texts with certain caveats. The confidence interval constructed using (8.43) is frequently referred to as a **Wald** confidence interval. Brown et al. (2001) surveyed eleven popular texts and found the following qualifications to use (8.43).

1. $n\pi, n(1 - \pi)$ are ≥ 5 or (10)
2. $n\pi(1 - \pi) \geq 5$ or (10)
3. $np, n(1 - p)$ are ≥ 5 or (10)
4. $p \pm 3\sqrt{p(1-p)/n}$ does not contain 0 or 1
5. n quite large

6. $n \geq 50$ unless π is very small

The article by Brown et al. (2001) points out why none of the standard qualifications serve their purpose and concludes the performance of confidence interval (8.43) is so erratic and that the qualifications given in influential texts are so defective that confidence interval (8.43) should not be used. One of the stronger arguments against using (8.43) is its poor coverage probability. The coverage probability of a confidence interval procedure for estimating π at a fixed value of π is

$$C_n(\pi) = \sum_{k=0}^n I(k, \pi) \binom{n}{k} \pi^k (1 - \pi)^{n-k} \quad (8.44)$$

where $I(k, \pi)$ equals 1 if the interval contains π when $X = k$ and equals 0 if it does not contain π . In other words, the coverage probability of a confidence interval is the proportion of all possible confidence intervals for a fixed π that contain π . Confidence intervals are constructed at a given confidence level $(1 - \alpha)$, which is referred to as the nominal coverage probability or nominal confidence level. In an ideal setting, the nominal confidence level will equal the coverage probability; however, when assumptions used to derive a confidence interval are not satisfied, the actual coverage probability can be either less than or greater than the nominal confidence level.

Example 8.24 \triangleright **Coverage Probability** \triangleleft Consider a random variable $X \sim \text{Bin}(n = 25, \pi)$, and define $P = \frac{X}{n}$.

- (a) Compute the coverage probability according to (8.44) for a 95% confidence interval using the Wald confidence interval from (8.43) if $\pi = 0.70$. Verify the coverage probability with the function `binom.coverage()` from the `binom` package.
- (b) Compute the coverage probability according to (8.44) for a 95% confidence interval using the Wald confidence interval from (8.43) if $\pi = 0.69$ instead of $\pi = 0.70$. Verify the coverage probability with the function `binom.coverage()` from the `binom` package.
- (c) Compute and graph the coverage probability for the Wald confidence interval, (8.43), using a confidence level of 95% with 2000 equally spaced values of π using base graphics and the function `binom.plot()` from the `binom` package.

Solution: The answers follow:

- (a) To compute $C_{25}(0.70)$, one must consider all the possible outcomes for X when $n = 25$. The random variable X can assume values $0, 1, 2, \dots, 25$, and for each value X a different observed value of p results, which one uses with (8.43) to compute a 95% confidence interval. R Code 8.18 on the following page computes the 26 possible confidence intervals using (8.43). The value stored in `cover` is a 0 when the parameter ($\pi = 0.70$) is not contained in the confidence interval and a 1 when it is. The values in `prob` are the corresponding binomial probabilities for an observed x . If $X = 15$, then $p = 0.60$, and the lower and upper bounds on a 95% confidence interval using (8.43) are $0.60 - 1.96\sqrt{\frac{0.60 \times (1-0.60)}{25}} = 0.408$ and $0.60 + 1.96\sqrt{\frac{0.60 \times (1-0.60)}{25}} = 0.792$, respectively. These values can be seen in the fifth row of `RES[12:24,]` in R Code 8.18 on the next page. Since $\pi = 0.70$ falls inside the `lcl` and `ucl` values of the fifth row of the output shown for `RES[12:24,]`, the fifth row of `RES[12:24,]` for the variable `cover` is a 1.

R Code 8.18

```

> n <- 25
> alpha <- 0.05
> x <- 0:n
> p <- x/n
> m.err <- qnorm(1 - alpha/2)*sqrt(p*(1 - p)/n)
> lcl <- p - m.err          # lower confidence limit
> ucl <- p + m.err          # upper confidence limit
> PI <- 0.70                 # PI = P(Success)
> prob <- dbinom(x, n, PI)   # binomial probability
> cover <- (PI >= lcl) & (PI <= ucl) # vector of 0s and 1s
> RES <- cbind(x, p, lcl, ucl, prob, cover)
> RES[12:24, ]                # show only rows 12-24

      x     p      lcl      ucl      prob cover
[1,] 11 0.44 0.2454199 0.6345801 0.004215583    0
[2,] 12 0.48 0.2841605 0.6758395 0.011475753    0
[3,] 13 0.52 0.3241605 0.7158395 0.026776757    1
[4,] 14 0.56 0.3654199 0.7545801 0.053553514    1
[5,] 15 0.60 0.4079635 0.7920365 0.091636012    1
[6,] 16 0.64 0.4518435 0.8281565 0.133635851    1
[7,] 17 0.68 0.4971447 0.8628553 0.165079581    1
[8,] 18 0.72 0.5439957 0.8960043 0.171193640    1
[9,] 19 0.76 0.5925865 0.9274135 0.147166462    1
[10,] 20 0.80 0.6432029 0.9567971 0.103016524    1
[11,] 21 0.84 0.6962931 0.9837069 0.057231402    1
[12,] 22 0.88 0.7526174 1.0073826 0.024279989    0
[13,] 23 0.92 0.8136550 1.0263450 0.007389562    0

> # P(Cover) = P(X = 13) + P(X = 14) + P(X = 15) + ... + P(X = 21)
> CP <- sum(dbinom(x[cover], n, PI)) # coverage probability
> CP

[1] 0.9492897

```

The coverage probability, according to (8.44) using (8.43) for a nominal 95% confidence level when $\pi = 0.70$, $C_{25}(0.70)$, is 0.9493.

```

> library(binom)
> binom.coverage(p = 0.70, n = 25, conf = 0.95, method = "asymptotic")

      method     p     n  coverage
1 asymptotic 0.7 25  0.9492897

```

The function `binom.coverage()` returns the same value for coverage probability as the one computed in R Code 8.18.

(b) To compute $C_{25}(0.69)$ for a nominal 95% confidence level using (8.43), one needs to change the value assigned to PI in R Code 8.18 as shown in R Code 8.19 on the facing page.

R Code 8.19

```

> n <- 25
> alpha <- 0.05
> x <- 0:n
> p <- x/n
> m.err <- qnorm(1 - alpha/2)*sqrt(p*(1 - p)/n)
> lcl <- p - m.err          # lower confidence limit
> ucl <- p + m.err          # upper confidence limit
> PI <- 0.69                 # PI = P(Success)
> prob <- dbinom(x, n, PI)   # binomial probability
> cover <- (PI >= lcl) & (PI <= ucl) # vector of 0s and 1s
> RES <- cbind(x, p, lcl, ucl, prob, cover)
> RES[12:23, ]               # show only rows 12-23

      x     p      lcl      ucl      prob cover
[1,] 11 0.44 0.2454199 0.6345801 0.00569489    0
[2,] 12 0.48 0.2841605 0.6758395 0.01478834    0
[3,] 13 0.52 0.3241605 0.7158395 0.03291599    1
[4,] 14 0.56 0.3654199 0.7545801 0.06279825    1
[5,] 15 0.60 0.4079635 0.7920365 0.10250294    1
[6,] 16 0.64 0.4518435 0.8281565 0.14259482    1
[7,] 17 0.68 0.4971447 0.8628553 0.16802919    1
[8,] 18 0.72 0.5439957 0.8960043 0.16622242    1
[9,] 19 0.76 0.5925865 0.9274135 0.13630803    1
[10,] 20 0.80 0.6432029 0.9567971 0.09101859    1
[11,] 21 0.84 0.6962931 0.9837069 0.04823566    0
[12,] 22 0.88 0.7526174 1.0073826 0.01952059    0

> # P(Cover) = P(X = 13) + P(X = 14) + P(X = 15) + ... + P(X = 20)
> CPb <- sum(dbinom(x[cover], n, PI)) # coverage probability
> CPb

[1] 0.9023902

```

From the eleventh row of `RES[12:23,]`, note that $\pi = 0.69$ falls below the `lcl` value. Changing the value of π from 0.70 to 0.69 changes the probability coverage from 0.9493 to 0.9024.

```

> library(binom)
> binom.coverage(p = 0.69, n = 25, conf = 0.95, method = "asymptotic")

      method     p     n coverage
1 asymptotic 0.69 25 0.9023902

```

The function `binom.coverage()` returns the same value for coverage probability as the one computed in R Code 8.19.

(c) In part (b), one saw how a small difference in π changed the coverage probability from 0.9493 (close to the nominal 95% confidence level) to 0.9024. To understand the coverage probability of (8.43) for a fine grid of π values, consider R Code 8.20 on the following page which uses a for loop to compute the coverage probabilities for 2000 evenly spaced values of π from 1/2000 to 1 – 1/2000. The plot of the resulting coverage probabilities versus π values

is shown in Figure 8.12 on the next page. It is clear from Figure 8.12 on the facing page that the coverage probabilities for most values of π fall below the nominal 95% confidence level, and for values of π close to 0 or 1, the corresponding coverage probabilities are close to 0.

R Code 8.20

```
> n <- 25
> alpha <- 0.05
> CL <- 1 - alpha
> z <- qnorm(1 - alpha/2)
> x <- 0:n
> p <- x/n
> m.err <- z*sqrt(p*(1 - p)/n)
> lcl <- p - m.err
> ucl <- p + m.err
> m <- 2000
> PI <- seq(1/m, 1 - 1/m, length = m)
> p.cov <- numeric(m)
> for (i in 1:m)
+ {
+   cover <- (PI[i] >= lcl) & (PI[i] <= ucl)
+   p.rel <- dbinom(x[cover], n, PI[i])
+   p.cov[i] <- sum(p.rel)
+ }
> plot(PI, p.cov, type = "l", ylim = c(0.0, 1.05), main = "n = 25",
+       xlab = expression(pi), ylab = "Coverage Probability")
> lines(c(1/m, 1 - 1/m), c(1 - alpha, 1 - alpha), col = "red",
+        lty = "dashed")
> text(0.5, CL + 0.05, paste("Targeted Confidence Level =", CL))
> #
> binom.plot(n = 25, method = binom.asymp, np = 2000)
```

Note that the function `binom.plot()` denotes the population proportion with p . Consequently, one sees π in the left graph of Figure 8.12 on the next page on the x axis and p in the right graph of Figure 8.12 on the facing page on the x axis.



8.4.1.1 Score Confidence Interval

A more accurate confidence interval for π as judged by coverage probability can be obtained by solving for the values that satisfy (8.45) instead of replacing $\sigma_{\hat{\pi}(\mathbf{x})}$ with its MLE, $\hat{\sigma}_{\hat{\pi}(\mathbf{x})}$. The resulting confidence interval is referred to as a **score confidence interval** or a **Wilson confidence interval**.

$$\mathbb{P}\left(P - z_{1-\alpha/2}\sqrt{\frac{\pi(1-\pi)}{n}} \leq \pi \leq P + z_{1+\alpha/2}\sqrt{\frac{\pi(1-\pi)}{n}}\right) = 1 - \alpha \quad (8.45)$$

Setting

$$-z_{1-\alpha/2} = \frac{p - \pi}{\sqrt{\pi(1-\pi)/n}} \quad (8.46)$$

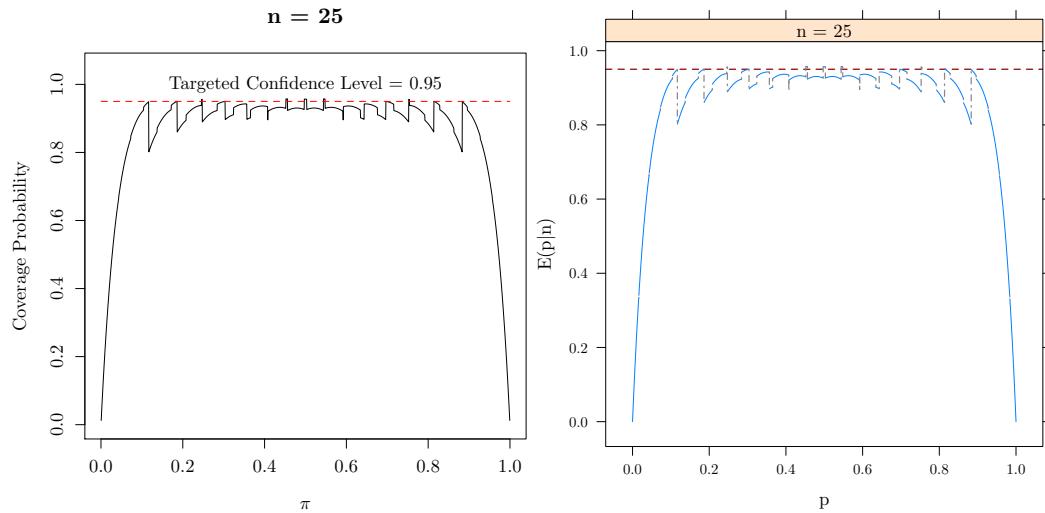


FIGURE 8.12: Coverage probability for a Wald confidence interval using (8.43) with a nominal 95% confidence level. The left graph was created with base graphics and the right graph was created from a call to the function `binom.plot()`.

leads to the quadratic equation

$$(z_{1-\alpha/2}^2 + n)\pi^2 - (2np + z_{1-\alpha/2})\pi + np^2 = 0,$$

which, when solved, returns the lower and upper confidence limits provided in (8.47). The confidence interval in (8.47) is called both a score interval and a Wilson interval in the literature. Note that the same quadratic equation is returned if the right hand side of (8.46) is set to $z_{1-\alpha/2}$.

$$\boxed{CI_{1-\alpha}(\pi) = \left[\frac{p + \frac{z_{1-\alpha/2}^2}{2n} - z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{\left(1 + \frac{z_{1-\alpha/2}^2}{n}\right)}, \frac{p + \frac{z_{1-\alpha/2}^2}{2n} + z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{\left(1 + \frac{z_{1-\alpha/2}^2}{n}\right)} \right]} \quad (8.47)$$

The center of (8.47) (the average of the two end points) can be expressed as a weighted average of the observed proportion, p , and $1/2$. As n increases, more weight is given to p .

$$\frac{p + \frac{z_{1-\alpha/2}^2}{2n}}{1 + \frac{z_{1-\alpha/2}^2}{n}} = \frac{\frac{1}{2n}(2pn + z_{1-\alpha/2}^2)}{\frac{1}{2n}(2n + 2z_{1-\alpha/2}^2)} = \frac{2pn + z_{1-\alpha/2}^2}{2(n + z_{1-\alpha/2}^2)} = p \left(\frac{n}{n + z_{1-\alpha/2}^2} \right) + \frac{1}{2} \left(\frac{z_{1-\alpha/2}^2}{n + z_{1-\alpha/2}^2} \right)$$

If the sample size is large, $z_{1-\alpha/2}^2/2n$ is negligible compared to p , $z_{1-\alpha/2}^2/4n^2$ under the square root is negligible compared to $p(1-p)/n$, and $z_{1-\alpha/2}^2/n$ is negligible compared to 1. If the negligible terms are ignored, the confidence interval formula in (8.43) emerges. When

R uses (8.47) to construct confidence intervals, under certain conditions, it also applies a Yates' continuity correction to p so that the p used in the lower limit is $p_L = p - \frac{1}{2n}$ and the p used in the upper limit is $p_U = p + \frac{1}{2n}$. The Wilson or score interval in (8.47) is preferred over the Wald confidence interval given in (8.43) since the Wilson interval returns confidence intervals whose coverage probability is closer to the user specified $1 - \alpha$ level. See Figure 8.13 on the next page. The R function `prop.test()` when used with appropriate arguments will compute a confidence interval using (8.47). The `binom.confint()` function from the `binom` package when passed the argument `methods = "wilson"` will also compute a confidence interval using (8.47).

8.4.1.2 Agresti-Coull Confidence Interval for the Population Proportion

Agresti and Coull (1998) proposed a nice compromise between the Wilson interval given in (8.47) and the Wald confidence interval given in (8.43) that has a coverage probability close to the Wilson interval over a wide range of parameters and sample sizes. The Agresti-Coull confidence interval is

$$CI_{1-\alpha}(\pi) = \left[\tilde{p} - z_{1-\alpha/2} \sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}}, \tilde{p} + z_{1-\alpha/2} \sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}} \right] \quad (8.48)$$

where X denotes the number of successes in a sample of size n , $\tilde{n} = n + z_{1-\alpha/2}^2$, and $\tilde{p} = \frac{1}{\tilde{n}}(X + \frac{1}{2}z_{1-\alpha/2}^2)$. The confidence interval given in (8.48) is referred to as the **Agresti-Coull** confidence interval. The `binom.confint()` function from the `binom` package when passed the argument `methods = "ac"` will compute a confidence interval using (8.48).

8.4.1.3 Clopper-Pearson Interval for the Population Proportion

The Clopper-Pearson interval is often referred to as an exact confidence interval for π . The Clopper-Pearson interval is not exact with respect to coverage probability. In fact, the coverage probability of the Clopper-Pearson interval is always at or above the nominal confidence level. It is exact because it uses the exact distribution (binomial) to compute a confidence interval rather than a normal approximation to the exact distribution as do the Wald, Wilson, and Agresti-Coull confidence intervals. The Clopper-Pearson confidence interval is

$$CI_{1-\alpha}(\pi) = [\beta_{\alpha/2,x,n-x+1}, \beta_{1-\alpha/2,x+1,n-x}] \quad (8.49)$$

where x is the number out of n observed successes and $\beta_{\alpha/2,x,n-x+1}$ and $\beta_{1-\alpha/2,x+1,n-x}$ are the $\alpha/2$ and $1 - \alpha/2$ percentiles of the standard $\beta(\alpha,\beta)$ distribution. The function `binom.confint()` from the `binom` package will return a Clopper-Pearson confidence interval when the user provides the argument `methods = "exact"`.

8.4.1.4 So Which Confidence Interval Do I Use?

The average coverage probability for the values displayed in Figure 8.13 on the facing page under the Agresti-Coull panel, Clopper-Pearson panel, Wald panel, and Wilson panel are 0.9618, 0.977, 0.8459, and 0.953, respectively. Similar results are returned for different values of n and different confidence levels. Among the Wald, Agresti-Coull, Clopper-Pearson,

and Wilson confidence intervals, the Wilson confidence intervals generally have an average coverage probability closer to the nominal confidence level for a wide range of n and confidence levels. Of course, one can always increase the probability coverage of a given method by increasing the width of the confidence interval. So does one of the four methods have uniformly smaller expected length? Unfortunately, the answer is no; however, Figure 8.14 on the next page indicates that the Wilson interval has one of the smaller expected lengths except at values close to 0 and 1 where the expected lengths of the Wald interval are shorter. If the user wants a coverage probability close to the advertised confidence level, they should use a Wilson interval. If the user wants to make sure the coverage probability never drops below the advertised confidence level they should use the Clopper-Pearson interval.

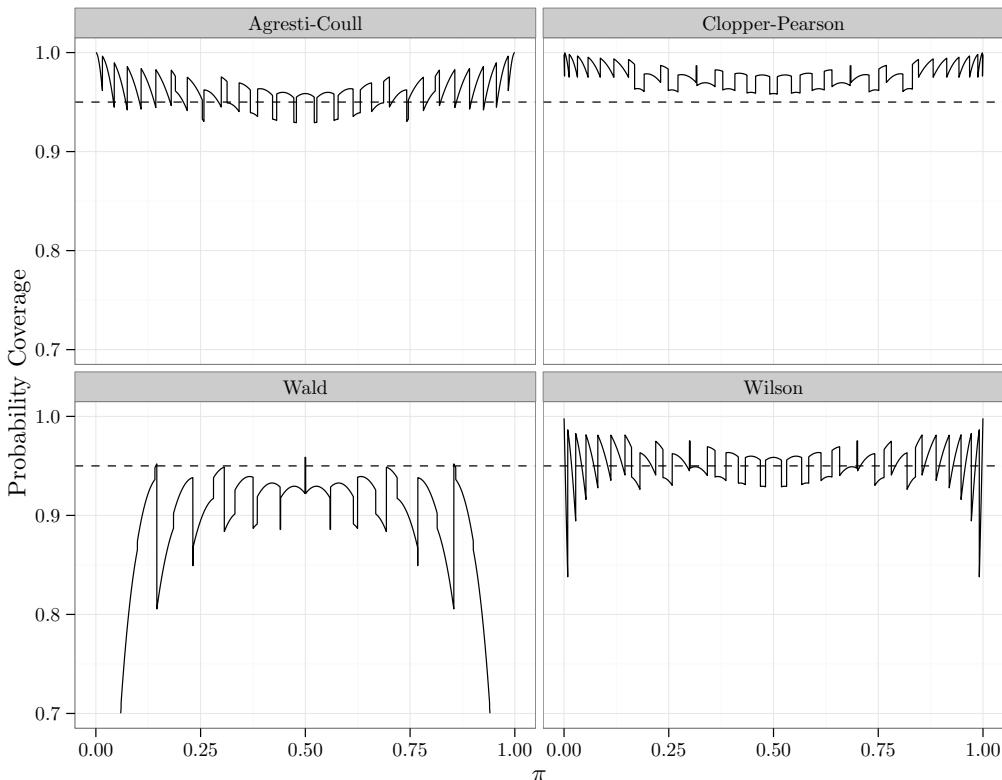


FIGURE 8.13: Coverage probability for the population proportion using the Agresti-Coull, Clopper-Pearson, Wald, and Wilson 95% confidence intervals when $n = 20$

Example 8.25 A professor is interested in what percent of students pass an introductory statistics class. He takes a random sample of 40 introductory statistics students and finds that 26 passed. Help the professor construct 95% confidence intervals for the true percent of students who pass using

- (a) The Wald confidence interval for π based on the MLE of $\sigma_{\hat{\pi}(x)}$ given in (8.43).
- (b) The Wilson confidence interval for π given in (8.47).

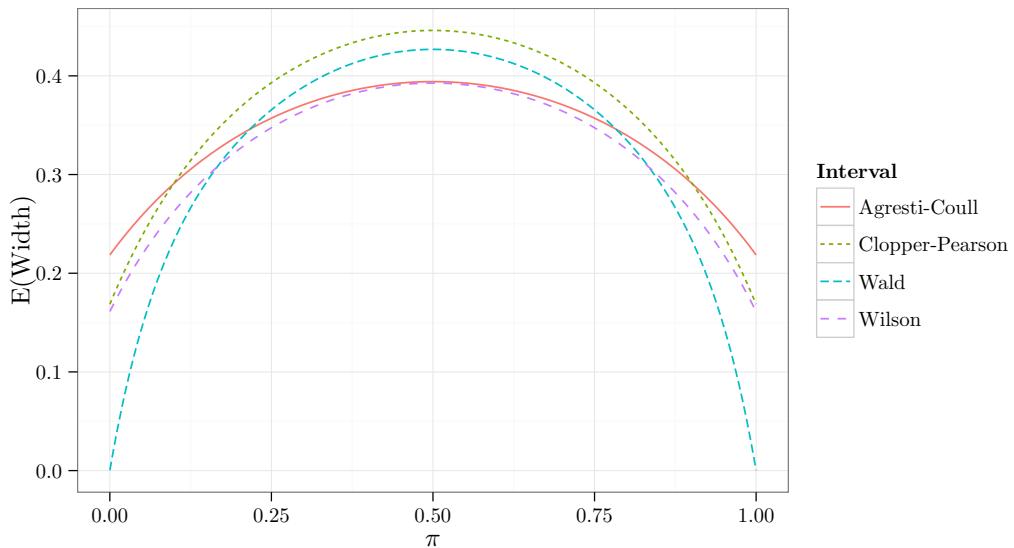


FIGURE 8.14: Expected width of the Agresti-Coull, Clopper-Pearson, Wald, and Wilson 95% confidence intervals when $n = 20$

- (c) The Wilson confidence interval for π with continuity corrections applied to the ps (use p_L and p_U).
- (d) The Agresti-Coull confidence interval for π .
- (e) The Clopper-Pearson confidence interval for π .

Solution: Because all of the confidence intervals are to have 95% confidence, $z_{1-\alpha/2} = z_{1-0.05/2} = z_{0.975} = 1.96$.

- (a) The Wald 95% confidence interval for π is computed using $p = \frac{26}{40} = 0.65$ as

$$\begin{aligned} CI_{0.95}(\pi) &= \left[p - z_{0.975} \sqrt{\frac{p(1-p)}{n}}, p + z_{0.975} \sqrt{\frac{p(1-p)}{n}} \right] \\ &= \left[0.65 - 1.96 \sqrt{\frac{(0.65)(1-0.65)}{40}}, 0.65 + 1.96 \sqrt{\frac{(0.65)(1-0.65)}{40}} \right] \\ &= [0.5022, 0.7978]. \end{aligned}$$

The requested interval is computed in R Code 8.21 using (8.43) as well as the function `binom.confint()`.

R Code 8.21

```
> n <- 40
> x <- 26
> p <- x/n          # sample proportion passing
> z <- qnorm(0.975) # z_{0.975}
```

```

> n <- 40
> CI <- p + c(-1, 1)*z*sqrt(p*(1 - p)/n)
> CI
[1] 0.5021883 0.7978117

> # Or
> library(binom)
> binom.confint(x = 26, n = 40, conf.level = 0.95, methods = "asymptotic")

      method  x  n mean      lower      upper
1 asymptotic 26 40 0.65 0.5021883 0.7978117

```

(b) The Wilson 95% confidence interval for π is

$$\begin{aligned}
CI_{0.95}(\pi) &= \left[\frac{p + \frac{z_{1-\alpha/2}^2}{2n} - z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{\left(1 + \frac{z_{1-\alpha/2}^2}{n}\right)}, \right. \\
&\quad \left. \frac{p + \frac{z_{1-\alpha/2}^2}{2n} + z_{1-\alpha/2} \sqrt{\frac{p(1-p)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{\left(1 + \frac{z_{1-\alpha/2}^2}{n}\right)} \right] \\
&= \left[\frac{0.65 + \frac{1.96^2}{(2)(40)} - 1.96 \sqrt{\frac{0.65(1-0.65)}{40} + \frac{1.96^2}{(4)(40^2)}}}{\left(1 + \frac{1.96^2}{40}\right)}, \right. \\
&\quad \left. \frac{0.65 + \frac{1.96^2}{(2)(40)} + 1.96 \sqrt{\frac{0.65(1-0.65)}{40} + \frac{1.96^2}{(4)(40^2)}}}{\left(1 + \frac{1.96^2}{40}\right)} \right] = [0.4951, 0.7787]
\end{aligned}$$

The requested interval is computed in R Code 8.22 using the functions `prop.test()` and `binom.confint()`.

R Code 8.22

```

> prop.test(x = 26, n = 40, correct = FALSE, conf.level = 0.95)$conf
[1] 0.4950588 0.7786547
attr(,"conf.level")
[1] 0.95

> # Or
> binom.confint(x = 26, n = 40, conf.level = 0.95, methods = "wilson")

      method  x  n mean      lower      upper
1 wilson 26 40 0.65 0.4950588 0.7786547

```

(c) Using the values $p_L = p - \frac{1}{2n} = 0.65 - \frac{1}{(2)(40)} = 0.6375$ and $p_U = p + \frac{1}{2n} = 0.65 + \frac{1}{(2)(40)} =$

0.6625, the 95% Wilson confidence interval with continuity corrections is

$$\begin{aligned}
 CI_{0.95}(\pi) &= \left[\frac{p_L + \frac{z_{1-\alpha/2}^2}{2n} - z_{1-\alpha/2} \sqrt{\frac{p_L(1-p_L)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{\left(1 + \frac{z_{1-\alpha/2}^2}{n}\right)}, \right. \\
 &\quad \left. \frac{p_U + \frac{z_{1-\alpha/2}^2}{2n} + z_{1-\alpha/2} \sqrt{\frac{p_U(1-p_U)}{n} + \frac{z_{1-\alpha/2}^2}{4n^2}}}{\left(1 + \frac{z_{1-\alpha/2}^2}{n}\right)} \right] \\
 &= \left[\frac{0.6375 + \frac{1.96^2}{(2)(40)} - 1.96 \sqrt{\frac{0.6375(1-0.6375)}{40} + \frac{1.96^2}{(4)(40^2)}}}{\left(1 + \frac{1.96^2}{40}\right)}, \right. \\
 &\quad \left. \frac{0.6625 + \frac{1.96^2}{(2)(40)} + 1.96 \sqrt{\frac{0.6625(1-0.6625)}{40} + \frac{1.96^2}{(4)(40^2)}}}{\left(1 + \frac{1.96^2}{40}\right)} \right] = [0.4826, 0.7890].
 \end{aligned}$$

The interval with continuity correction is computed with R by typing

```
> prop.test(x = 26, n = 40, correct = TRUE, conf.level = 0.95)$conf
[1] 0.4826446 0.7889540
attr("conf.level")
[1] 0.95
```

(d) The Agresti-Coull 95% confidence interval for π is computed using $\tilde{n} = 26 + z_{0.975}^2 = 29.8415$ and $\tilde{p} = \frac{1}{\tilde{n}} (26 + \frac{1}{2} z_{0.975}^2) = 0.6369$ as

$$\begin{aligned}
 CI_{0.95}(\pi) &= \left[\tilde{p} - z_{0.975} \sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}}, \tilde{p} + z_{0.975} \sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{n}}} \right] \\
 &= \left[0.6369 - 1.96 \sqrt{\frac{(0.6369)(1-0.6369)}{43.8415}}, \right. \\
 &\quad \left. 0.6369 + 1.96 \sqrt{\frac{(0.6369)(1-0.6369)}{43.8415}} \right] \\
 &= [0.4945, 0.7792].
 \end{aligned}$$

The requested interval is computed in R Code 8.23 using (8.48) as well as the function `binom.confint()`.

R Code 8.23

```
> z <- qnorm(0.975)
> ntilde <- 40 + z^2
> ntilde
[1] 43.84146

> ptild <- (1/ntilde)*(26 + 1/2*z^2)
> ptild
```

```
[1] 0.6368568
> me <- z*sqrt(ptilde*(1 - ptilde)/ntilde)
> CI <- ptilde + c(-1, 1)*me
> CI

[1] 0.4945041 0.7792094

> # Or
> binom.confint(x = 26, n = 40, methods = "ac")

      method  x  n mean      lower      upper
1 agresti-coull 26 40 0.65 0.4945041 0.7792094
```

- (e) The Clopper-Pearson 95% confidence interval for π is computed using $x = 26$ and $n = 40$ as

$$\begin{aligned} CI_{0.95}(\pi) &= [\beta_{\alpha/2,x,n-x+1}, \beta_{1-\alpha/2,x+1,n-x}] \\ &= [\beta_{0.05/2,26,40-26+1}, \beta_{1-0.05/2,26+1,40-26}] \\ &= [0.4832, 0.7937]. \end{aligned}$$

The requested interval is computed in R Code 8.24 using the function `qbeta()` to find the values for (8.49) as well as the function `binom.confint()`.

R Code 8.24

```
> CI <- c(qbeta(0.025, 26, 40 - 26 + 1), qbeta(0.975, 26 + 1, 40 - 26))
> CI

[1] 0.4831555 0.7937175

> # Or
> binom.confint(x = 26, n = 40, conf = 0.95, method = "exact")

      method  x  n mean      lower      upper
1  exact 26 40 0.65 0.4831555 0.7937175
```

So, depending on which confidence interval the professor prefers, he can be 95% confident that the proportion of students who pass lies in $[0.5022, 0.7978]$, $[0.4951, 0.7787]$, $[0.4826, 0.7890]$, $[0.4945, 0.7792]$, or $[0.4832, 0.7937]$. ■

Example 8.26 A computer firm would like to construct three Wilson confidence intervals for the proportion of supermarkets that use a computerized database to manage their warehouses. Suppose 200 supermarkets are surveyed, and 157 of the 200 supermarkets have computerized inventories. Construct 90%, 95%, and 99% confidence intervals for the true proportion of supermarkets that use a computerized database to manage the inventory of their warehouses.

Solution: The requested Wilson confidence intervals using (8.47) are computed in R Code 8.25 on the following page.

R Code 8.25

```
> CI1 <- prop.test(x = 157, n = 200, conf = 0.90, correct = FALSE)$conf
> CI1

[1] 0.7335816 0.8288105
attr(,"conf.level")
[1] 0.9

> # Or
> binom.confint(x = 157, n = 200, conf = 0.90, methods = "wilson")

  method   x   n   mean     lower      upper
1 wilson 157 200 0.785 0.7335816 0.8288105

> CI2 <- prop.test(x = 157, n = 200, conf = 0.95, correct = FALSE)$conf
> CI2

[1] 0.7229769 0.8362812
attr(,"conf.level")
[1] 0.95

> # Or
> binom.confint(x = 157, n = 200, conf = 0.95, methods = "wilson")

  method   x   n   mean     lower      upper
1 wilson 157 200 0.785 0.7229769 0.8362812

> CI3 <- prop.test(x = 157, n = 200, conf = 0.99, correct = FALSE)$conf
> CI3

[1] 0.7016667 0.8500310
attr(,"conf.level")
[1] 0.99

> # Or
> binom.confint(x = 157, n = 200, conf = 0.99, methods = "wilson")

  method   x   n   mean     lower      upper
1 wilson 157 200 0.785 0.7016667 0.850031
```

The computer firm is 90% confident the population proportion of supermarkets that use a computerized database to manage their warehouses lies in [0.7336, 0.8288], 95% confident this population proportion lies in [0.723, 0.8363], and 99% confident this population proportion lies in [0.7017, 0.85]. Take special note that the widths of the confidence intervals increase as the confidence level increases. ■

Example 8.27 ▷ Confidence Interval and Sample Size for π The Department of Agriculture wants to estimate the proportion of rural farm owners who are under 40 years of age. They take a random sample of 2000 farms and find that 400 of the 2000 owners are under the age of 40.

- (a) Construct a 95% Wald confidence interval for π using the asymptotic confidence interval for π based on the MLE of $\hat{\sigma}_{\hat{\pi}(x)}$ given in (8.43).

- (b) Determine the required sample size so that the maximum margin of error is within 0.015 of the true value of π for a 95% confidence level.

Solution: Note that $p = \frac{400}{2000} = 0.20$.

- (a) A 95% Wald confidence interval for π using (8.43) is

$$\begin{aligned} CI_{0.95}(\pi) &= \left[p - z_{1-0.05/2} \sqrt{\frac{p(1-p)}{n}}, p + z_{1-0.05/2} \sqrt{\frac{p(1-p)}{n}} \right] \\ &= \left[0.2 - (1.96) \sqrt{\frac{0.2(1-0.2)}{2000}}, 0.2 + (1.96) \sqrt{\frac{0.2(1-0.2)}{2000}} \right] \\ &= [0.1825, 0.2175]. \end{aligned}$$

To verify the confidence interval with R, type

```
> x <- 400
> n <- 2000
> p <- x/n
> z <- qnorm(0.975) # z_{0.975}
> CI <- p + c(-1, 1)*z*sqrt(p*(1 - p)/n)
> CI
[1] 0.1824695 0.2175305

> # Or
> binom.confint(x = 400, n = 2000, conf = 0.95, methods = "asymptotic")
   method   x     n mean      lower      upper
1 asymptotic 400 2000  0.2 0.1824695 0.2175305
```

- (b) In order to construct a confidence interval such that the maximum margin of error does not exceed 0.015, one needs to ensure that

$$(1.96) \sqrt{\frac{p(1-p)}{n}} < 0.015. \quad (8.50)$$

To maximize the margin of error, use $p = \frac{1}{2}$ regardless of any prior information concerning p . Using a value for p of $\frac{1}{2}$ will ensure the margin of error is maximized at a given confidence level. To see why this is true, consider plotting $p \times (1-p)$ versus p . This can be done by typing

```
> f <- function(x){sqrt(x*(1 - x))} # x takes place of p in f
> curve(expr = f, from = 0, to = 1, n = 500)
```

Consequently, solving (8.50) for n yields 4268.4. To guarantee the maximum margin of error is within 0.015 at a 95% confidence level, always take the ceiling of n (use the next largest integer). In this case, a sample of size 4269 will guarantee the maximum margin of error will be less than 0.015 at a 95% confidence level. That is,

$$(1.96) \sqrt{\frac{\left(\frac{1}{2}\right)\left(\frac{1}{2}\right)}{4269}} = 0.01499902 < 0.015.$$

The function `nsize()` from the `PASWR2` package can also be used to find the required sample size as seen in R Code 8.26.

R Code 8.26

```
> nsizer(b = 0.015, type = "pi", conf.level = 0.95)
```

The required sample size (n) to estimate the population proportion of successes with a 0.95 confidence interval so that the margin of error is no more than 0.015 is 4269 .



8.4.2 Confidence Interval for a Difference in Population Proportions

In this section, the focus is on two populations, X and Y , from which random samples of sizes n_X and n_Y , respectively, are taken. If π_X and π_Y are the population proportions of successes and P_X and P_Y are the respective sample proportions of successes, then the resulting sampling distributions of P_X and P_Y , provided n_X and n_Y are sufficiently large, are approximately normal. That is,

$$P_X \sim N\left(\pi_X, \sqrt{\frac{\pi_X(1-\pi_X)}{n_X}}\right) \quad \text{and} \quad P_Y \sim N\left(\pi_Y, \sqrt{\frac{\pi_Y(1-\pi_Y)}{n_Y}}\right).$$

Since the sampling distributions of both P_X and P_Y are approximately normal, the sampling distribution for the difference between P_X and P_Y will also be approximately normal. Specifically,

$$P_X - P_Y \sim N\left(\pi_X - \pi_Y, \sqrt{\frac{\pi_X(1-\pi_X)}{n_X} + \frac{\pi_Y(1-\pi_Y)}{n_Y}}\right) \quad (8.51)$$

according to Theorem 5.1 on page 320.

Using a similar approach to the one presented for the construction of a confidence interval for the difference between two means, construct a $(1 - \alpha) \cdot 100\%$ asymptotic confidence interval for $\pi_X - \pi_Y$ as shown in (8.52). The rationale for replacing π_X and π_Y with p_X and p_Y in $\sqrt{\frac{\pi_X(1-\pi_X)}{n_X} + \frac{\pi_Y(1-\pi_Y)}{n_Y}}$ is the invariance property of maximum likelihood estimators (property 2 on page 435), where $\hat{\pi}_X = P_X$ and $\hat{\pi}_Y = P_Y$.

$$CI_{1-\alpha}(\pi_X - \pi_Y) = \left[(p_X - p_Y) - z_{1-\alpha/2} \sqrt{\frac{p_X(1-p_X)}{n_X} + \frac{p_Y(1-p_Y)}{n_Y}}, (p_X - p_Y) + z_{1-\alpha/2} \sqrt{\frac{p_X(1-p_X)}{n_X} + \frac{p_Y(1-p_Y)}{n_Y}} \right] \quad (8.52)$$

It is generally advisable to use the continuity correction $\frac{1}{2} \left(\frac{1}{n_X} + \frac{1}{n_Y} \right)$ with (8.52) any time

$$|p_X - p_Y| > \frac{1}{2} \left(\frac{1}{n_X} + \frac{1}{n_Y} \right). \quad (8.53)$$

The continuity correction is subtracted and added to the lower and upper confidence limits of (8.52), respectively. The R function `prop.test()` automatically applies the continuity correction when (8.53) is satisfied provided the user does not issue the argument `correct = FALSE`.

Example 8.28 A company wants to see if a certain change in the process for manufacturing component parts is beneficial. Samples are taken using both the existing and the new procedure to determine if the new process results in an improvement. The first sample is taken before the change has been implemented, and the second sample is taken after the change has been implemented. If 70 of 1400 elements are found to be defective in the first sample and 90 of 2000 elements are found to be defective in the second sample, find a 95% confidence interval for the true difference in the proportion of defective components between the existing and the new processes.

Solution: The sample proportions of successes are $p_X = \frac{70}{1400} = 0.05$ and $p_Y = \frac{90}{2000} = 0.045$. Using (8.52), the 95% confidence interval for the true difference in the proportions of defective components between the existing and the new processes is given in (8.54). Since the confidence interval contains 0, there is no reason to suspect the new procedure significantly reduces the proportion of defective items.

$$\begin{aligned}
 CI_{0.95}(\pi_X - \pi_Y) &= \\
 &\left[(0.05 - 0.045) - 1.96\sqrt{\frac{(0.05)(1 - 0.05)}{1400} + \frac{(0.045)(1 - 0.045)}{2000}}, \right. \\
 &\quad \left. (0.05 - 0.045) + 1.96\sqrt{\frac{(0.05)(1 - 0.05)}{1400} + \frac{(0.045)(1 - 0.045)}{2000}} \right] \\
 &= [-0.0096, 0.0196]. \tag{8.54}
 \end{aligned}$$

To construct a 95% confidence interval for $\pi_X - \pi_Y$ using (8.52), key in

```

> prop.test(x = c(70, 90), n = c(1400, 2000), conf.level = 0.95,
+           correct = FALSE)$conf
[1] -0.009590358  0.019590358
attr("conf.level")
[1] 0.95

```

Since $|p_X - p_Y| = |0.05 - 0.045| = 0.005 > \frac{1}{2}(\frac{1}{1400} + \frac{1}{2000}) = 0.0006$, 0.0006 should be subtracted from and added to the smaller and larger values reported in (8.54), respectively. Consequently, a continuity corrected 95% confidence interval for the true difference in the proportion of defective components between the existing and the new process is

$$CI_{0.95}(\pi_X - \pi_Y) = [-0.0096 - 0.0006, 0.0196 + 0.0006] = [-0.0102, 0.0202].$$

To produce the continuity corrected interval with R, enter

```

> prop.test(x = c(70, 90), n = c(1400, 2000), conf.level = 0.95,
+           correct = TRUE)$conf
[1] -0.0101975  0.0201975
attr("conf.level")
[1] 0.95

```



8.4.3 Confidence Interval for the Mean of a Poisson Random Variable

Recall that a Poisson random variable counts the number of occurrences over some period of time or region of space where the occurrences are relatively rare. When collecting occurrences from a Poisson distribution, it follows that the sample values will have a positive skew, since the Poisson distribution itself is skewed to the right. This will often rule out confidence interval formulas that require normality assumptions; however, for sufficiently large samples, one can use (8.37) on page 490 to construct confidence limits for the mean of a Poisson distribution. When using (8.37) for confidence interval construction for the mean of a Poisson random variable, first find the maximum likelihood estimator of λ . In Example 7.20 on page 426, the maximum likelihood estimator of λ for a Poisson distribution was found to be \bar{X} . That is, $\hat{\lambda}(\mathbf{X}) = \bar{X}$. To calculate the Fisher information $I_n(\lambda)$ using (7.50) on page 433 requires knowledge of the second-order derivative of the log-likelihood function with respect to λ . This second-order derivative was computed in Example 7.20 and is reproduced here for the reader's benefit:

$$\frac{\partial^2 \ln L(\lambda|\mathbf{X})}{\partial \lambda^2} = \frac{-\sum_{i=1}^n x_i}{\lambda^2}. \quad (8.55)$$

Taking the expected value of (8.55) yields the following, from which the Fisher information, $I_n(\lambda)^{-1} = \frac{\lambda}{n}$, is obtained:

$$-E\left[\frac{\partial^2 \ln L(\lambda|\mathbf{X})}{\partial \lambda^2}\right] = \frac{n\lambda}{\lambda^2} = \frac{n}{\lambda}. \quad (8.56)$$

Taking advantage of the asymptotic properties of MLE estimators allows one to write

$$\hat{\lambda}(\mathbf{X}) = \bar{X} \sim N\left(\lambda, \sqrt{\frac{\lambda}{n}}\right) \text{ as } n \rightarrow \infty.$$

One may then use (8.37), the confidence interval formula for MLEs, to construct a $(1 - \alpha) \cdot 100\%$ asymptotic confidence interval for λ as shown here where $\hat{\sigma}_{\hat{\lambda}(\mathbf{x})} = \sqrt{\frac{\bar{x}}{n}}$:

$$CI_{1-\alpha}(\lambda) = \left[\bar{x} - z_{1-\alpha/2} \sqrt{\frac{\bar{x}}{n}}, \bar{x} + z_{1-\alpha/2} \sqrt{\frac{\bar{x}}{n}} \right]. \quad (8.57)$$

One could obtain a similar confidence interval by recognizing that \bar{X} has a normal distribution with parameters μ and $\frac{\sigma}{\sqrt{n}}$ for large sample sizes according to the Central Limit Theorem. Since the mean for a Poisson is λ and the standard deviation of a Poisson random variable is $\sqrt{\lambda}$, it follows that

$$\bar{X}_{Pois} \sim N\left(\lambda, \frac{\sqrt{\lambda}}{\sqrt{n}}\right).$$

Example 8.29 Example 4.4 on page 258 provided evidence to suggest the number of goals scored in the regulation 90-minute periods of World Cup soccer matches from 1990 to

2002 have a Poisson distribution. Use the information in column `goals` of the data frame `SOCCKER` to construct a 90% confidence interval for the mean number of goals scored during a 90 minute regulation period.

Solution: The 90% confidence interval for λ is constructed using (8.57):

$$\begin{aligned} CI_{0.90}(\lambda) &= \left[\bar{x} - z_{1-0.10/2} \sqrt{\frac{\bar{x}}{n}}, \bar{x} + z_{1-0.10/2} \sqrt{\frac{\bar{x}}{n}} \right] \\ &= \left[2.4784 - 1.6449 \sqrt{\frac{2.4784}{232}}, 2.4784 + 1.6449 \sqrt{\frac{2.4784}{232}} \right] \\ &= [2.3084, 2.6485]. \end{aligned} \quad (8.58)$$

To compute the values in (8.58) with R, use the data frame `SOCCKER` as shown in R Code 8.27

R Code 8.27

```
> n <- sum(!is.na(SOCCKER$game)) # number of games
> xbar <- mean(SOCCKER$goals, na.rm = TRUE)
> z <- qnorm(0.95) # z_{0.95}
> CI <- xbar + c(-1, 1) * z * sqrt(xbar/n)
> CI

[1] 2.308439 2.648458
```

So, one is 90% confident the mean number of goals scored in a World Cup soccer match lies in [2.3084, 2.6485]. █

8.5 Problems

1. Is $[\bar{x} - 3, \bar{x} + 3]$ a confidence interval for the population mean of a normal distribution? Why or why not?
2. Explain how to construct a 95% confidence interval for the population mean of a normal distribution if σ is known.
3. Given a random sample $\{X_1, X_2, \dots, X_n\}$ from a normal population $N(\mu, \sigma)$, where σ is known:
 - (a) What is the confidence level for the interval $\bar{x} \pm 2.053749 \frac{\sigma}{\sqrt{n}}$?
 - (b) What is the confidence level for the interval $\bar{x} \pm 1.405072 \frac{\sigma}{\sqrt{n}}$?
 - (c) What is the value of the percentile $z_{\alpha/2}$ for a 99% confidence interval?
4. Given a random sample $\{X_1, X_2, \dots, X_n\}$ from a normal population $N(\mu, \sigma)$, where σ is known, consider the confidence interval $\bar{x} \pm z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}$ for μ .
 - (a) Given a fixed sample size n , explain the relationship between the confidence level and the precision of the confidence interval.
 - (b) Given a confidence level $(1 - \alpha)\%$, explain how the precision of the confidence interval changes with the sample size.
5. Given a normal population with known variance σ^2 , by what factor must the sample size be increased to reduce the length of a confidence interval for the mean by a factor of k ?
6. A historic data set studied by R.A. Fisher is the measurements in centimeters of four flower parts (sepal length, sepal width, petal length, and petal width) on 50 specimens for each of three species of irises (*Setosa*, *Versicolor*, and *Virginica*). The data are stored in the data frame **iris** (Fisher, 1936).
 - (a) Analyze the sepal lengths for *Setosa*, *Versicolor*, and *Virginica* irises, and comment on the characteristics of their distributions.
 - (b) Based on the analysis from part (a), construct an appropriate 99% confidence interval for the mean sepal length of Setosa irises.
7. Surface-water salinity measurements were taken in a bottom-sampling project in White-water Bay, Florida. These data are stored in the data frame **SALINITY** in the **PASWR2** package. Geographic considerations lead geologists to believe that the salinity variation should be normally distributed. If this is true, it means there is free mixing and interchange between open marine water and fresh water entering the bay (Davis, 1986).
 - (a) Construct a quantile-quantile plot of the data. Does this plot rule out normality?
 - (b) Construct a 95% confidence interval for the mean salinity variation.

8. The survival times in weeks for 20 male rats that were exposed to a high level of radiation are

152	152	115	109	137	88	94	77	160	165
125	40	128	123	136	101	62	153	83	69

Data are from Lawless (1982) and are stored in the data frame **RAT**.

- (a) Construct a quantile-quantile plot of the survival times. Based on the quantile-quantile plot, can normality be ruled out?
- (b) Construct a 92% confidence interval for the average survival time for male rats exposed to high levels of radiation.

9. A large company wants to estimate the proportion of its accounts that are paid on time.

- (a) How large a sample is needed to estimate the true proportion within 3% with a 96% confidence level?
- (b) Suppose 650 out of 800 accounts are paid on time. Construct 95% confidence intervals for the true proportion of accounts that are paid on time using an asymptotic confidence interval, a score confidence interval, an Agresti-Coull confidence interval, and the Clopper-Pearson confidence interval.

10. In a study conducted at Appalachian State University, students used digital oral thermometers to record their temperatures each day they came to class. A randomly selected day of student temperatures is provided in the following table and in the data frame **STATTEMPS**. Information is also provided with regard to subject gender and the hour of the day when the students' temperatures were measured.

	8 a.m. Class			9 a.m. Class		
Males	92.7	94.1	96.5	94.1	96.0	98.2
	93.2	97.1	93.7	96.5	94.4	
Females	96.9	94.0	93.7	96.5	94.3	93.9
	93.9	93.5	97.0	96.5	95.6	98.2
	97.2	92.0	96.6	96.4	96.3	95.1
	94.9	92.1		97.1	96.6	96.8

- (a) Construct a 95% confidence interval for the true average temperature difference between females and males. Does the interval contain the value zero? What does this suggest about gender temperature differences?
- (b) Construct a 95% confidence interval for the true average temperature difference between students taking their temperatures at 8 a.m. and students taking their temperatures at 9 a.m. Give a reason why one group appears to have a higher temperature reading.

11. The Cosmed K4b² is a portable metabolic system. A study at Appalachian State University compared the metabolic values obtained from the Cosmed K4b² to those of a

reference unit (Amatek) over a range of workloads from easy to maximal to test the validity and reliability of the Cosmed K4b². A small portion of the results for VO₂ (ml/kg/min) measurements taken at a 150 watt workload are stored in data frame **COSAMA** and in the following table:

Subject	Cosmed	Amatek	Subject	Cosmed	Amatek
1	31.71	31.20	8	30.33	27.95
2	33.96	29.15	9	30.78	29.08
3	30.03	27.88	10	30.78	28.74
4	24.42	22.79	11	31.84	28.75
5	29.07	27.00	12	22.80	20.20
6	28.42	28.09	13	28.99	29.25
7	31.90	32.66	14	30.80	29.13

- (a) Construct a quantile-quantile plot for the between-system differences.
 - (b) Are the VO₂ values reported for Cosmed and Amatek independent?
 - (c) Construct a 95% confidence interval for the average VO₂ system difference.
12. Let $\{X_1, \dots, X_{19}\}$ and $\{Y_1, \dots, Y_{15}\}$ be two random samples from a $N(\mu_X, \sigma)$ and a $N(\mu_Y, \sigma)$, respectively. Suppose that $\bar{x} = 57.3$, $s_X^2 = 8.3$, $\bar{y} = 65.6$, and $s_Y^2 = 9.7$. Find a 96% confidence interval for μ_X , μ_Y , and $\mu_X - \mu_Y$.
13. The water consumption in liters per family per day in a given city is a normally distributed random variable with unknown variance. Consider the following confidence intervals for the population mean obtained from a random sample of size n :
- $$[374.209, 545.791], \quad [340.926, 579.074], \quad [389.548, 530.452].$$
- (a) Find the value of the sample mean.
 - (b) If the intervals are obtained from the same random sample, match the confidence levels 90%, 95% and 99% with the corresponding confidence intervals.
14. The best-paid 20 tennis players in the world have earned millions of dollars during their careers and are famous for having won some of the four “Grand Slam” tournaments. Somewhat less famous players who are in positions 20 through 100 in the earnings’ rankings have also garnered large sums. The following data (in millions of dollars) correspond to the earnings of 15 randomly selected players classified somewhere in positions 20 through 100. They are also stored in the data frame **TOP20**.

10.10	8.80	8.64	7.67	6.34	6.03	5.90	5.68
5.51	5.38	5.31	4.92	4.54	4.02	3.86	

Compute a 94% confidence interval for the average earnings of players classified between positions 20 and 100 of the ranking. (Source: <http://www.atptennis.com/en/>) Be sure to apply an appropriate finite population correction factor to the margin of error.

15. The following data is the amount of nuclear energy (in TOE, tons of oil equivalent) produced in 12 randomly selected European countries during 2003. The values are also stored in the data frame **TOE**. In 2003, the EU was comprised of 15 countries. Be sure to apply an appropriate finite population correction factor to the margin of error.

12222	6674	15961	3994	2841	1036
1343	4608	5864	17390	22877	4457

Compute a 95% confidence interval for the 2003 mean European TOE assuming the amount of nuclear energy is normally distributed.

16. A group of engineers working with physicians in a research hospital is developing a new device to measure blood glucose levels. Based on measurements taken from patients in a previous study, the physicians assert that the new device provides blood glucose levels slightly higher than those provided by the old device. To corroborate their suspicion, 15 diabetic patients were randomly selected, and their blood glucose levels were measured with both the new and the old devices. The measurements, in mg/100 ml, appear in the following table and are stored in the data frame **GLUCOSE**:

Blood glucose levels

Patient	Old	New	Patient	Old	New
Patient 1	182.47	195.64	Patient 9	179.04	195.25
Patient 2	175.53	196.31	Patient 10	180.50	194.48
Patient 3	181.71	190.33	Patient 11	182.15	197.33
Patient 4	179.03	192.90	Patient 12	183.55	193.81
Patient 5	177.28	193.24	Patient 13	180.86	198.03
Patient 6	177.49	193.05	Patient 14	180.82	193.31
Patient 7	179.54	193.87	Patient 15	178.88	198.43
Patient 8	185.12	196.39			

- (a) Are the samples independent? Why or why not?
- (b) If the blood glucose level is a normally distributed random variable, compute a 95% confidence interval for the mean differences of the population.
- (c) Use the results in (b) to decide whether or not the two devices give the same results.

17. The European Union is developing new policies to promote research and development investment. A random sample of 15 countries' investments for the years 2002 and 2003 is taken, and the results (in millions of euros) are stored in the data frame **EURD** and shown in the following table:

Country	2002	2003
Belgium	5200.737	5177.444
Czech Republic	959.362	1012.579
Estonia	55.699	66.864
France	34527.000	34569.095
Cyprus	33.791	40.969
Latvia	41.532	37.724
Lithuania	99.642	110.580
Hungary	705.754	693.057
Malta	11.861	11.453
Portugal	1029.010	1019.580
Slovenia	360.419	377.435
Slovakia	148.335	169.105
Bulgaria	81.228	88.769
Croatia	270.606	291.856
Romania	183.686	202.941

- (a) Compute a 95% confidence interval for the mean difference between 2003 and 2002 investments. Assume that the countries sampled were those that were EU countries in 2014 (28) but were not necessarily EU countries in 2002. In 2002, the EU was comprised of 15 countries. Apply an appropriate finite population factor to the margin of error when reporting the 95% confidence interval.
- (b) Use (a) to decide if the new policies are increasing investments.

18. The following data were taken to measure the unknown pH values μ of a solution in a chemical experiment:

$$8.01, 8.05, 7.96, 8.04, 8.03, 8.03, 8.02, 7.98, 8.05, 8.03.$$

If the pH meter has a systematic error, Δ , and a normally distributed random error, $\varepsilon \sim N(0, \sigma)$, then it can be assumed that the observations come from a normal random variable, $X \sim N(\mu + \Delta, \sigma)$.

- (a) Compute a 95% confidence interval for μ when $\Delta = 0$ and $\sigma = 0.05$. Compute the interval assuming that the variance is unknown.
- (b) Repeat part (a) with $\Delta = 0.2$.

19. When sampling from a normal distribution, what sample size will ensure that the interval $\bar{x} \pm s$ attains at least a 95% confidence level?

20. Let $\{X_1, \dots, X_n\}$ be a simple random sample from a normal distribution $N(\mu, \sigma^2)$, and consider the following random variables:

$$X = \min_{1 \leq i \leq n} \{x_i\}, \quad Y = \max_{1 \leq i \leq n} \{x_i\}.$$

- (a) Set the seed value at 69, and generate $m = 100$ samples of size $n = 5$ from a normal population $N(\mu = 5, \sigma = 2)$. Compute the number of intervals of the types $[X, Y]$ containing the real value $\mu = 5$. If the theoretical coverage of these intervals is 94% for a sample of size $n = 5$, do the empirical results agree with the theoretical coverage? Hint: Modify the code to `cisim()` to help answer the questions.
- (b) Set the seed value at 18, and generate $m = 100$ samples of size $n = 5$ from a normal population $N(\mu = 5, \sigma = 2)$. Compute the confidence intervals of the type $[X, Y]$ and $[\bar{X} + z_{0.03} \frac{\sigma}{\sqrt{n}}, \bar{X} + z_{0.97} \frac{\sigma}{\sqrt{n}}]$. Construct a plot with the length of both types of intervals. Repeat the exercise with samples of size $n = 10$, $n = 50$, and $n = 100$. Which type of confidence interval is preferred? Why?

21. Given the following data

$$25.3 \quad 23.8 \quad 27.5 \quad 23.2 \quad 24.5 \quad 25.3 \quad 24.6 \quad 26.8 \quad 25.9 \quad 29.2,$$

- (a) State the assumption(s) needed to construct a confidence interval for the population variance.
- (b) Assuming your assumption(s) in (a) are satisfied, construct a 95% confidence interval for σ .
- (c) Assuming that $\mu = 25$, construct a 95% confidence interval for σ .

22. Schizophrenia is believed to cause changes in dopamine levels. Twenty-five patients with schizophrenia were classified as psychotic or non-psychotic after being treated with an antipsychotic drug. Samples of cerebral fluid were taken from each patient and assayed for dopamine b -hydroxylase (DBH) activity. The dopamine measurements for the two groups are in nmol/(ml)(h)/(mg) of protein and are stored in the data frame **SCHIZO** as well as in the following table (Sternberg et al., 1982).

Judged Non-Psychotic			Judged Psychotic	
0.0104	0.0105	0.0112	0.0150	0.0204
0.0116	0.0130	0.0145	0.0208	0.0222
0.0154	0.0156	0.0170	0.0226	0.0245
0.0180	0.0200	0.0200	0.0270	0.0275
0.0210	0.0230	0.0252	0.0306	0.0320

- (a) Construct side-by-side boxplots of the two groups. Based on the boxplots, comment on the relative shapes of the two distributions.
- (b) Construct quantile-quantile plots for the two groups, and comment on whether or not the plots support the analysis in part (a).
- (c) Construct a 95% confidence interval for the true ratio of psychotic to non-psychotic variances.

- (d) Based on the confidence interval for the ratio of variances, should the variances be pooled to construct a 95% confidence interval for the true dopamine level difference between psychotic and non-psychotic patients?
- (e) Construct a 95% confidence interval for the true dopamine level difference between psychotic and non-psychotic patients.
- (f) Does the confidence interval contain zero? What does this say about the effectiveness of the antipsychotic drug?

23. Assuming two independent random samples of sizes 22 and 45 with variance estimates of $s_1^2 = 38.7$ and $s_2^2 = 45.6$, respectively, have been taken, construct a 95% confidence interval for σ .

24. Those teams who win Formula 1 championships have pit crews who change tires as fast as possible. The data frame **FORMULA1** and the following table contain the times (in seconds) that the pit crews of two different teams spent changing tires in 10 randomly selected races per team.

Team 1	5.613	6.130	5.422	5.947	5.514	5.322	5.690	5.243	5.920	5.859
Team 2	5.934	5.335	5.826	4.821	5.664	5.292	5.257	6.245	5.981	5.197

- (a) Assuming that the times are normally distributed, compute a 95% confidence interval for the variance ratio σ_1^2/σ_2^2 . Are the population variances equal?
- (b) Use the results in part (a) to compute a 95% confidence interval for the difference of the population means $\mu_2 - \mu_1$. What does the result signify?

25. Let $\{X_1, X_2, \dots, X_n\}$ be a random sample from a normal population $N(\mu, \sigma)$, where μ and σ are unknown. Find the value of the sample size n if $[0.59s^2, 2s^2]$ is to be at least a 94% confidence interval for σ^2 .

26. Use a seed equal to 55, and simulate $m = 100$ samples of size $n = 800$ from a $N(15, \sigma = \sqrt{6})$. Calculate the confidence intervals for σ^2 at the $1 - \alpha = 0.96$ confidence level. Plot the confidence intervals, and calculate the number of times the parameter is not contained in the simulated confidence intervals. (Hint: use **cisim()**.)

27. Use a seed equal to 121, and simulate $m_x = 100$ samples of size $n_x = 1500$ from a $N(3, \sigma = \sqrt{5})$ and $m_y = 100$ samples of size $n_y = 1500$ from a $N(6, \sigma = \sqrt{7})$. Calculate the confidence intervals for σ_x^2/σ_y^2 with a $1 - \alpha = 0.94$ confidence level. Plot the intervals and calculate the number of times the parameter ratio is not in the simulated confidence interval. Hint: modify the function **cimsim()**.

28. The drug sulfinpyrazone was studied for its efficacy in preventing death after myocardial infarction (Anturane Reinfarction Trial Research Group, 1980). Construct a 90% confidence interval for the true proportion of deaths between patients who have suffered a myocardial infarction who were administered sulfinpyrazone and patients who were administered a placebo after myocardial infarctions. Based on the confidence interval, does sulfinpyrazone

appear to reduce the proportion of deaths among patients who have suffered a myocardial infarction?

	Death (all causes)	Survivors
Sulphipyrazone	44	580
Placebo	62	563

29. From a random sample of 2000 Internet domains registered in a country during the last few years, 300 were “.org” domains. Compute a 98% confidence interval for the proportion of “.org” domains registered in that country during the last few years.
30. Use a seed equal to 10, and simulate 10,000 samples of size $n_x = 65$ from a $N(4, \sigma_x = \sqrt{2})$ distribution and 300 samples of size $n_y = 90$ from a $N(5, \sigma_y = \sqrt{3})$. Check that $\frac{s_x^2/\sigma_x^2}{s_y^2/\sigma_y^2}$ follows an $F_{64,89}$ distribution.
31. Use a seed equal to 6, and simulate $m = 10,000$ samples of size $n = 1000$ from a $Bin(1, \pi = 0.4)$ distribution. Show that the sampling proportion is approximately normally distributed.
32. How large a sample is needed to ensure the bound on the error of estimation for the population proportion is no more than 3 percentage points for a 98% confidence interval?
33. A large company wants to estimate the proportion of its accounts that are paid on time.
- How large a sample is needed to estimate the true proportion within 4% with a 95% confidence interval?
 - Suppose 650 out of 800 accounts are paid on time. Construct a 98% confidence interval for the true proportion of accounts that are paid on time.
34. A sociology research center conducts a survey to discern whether the proportion of vegetarians is larger in urban or rural areas. Of the 180 people from urban areas, 32 were vegetarians. Of the 75 from rural areas, 17 were vegetarians. Construct a 95% confidence interval for the difference between urban and rural vegetarian proportions.
35. Schizophrenia and other psychoses are complex and debilitating diseases, which affect about 2% of the population. Two of the approaches used, as well as in other medical diseases, to reduce clinical heterogeneity among psychoses are categorical and dimensional. The first one assumes that there exist different subgroups within psychosis and the second one assumes that schizophrenia dimensions fall on a dimensional continuum within psychosis. A sample of 660 consecutively admitted patients in Hospital Virgen del Camino (Pamplona, Spain) is available with the following diagnoses: 358 schizophrenic patients, 61 with schizopreniform disorder, 37 with schizoaffective disorder, 64 with bipolar disorder, 24 with delusional disorder, 54 with brief psychotic disorder, and 32 with atypical psychosis. Compute a 95% confidence interval for the proportion of the different types of patients (Cuesta et al., 2007).
36. Find the required sample size (n) to estimate the proportion of students spending more than €10 a week on entertainment with a 95% confidence interval so that the margin of error is no more than 0.02.

Chapter 9

Hypothesis Testing

9.1 Introduction

A **hypothesis test** in the Neyman-Pearson paradigm is a decision criterion that allows practitioners of statistics to select between two complementary hypotheses. Before conducting the hypothesis test, define the **null hypothesis**, H_0 , which is assumed to be true prior to conducting the hypothesis test. The null hypothesis is compared to another hypothesis, called the **alternative hypothesis**, and denoted H_1 . The alternative hypothesis is often called the research hypothesis since the theory or what is believed to be true about the parameter is specified in the alternative hypothesis. Both hypotheses define complementary subsets of the parameter space Θ where the parameter θ is defined. The null hypothesis defines the region $[\theta \in \Theta_0]$ and the alternative hypothesis defines the region $[\theta \in \Theta_1]$. The subsets Θ_0 and Θ_1 are mutually exclusive by definition, and they are complementary since $\Theta_0 \cup \Theta_1 = \Theta$. When a hypothesis uniquely specifies the distribution of the population from which the sample is taken, the hypothesis is said to be **simple**. For a simple hypothesis, Θ_0 is composed of a single element. Any hypothesis that is not a simple hypothesis is called a **composite hypothesis**. A composite hypothesis does not completely specify the population distribution. Of the various combinations of hypotheses that could be examined, the case where the null hypothesis is simple and the alternative hypothesis is composite will be the focus of this text. Hypothesis tests will generally take a form similar to those in Table 9.1, where θ_0 is a single numerical value. For alternative hypotheses (A) and (B), which are lower one-sided and upper one-sided, respectively, the hypothesis test is called a **one-tailed test**. For the alternative hypothesis in (C), a two-sided alternative, the hypothesis test is called a **two-tailed test**.

Table 9.1: Form of hypothesis tests

Null Hypothesis	Alternative Hypothesis	Type of Alternative
	(A) $H_1 : \theta < \theta_0$	lower one-sided
$H_0 : \theta = \theta_0$	(B) $H_1 : \theta > \theta_0$	upper one-sided
	(C) $H_1 : \theta \neq \theta_0$	two-sided

Example 9.1 If $H_0 : \pi = 0.4$ in a $Bernoulli(\pi)$ distribution, the null hypothesis is simple since the hypothesis $H_0 : \pi = 0.4$ uniquely specifies the distribution as $Bernoulli(0.4)$. If $H_1 : \pi < 0.4$, the hypothesis is composite since π can take any value in the interval $[0, 0.4)$.

The goal in hypothesis testing is to decide which one of the two hypotheses (null or

alternative) is true. To help decide between the two hypotheses, calculate a test statistic based on the sample information from the experiment. The possible values for the test statistic are split into two mutually exclusive subsets R and R^c . R is the **rejection region** and R^c is referred to as the **acceptance region**. The boundary values of the rejection region are called **critical values**. If the test statistic falls in the acceptance region, accept the null hypothesis. If the value of the test statistic falls in the rejection region, reject the null hypothesis and accept the alternative hypothesis.

There are two basic ways to think of a hypothesis test. First, one can think of it as a two-decision problem where the researcher will choose one of two hypotheses to be true. This is the historical approach due to Jerzy Neyman and Egon Pearson (Johnson and Kotz, 2011). The second method, due to Ronald Fisher, determines how much evidence exists in the data against the null hypothesis (Johnson and Kotz, 2011). The null hypothesis is never accepted but is merely a hypothesis of “no difference.” The test will determine if the data that have been collected could be due to chance alone if the null hypothesis were true; and if this is not likely, the researcher has statistically significant evidence that the alternative hypothesis is true. A hypothesis test where the null hypothesis is never accepted but merely “not rejected” is called a significance test.

Example 9.2 The weight of a ball-bearing fluctuates between 1.5 g and 4.5 g. One wants to test whether the distribution of the weight for the ball-bearing has a mean of either 2 g ($H_0 : \mu = 2$) or 2.5 g ($H_1 : \mu = 2.5$). A random sample of size one is taken. If the weight of the ball-bearing is greater than 2.3 g, the null hypothesis that the mean weight of the ball-bearings is 2 g is rejected, and the alternative hypothesis that the mean weight of the ball-bearings is 2.5 g is accepted. Specify the sample space, the rejection region, and the acceptance region for this experiment.

Solution: The sample space is given by the interval [1.5, 4.5]. The rejection region is the subinterval $R = (2.3, 4.5]$, and the acceptance region is the subinterval $R^c = [1.5, 2.3]$. Note that $R^c \cup R = [1.5, 2.3] \cup (2.3, 4.5] = [1.5, 4.5]$. ■

9.2 Type I and Type II Errors

The decision one reaches using a hypothesis test is always subject to error. That is, when a decision is reached to reject the null hypothesis and accept the alternative hypothesis, this may be the correct decision or a mistake (error). Likewise, if the null hypothesis is not rejected but rather accepted, an error could also be made. Simply put, one can never be sure of the truth, since the decision in a hypothesis test to reject or not to reject a hypothesis is based on sample information. To get a better grasp of the errors one might make with a hypothesis test, consider the following hypothetical legal situation.

An individual is on trial for a capital offense. In the United States’ judicial system, an individual is considered innocent until proven guilty of an offense. Consequently, the null hypothesis in this case is that the individual is innocent and the alternative hypothesis is that the person is guilty. After the prosecuting and defending attorneys present their evidence, the jury makes a decision either to convict or not to convict the individual of the capital offense. If the prosecuting attorney presents a strong case, the jury is likely to convict the defendant; however, just because the jury convicts the defendant, it does not mean that the defendant is actually guilty of the capital offense. Likewise, if the jury does not convict the defendant of the capital offense, this does not imply the individual

is innocent. To better see the possible consequences of the decisions the jury may reach, consider Table 9.2.

Table 9.2: Possible outcomes and their consequences for a trial by jury

True State of the Defendant (Reality)	Jury's Decision	
	Reject H_0 (guilty)	Accept H_0 (not guilty)
H_0 True (innocent)	A. error	C. correct
H_0 False (guilty)	B. correct	D. error

- A. If the null hypothesis is true and it is rejected, the decision is incorrect. In other words, by rejecting a true null hypothesis, an error has been made. In statistics, this type of error is called a **type I error**. The probability of committing a type I error is α . In the legal example, a type I error would be to convict an innocent defendant.
- B. If the null hypothesis is false and it is rejected, the decision is correct. In the legal arena, this translates into a jury convicting a guilty defendant.
- C. If the null hypothesis is true and the null hypothesis is accepted, the decision is correct. In the legal example, if the defendant is innocent and the jury decides the defendant is not guilty of the charge, the jury's decision is correct.
- D. If the null hypothesis is false and it is not rejected, the decision is incorrect. By failing to reject a false null hypothesis, an error has been made. In statistics, this error is called a **type II error**. The probability of committing a type II error is β . In the legal scenario, a type II error is made when a guilty person is not convicted.

The probability of committing a type I error (rejecting H_0 when it is true) is called the **level of significance** for a hypothesis test. The level of significance is also known as the size of the test and is denoted by α , where

$$\alpha = \mathbb{P}(\text{type I error}) = \mathbb{P}(\text{reject } H_0 | H_0 \text{ is true}) = \mathbb{P}(\text{accept } H_1 | H_0 \text{ is true}).$$

The probability of committing a type II error is β , where

$$\begin{aligned}\beta &= \mathbb{P}(\text{type II error}) = \mathbb{P}(\text{fail to reject } H_0 | H_0 \text{ is false}) \\ &= \mathbb{P}(\text{accept } H_0 | H_1 \text{ is true}).\end{aligned}$$

The relationship between type I and type II errors is shown in Table 9.3 on the next page.

If the researcher fails to reject the null hypothesis when the null hypothesis is true, note that no error is committed. Specifically, the correct decision should be reached in roughly $(1 - \alpha) \times 100\%$ of all trials. Using the same logic, approximately $(1 - \beta) \times 100\%$ of the times sample data are evaluated in a test of hypothesis, a false null hypothesis will be rejected.

Since a type I error is frequently considered to be more serious than a type II error and the probability of a type I error is easier to control than the probability of a type II

Table 9.3: Relationship between type I and type II errors

		Decision	
		Reject H_0	Fail To Reject H_0
Null Hypothesis	True	Type I Error $\mathbb{P}(\text{Type I Error}) = \alpha$ (Level of Significance)	Correct Decision $\mathbb{P}(\text{Accept } H_0 H_0) = 1 - \alpha$ (Confidence Level)
	False	Correct Decision $\mathbb{P}(\text{Accept } H_1 H_1) = 1 - \beta$ (Power of the Test)	Type II Error $\mathbb{P}(\text{Type II Error}) = \beta$

error, it is common practice for researchers to specify *a priori* the largest probability of a type I error they are willing to accept and subsequently to use this value as their level of significance to make a decision when they conduct their hypothesis testing. The North American judicial system certainly considers convicting an innocent person to be a worse error than allowing a guilty person to walk free; however, a type I error is not always more critical than a type II error. Suppose one is going to go skydiving. In this scenario, the null hypothesis is that the parachute will open and the alternative hypothesis is that the parachute will not open. Certainly a type II error (concluding the parachute will open when it will not) is more critical than a type I error (concluding the parachute will not open when it will).

Example 9.3 Given a normal distribution with unknown mean μ and known standard deviation $\sigma = 2$, one wishes to test the null hypothesis $H_0 : \mu = 1$ versus the alternative hypothesis $H_1 : \mu = 4$. A sample of size one is taken where the rejection region is considered to be the interval $(2, \infty)$. In other words, if the sample value is greater than 2, the null hypothesis is rejected. On the other hand, if the sample value is less than or equal to two, one fails to reject the null hypothesis. Determine α and β for this experiment.

Solution: Although there is no way to know if the decision made with regard to the null hypothesis is correct, there is a reasonable criterion that allows the determination of the probability of making type I and type II errors.

Determine α — The probability of committing a type I error, the level of significance, is the probability that the sample value falls in the rejection region $(2, \infty)$ when $H_0 : \mu = 1$ is true. To find α , it is necessary to find $\mathbb{P}(X_1 > 2 | N(1, 2))$. See Figure 9.1 for a graphical representation of the type I error. Note that

$$\alpha = \mathbb{P}(X_1 > 2 | N(1, 2)) = \mathbb{P}\left(\frac{X_1 - 1}{2} > \frac{2 - 1}{2}\right) = \mathbb{P}(Z > 0.5) = 0.3085.$$

To find α with R, key in

```
> ALPHA <- 1 - pnorm(2, 1, 2)
> ALPHA
```

```
[1] 0.3085375
```

Note: R returns the area to the left of a given value when using the function `pnorm()`. The user can also find the area to the right of a given value (quantile) by using the argument `lower.tail = FALSE`. Consequently, one might have used the `lower.tail = FALSE` argument with R's `pnorm()` function to find the answer.

```
> ALPHA <- pnorm(2, 1, 2, lower.tail = FALSE)
> ALPHA
[1] 0.3085375
```

Determine β — The probability of making a type II error is the probability of failing to reject $H_0 : \mu = 1$ when in actuality $H_1 : \mu = 4$. In other words, although $\mu = 4$, the null hypothesis is not rejected because the test statistic does not fall in the rejection region but does lie in the region $(-\infty, 2]$. For a graphical representation of the type II error, see Figure 9.1. Mathematically this is written

$$\beta = \mathbb{P}(X_1 \leq 2 | N(4, 2)) = \mathbb{P}(Z \leq -1) = 0.1587.$$

To find β with R, enter

```
> BETA <- pnorm(2, 4, 2)
> BETA
[1] 0.1586553
```

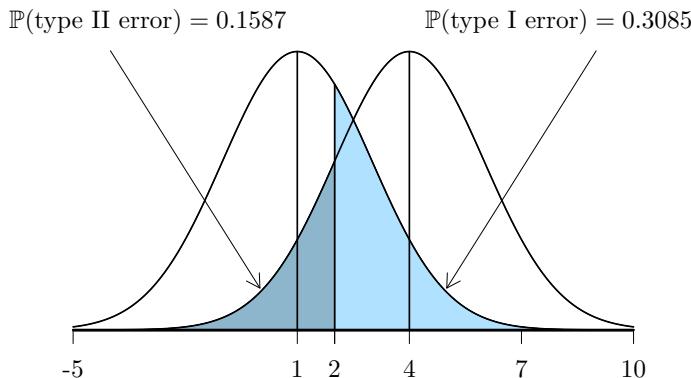


FIGURE 9.1: Graphical representation of type I and type II errors when $H_0 : \mu = 1$ versus $H_1 : \mu = 4$

Since the probabilities of committing type I and type II errors for a fixed sample size are dependent, it is usually impossible to make both type I and type II errors arbitrarily small. Out of convenience, the tests considered are restricted to only those tests that control the type I error at a given significance level and subsequently select from these tests the test with the most power. Researchers typically fix the probability of committing a type I error at the 0.01, 0.05, or 0.1 significance level; however, these are merely values that were tabulated early in the history of statistics and have been used mainly for convenience rather than through any actual merit. Since there are as many tests as there are partitions

of the sample space, the number of tests one may have to evaluate to decide between two competing hypotheses might be huge. For this very reason, certain partitions will produce results that are more appealing in the sense of supporting a specific hypothesis.

9.3 Power Function

Given a composite alternative hypothesis $H_1 : \theta \in \Theta_1$, the power of the test, $\text{Power}(\theta)$, is

$$\begin{aligned}\text{Power}(\theta) &= \mathbb{P}(\text{reject } H_0 | H_0 \text{ is false}) = \mathbb{P}(\text{accept } H_1 | H_1) \\ &= 1 - \beta(\theta),\end{aligned}\tag{9.1}$$

where $\beta(\theta)$ is the probability of a type II error at a given θ . Loosely speaking, the power of a test is the probability the test detects differences when differences exist. Note that $\text{Power}(\theta)$ is a function of the parameter θ , which has for each value of θ in the alternative hypothesis, $\theta \in \Theta_1$, the power that a simple alternative hypothesis would have for that value of θ . When the null hypothesis is simple, $\theta = \theta_0$, the power of the test at θ_0 is the same as the significance level, that is, $\text{Power}(\theta_0) = \alpha$.

Example 9.4 Given the density function

$$f(x|\theta) = \theta e^{-\theta x}, \quad x \geq 0, \quad \theta \in \varnothing,$$

- (a) Consider a test of hypothesis where $H_0 : \theta = 2$ versus $H_1 : \theta > 2$. Using a random sample of size one, find k such that, if $x \leq k$, the test is conducted at the $\alpha = 0.05$ level.
- (b) Further, determine the power function of this test.

Solution: The solutions are as follows:

- (a) First, set up the integral to find the value of k that yields a significance level of 0.05:

$$\alpha = \mathbb{P}(X_1 < k | H_0) = \int_0^k 2e^{-2x_1} dx_1 = 1 - e^{-2k} = 0.05$$

The solution for k is $k = 0.0256$.

```
> qexp(0.05, 2)
[1] 0.02564665
```

- (b) The power of the test is

$$\begin{aligned}\text{Power}(\theta) &= 1 - \beta(\theta) = \mathbb{P}(\text{accept } H_1 \text{ when it is true}) = \mathbb{P}(X_1 < 0.0256 | H_1) \\ &= \int_0^{0.0256} \theta e^{-\theta x_1} dx_1 = 1 - e^{-0.0256\theta}.\end{aligned}$$

Note that the answer clearly illustrates that it is not possible to obtain a single value for the power of a composite alternative hypothesis since the answer itself is a function of θ . In

other words, for each value of the parameter θ compatible with the alternative hypothesis (in this case $\theta > 2$), a value for the power function is obtained that corresponds to that simple hypothesis. As the parameter θ takes on values greater than two, the power function approaches one. ■

Example 9.5 ▷ Achievement Test ◁ Test the null hypothesis that for a certain age group the mean score on an achievement test (scores follow a normal distribution with $\sigma = 6$) is equal to 40 against the alternative that it is not equal to 40.

- Find the probability of type I error for $n = 9$ if the null hypothesis is rejected when the sample mean is less than 36 or greater than 44.
- Find the probability of type I error for $n = 36$ if the null hypothesis is rejected when the sample mean is less than 38 or greater than 42.
- Plot the power functions for $n = 9$ and $n = 36$ for values of μ between 30 and 50.

Solution: The solutions are as follows:

- The probability of a type I error for $n = 9$ if the null hypothesis is rejected when the sample mean is less than 36 or greater than 44 is

$$\begin{aligned}\mathbb{P}(\text{Type I error}) &= \mathbb{P}\left(\bar{X} < 36 \mid \bar{X} \sim N\left(40, \frac{6}{\sqrt{9}}\right)\right) + \mathbb{P}\left(\bar{X} > 44 \mid \bar{X} \sim N\left(40, \frac{6}{\sqrt{9}}\right)\right) \\ &= \mathbb{P}\left(Z < \frac{36 - 40}{2}\right) + \mathbb{P}\left(Z > \frac{44 - 40}{2}\right) \\ &= \mathbb{P}(Z < -2) + \mathbb{P}(Z > 2) = 0.02275 + 0.02275 = 0.04550.\end{aligned}$$

To compute the answer with R, key in

```
> pnorm(36, 40, 6/sqrt(9)) + 1 - pnorm(44, 40, 6/sqrt(9))
[1] 0.04550026
```

- The probability of type I error for $n = 36$ if the null hypothesis is rejected when the sample mean is less than 38 or greater than 42 is

$$\begin{aligned}\mathbb{P}(\text{Type I error}) &= \mathbb{P}\left(\bar{X} < 38 \mid \bar{X} \sim N\left(40, \frac{6}{\sqrt{36}}\right)\right) + \mathbb{P}\left(\bar{X} > 42 \mid \bar{X} \sim N\left(40, \frac{6}{\sqrt{36}}\right)\right) \\ &= \mathbb{P}\left(Z < \frac{38 - 40}{1}\right) + \mathbb{P}\left(Z > \frac{42 - 40}{1}\right) \\ &= \mathbb{P}(Z < -2) + \mathbb{P}(Z > 2) = 0.02275 + 0.02275 = 0.04550.\end{aligned}$$

To compute the answer with R, enter

```
> pnorm(38, 40, 6/sqrt(36)) + 1 - pnorm(42, 40, 6/sqrt(36))
[1] 0.04550026
```

- The power function for $n = 9$ is

$$\text{Power}(\mu) = \mathbb{P}\left(\bar{X} < 36 \mid \bar{X} \sim N\left(\mu, \frac{6}{\sqrt{9}}\right)\right) + \mathbb{P}\left(\bar{X} > 44 \mid \bar{X} \sim N\left(\mu, \frac{6}{\sqrt{9}}\right)\right).$$

The power function for $n = 36$ is

$$\text{Power}(\mu) = \mathbb{P}\left(\bar{X} < 38 \mid \bar{X} \sim N\left(\mu, \frac{6}{\sqrt{36}}\right)\right) + \mathbb{P}\left(\bar{X} > 42 \mid \bar{X} \sim N\left(\mu, \frac{6}{\sqrt{36}}\right)\right).$$

To produce a plot similar to the one in Figure 9.2 with base R graphics, use R Code 9.1. The actual graph in Figure 9.2 was created with `ggplot2` graphics. It is left to the reader as an exercise to recreate Figure 9.2 using `ggplot2` graphics.

R Code 9.1

```
> mu <- seq(30, 50, 0.01)
> power9 <- 1 - pnorm(44, mu, 6/sqrt(9)) + pnorm(36, mu, 6/sqrt(9))
> power36 <- 1 - pnorm(42, mu, 6/sqrt(36)) + pnorm(38, mu,
+   6/sqrt(36))
> plot(mu, power9, type = "l", ylab = expression(Power(mu)),
+   xlab = expression(mu), ylim = c(0, 1))
> lines(mu, power36, type = "l", lty = "dashed")
> arrows(32, 0.6, 34.2, 0.78, lwd = 2, length = 0.05)
> arrows(32, 0.35, 37, 0.78, lwd = 2, length = 0.05)
> arrows(40, 0.65, 40, 0.06, lwd = 2, length = 0.05)
> text(32, 0.58, expression(n == 9))
> text(32, 0.33, expression(n == 36))
> text(40, 0.7, expression(alpha == 0.045))
```

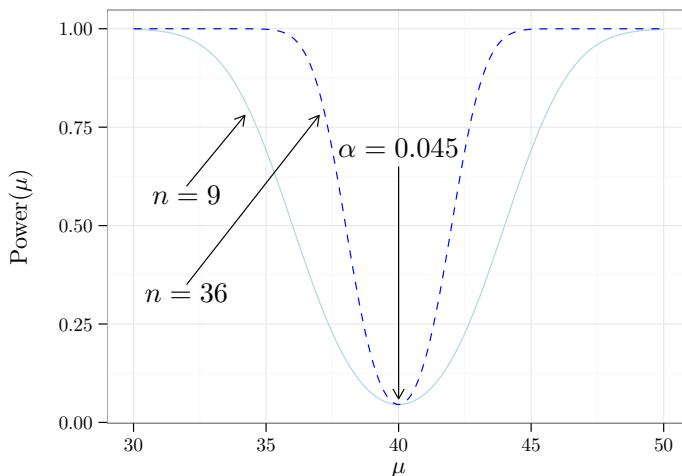


FIGURE 9.2: Graphical representation of the power function, $\text{Power}(\mu)$, for both scenarios in Example 9.5 on the preceding page

Note that $\text{Power}(\mu_0) = \alpha$ for both power functions depicted in Figure 9.2. In general, as the true μ is farther from the hypothesized μ in H_0 , the power of a test will increase. Additionally, the power function approaches 1 faster for larger n as the true μ moves farther from the hypothesized μ in H_0 . ■

9.4 Uniformly Most Powerful Test

First, note that tests with identical α values do not necessarily have identical power for a fixed sample size as in Example 9.6.

Example 9.6 Given a $N(\mu, 1)$ population from which one takes a simple random sample of size 1, test the null hypothesis $H_0 : \mu = 1$ versus the alternative hypothesis $H_1 : \mu = 2$. Determine the significance level and the power of the test for the following rejection regions:

- (a) $(2.0364, \infty)$
- (b) $(1.1000, 1.3000) \cup (2.4617, \infty)$.

Solution: The answers are as follows:

- (a) Since $R = (2.0364, \infty)$,

$$\begin{aligned}\alpha &= \mathbb{P}(X > 2.0364 | N(1, 1)) = \mathbb{P}\left(\frac{X - 1}{1} > \frac{2.0364 - 1}{1}\right) = \mathbb{P}(Z > 1.0364) = 0.15, \\ \beta &= \mathbb{P}(X \leq 2.0364 | N(2, 1)) = \mathbb{P}\left(\frac{X - 2}{1} \leq \frac{2.0364 - 2}{1}\right) = \mathbb{P}(Z \leq 0.0364) = 0.5145,\end{aligned}$$

and the power of the test is $1 - \beta = 1 - 0.5145 = 0.4855$. See Figure 9.3 for a graphical representation of the type I and type II errors.

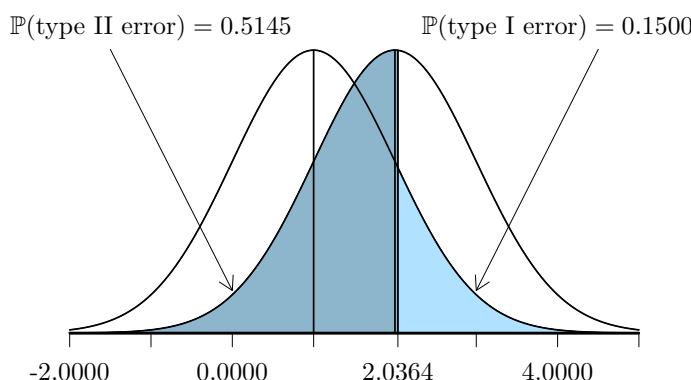


FIGURE 9.3: Graphical representation of type I and type II errors when $H_0 : \mu = 1$ versus $H_1 : \mu = 2$ with rejection region $(2.0364, \infty)$

- (b) Since the rejection region is $(1.1000, 1.3000) \cup (2.4617, \infty)$, the probability of committing

a type I error is

$$\begin{aligned}
 \alpha &= \mathbb{P}(1.1000 < X < 1.3000 | N(1, 1)) + \mathbb{P}(X > 2.4617 | N(1, 1)) \\
 &= \mathbb{P}\left(\frac{1.1000 - 1}{1} < Z < \frac{1.3000 - 1}{1}\right) + \mathbb{P}\left(\frac{X - 1}{1} > \frac{2.4617 - 1}{1}\right) \\
 &= \mathbb{P}(0.1000 < Z < 0.3000) + \mathbb{P}(Z > 1.4617) \\
 &= \mathbb{P}(Z < 0.3000) - \mathbb{P}(Z < 0.1000) + \mathbb{P}(Z > 1.4617) \\
 &= 0.6179 - 0.5398 + 0.0719 = 0.15,
 \end{aligned}$$

and the probability of committing a type II error is

$$\begin{aligned}
 \beta &= \mathbb{P}(X \leq 1.1000 | N(2, 1)) + \mathbb{P}(1.3000 \leq X \leq 2.4617 | N(2, 1)) \\
 &= \mathbb{P}\left(\frac{X - 2}{1} \leq \frac{1.1000 - 2}{1}\right) + \mathbb{P}\left(\frac{1.3000 - 2}{1} \leq \frac{X - 2}{1} \leq \frac{2.4617 - 2}{1}\right) \\
 &= \mathbb{P}(Z \leq -0.9000) + \mathbb{P}(-0.7000 \leq Z \leq 0.4617) \\
 &= \mathbb{P}(Z \leq -0.9000) + \mathbb{P}(Z \leq 0.4617) - \mathbb{P}(Z \leq -0.7000) \\
 &= 0.1841 + 0.6616 - 0.242 = 0.6199.
 \end{aligned}$$

A graphical representation of the type I and type II errors is provided in Figure 9.4 on the facing page. To find α and β with R, type

```

> ALPHA <- pnorm(1.3, 1, 1) - pnorm(1.1, 1, 1) + pnorm(2.4617,
+      1, 1, lower = FALSE)
> ALPHA
[1] 0.1499953

> BETA <- pnorm(1.1, 2, 1) + pnorm(2.4617, 2, 1) - pnorm(1.3,
+      2, 1)
> BETA
[1] 0.6199482

```

It follows that the power of the test is $1 - \beta = 1 - 0.6199 = 0.3801$. ■

It is clear to see from the previous example that, with the same level of significance (0.1500), the test with a rejection region of $(1.1000, 1.3000) \cup (2.4617, \infty)$ has less power than the test that uses a rejection region of $(2.0364, \infty)$. The probabilities of committing type I and type II errors for the rejection regions $(2.0364, \infty)$ and $(1.1000, 1.3000) \cup (2.4617, \infty)$ are shown in Figures 9.3 and 9.4, respectively. In general, it is possible to have a test that is “better” in the sense of having more power than another test even though both tests have the same significance level. So, the researcher wants to find a **uniformly most powerful** test that has more power than all other tests that have the correct significance level, α , if such a test exists. To be complete, it is important to note that uniformly most powerful tests do not always exist. A generalization that can be made from Example 9.6 is that one-sided tests with the same sample size as two-sided tests will always have more power for the same α level.

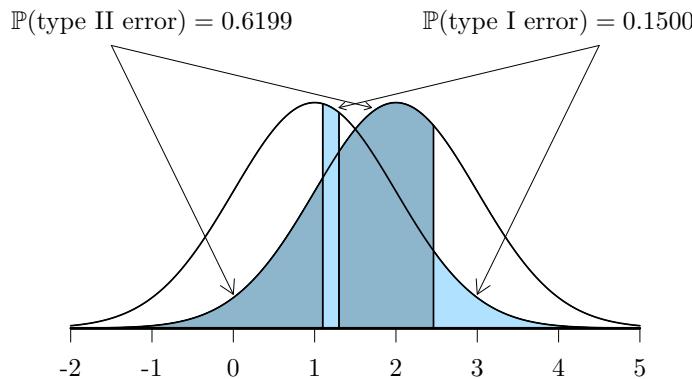


FIGURE 9.4: Graphical representation of type I and type II errors when $H_0 : \mu = 1$ versus $H_1 : \mu = 2$ with rejection region $(1.1000, 1.3000) \cup (2.4617, \infty)$

9.5 \wp -Value or Critical Level

Fisher's advocates object to establishing *a priori* the level of significance when testing a hypothesis. Instead, they prefer to make their decisions to reject or fail to reject the null hypothesis based on \wp -values. The critical level or **\wp -value** is defined as the probability of observing a difference as extreme or more extreme than the difference observed under the assumption that the null hypothesis is true. Virtually all statistical software packages will return a \wp -value when testing a hypothesis. The values of the statistic $t(\mathbf{x})$ observed from the sample and \wp -value calculations are summarized in Table 9.4.

Table 9.4: Calculation of \wp -values for continuous distributions

Alternative hypothesis	\wp -value
$H_1 : \theta < \theta_0$	$\mathbb{P}(T \leq t_{\text{obs}} H_0)$
$H_1 : \theta > \theta_0$	$\mathbb{P}(T \geq t_{\text{obs}} H_0)$
$H_1 : \theta \neq \theta_0$	$2 \min \left\{ \mathbb{P}(T \leq t_{\text{obs}} H_0), \mathbb{P}(T \geq t_{\text{obs}} H_0) \right\}$

It is important to note that the \wp -value is not fixed *a priori*, but rather is determined after the sample is taken. A small \wp -value indicates that observing differences as large or larger than the one found in the sample is rare, and thus does not occur by chance alone. A small \wp -value lends support to H_1 ; so, given a fixed significance level α , reject H_0 whenever the \wp -value $< \alpha$. In Fisher's paradigm, hypothesis tests are tests of significance, where a \wp -value is calculated without regard to a fixed rejection region. The Neyman-Pearson paradigm uses a specified α level to calculate a rejection region that is used in conjunction with a standardized test statistic to reach a statistical conclusion.

9.6 Tests of Significance

Using the following steps incorporates ideas from both Fisher and Neyman and Pearson for solving **test of hypothesis**-type problems. The steps allow others to follow the reasoning one uses to reach a statistical decision.

Step 1: Hypotheses — State the null and alternative hypotheses.

First, establish the null hypothesis, $H_0 : \theta = \theta_0$. Next, determine the form of the alternative hypothesis, H_1 . The forms H_1 can take are found in Table 9.1 on page 519, where evidence is to be found that θ is less than, greater than, or not equal to the θ_0 specified in H_0 . If one wishes to specify a value for which H_1 is true, that value is denoted with either θ_1 or $\theta_1(X, Y, \dots)$.

Step 2: Test Statistic — Select an appropriate test statistic and determine the sampling distribution of the test statistic or the standardized test statistic under the assumption that the null hypothesis is true.

Choose a test statistic, $\hat{\theta}$, generally one such that the expected value of the test statistic is equal to the parameter in H_0 . For example, if testing μ , $\hat{\theta} = \bar{X}$; or, if testing π , $\hat{\theta} = P$.

A common standardized test statistic will take the form

$$T = t(\mathbf{X}) = \frac{\hat{\theta}(\mathbf{X}) - \theta_0}{\sqrt{Var[\hat{\theta}(\mathbf{X})]}}.$$

Other test statistics will present themselves when testing hypotheses regarding variances.

Step 3: Rejection Region Calculations — If the computations are to be done by hand, use the specified α level to compute the critical value and to determine the rejection region for the standardized test statistic. If the computations are to be done by a computer, do not do this.

Then, calculate the value of $t(\mathbf{X})$, assuming H_0 is true. The value of the statistic $t(\mathbf{X})$ observed from the sample is denoted $t(\mathbf{x}) = t_{\text{obs}}$.

Step 4: Statistical Conclusion — If a rejection region was not computed in step 3, calculate the p -value. The procedure for calculating the p -value is found in Section 9.5 on the previous page.

Use the rejection region or the p -value to determine if the evidence warrants rejecting the null hypothesis. If t_{obs} falls into the rejection region, reject H_0 ; if not, fail to reject H_0 . If the p -value is less than α , reject H_0 ; if not, fail to reject H_0 .

Step 5: English Conclusion — State in plain English what the conclusion reached in step 4 means. This statement will always be about the alternative hypothesis. That is, the evidence will either warrant concluding the alternative hypothesis or the evidence will not be sufficient to conclude the alternative hypothesis is true.

There are two distributions that occur frequently in hypothesis testing involving means: a standard normal distribution and a t -distribution. When the standardized test statistic follows a standard normal distribution, the hypothesis test will typically be called a **one-sample z -test** or a **two-sample z -test**, depending on whether there are one or two samples. Likewise, if the standardized test statistic follows a t -distribution, the test will be a **one-sample t -test**, a **two-sample t -test**, or a **paired t -test**. The general form for a z -test statistic is

$$\frac{\text{statistic} - \mu_{\text{statistic}}}{\sigma_{\text{statistic}}} \quad (9.2)$$

while the general form of a t -test statistic is

$$\frac{\text{statistic} - \mu_{\text{statistic}}}{\hat{\sigma}_{\text{statistic}}}. \quad (9.3)$$

Duality of Confidence Intervals and Tests of Significance When confidence intervals were constructed in Chapter 8, there was often a statistic $\hat{\theta}(\mathbf{X})$ that had a known distribution, where θ was the mean of $\hat{\theta}(\mathbf{X})$ and $\sigma_{\hat{\theta}(\mathbf{X})}$ was the square root of the variance of $\hat{\theta}(\mathbf{X})$. From this statistic's distribution, a pivot was constructed that took the form $\frac{\hat{\theta}(\mathbf{X}) - \theta}{\sigma_{\hat{\theta}(\mathbf{X})}}$ with a known distribution (denoted, in general, as T). One would use this pivot to construct a $(1 - \alpha) \cdot 100\%$ confidence interval:

$$CI_{1-\alpha}(\theta) = \left[\hat{\theta}(\mathbf{x}) + t_{\alpha/2} \cdot \sigma_{\hat{\theta}(\mathbf{x})}, \hat{\theta}(\mathbf{x}) + t_{1-\alpha/2} \cdot \sigma_{\hat{\theta}(\mathbf{x})} \right].$$

In testing hypotheses, when the standardized test statistic has the same form as the pivot used to construct a confidence interval, namely $t_{\text{obs}} = \frac{\hat{\theta}(\mathbf{x}) - \theta_0}{\sigma_{\hat{\theta}(\mathbf{x})}}$, and the confidence intervals and the acceptance region for the null hypothesis are based on the same distribution, there exists a duality between $(1 - \alpha) \cdot 100\%$ confidence intervals and α -level hypothesis tests. That is, when θ_0 is in the confidence interval, $H_0 : \theta = \theta_0$ is not rejected. This is summarized in general in Table 9.5.

Table 9.5: Duality of $(1 - \alpha) \cdot 100\%$ confidence intervals and α -level tests of significance

Alternative Hypothesis	Fail to Reject H_0 Region	$(1 - \alpha) \cdot 100\%$ Confidence Interval
$H_1 : \theta < \theta_0$	$t_{\text{obs}} \geq t_\alpha$	$(-\infty, \hat{\theta}(\mathbf{x}) - t_\alpha \cdot \sigma_{\hat{\theta}(\mathbf{x})}]$
$H_1 : \theta > \theta_0$	$t_{\text{obs}} \leq t_{1-\alpha}$	$(\hat{\theta}(\mathbf{x}) - t_{1-\alpha} \cdot \sigma_{\hat{\theta}(\mathbf{x})}, \infty)$
$H_1 : \theta \neq \theta_0$	$t_{\alpha/2} \leq t_{\text{obs}} \leq t_{1-\alpha/2}$	$\left[\hat{\theta}(\mathbf{x}) + t_{\alpha/2} \cdot \sigma_{\hat{\theta}(\mathbf{x})}, \hat{\theta}(\mathbf{x}) + t_{1-\alpha/2} \cdot \sigma_{\hat{\theta}(\mathbf{x})} \right]$

9.7 Hypothesis Tests for Population Means

The hypothesis tests for different averages parallel their respective confidence intervals. Again, underlying distributions are assumed to be normal, and various tests are done depending on whether variances are known or unknown and sample sizes are similar or quite different. It is important to check the underlying normality, because the procedures in this section depend on that assumption's truth. If normality is not present, nonparametric methods discussed in Chapter 10 will give more accurate results.

9.7.1 Test for the Population Mean When Sampling from a Normal Distribution with Known Population Variance

The null hypothesis for testing the mean when sampling from a normal distribution with known variance is $H_0 : \mu = \mu_0$, where μ_0 is a particular value. It is important to emphasize that a normal distribution as well as a known variance are being assumed. Seldom, if ever, will the distribution and its variance be known with certainty; however, a firm foundation in how significance tests are conducted with these assumptions will provide a base on which more hypothesis testing procedures can be built.

The basic idea behind a test of significance for the mean when working with a random sample of size n is to determine how likely the values observed in the sample are to occur. Typically, the sampling distribution of \bar{X} , which is $N(\mu_0, \sigma/\sqrt{n})$, is used to construct a standardized test statistic since one is sampling from a normal distribution under the assumption that the null hypothesis is true. Further, the Central Limit Theorem states that the sampling distribution of \bar{X} approaches a normal distribution even if the original population is not normal, provided the sample size n is sufficiently large. The standardized test statistic under the assumption that H_0 is true is

$$Z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} \sim N(0, 1).$$

The formula to calculate its observed value as well as the three possible alternative hypotheses and their rejection regions are described in Table 9.6 on the next page.

Example 9.7 A random sample of size $n = 30$ is taken from a distribution known to be $N(\mu, \sigma = 2)$. If the $\sum_{i=1}^{30} x_i = 56$,

- (a) Test the null hypothesis $H_0 : \mu = 1.8$ versus the alternative hypothesis $H_1 : \mu > 1.8$ at the $\alpha = 0.05$ significance level.
- (b) Find $\beta(\mu_1 = 3)$ and $\text{Power}(\mu_1 = 3)$.

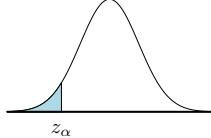
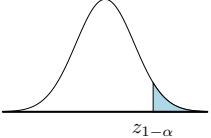
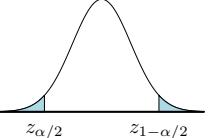
Solution: The answers are as follows:

- (a) Use the five-step procedure.

Step 1: **Hypotheses** — $H_0 : \mu = 1.8$ versus $H_1 : \mu > 1.8$.

Step 2: **Test Statistic** — The test statistic chosen is \bar{X} because $E[\bar{X}] = \mu$. The value of this test statistic is $\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{56}{30} = 1.8667$. The standardized test statistic and its distribution under the assumption H_0 is true are $Z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} \sim N(0, 1)$.

Table 9.6: Summary for testing the mean when sampling from a normal distribution with known variance (one-sample z -test)

Null Hypothesis — $H_0 : \mu = \mu_0$	Standardized Test Statistic's Value — $z_{\text{obs}} = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$		
Alternative Hypothesis	$H_1 : \mu < \mu_0$	$H_1 : \mu > \mu_0$	$H_1 : \mu \neq \mu_0$
Rejection Region	$z_{\text{obs}} < z_\alpha$	$z_{\text{obs}} > z_{1-\alpha}$	$ z_{\text{obs}} > z_{1-\alpha/2}$
Graphical Representation of Rejection Region			

Step 3: **Rejection Region Calculations** — Because the standardized test statistic is distributed $N(0, 1)$ and H_1 is an upper one-sided hypothesis, the rejection region is $z_{\text{obs}} > z_{1-\alpha} = z_{0.95} = 1.6449$. The value of the standardized test statistic is $z_{\text{obs}} = \frac{\frac{56}{30} - 1.8}{2/\sqrt{30}} = 0.1826$.

Step 4: **Statistical Conclusion** — The φ -value is $\mathbb{P}(Z \geq 0.1826) = 0.4276$.

- I. From the rejection region, fail to reject H_0 because 0.1826 is not greater than 1.6449.
- II. From the φ -value, fail to reject H_0 because the φ -value = 0.4276 is greater than 0.05.

Fail to reject H_0 .

Step 5: **English Conclusion** — There is not evidence to suggest that the mean is greater than 1.8.

See R Code 9.2 to find $z_{0.95}$ and the φ -value for a z_{obs} value of 0.1826 for a right tail alternative hypothesis.

R Code 9.2

```
> z <- qnorm(0.95)                      # Critical Value
> zobs <- ((56/30) - 1.8)/(2/sqrt(30))  # z_obs
> pvalue <- pnorm(zobs, lower = FALSE)    # p-value
> c(z = z, zobs = zobs, pvalue = pvalue)

      z      zobs      pvalue
1.6448536 0.1825742 0.4275661
```

(b) $\beta(\mu_1 = 3)$ and $\text{Power}(\mu_1 = 3)$ are

$$\begin{aligned}
 \beta(\mu_1 = 3) &= \mathbb{P}(\text{Type II error}) = \mathbb{P}\left(\text{Fail to reject } H_0 \mid \bar{X} \sim N\left(3, \frac{2}{\sqrt{30}}\right)\right) \\
 &= \mathbb{P}\left(\frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}} \leq z_{1-\alpha} \mid \bar{X} \sim N\left(3, \frac{2}{\sqrt{30}}\right)\right) \\
 &= \mathbb{P}\left(\bar{X} \leq z_{0.95} \frac{\sigma}{\sqrt{n}} + \mu_0 \mid \bar{X} \sim N\left(3, \frac{2}{\sqrt{30}}\right)\right) \\
 &= \mathbb{P}\left(\frac{\bar{X} - 3}{\frac{2}{\sqrt{30}}} \leq \frac{\frac{(1.6449)(2)}{\sqrt{30}} + 1.8 - 3}{\frac{2}{\sqrt{30}}}\right) \\
 &= \mathbb{P}(Z \leq -1.6415) = 0.0503 \text{ and}
 \end{aligned}$$

$$\text{Power}(\mu_1 = 3) = 1 - \beta(\mu_1 = 3) = 1 - 0.0503 = 0.9497.$$

R Code 9.3 computes $\beta(\mu_1 = 3)$ and $\text{Power}(\mu_1 = 3)$.

R Code 9.3

```

> beta3 <- pnorm(qnorm(0.95, 1.8, 2/sqrt(30)), 3, 2/sqrt(30))
> power3 <- 1 - beta3
> beta3

[1] 0.05034873

> power3

[1] 0.9496513

```

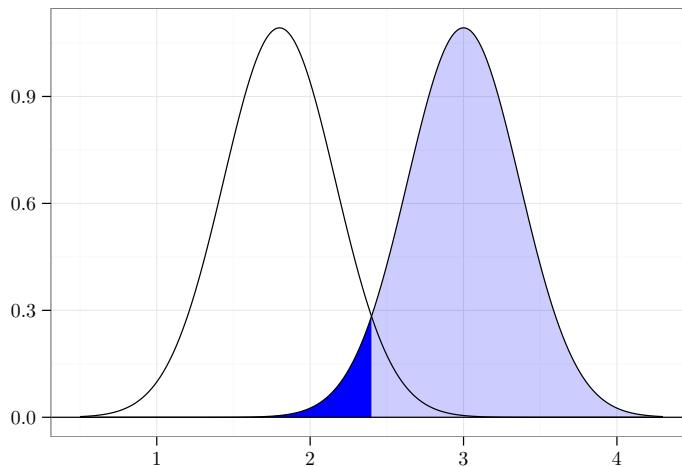
A graph that depicts $\beta(\mu_1 = 3)$ and $\text{Power}(\mu_1 = 3)$ is created in Figure 9.5 on the facing page. Two small functions are created in R Code 9.4 that shade the areas for $\beta(\mu_1 = 3)$ and $\text{Power}(\mu_1 = 3)$.

R Code 9.4

```

> library(ggplot2)
> p <- ggplot(data = data.frame(x = c(0.5, 4.3)), aes(x = x))
> dnorm_func <- function(x){
+   y <- dnorm(x, 3, 2/sqrt(30))
+   y[x < qnorm(0.95, 1.8, 2/sqrt(30))] <- NA
+   return(y)
+ }
> dnorm_func1 <- function(x){
+   y <- dnorm(x, 3, 2/sqrt(30))
+   y[x >= qnorm(0.95, 1.8, 2/sqrt(30))] <- NA
+   return(y)
+ }
> p + stat_function(fun = dnorm_func, geom = "area", fill = "blue",
+   alpha = 0.2, n = 500) +

```

FIGURE 9.5: Graph depicting $\beta(\mu_1 = 3)$ (dark shading) and Power($\mu_1 = 3$) (light shading)

```
+   stat_function(fun = dnorm_func1, geom = "area", fill = "blue") +
+   geom_hline(yintercept = 0) +
+   stat_function(fun = dnorm, args = list(1.8, 2/ sqrt(30)), n = 500) +
+   stat_function(fun = dnorm, args = list(3, 2/sqrt(30)), n = 500) +
+   theme_bw() +
+   labs(x = "", y = "")
```



9.7.2 Test for the Population Mean When Sampling from a Normal Distribution with Unknown Population Variance

The null hypothesis is still $H_0 : \mu = \mu_0$ when working with data from a normal distribution with unknown variance; however, the standardized test statistic under the assumption that H_0 is true is now

$$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}} \sim t_{n-1}.$$

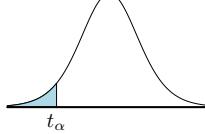
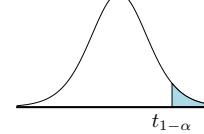
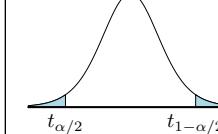
The formula to calculate its observed value as well as the three possible alternative hypotheses and their rejection regions are described in Table 9.7 on the following page. The computation of β and Power with the t -test is not nearly as easy as with the standard normal distribution. This is due to the fact that when the null hypothesis is false, the random variable $\frac{\bar{X}-\mu_0}{S/\sqrt{n}}$ has what is known as a non-central t -distribution denoted $t_{n-1;\gamma}^*$, with non-centrality parameter

$$\gamma = \frac{\mu_1 - \mu_0}{\frac{\sigma}{\sqrt{n}}},$$

where μ_1 is the true value of μ . To compute the power for a t -test, one must provide some estimate of σ for the non-centrality parameter. It is also possible to compute the power for a two-tailed t -test using the non-central F -distribution. The relationship between the non-central t -distribution and the non-central F -distribution is given in (9.4).

$$\mathbb{P}((t_{n-1;\gamma}^* < t_{\alpha/2;n-1}) \cup (t_{n-1;\gamma}^* > t_{1-\alpha/2;n-1})) = \mathbb{P}(F_{1,n-1;\gamma^2} > (t_{1-\alpha/2;n-1})^2) \quad (9.4)$$

Table 9.7: Summary for testing the mean when sampling from a normal distribution with unknown variance (one-sample t -test)

Null Hypothesis — $H_0 : \mu = \mu_0$	Standardized Test Statistic's Value — $t_{\text{obs}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$		
Alternative Hypothesis	$H_1 : \mu < \mu_0$	$H_1 : \mu > \mu_0$	$H_1 : \mu \neq \mu_0$
Rejection Region	$t_{\text{obs}} < t_{\alpha;n-1}$	$t_{\text{obs}} > t_{1-\alpha;n-1}$	$ t_{\text{obs}} > t_{1-\alpha/2;n-1}$
Graphical Representation of Rejection Region			
Note that the degrees of freedom for the t values in all the graphical representations are $n - 1$.			

Example 9.8 A random sample of size $n = 25$ is taken from a distribution known to be $N(\mu, \sigma)$. If the $\sum_{i=1}^n x_i = 100$ and the $\sum_{i=1}^n x_i^2 = 600$,

- (a) Test the null hypothesis $H_0 : \mu = 2.5$ versus the alternative hypothesis $H_1 : \mu \neq 2.5$ at the $\alpha = 0.05$ significance level.
- (b) Find Power($\mu_1 = 4$) if it is assumed $\sigma = 2.5$.
- (c) Use R to simulate a $t_{24;\gamma=3}^*$ distribution, and use it to compute the simulated power of the test in (b).

Solution: The answers are as follows:

- (a) To solve this part, use the five-step procedure.

Step 1: **Hypotheses** — These are given in the problem as

$$H_0 : \mu = 2.5 \text{ versus } H_1 : \mu \neq 2.5.$$

Step 2: **Test Statistic** — The test statistic chosen is \bar{X} because $E[\bar{X}] = \mu$. The value of this test statistic is $\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{100}{25} = 4$. The standardized test statistic and its distribution under the assumption H_0 is true are $\frac{\bar{X} - \mu_0}{S/\sqrt{n}} \sim t_{25-1}$.

Step 3: **Rejection Region Calculations** — Because the standardized test statistic is distributed t_{24} and H_1 is a two-tailed hypothesis, the rejection region is $|t_{\text{obs}}| >$

$t_{1-0.05/2;24} = t_{0.975;24} = 2.0639$. The value of the standardized test statistic is $t_{\text{obs}} = \frac{\bar{x}-\mu_0}{s/\sqrt{n}} = \frac{4-2.5}{2.8868/\sqrt{25}} = 2.5981$. (The value for s is calculated $\sqrt{\frac{\sum_{i=1}^n x_i^2 - n\bar{x}^2}{n-1}} = \sqrt{\frac{600-(25)(4^2)}{25-1}} = 2.8868$.)

Step 4: **Statistical Conclusion** — The p -value is $2 \cdot \mathbb{P}(t_{24} \geq 2.5981) = 0.01577$.

- I. From the rejection region, reject H_0 because $t_{\text{obs}} = 2.5981$ is greater than 2.0639.
- II. From the p -value, reject H_0 because the p -value = 0.01577 is less than 0.05.

Reject H_0 .

Step 5: **English Conclusion** — There is evidence to suggest that the mean is not equal to 2.5.

To use R to find $t_{0.975,24}$ and the p -value for a t_{obs} value of 2.5981 for a two-tailed alternative hypothesis, type

```
> ct <- qt(0.975, 24)
> S <- sqrt((600 - 25*4^2)/(25 - 1))
> tobs <- (4 - 2.5)/(S/sqrt(25))
> pvalue <- pt(tobs, 25 - 1, lower = FALSE)*2
> c(ct = ct, S = S, tobs = tobs, pvalue = pvalue)

      ct          S          tobs        pvalue
2.06389856 2.88675135 2.59807621 0.01577286
```

(b) Before computing $\text{Power}(\mu_1 = 4)$, first determine the non-centrality parameter:

$$\gamma = \frac{\mu_1 - \mu_0}{\frac{\sigma}{\sqrt{n}}} = \frac{4.0 - 2.5}{\frac{2.5}{\sqrt{25}}} = 3.0.$$

Let $T = t(\mathbf{X}) = \frac{\bar{X}-\mu_0}{S/\sqrt{n}}$. Then

$$\begin{aligned} \text{Power}(\mu_1 = 4) &= \mathbb{P}(\text{Reject } H_0 | H_1) \\ &= \mathbb{P}\left((T < t_{\alpha/2; n-1}) \cup (T > t_{1-\alpha/2; n-1}) \mid T \sim t_{n-1; \gamma}^*\right) \\ &= \mathbb{P}((t_{24; 3}^* < t_{0.025; 24}) \cup (t_{24; 3}^* > t_{0.975; 24})) \\ &= \mathbb{P}((t_{24; 3}^* < -2.0639) \cup (t_{24; 3}^* > 2.0639)) \\ &= \mathbb{P}((t_{24; 3}^* < -2.0639) + (t_{24; 3}^* > 2.0639)) = 0.8207. \end{aligned}$$

A graphical representation of the $\text{Power}(\mu_1 = 4)$ is depicted in Figure 9.6 on the next page. The $\text{Power}(\mu_1 = 4)$ using both the non-central t -distribution and the non-central F distribution will be computed. To find $\mathbb{P}((t_{24; 3}^* < t_{0.025; 24}) \cup (t_{24; 3}^* > t_{0.975; 24}))$ with R, enter

```
> pt(qt(0.025, 24), 24, 3) + pt(qt(0.975, 24), 24, 3, lower = FALSE)
```

```
[1] 0.8207219
```

Using the relationship between t -distributions and F -distributions given in (9.4), the Power($\mu_1 = 4$) is expressed as

$$\begin{aligned}\mathbb{P}((t_{24;3}^* < t_{0.025;24}) \cup (t_{24;3}^* > t_{0.975;24})) &= \mathbb{P}(F_{1,24;\gamma=3^2} > (t_{1-\alpha/2;n-1})^2) \\ &= \mathbb{P}(F_{1,24;\gamma=3^2} > (t_{0.975;24})^2) \\ &= \mathbb{P}(F_{1,24;\gamma=3^2} > (2.0639)^2 = 4.2597) \\ &= 0.8207.\end{aligned}$$

To find $\mathbb{P}(F_{1,24;9} > 4.2597) = 1 - \mathbb{P}(F_{1,24;9} < 4.2597)$ with R, key in

```
> pf(qt(0.975, 24)^2, 1, 24, 9, lower = FALSE)
[1] 0.8207219
```

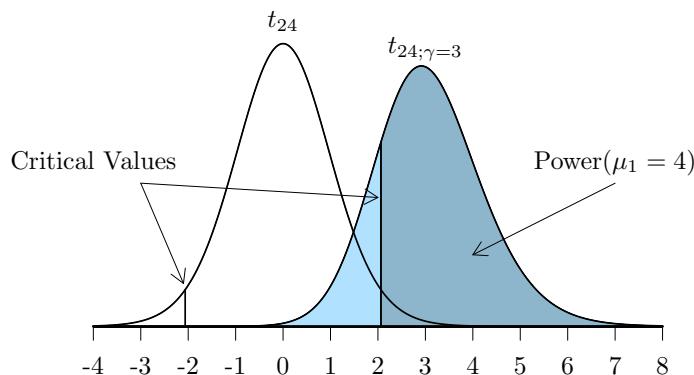


FIGURE 9.6: Central t -distribution and non-central t -distribution with $\gamma = 3$

One can also use the R function `power.t.test()` to compute the answer as follows:

```
> power.t.test(n = 25, delta = 1.5, sd = 2.5, type = "one.sample",
+ strict = TRUE)
```

One-sample t test power calculation

```
n = 25
delta = 1.5
sd = 2.5
sig.level = 0.05
power = 0.8207219
alternative = two.sided
```

(c) R Code 9.5 computes the simulated Power($\mu_1 = 4$), $\widehat{\text{Power}}(\mu_1 = 4)$.

R Code 9.5

```
> set.seed(7)
> SIMS <- 50000
```

```

> n <- 25
> tstar <- numeric(SIMS)
> for(i in 1:SIMS){
+   rs <- rnorm(n, 4, 2.5)
+   tstar[i] <- (mean(rs) - 2.5) / (sd(rs) / sqrt(n))
+ }
> power <- mean(tstar < qt(0.025, n - 1)) +
+   mean(tstar > qt(0.975, n - 1))
> power
[1] 0.81918

```

The simulated $\text{Power}(\mu_1 = 4) = \widehat{\text{Power}}(\mu_1 = 4) = 0.8192$. The simulated power is shown in Figure 9.7 and is created with R Code 9.6.

R Code 9.6

```

> DF <- data.frame(x = tstar)
> x.dens <- density(tstar)
> df.dens <- data.frame(x = x.dens$x, y = x.dens$y)
> p <- ggplot(data = DF) +
+   geom_density(aes(x = x, y = ..density..), fill = "blue",
+   alpha = 0.2)
> p + geom_area(data = subset(df.dens, x >= qt(0.975, 24) &
+   x <= max(DF$x)), aes(x = x, y = y), fill = "blue",
+   alpha = 1.0) +
+   stat_function(fun = dt, args = list(df = 24)) +
+   xlim(-4, 10) +
+   geom_vline(xintercept = qt(c(0.025, 0.975), 24), lty = "dashed") +
+   labs(x = "", y = "") +
+   theme_bw()

```

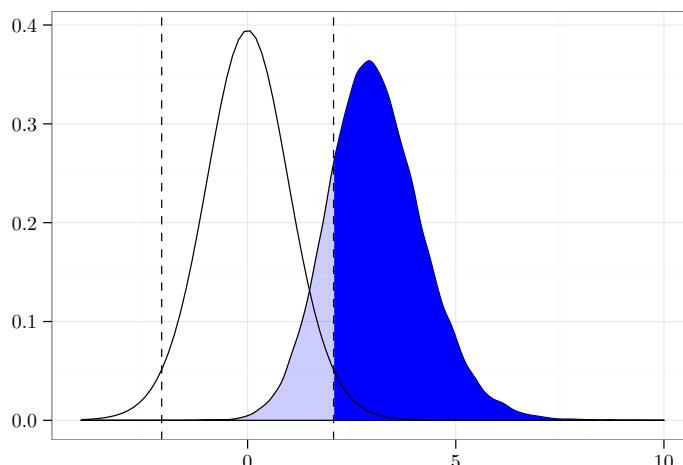


FIGURE 9.7: Central t -distribution and simulated non-central t -distribution with $\gamma = 3$. Dashed vertical lines represent critical values for testing the null hypothesis.

Example 9.9 ▷ One-Sample t-Test: Fertilizers ◁ A farmer wants to test if a new brand of fertilizer increases his wheat yields per plot. He puts the new fertilizer on 15 equal plots and records the subsequent yields for the 15 plots. If his traditional yield is two bushels per plot, conduct a test of significance for μ at the $\alpha = 0.05$ significance level after verifying the data follow a normal distribution. The yields for the 15 fields are

2.5 3.0 3.1 4.0 1.2 5.0 4.1 3.9 3.2 3.3 2.8 4.1 2.7 2.9 3.7

Solution: To solve this problem, start by verifying the normality assumption of the data using exploratory data analysis (`eda()`). The results from applying the function `eda()` to the wheat yields per plot are provided in Figure 9.8. Based on the graphical output from the function `eda()`, it is not unreasonable to assume that wheat yield follows a normal distribution. Now, proceed with the five-step procedure.

EXPLORATORY DATA ANALYSIS

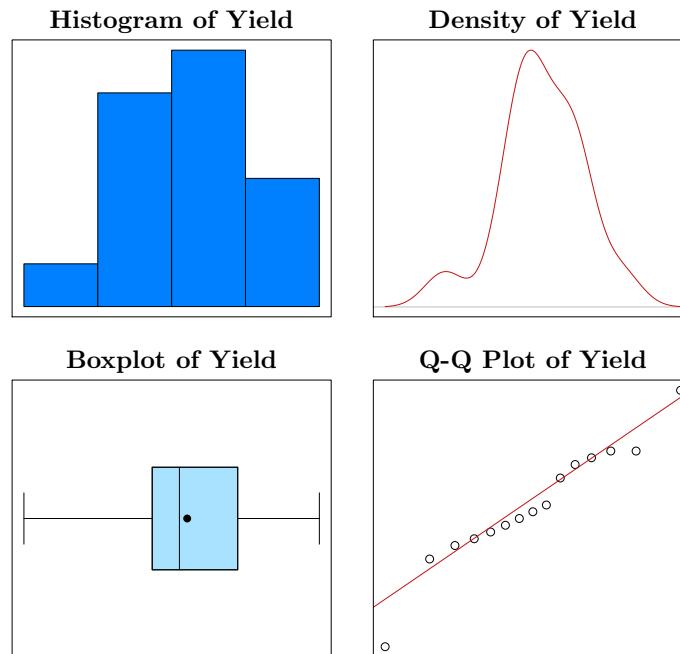


FIGURE 9.8: Exploratory data analysis of the wheat yield per plot values

Step 1: Hypotheses — To test if wheat yield is increased, the hypotheses are

$$H_0 : \mu = 2 \text{ versus } H_1 : \mu > 2.$$

Step 2: Test Statistic — The test statistic chosen is \bar{X} because $E[\bar{X}] = \mu$. The value of this test statistic is $\bar{x} = \frac{\sum_{i=1}^n x_i}{n} = \frac{49.5}{15} = 3.3$. The standardized test statistic under the assumption that H_0 is true and its distribution are $\frac{\bar{X} - \mu_0}{S/\sqrt{n}} \sim t_{15-1}$.

Step 3: Rejection Region Calculations — Because the standardized test statistic is distributed t_{14} and H_1 is an upper one-sided hypothesis, the rejection region is $t_{\text{obs}} > t_{1-0.05, 14} = t_{0.95, 14} = 1.7613$. The value of the standardized test statistic is $t_{\text{obs}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{3.3 - 2}{0.8920/\sqrt{15}} = 5.6443$.

Step 4: Statistical Conclusion — The p -value is $\mathbb{P}(t_{14} \geq 5.6443) = 0$.

- I. From the rejection region, reject H_0 because $t_{\text{obs}} = 5.6443$ is greater than 1.7613.
- II. From the p -value, reject H_0 because the p -value 0 is less than 0.05.

Reject H_0 .

Step 5: English Conclusion — There is evidence to suggest that the mean yield with the new fertilizer is greater than two bushels per plot.

To perform calculations with R, enter

```
> ct <- qt(0.95, 14) # critical value
> Yield <- c(2.5, 3, 3.1, 4, 1.2, 5, 4.1, 3.9,
+           3.2, 3.3, 2.8, 4.1, 2.7, 2.9, 3.7)
> xbar <- mean(Yield)
> S <- sd(Yield)
> tobs <- (xbar - 2) / (S / sqrt(15))
> pvalue <- pt(tobs, 15 - 1, lower = FALSE)      # pvalue
> pvalue

[1] 3.026149e-05

> c(CriticalValue = ct, xbar = xbar, S = S, tobs = tobs)

CriticalValue      xbar          S          tobs
1.7613101     3.3000000   0.8920282   5.6443041
```

To compute the value of the standardized test statistic and its corresponding p -value with the function `t.test()`, type

R Code 9.7

```
> t.test(Yield, alternative = "greater", mu = 2)
```

One Sample t-test

```
data: Yield
t = 5.6443, df = 14, p-value = 3.026e-05
alternative hypothesis: true mean is greater than 2
95 percent confidence interval:
 2.894334      Inf
sample estimates:
mean of x
 3.3
```

Note that the upper limit of the confidence interval in the R output is `Inf`, indicating that the limit on the right side of the confidence interval is ∞ . Also, the calculation of the lower limit uses (8.10) on page 464 modified for a one-sided confidence interval. ■

9.7.3 Test for the Difference in Population Means When Sampling from Independent Normal Distributions with Known Variances

When sampling from two normal distributions with known variances, the null hypothesis for testing the difference between two means is $H_0 : \mu_X - \mu_Y = \delta_0$, and the standardized test statistic under the assumption that H_0 is true is

$$Z = \frac{\bar{X} - \bar{Y} - \delta_0}{\sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}} \sim N(0, 1).$$

The formulas to calculate its observed value as well as the three possible alternative hypotheses and their rejection regions are described in Table 9.8. Note that testing the equality of two means ($H_0 : \mu_X = \mu_Y$) is the same as specifying $\delta_0 = 0$ in the null hypothesis $H_0 : \mu_X - \mu_Y = \delta_0$.

Table 9.8: Summary for test for differences in means when taking independent samples from normal distributions with known variances (two-sample z -test)

$$\text{Null Hypothesis} — H_0 : \mu_X - \mu_Y = \delta_0$$

$$\begin{array}{ll} \text{Standardized} & \\ \text{Test Statistic's} & z_{\text{obs}} = \frac{\bar{x} - \bar{y} - \delta_0}{\sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}} \\ \text{Value} & \end{array}$$

Alternative Hypothesis	Rejection Region
$H_1 : \mu_X - \mu_Y < \delta_0$	$z_{\text{obs}} < z_\alpha$
$H_1 : \mu_X - \mu_Y > \delta_0$	$z_{\text{obs}} > z_{1-\alpha}$
$H_1 : \mu_X - \mu_Y \neq \delta_0$	$ z_{\text{obs}} > z_{1-\alpha/2}$

Example 9.10 A researcher wishes to see if it is reasonable to believe that engineering majors have higher math SAT scores than English majors. She takes two random samples. The first sample consists of 64 engineering majors' SAT math scores (X). Typically, these scores follow a normal distribution with a known standard deviation of $\sigma_X = 100$ but with an unknown mean. The second sample consists of 144 observations of English majors' SAT scores (Y). These also follow a normal distribution with a standard deviation of $\sigma_Y = 108$ with an unknown mean as well.

- (a) Test the null hypothesis of equality of means at the 10% significance level ($\alpha = 0.1$) if one knows the difference in sample means is 20.

- (b) Find the power of the test in part (a) if $\mu_1(X, Y) = \mu_X - \mu_Y = 40$. (Note that $\mu_0(X, Y) = 0$ from H_0 .)

Solution: The answers are as follows:

- (a) Use the five-step procedure.

Step 1: **Hypotheses** — To test if engineering majors have a higher average math SAT score than English majors, the hypotheses are

$$H_0 : \mu_X - \mu_Y = 0 \text{ versus } H_1 : \mu_X - \mu_Y > 0.$$

Step 2: **Test Statistic** — The test statistic chosen is $\bar{X} - \bar{Y}$ because $E[\bar{X} - \bar{Y}] = \mu_X - \mu_Y$. The value of this test statistic is 20 according to the problem. The standardized test statistic under the assumption that H_0 is true and its distribution are

$$\frac{\bar{X} - \bar{Y} - \delta_0}{\sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}} \sim N(0, 1).$$

Step 3: **Rejection Region Calculations** — Because the standardized test statistic is distributed $N(0, 1)$ and H_1 is an upper one-sided hypothesis, the rejection region is $z_{\text{obs}} > z_{1-0.1} = z_{0.9} = 1.2816$. The value of the standardized test statistic is

$$z_{\text{obs}} = \frac{\bar{x} - \bar{y} - \delta_0}{\sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}} = \frac{20 - 0}{\sqrt{\frac{100^2}{64} + \frac{108^2}{144}}} = \frac{20}{15.4029} = 1.2985.$$

Step 4: **Statistical Conclusion** — The φ -value is $\mathbb{P}(Z \geq 1.2985) = 0.0971$.

- I. From the rejection region, reject H_0 because $z_{\text{obs}} = 1.2985$ is greater than 1.2816.
- II. From the φ -value, reject H_0 because the φ -value = 0.0971 is less than 0.1.

Reject H_0 .

Step 5: **English Conclusion** — There is evidence to suggest that the difference between the average math SAT score for engineering majors and that of the average math SAT score for English majors is greater than zero; therefore, the evidence suggests engineering majors have a higher average math SAT score.

- (b) Find the power of the test in part (a) if

$$H_1 : \bar{X} - \bar{Y} \sim N(\mu_1(X, Y) = 40, \sigma_{\bar{X}-\bar{Y}}).$$

Recall that $\mu_0(X, Y) = 0$ in H_0 and $\sigma_{\bar{X}-\bar{Y}} = \sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}} = 15.4029$.

$$\begin{aligned}
\beta(\mu_1(X, Y)) &= \mathbb{P}(\text{Fail to Reject } H_0 | H_1) \\
&= \mathbb{P}\left(\frac{\bar{X} - \bar{Y} - \mu_0(X, Y)}{\sigma_{\bar{X}-\bar{Y}}} \leq z_{0.9} | H_1\right) \\
&= \mathbb{P}\left(\frac{\bar{X} - \bar{Y} - 0}{\sigma_{\bar{X}-\bar{Y}}} \leq z_{0.9} | H_1\right) \\
&= \mathbb{P}(\bar{X} - \bar{Y} \leq z_{0.9} \sigma_{\bar{X}-\bar{Y}} | H_1) \\
&= \mathbb{P}\left(\frac{\bar{X} - \bar{Y} - \mu_1(X, Y)}{\sigma_{\bar{X}-\bar{Y}}} \leq \frac{z_{0.9} \sigma_{\bar{X}-\bar{Y}} - \mu_1(X, Y)}{\sigma_{\bar{X}-\bar{Y}}} | H_1\right) \\
&= \mathbb{P}\left(Z \leq \frac{(1.2816)(15.4029) - 40}{15.4029}\right) \\
&= \mathbb{P}(Z \leq -1.3154) = 0.0942.
\end{aligned}$$

So, the power is

$$\text{Power}(\mu_1(X, Y)) = 1 - \beta(\mu_1(X, Y)) = 1 - 0.0942 = 0.9058.$$

To find the power with R, enter

```

> sig <- sqrt(100^2/64 + 108^2/144)
> cv <- qnorm(0.9, 0, sig) # critical value
> BETA <- pnorm(cv, 40, sig)
> POWER <- 1 - BETA
> POWER
[1] 0.9058052

```



9.7.4 Test for the Difference in Means When Sampling from Independent Normal Distributions with Variances That Are Unknown but Assumed Equal

Recall that when random samples of size n_X and n_Y , respectively, are taken from two normal distributions $N(\mu_X, \sigma)$ and $N(\mu_Y, \sigma)$, where σ is unknown, the random variable

$$T = \frac{[(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)]}{\sqrt{S_p^2 \left(\frac{1}{n_X} + \frac{1}{n_Y}\right)}} \sim t_{n_X+n_Y-2}$$

by Theorem 6.4 on page 396, where $S_p^2 = \frac{(n_X-1)S_X^2 + (n_Y-1)S_Y^2}{n_X+n_Y-2}$. The null hypothesis used to test for a difference of means between two normal distributions where the variances are assumed to be unknown but equal is $H_0 : \mu_X - \mu_Y = \delta_0$. When H_0 is false, the random variable T has a non-central t -distribution with non-centrality parameter

$$\gamma = \frac{\mu_1(X, Y) - \mu_0(X, Y)}{\sigma_{\bar{X}-\bar{Y}}}$$

where $\mu_1(X, Y)$ is the value of $\mu_X - \mu_Y$ under H_1 and $\mu_0(X, Y) = \delta_0$. This distribution is denoted $t_{n_X+n_Y-2; \gamma}^*$. The value of the standardized test statistic is written

$$t_{\text{obs}} = \frac{\bar{x} - \bar{y} - \delta_0}{\sqrt{\frac{(n_X-1)s_X^2 + (n_Y-1)s_Y^2}{n_X+n_Y-2}} \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}} = \frac{\bar{x} - \bar{y} - \delta_0}{s_p \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}}.$$

The three possible alternative hypotheses and the corresponding rejection regions are in Table 9.9. Use of the pooled t -test should only be undertaken when the variances of X and Y are almost certainly equal.

Table 9.9: Summary for test for differences in means when taking independent samples from normal distributions with unknown but assumed equal variances (two-sample pooled t -test)

$$\text{Null Hypothesis} — H_0 : \mu_X - \mu_Y = \delta_0$$

$$\begin{array}{l} \text{Standardized Test} \\ \text{Statistic's Value} \end{array} — t_{\text{obs}} = \frac{\bar{x} - \bar{y} - \delta_0}{s_p \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}}$$

Alternative Hypothesis	Rejection Region
$H_1 : \mu_X - \mu_Y < \delta_0$	$t_{\text{obs}} < t_{\alpha; n_X+n_Y-2}$
$H_1 : \mu_X - \mu_Y > \delta_0$	$t_{\text{obs}} > t_{1-\alpha; n_X+n_Y-2}$
$H_1 : \mu_X - \mu_Y \neq \delta_0$	$ t_{\text{obs}} > t_{1-\alpha/2; n_X+n_Y-2}$

Example 9.11 \triangleright ***t-Test, $\sigma_X = \sigma_Y$ Assumed: School Satisfaction*** \triangleleft A questionnaire is devised by the Board of Governors to measure the level of satisfaction for graduates from two competing state schools. Past history indicates the variance in satisfaction levels for both schools is equal. The questionnaire is randomly administered to 11 students from State School X and 15 students from State School Y (the results have been ordered and stored in data frame **STCHOOL**).

School X: 69 75 76 80 81 82 86 89 91 92 97

School Y: 59 62 66 70 70 75 75 77 78 79 81 84 84 86 94

- (a) Test to see if there are significant differences between the mean satisfaction levels for graduates of the two competing state schools using a significance level of 5%.
- (b) Find the power for $\mu_1(X, Y) = \mu_X - \mu_Y = 10$ for the test in (a) if it is assumed $\sigma_X = \sigma_Y = 9$.

Solution: The answers are as follows:

- (a) To solve this part, start by verifying the reasonableness of the normality assumption. The side-by-side boxplots and normal quantile-quantile plots depicted in Figure 9.9 suggest

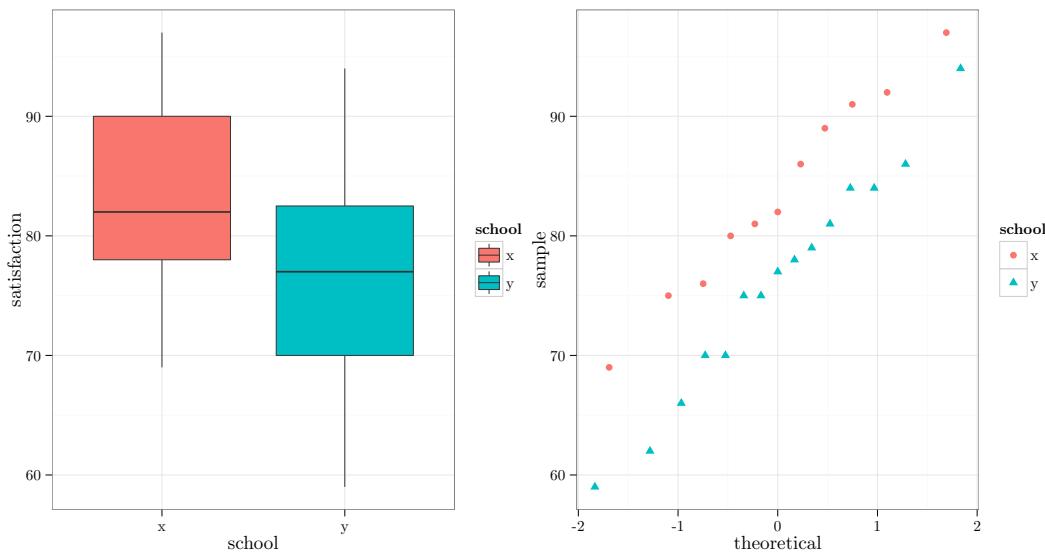


FIGURE 9.9: Side-by-side boxplots and normal quantile-quantile plots of the satisfaction level for graduates from State School X and State School Y

it is reasonable to assume the satisfaction levels for graduates from both state schools follow normal distributions with nearly identical interquartile ranges.

Five-Step Procedure:

Step 1: Hypotheses — Since the problem gives no reason to suspect graduates from School X are any more satisfied than graduates from School Y, use a two-tailed alternative hypothesis:

$$H_0 : \mu_X - \mu_Y = 0 \text{ versus } H_1 : \mu_X - \mu_Y \neq 0.$$

Step 2: Test Statistic — The test statistic chosen is $\bar{X} - \bar{Y}$ because $E[\bar{X} - \bar{Y}] = \mu_X - \mu_Y$. The value of this test statistic is $83.4545 - 76 = 7.4545$. The standardized test statistic under the assumption that H_0 is true and its distribution are

$$\frac{[(\bar{X} - \bar{Y}) - \delta_0]}{\sqrt{S_p^2 \left(\frac{1}{n_X} + \frac{1}{n_Y} \right)}} \sim t_{n_X + n_Y - 2}.$$

Step 3: Rejection Region Calculations — Because the standardized test statistic is distributed $t_{n_X + n_Y - 2}$ and H_1 is a two-sided hypothesis, the rejection region is $|t_{\text{obs}}| > t_{1 - 0.05/2; 11+15-2} = t_{0.975; 24} = 2.0639$. Note that $s_X = 8.4067$, $s_Y = 9.4491$, and the pooled standard deviation is $s_p = 9.0294$. The value of the standardized test statistic is

$$t_{\text{obs}} = \frac{\bar{x} - \bar{y} - \delta_0}{s_p \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}} = \frac{83.4545 - 76 - 0}{9.0294 \sqrt{\frac{1}{11} + \frac{1}{15}}} = 2.0798.$$

Step 4: Statistical Conclusion — The φ -value is $2 \times \mathbb{P}(t_{24} \geq 2.0798) = 2 \times 0.0242 = 0.0484$.

- I. From the rejection region, reject H_0 because $|t_{\text{obs}}| = 2.0798$ is greater than 2.0639.
- II. From the ϕ -value, reject H_0 because the ϕ -value = 0.0484 is less than 0.05.

Reject H_0 .

Step 5: **English Conclusion** — There is evidence to suggest the average satisfaction levels of State School X and State School Y are different.

To compute the value of the standardized test statistic and its corresponding ϕ -value with the function `t.test()`, key in

```
> t.test(satisfaction ~ school, data = STSCHOOL, var.equal = TRUE)
```

Two Sample t-test

```
data: satisfaction by school
t = 2.0798, df = 24, p-value = 0.0484
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.05691592 14.85217499
sample estimates:
mean in group x mean in group y
 83.45455      76.00000
```

The confidence interval is calculated using (8.15) on page 474 and does not include 0. Thus, a conclusion based on this interval would be identical to that in step 5 of the five-step procedure used to solve this problem.

(b) Before computing $\text{Power}(\mu_1(X, Y) = 10)$, first determine the non-centrality parameter:

$$\gamma = \frac{\mu_1(X, Y) - \mu_0(X, Y)}{\sigma_{\bar{X}-\bar{Y}}} = \frac{10 - 0}{\sqrt{\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}}} = \frac{10}{\sqrt{\frac{9^2}{11} + \frac{9^2}{15}}} = \frac{10}{3.5726} = 2.7991.$$

Let $T = t(\mathbf{X}, \mathbf{Y}) = \frac{\bar{X} - \bar{Y} - \mu_0(X, Y)}{S_p \sqrt{\frac{1}{n_X} + \frac{1}{n_Y}}}$. Then

$$\begin{aligned} \text{Power}(\mu_1(X, Y) = 10) &= \mathbb{P}(\text{Reject } H_0 | H_1) \\ &= \mathbb{P}\left((T < t_{\alpha/2; n_X+n_Y-2}) \mid T \sim t_{n_X+n_Y-2; \gamma}^*\right) + \\ &\quad \mathbb{P}\left((T > t_{1-\alpha/2; n_X+n_Y-2}) \mid T \sim t_{n_X+n_Y-2; \gamma}^*\right) \\ &= \mathbb{P}((t_{24; 2.7991}^* < t_{0.025; 24})) + \mathbb{P}((t_{24; 2.7991}^* > t_{0.975; 24})) \\ &= \mathbb{P}((t_{24; 2.7991}^* < -2.0639)) + \mathbb{P}((t_{24; 2.7991}^* > 2.0639)) = 0.7660. \end{aligned}$$

Find the $\text{Power}(\mu_1(X, Y) = 10)$ using the non-central t -distribution and the non-central F distribution. To calculate the quantity $\mathbb{P}((t_{24; 2.7991}^* < t_{0.025; 24})) + \mathbb{P}((t_{24; 2.7991}^* > t_{0.975; 24}))$ with R, enter

```
> gamma <- 10/sqrt(9^2/11 + 9^2/15)
> gamma
[1] 2.799064

> power <- pt(qt(0.025, 24), 24, gamma) + pt(qt(0.975, 24),
+      24, gamma, lower = FALSE)
> power
[1] 0.7659719
```

Using the relationship between t -distributions and F distributions given in (9.4), write

$$\begin{aligned}\mathbb{P}((t_{24;2.7991}^* < t_{0.025;24}) \cup (t_{24;2.7991}^* > t_{0.975;24})) &= \mathbb{P}(F_{1,24;\gamma=2.7991^2} > (t_{1-\alpha/2;n-1})^2) \\ &= \mathbb{P}(F_{1,24;\gamma=2.7991^2} > (t_{0.975;24})^2) \\ &= \mathbb{P}(F_{1,24;\gamma=2.7991^2} > (2.0639)^2 = 4.26) \\ &= 0.7660.\end{aligned}$$

To find $\mathbb{P}(F_{1,24;2.7991^2} > 4.26) = 1 - \mathbb{P}(F_{1,24;7.8348} < 4.26)$ with R, key in

```
> 1 - pf(qt(0.975, 24)^2, 1, 24, gamma^2)
[1] 0.7659719
```



9.7.5 Test for a Difference in Means When Sampling from Independent Normal Distributions with Variances That Are Unknown and Not Assumed Equal

Recall that when random samples of size n_X and n_Y , respectively, are taken from two normal distributions $N(\mu_X, \sigma)$, and $N(\mu_Y, \sigma)$, where σ is known, the random variable

$$Z = \frac{[(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)]}{\sqrt{\left(\frac{\sigma_X^2}{n_X} + \frac{\sigma_Y^2}{n_Y}\right)}} \sim N(0, 1).$$

In real problems, the values of the population variances are seldom known. Further, the random variable

$$\frac{[(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)]}{\sqrt{\left(\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}\right)}}, \quad (9.5)$$

does not have a known distribution; however, the random variable in (9.5) can be approximated with a t -distribution with ν degrees of freedom, where

$$\nu = \frac{\left(\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}\right)^2}{\frac{(s_X^2/n_X)^2}{n_X - 1} + \frac{(s_Y^2/n_Y)^2}{n_Y - 1}}. \quad (9.6)$$

The approximation of the random variable (9.5) with a t_ν is known as the **Welch-Satterthwaite** method. Output from R using this technique is simply labeled **Welch**.

The null hypothesis used to test for a difference of means between two independent normal distributions where the variances are unknown and unequal is $H_0 : \mu_X - \mu_Y = \delta_0$. The value of the standardized test statistic using the Welch-Satterthwaite method is written

$$t_{\text{obs}} = \frac{\bar{x} - \bar{y} - \delta_0}{\sqrt{\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}}}.$$

The three possible alternative hypotheses and the corresponding rejection regions are in Table 9.10.

Table 9.10: Summary for test for differences in means when taking independent samples from normal distributions with unknown and unequal variances (Welch test)

$$\text{Null Hypothesis} — H_0 : \mu_X - \mu_Y = \delta_0$$

$$\begin{array}{ll} \text{Standardized Test} & t_{\text{obs}} = \frac{\bar{x} - \bar{y} - \delta_0}{\sqrt{\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}}} \\ \text{Statistic's Value} & \end{array}$$

Alternative Hypothesis	Rejection Region
$H_1 : \mu_X - \mu_Y < \delta_0$	$t_{\text{obs}} < t_{\alpha;\nu}$
$H_1 : \mu_X - \mu_Y > \delta_0$	$t_{\text{obs}} > t_{1-\alpha;\nu}$
$H_1 : \mu_X - \mu_Y \neq \delta_0$	$ t_{\text{obs}} > t_{1-\alpha/2;\nu}$

Example 9.12 A bottled water company acquires its water from two independent sources, X and Y. The company suspects that the sodium content in the water from source X is less than the sodium content for water from source Y. An independent agency measures the sodium content in 20 samples from source X and 10 samples from source Y and stores them in data frame `WATER`. Is there statistical evidence to suggest the average sodium content in the water from source X is less than the average sodium content in the water from source Y? The measurements for the sodium values are mg/L. Use an α level of 0.05 to test the appropriate hypotheses.

Source X: 84 73 92 84 95 74 80 86 80 77
86 72 62 54 77 63 85 59 66 79

Source Y: 78 79 84 82 80 85 81 83 79 81

Solution: To solve this problem, start by verifying the reasonableness of the normality assumption. The side-by-side boxplots and normal quantile-quantile plots depicted in Figure 9.10 on the next page suggest it is reasonable to assume the sodium values for both sources follow normal distributions; however, it is clear from the boxplots that the variances are very different. Now, proceed with the five-step procedure.

Step 1: Hypotheses — Since the problem wants to test to see if the mean sodium content from source X is less than the mean sodium content from source Y, use a lower

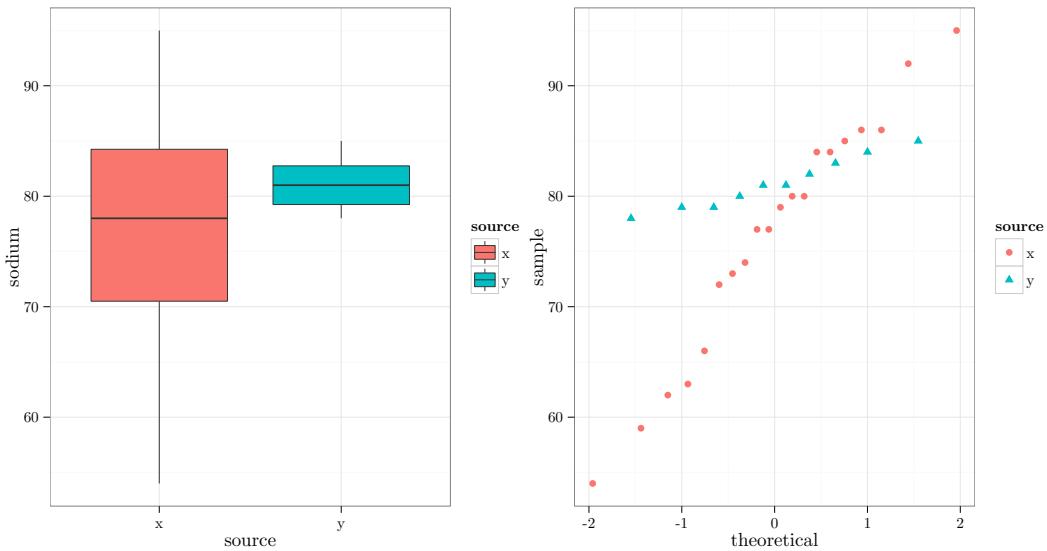


FIGURE 9.10: Side-by-side boxplots and normal quantile-quantile plots of the sodium content for source X and source Y

one-sided alternative hypothesis.

$$H_0 : \mu_X - \mu_Y = 0 \text{ versus } H_1 : \mu_X - \mu_Y < 0.$$

Step 2: **Test Statistic** — The test statistic chosen is $\bar{X} - \bar{Y}$ because $E[\bar{X} - \bar{Y}] = \mu_X - \mu_Y$. The value of this test statistic is $76.4 - 81.2 = -4.8$. The standardized test statistic under the assumption that H_0 is true and its approximate distribution are

$$\frac{[(\bar{X} - \bar{Y}) - \delta_0]}{\sqrt{\left(\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}\right)}} \approx t_\nu.$$

Step 3: **Rejection Region Calculations** — Because the standardized test statistic is distributed approximately t_ν and H_1 is a lower one-sided hypothesis, the rejection region is $t_{\text{obs}} < t_{0.05; 22.069} = -1.7169$. The degrees of freedom are

$$\nu = \frac{\left(\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}\right)^2}{\frac{(s_X^2/n_X)^2}{n_X-1} + \frac{(s_Y^2/n_Y)^2}{n_Y-1}} = \frac{\left(\frac{122.7789}{20} + \frac{5.2889}{10}\right)^2}{\frac{(122.7789/20)^2}{20-1} + \frac{(5.2889/10)^2}{10-1}} = 22.069,$$

and the value of the standardized test statistic is

$$t_{\text{obs}} = \frac{\bar{x} - \bar{y} - \delta_0}{\sqrt{\frac{s_X^2}{n_X} + \frac{s_Y^2}{n_Y}}} = \frac{76.4 - 81.2 - 0}{\sqrt{\frac{122.7789}{20} + \frac{5.2889}{10}}} = -1.8589.$$

Step 4: **Statistical Conclusion** — The φ -value is $\mathbb{P}(t_{22.069} \leq -1.8589) = 0.0382$.

- I. From the rejection region, reject H_0 because $t_{\text{obs}} = -1.8589$ is less than -1.7169 .

II. From the φ -value, reject H_0 because the φ -value = 0.0382 is less than 0.05.

Reject H_0 .

Step 5: **English Conclusion** — There is evidence to suggest the average sodium content for source X is less than the average sodium content for source Y.

To compute the value of the standardized test statistic and its corresponding φ -value with the function `t.test()`, type

```
> t.test(sodium ~ source, data = WATER, var.equal = FALSE, alt = "less")
```

```
Welch Two Sample t-test

data: sodium by source
t = -1.8589, df = 22.069, p-value = 0.03822
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
-Inf -0.3665724
sample estimates:
mean in group x mean in group y
76.4           81.2
```

The confidence interval R calculates agrees with the one from (8.16) on page 476 modified for a one-sided confidence interval. Note that the values included in the confidence interval are all less than zero, which would give a conclusion identical to that found in step 5 of the five-step procedure. ■

9.7.6 Test for the Mean Difference When the Differences Have a Normal Distribution

If one wants to test whether there has been some change in a single group of subjects or if there exists some difference between two dependent samples, one can compute the net change from one condition to the next and do a paired t -test provided certain normality assumptions are satisfied. Recall from Section 8.2.7 on page 480 that when a researcher is presented with paired samples, the standard approach is to analyze the differences between the paired data. Provided the distribution of population differences is

$$D \sim N(\mu_D = \mu_X - \mu_Y, \sigma_D),$$

the random variable

$$T = \frac{\bar{D} - \mu_D}{S_D / \sqrt{n_D}} \sim t_{n-1}.$$

The null hypothesis for testing a difference of means with dependent samples is $H_0 : \mu_D = \mu_X - \mu_Y = \delta_0$, and the value of the standardized test statistic is written

$$t_{\text{obs}} = \frac{\bar{d} - \delta_0}{s_D / \sqrt{n_D}}.$$

The three alternative hypotheses and the rejection regions for H_0 are in Table 9.11 on the following page. The paired t -test has a smaller variance than does an independent

two-sample t -test when the data are dependent and is a special case of the experimental design known as the randomized block design. The matched differences are known as **blocks**. Blocks should be used any time the differences within a block are relatively homogeneous compared to the differences within the particular treatment. When blocks are used appropriately, differences noted in the paired observations can subsequently be attributed to differences in treatments.

Table 9.11: Summary for testing the mean of the differences between two dependent samples when the differences follow a normal distribution with unknown variance (paired t -test)

$$\text{Null Hypothesis} — H_0 : \mu_D = \mu_X - \mu_Y = \delta_0$$

$$\begin{array}{ll} \text{Standardized} & \\ \text{Test Statistic's} & — t_{\text{obs}} = \frac{\bar{d} - \delta_0}{s_D / \sqrt{n_D}} \\ \text{Value} & \end{array}$$

Alternative Hypothesis	Rejection Region
$H_1 : \mu_D < \delta_0$	$t_{\text{obs}} < t_{\alpha;n-1}$
$H_1 : \mu_D > \delta_0$	$t_{\text{obs}} > t_{1-\alpha;n-1}$
$H_1 : \mu_D \neq \delta_0$	$ t_{\text{obs}} > t_{1-\alpha/2;n-1}$

Example 9.13 The data frame **barley** in the **lattice** package lists barley yield in bushels per acre for the years 1931 and 1932 for ten varieties of barley grown at six sites. Is there evidence to suggest the average barley yield in 1932 for the Morris site is greater than the average barley yield in 1932 for the Crookston site? Use the five-step procedure to test the appropriate hypotheses using an $\alpha = 0.05$ significance level.

Solution: Note that the same ten varieties are grown at both the Morris and the Crookston sites. Consequently, the yields at the two sites are dependent on the varieties. That is, variety acts as a block. It stands to reason that one can expect less variability between two similar plots growing the same variety than the variability within each of the plots growing different varieties. Start the analysis by verifying the normality assumption required to use a paired t -test. The results from applying the function **eda()** to the differences between the 1932 barley yields from the Morris and Crookston sites are provided in Figure 9.11 on the next page. Based on the graphical output from the function **eda()**, it is not unreasonable to assume the differences between the 1932 barley yields from the Morris and Crookston sites follow a normal distribution. Now, proceed with the five-step procedure.

Step 1: **Hypotheses** — To test if the average 1932 barley yield from Morris is greater than the average 1932 barley yield from Crookston, the hypotheses are

$$H_0 : \mu_D = 0 \text{ versus } H_1 : \mu_D > 0.$$

Step 2: **Test Statistic** — The test statistic chosen is \bar{D} because $E[\bar{D}] = \mu_D$. The value of this test statistic is $\bar{d} = 10.3333$. The standardized test statistic under the assumption that H_0 is true and its distribution are $\frac{\bar{D} - \delta_0}{s_D / \sqrt{n_D}} \sim t_{10-1}$.

EXPLORATORY DATA ANALYSIS

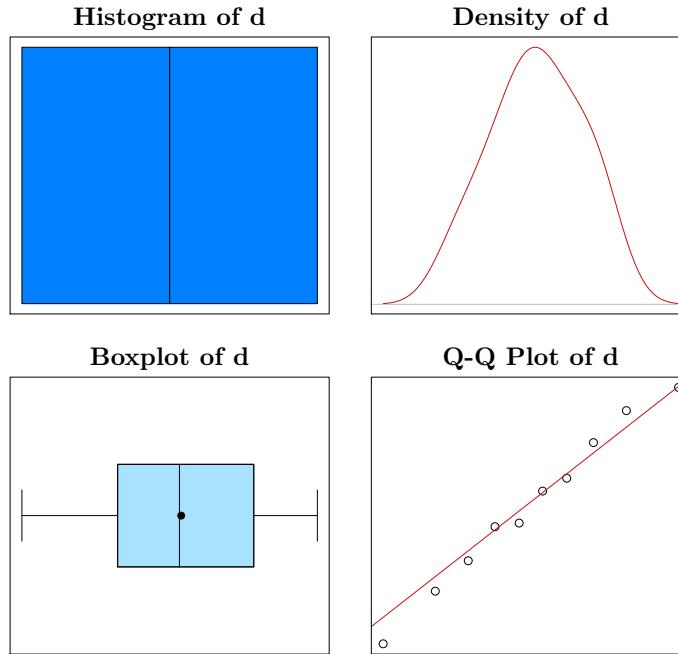


FIGURE 9.11: Exploratory data analysis of the differences between 1932 barley yields from the Morris and Crookston sites

Step 3: Rejection Region Calculations — Because the standardized test statistic is distributed t_9 and H_1 is an upper one-sided hypothesis, the rejection region is $t_{\text{obs}} > t_{1-0.05;9} = t_{0.95;9} = 1.8331$. The value of the standardized test statistic is $t_{\text{obs}} = \frac{\bar{d}-\delta_0}{s_D/\sqrt{n_D}} = \frac{10.3333-0}{5.1931/\sqrt{10}} = 6.2924$.

Step 4: Statistical Conclusion — The φ -value is $\mathbb{P}(t_9 \geq 6.2924) = 1e - 04$.

- From the rejection region, reject H_0 because $t_{\text{obs}} = 6.2924$ is greater than 1.8331.
- From the φ -value, reject H_0 because the φ -value $1e - 04$ is less than 0.05.

Reject H_0 .

Step 5: English Conclusion — There is evidence to suggest that the 1932 mean barley yield for Morris is greater than the 1932 mean barley yield for Crookston.

To compute the rejection region, value of the standardized test statistic, and its corresponding φ -value with `t.test()`, see R Code 9.8

R Code 9.8

```
> ct <- qt(0.95, 9) # Critical Value
> ct
[1] 1.833113
```

```
> yieldMor32 <- barley$yield[barley$year == "1932" & barley$site ==
+     "Morris"]
> yieldCro32 <- barley$yield[barley$year == "1932" & barley$site ==
+     "Crookston"]
> d <- yieldMor32 - yieldCro32
> t.test(d, alternative = "greater") # t-test on d
```

One Sample t-test

```
data: d
t = 6.2924, df = 9, p-value = 7.113e-05
alternative hypothesis: true mean is greater than 0
95 percent confidence interval:
 7.323012      Inf
sample estimates:
mean of x
10.33333
```

An alternative approach that yields identical results is to use the `paired = TRUE` argument with `t.test()`. The values can be passed to `t.test()` as either vectors or as a formula. R Code 9.9 uses both approaches after creating the data frame `DF`. Do note that the levels of the `ind` variable in the data frame `DF` are alphabetical with `yieldCro32` first and `yieldMor32` second. Consequently, one must use `alternative = less` when passing a formula to `t.test()` for an equivalent test to the one when vectors are passed to `t.test()` that are not in alphabetical order and the argument `paired = TRUE` is used.

R Code 9.9

```
> yieldMor32 <- barley$yield[barley$year == "1932" & barley$site ==
+     "Morris"]
> yieldCro32 <- barley$yield[barley$year == "1932" & barley$site ==
+     "Crookston"]
> t.test(yieldMor32, yieldCro32, paired = TRUE, alternative = "greater")
```

Paired t-test

```
data: yieldMor32 and yieldCro32
t = 6.2924, df = 9, p-value = 7.113e-05
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 7.323012      Inf
sample estimates:
mean of the differences
10.33333

> DF <- stack(data.frame(yieldMor32, yieldCro32))
> head(DF) # show first 6 rows of DF

  values      ind
1 34.36666 yieldMor32
2 35.13333 yieldMor32
```

```

3 35.03333 yieldMor32
4 38.83333 yieldMor32
5 46.63333 yieldMor32
6 43.53334 yieldMor32

> t.test(values ~ ind, data = DF, paired = TRUE, alternative = "less")

Paired t-test

data: values by ind
t = -6.2924, df = 9, p-value = 7.113e-05
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
-Inf -7.323012
sample estimates:
mean of the differences
-10.33333

```

Note that the confidence interval calculated by `t.test()` is using (8.21) on page 480 modified for one-sided confidence intervals. The interval calculated agrees with our conclusion from step 5 because it contains values that are exclusively greater than zero. ■

9.8 Hypothesis Tests for Population Variances

The hypothesis tests for a single variance and a ratio of variances are based on the same pivots as those used for the related confidence intervals. An underlying normal population is required for the hypothesis tests described in this section to give accurate results. If this assumption is violated, nonparametric methods should be used to come to reasonable conclusions based on the data.

9.8.1 Test for the Population Variance When Sampling from a Normal Distribution

The tests for population means presented up to this point have assumed the sampling distributions for their corresponding statistics follow a normal distribution; however, the tests for means are fairly robust to violations in normality assumptions. In contrast, the normality assumption for testing a hypothesis about variance is not robust to departures from normality. Consequently, one should proceed with caution when testing a hypothesis about the variance especially since non-normality is difficult to detect when working with small to moderate size samples. As a minimum, one should look at a normal quantile-quantile plot to make sure normality is plausible before testing a hypothesis concerning the population variance.

Provided X_1, X_2, \dots, X_n is a random sample from a $N(\mu, \sigma^2)$ distribution, the random variable

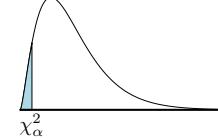
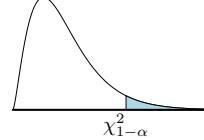
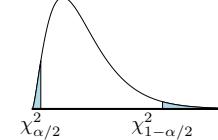
$$\frac{(n-1)S^2}{\sigma^2} \sim \chi_{n-1}^2.$$

The null hypothesis for testing the population variance is $H_0 : \sigma^2 = \sigma_0^2$, and the value for the test statistic is

$$\chi_{\text{obs}}^2 = \frac{(n - 1)s^2}{\sigma_0^2}.$$

The three alternative hypotheses and the rejection regions for H_0 are in Table 9.12.

Table 9.12: Summary for testing the population variance when sampling from a normal distribution

Null Hypothesis — $H_0 : \sigma^2 = \sigma_0^2$	Standardized Test Statistic's Value — $\chi_{\text{obs}}^2 = \frac{(n - 1)s^2}{\sigma_0^2}$		
Alternative Hypothesis	$H_1 : \sigma^2 < \sigma_0^2$	$H_1 : \sigma^2 > \sigma_0^2$	$H_1 : \sigma^2 \neq \sigma_0^2$
Rejection Region	$\chi_{\text{obs}}^2 < \chi_{\alpha;n-1}^2$	$\chi_{\text{obs}}^2 > \chi_{1-\alpha;n-1}^2$	$\chi_{\text{obs}}^2 < \chi_{\alpha/2;n-1}^2 \cup \chi_{\text{obs}}^2 > \chi_{1-\alpha/2;n-1}^2$
Graphical Representation of Rejection Region			
Note that the degrees of freedom for all the χ^2 values are $n - 1$.			

Example 9.14 The quality control office of a large hardware manufacturer received more than twice the number of complaints it usually receives in reference to the diameter variability of its 4 cm washers. In light of the complaints, the quality control manager wants to ascertain whether or not there has been an increase in the diameter variability of the company's washers manufactured this month versus last month, where the variance was 0.004 cm^2 . The manager takes a random sample of 20 washers manufactured this month. The results are recorded in Table 9.13 and stored in the data frame **WASHER**. Conduct an appropriate hypothesis test using a significance level of $\alpha = 0.05$.

Table 9.13: Diameters for 20 randomly selected washers (**WASHER**)

4.06	4.02	4.04	4.04	3.97	3.87	4.03	3.85	3.91	3.98
3.96	3.90	3.95	4.11	4.00	4.12	4.00	3.98	3.92	4.02

Solution: Prior to using a test that is very sensitive to departures in normality, as a minimum, create a quantile-quantile plot to verify the assumption of normality. The results from applying `eda()`, shown in Figure 9.12 on the next page, suggest the diameters of the washers follow a normal distribution. Now, continue with the five-step procedure.

EXPLORATORY DATA ANALYSIS

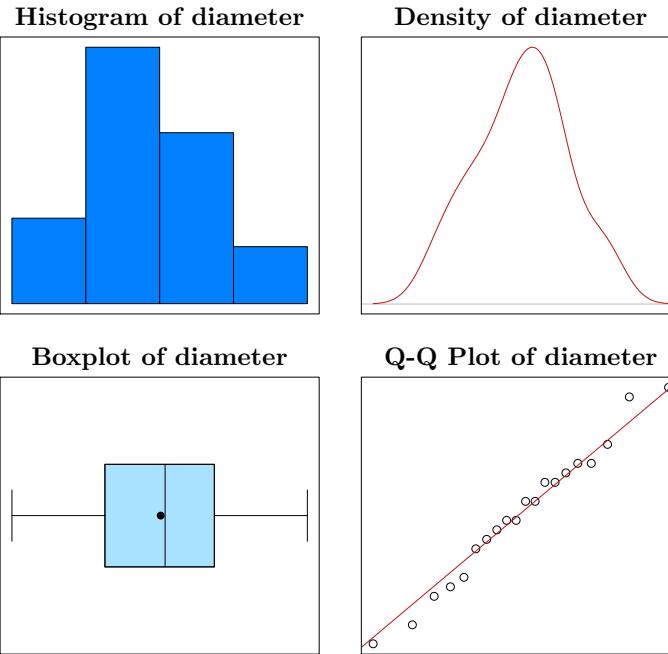


FIGURE 9.12: Graphs from using `eda()` on the washers' diameters

Step 1: Hypotheses — The null and alternative hypotheses to test whether the diameter variability of the company's washers manufactured this month is greater than the variability last month, where the variance was 0.004 cm^2 , are

$$H_0 : \sigma^2 = 0.004 \text{ versus } H_1 : \sigma^2 > 0.004.$$

Step 2: Test Statistic — The test statistic chosen is S^2 because $E[S^2] = \sigma^2$. The value of this test statistic is $s^2 = 0.0053$. The standardized test statistic under the assumption that H_0 is true and its distribution are $\frac{(n-1)s^2}{\sigma^2} \sim \chi_{n-1}^2$.

Step 3: Rejection Region Calculations — Because the standardized test statistic is distributed χ_{19}^2 and H_1 is an upper one-sided hypothesis, the rejection region is $\chi_{0.95;19}^2 > \chi_{0.95;19}^2 = 30.1435$. The value of the standardized test statistic is $\chi_{\text{obs}}^2 = \frac{(20-1)(0.0053)}{0.004} = 25.2637$.

Step 4: Statistical Conclusion — The ϕ -value is $\mathbb{P}(\chi_{19}^2 \geq 25.2637) = 0.152$.

- From the rejection region, fail to reject H_0 because $\chi_{\text{obs}}^2 = 25.2637$ is less than 30.1435.
- From the ϕ -value, fail to reject H_0 because the ϕ -value = 0.152 is greater than 0.05.

Fail to reject H_0 .

Step 5: **English Conclusion** — There is insufficient evidence to suggest the variance for washers manufactured this month increased from the variance of washers manufactured last month.

To compute the critical value, the standardized test statistic's value, and the corresponding p -value with R, use the variable `diameter`, which contains the variance of last month's washers' diameters.

```
> cv <- qchisq(0.95,19)           # Critical Value
> s2 <- var(WASHER$diameter)
> n <- sum(!is.na(WASHER$diameter))
> n

[1] 20

> Chi20bs <- (n - 1)*s2 / 0.004      # Standardized Test Statistic's Value
> Chi20bs

[1] 25.26375

> pvalue <- pchisq(Chi20bs, n-1, lower = FALSE)
> c(CriticalValue = cv, Chi20bs = Chi20bs, pvalue = pvalue)

CriticalValue      Chi20bs      pvalue
 30.1435272    25.2637500    0.1520425
```



9.8.2 Test for Equality of Variances When Sampling from Independent Normal Distributions

This section addresses the issue of comparing the variances of two distributions. This problem is encountered when comparing instrument precisions or uniformity of products. Another application is to check the assumption of equal variances for the pooled t -test; however, as mentioned earlier, the pooled t -test should only be used when equality of variances is beyond doubt. Consequently, this text will not place as large an emphasis on that use of the test as some other texts do. Provided X_1, X_2, \dots, X_{n_X} and Y_1, Y_2, \dots, Y_{n_Y} are independent random samples from $N(\mu_X, \sigma_X^2)$ and $N(\mu_Y, \sigma_Y^2)$ distributions, respectively, the random variable

$$F = \frac{S_X^2/\sigma_X^2}{S_Y^2/\sigma_Y^2} \sim F_{n_X-1, n_Y-1}.$$

The null hypothesis for testing the equality of two population variances is $H_0 : \sigma_X^2 = \sigma_Y^2$, which is equivalent to testing $H_0 : \sigma_X^2/\sigma_Y^2 = 1$. The value for the test statistic when the variances are assumed equal is

$$f_{\text{obs}} = \frac{s_X^2}{s_Y^2}.$$

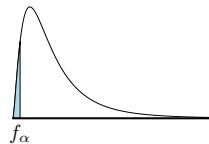
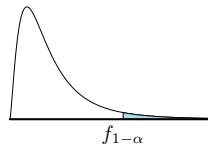
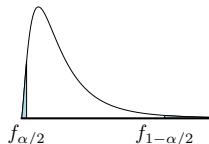
The three alternative hypotheses and the rejection regions for H_0 are in Table 9.14 on the facing page.

Example 9.15 \triangleright **F-Test: Breathalyzers** \lhd In an effort to reduce the number of drunk drivers associated with fraternal organizations, the fraternity council wants to distribute portable breathalyzers to all the fraternities on campus. There are two companies that are

Table 9.14: Summary for test for equality of variances when sampling from independent normal distributions

$$\text{Null Hypothesis} — H_0 : \sigma_X^2 = \sigma_Y^2 \quad \text{Standardized Test Statistic's Value} — f_{\text{obs}} = \frac{s_X^2}{s_Y^2}$$

Note that all f values in this table have degrees of freedom $n_X - 1, n_Y - 1$.

Alternative Hypothesis	$H_1 : \sigma_X^2 < \sigma_Y^2$	$H_1 : \sigma_X^2 > \sigma_Y^2$	$H_1 : \sigma_X^2 \neq \sigma_Y^2$
Rejection Region	$f_{\text{obs}} < f_\alpha$	$f_{\text{obs}} > f_{1-\alpha}$	$f_{\text{obs}} < f_{\alpha/2}$ or $f_{\text{obs}} > f_{1-\alpha/2}$
Graphical Representation of Rejection Region			

bidding to provide these breathalyzers. The fraternity council has decided to purchase all of its breathalyzers from the company whose breathalyzers have the smaller variance. Based on advertisement, the fraternity council suspects breathalyzer X to have a smaller variance than breathalyzers from company Y. Each company provides ten portable breathalyzers to the fraternity council. Two 180-pound volunteers each consumed a 12-ounce beer every 15 minutes for one hour. One hour after the fourth beer was consumed, each volunteer's blood alcohol was measured with a different breathalyzer from the same company. The numbers recorded in data frame **BAC** are the sorted blood alcohol content values reported with breathalyzers from company X and company Y. Test the appropriate hypotheses using a 5% significance level. (Note: The units of measurement for blood alcohol content, BAC, are grams of alcohol per liter of blood, g/L.)

Company X: 0.08 0.09 0.09 0.10 0.10 0.10 0.10 0.11 0.11 0.12
 Company Y: 0.00 0.03 0.04 0.04 0.05 0.05 0.06 0.07 0.08 0.08

Solution: Prior to using a test that is very sensitive to departures in normality, the function `eda()` is applied to the ten blood alcohol readings using breathalyzers from company X and the ten blood alcohol readings recorded using breathalyzers from company Y. Based on the results displayed in Figure 9.13 on the next page, it seems reasonable to assume the blood alcohol values breathalyzers report from both companies X and Y follow normal distributions. Although the blood alcohol values reported with company Y analyzers are slightly skewed to the left, one must remember that only ten values were used in the construction of the graphs and that graphs constructed with small sample sizes, even when sampling from normal distributions, will often appear skewed. When working with small sample sizes, one may want to test formally the hypothesis of normality with a function like `shapiro.test()`, which is explained more fully in Section 10.7.3 of Chapter 10. The Shapiro-Wilk Normality Test when applied to the blood alcohol values from Company Y also indicates normality is plausible based on the relatively large p -value (0.5489); therefore, proceed with the five-step procedure.

```
> shapiro.test(BAC$Y)
```

Shapiro-Wilk normality test

```
data: BAC$Y
W = 0.93963, p-value = 0.5489
```

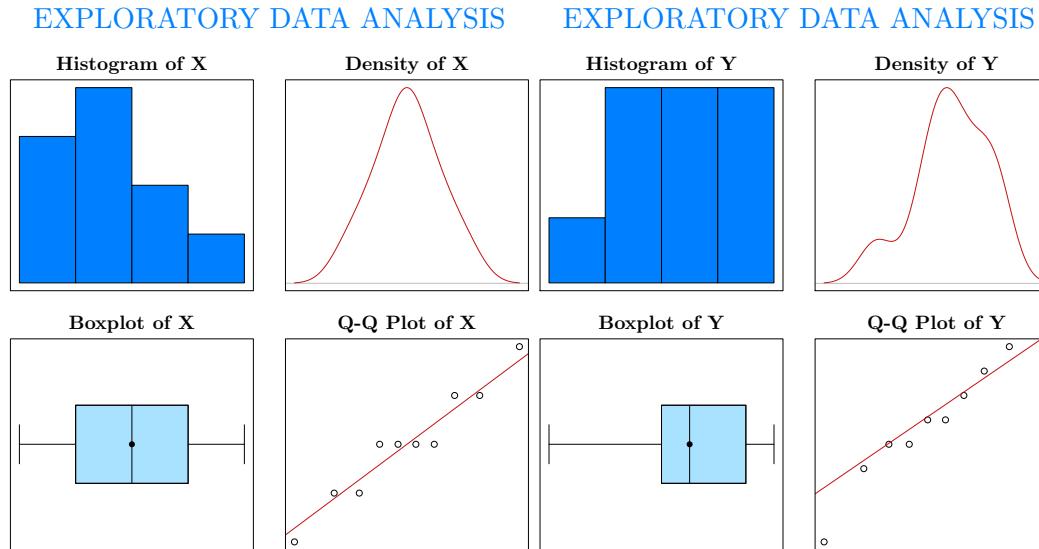


FIGURE 9.13: Exploratory data analysis for the blood alcohol values using the breathalyzers from company X and company Y on two volunteers after drinking four beers

Step 1: **Hypotheses** — The null and alternative hypotheses to test whether the variability in blood alcohol values using company X's breathalyzers is less than the variability in blood alcohol values using company Y's breathalyzers are

$$H_0 : \sigma_X^2 = \sigma_Y^2 \text{ versus } H_1 : \sigma_X^2 < \sigma_Y^2.$$

Step 2: **Test Statistic** — The test statistics chosen are S_X^2 and S_Y^2 since $E[S_X^2] = \sigma_X^2$ and $E[S_Y^2] = \sigma_Y^2$. The values of these test statistics are $s_X^2 = 0.0001333333$ and $s_Y^2 = 0.0006$. The standardized test statistic under the assumption that H_0 is true and its distribution are $S_X^2/S_Y^2 \sim F_{10-1,10-1}$.

Step 3: **Rejection Region Calculations** — Because the standardized test statistic is distributed $F_{9,9}$ and H_1 is a lower one-sided hypothesis, the rejection region is $f_{\text{obs}} < F_{0.05;9,9} = 0.3146$. The value of the standardized test statistic is $f_{\text{obs}} = (0.0001333333)/(0.0006) = 0.2222$.

Step 4: **Statistical Conclusion** — The α -value is $\mathbb{P}(F_{9,9} \leq 0.2222) = 0.0176$.

- From the rejection region, reject H_0 because $f_{\text{obs}} = 0.2222$ is less than 0.3146.

II. From the p -value, reject H_0 because the p -value = 0.0176 is less than 0.05.

Reject H_0 .

Step 5: English Conclusion — The evidence suggests the variability of blood alcohol values using breathalyzers from company X is less than the variance for blood alcohol values using breathalyzers from company Y.

To compute the critical value, the standardized test statistic's value, and the corresponding p -value with R, enter

```
> cv <- qf(0.05, 9, 9)      # Critical Value
> X <- BAC$X
> Y <- BAC$Y
> VX <- var(X)
> VY <- var(Y)
> RV <- VX / VY           # Standardized Test Statistic's Value
> pvalue <- pf(RV, 9, 9)    # P-Value
> c(CriticalValue = cv, VarianceX = VX, VarianceY = VY,
+   TestStat = RV, Pvalue = pvalue)

CriticalValue      VarianceX      VarianceY      TestStat      Pvalue
0.3145749062  0.0001333333  0.0006000000  0.2222222222  0.0176434906
```

To test the appropriate hypothesis using the R function `var.test()`, key in

```
> var.test(X, Y, alternative = "less")

F test to compare two variances

data: X and Y
F = 0.22222, num df = 9, denom df = 9, p-value = 0.01764
alternative hypothesis: true ratio of variances is less than 1
95 percent confidence interval:
0.0000000 0.7064207
sample estimates:
ratio of variances
0.2222222

> # OR using a formula
> DF <- stack(data.frame(X, Y))
> var.test(values ~ ind, data = DF, alternative = "less")

F test to compare two variances

data: values by ind
F = 0.22222, num df = 9, denom df = 9, p-value = 0.01764
alternative hypothesis: true ratio of variances is less than 1
95 percent confidence interval:
0.0000000 0.7064207
sample estimates:
ratio of variances
0.2222222
```

```
> rm(X, Y) # Clean up
```

The confidence interval is calculated with (8.31) on page 487, modified for a one-sided confidence interval. Note that the interval agrees with our step 5 conclusion as it contains values that are exclusively less than 1, implying $\frac{\sigma_X^2}{\sigma_Y^2} < 1$. ■

9.9 Hypothesis Tests for Population Proportions

Tests of hypotheses concerning proportions are encountered in many areas. For example, manufacturing firms often test the percent of defective items in their products, and politicians often test that their support base will garner them a certain proportion of votes in an election. Many other examples exist. In this section, the problem of testing a hypothesis where the proportion of successes in a binomial experiment (π) is equal to some value (π_0) is considered.

9.9.1 Testing the Proportion of Successes in a Binomial Experiment (Exact Test)

Given a random variable $Y \sim \text{Bin}(n, \pi)$, an exact test for the null hypothesis $H_0 : \pi = \pi_0$, is constructed. The three possible alternative hypotheses and the φ -value formulas associated with each alternative hypothesis are given in Table 9.15. Note the use of the indicator function in the computation of the two-sided φ -value. Although it is possible to calculate rejection regions for this exact test, due to the discrete nature of Y , it is unlikely a critical region can be established whose size is exactly equal to the prescribed α ; therefore, φ -values are generally preferred when working with exact problems over the defining of rejection regions.

Table 9.15: Summary for testing the proportion of successes in a binomial experiment (number of successes is $Y \sim \text{Bin}(n, \pi)$)

Null Hypothesis — $H_0 : \pi = \pi_0$

Test Statistic's Value — y_{obs} = number of observed successes

Alternative Hypothesis	φ -Value Formula
$H_1 : \pi < \pi_0$	$\mathbb{P}(Y \leq y_{\text{obs}} H_0) = \sum_{i=0}^{y_{\text{obs}}} \binom{n}{i} \pi_0^i (1 - \pi_0)^{n-i}$
$H_1 : \pi > \pi_0$	$\mathbb{P}(Y \geq y_{\text{obs}} H_0) = \sum_{i=y_{\text{obs}}}^n \binom{n}{i} \pi_0^i (1 - \pi_0)^{n-i}$
$H_1 : \pi \neq \pi_0$	$\sum_{i=0}^n I(\mathbb{P}(Y = i) \leq \mathbb{P}(Y = y_{\text{obs}})) \cdot \binom{n}{i} \pi_0^i (1 - \pi_0)^{n-i}$

It is also possible to compute an exact confidence interval for π ; however, due to the discrete nature of Y , the actual confidence level (coverage probability) of the interval is often considerably higher than the stated confidence level. An exact $(1 - \alpha) \cdot 100\%$ confidence interval for π requires each one-sided φ -value in an exact binomial test to exceed $\alpha/2$. Except when $y = 0$ and the lower bound is zero, and when $y = n$ and the upper bound is 1, the lower and upper endpoints for an exact $(1 - \alpha) \cdot 100\%$ confidence interval for π are the solutions in π_0 to the equations

$$\sum_{k=0}^y \binom{n}{k} \pi_0^k (1 - \pi_0)^{n-k} \geq \frac{\alpha}{2} \quad \text{and} \quad \sum_{k=y}^n \binom{n}{k} \pi_0^k (1 - \pi_0)^{n-k} \geq \frac{\alpha}{2}. \quad (9.7)$$

For values of $y = 1, 2, \dots, n - 1$, it can be shown that the solutions to (9.7) yield the lower and upper endpoint expressions for the confidence interval given in (9.8), which is known as the Clopper-Pearson interval.

$$CI_{1-\alpha}(\pi) =$$

$$\left[\left(1 + \frac{n - y_{\text{obs}} + 1}{y_{\text{obs}} F_{\alpha/2; 2y_{\text{obs}}, 2(n-y_{\text{obs}}+1)}} \right)^{-1}, \left(1 + \frac{n - y_{\text{obs}}}{(y_{\text{obs}} + 1) F_{1-\alpha/2; 2(y_{\text{obs}}+1), 2(n-y_{\text{obs}})}} \right)^{-1} \right] \quad (9.8)$$

The function `binom.test()` performs an exact binomial test and uses the criterion in Table 9.15 on the facing page, called the **likelihood** method, to compute its φ -values. The **likelihood** method the `binom.test()` uses to compute φ -values for two-sided alternatives differs from the general criterion of

$$2 \min [\mathbb{P}(Y \leq y_{\text{obs}} | H_0), \mathbb{P}(Y \geq y_{\text{obs}} | H_0)],$$

used up to now with two-sided alternatives and continuous distributions. It is of interest to note that a φ -value computed using the general criterion will agree with the `binom.test()` for two-sided tests, which uses the criterion in Table 9.15 on the preceding page when the distribution is symmetric, that is, when $\pi = 0.5$.

Example 9.16 ▷ **Exact Binomial Test: Graduates' Jobs** ◷ A recent report claimed that 20% of all college graduates find a job in their chosen field of study within one year of graduation. A random sample of 500 graduates found that 90 obtained work in their field within one year of graduation.

- (a) Is there statistical evidence to refute the claim at the $\alpha = 0.05$ level?
- (b) Compute an exact 95% confidence interval for the true proportion of college graduates who find work in their chosen field of study within one year of graduation.

Solution: The answers are as follows:

- (a) Use the five-step procedure.

Step 1: **Hypotheses** — The null and alternative hypotheses to test whether or not 20% of college graduates find work in their chosen field within one year of graduation are

$$H_0 : \pi = 0.20 \text{ versus } H_1 : \pi \neq 0.20.$$

Step 2: **Test Statistic** — The test statistic chosen is Y , where Y is the number of college graduates finding work in their chosen field within one year of graduation. Provided H_0 is true, $Y \sim \text{Bin}(n, \pi_0)$. The value of the test statistic is $y_{\text{obs}} = 90$.

Step 3: **Rejection Region Calculations** — Rejection is based on the ϕ -value, so none are required.

Step 4: **Statistical Conclusion** — The ϕ -value is 0.2881.

$$\begin{aligned}\phi\text{-value} &= \sum_{i=0}^n I(\mathbb{P}(Y = i) \leq \mathbb{P}(Y = y_{\text{obs}})) \cdot \binom{n}{i} \pi_0^i (1 - \pi_0)^{n-i} \\ &= \sum_{i=0}^{500} I(\mathbb{P}(Y = i) \leq \mathbb{P}(Y = 90)) \cdot \binom{500}{i} 0.20^i (1 - 0.20)^{500-i} \\ &= 0.2881 \quad \text{Computed with R}\end{aligned}$$

Thus, one fails to reject H_0 because 0.2881 is greater than 0.05. R Code 9.10 computes this ϕ -value and shows the output from using the function `binom.test()`.

Fail to reject H_0 .

Step 5: **English Conclusion** — There is not sufficient evidence to suggest the proportion of college graduates finding work in their chosen fields of study within one year of graduation is something other than 20%.

R Code 9.10

```
> probs <- dbinom(0:500, 500, 0.2)
> pvalue <- sum(probs[probs <= dbinom(90, 500, 0.2)])
> pvalue
[1] 0.2880566

> binom.test(x = 90, n = 500, p = 0.2)

Exact binomial test

data: 90 and 500
number of successes = 90, number of trials = 500, p-value = 0.2881
alternative hypothesis: true probability of success is not equal to 0.2
95 percent confidence interval:
 0.1473006 0.2165364
sample estimates:
probability of success
                  0.18
```

(b) An exact 95% confidence interval is constructed using (9.8):

$$\begin{aligned}
CI_{1-0.05}(\pi) &= \left[\left(1 + \frac{500 - 90 + 1}{(90)F_{0.05/2;2(90),2(500-90+1)}} \right)^{-1}, \right. \\
&\quad \left. \left(1 + \frac{500 - 90}{(90 + 1)F_{1-0.05/2;2(90+1),2(500-90)}} \right)^{-1} \right] \\
&= \left[\left(1 + \frac{411}{(90)(0.7889)} \right)^{-1}, \left(1 + \frac{410}{(91)(1.2452)} \right)^{-1} \right] \\
&= [0.1473, 0.2165].
\end{aligned}$$

One is 95% confident that the true proportion of college graduates finding work in their chosen fields of study within one year of graduation lies in [0.1473, 0.2165]. Note that this confidence interval, also calculated in R Code 9.10 on the preceding page, contains the hypothesized value of 0.20, corroborating the decision to fail to reject the null hypothesis.

9.9.2 Testing the Proportion of Successes in a Binomial Experiment (Normal Approximation)

In Section 9.9.1, an exact test and confidence interval were presented for the proportion of successes in a binomial experiment where the random variable $Y \sim \text{Bin}(n, \pi)$. Specifically, $Y = \sum_{i=1}^n X_i$, where $X_i \sim \text{Bernoulli}(\pi)$. A discussion of the properties of Y can be found in Section 6.5.3, starting on page 377. The numerical computations required by exact methods make a computer essentially indispensable, especially when presented with a large sample. Fortunately, for those who do not have access to a computer, approximations to exact distributions are possible for large samples. This is the focus of the current section.

Recall that the asymptotic properties of MLE estimators allow one to write

$$P = \frac{Y}{n} \sim N \left(\pi, \sqrt{\frac{\pi(1-\pi)}{n}} \right) \text{ as } n \rightarrow \infty.$$

Provided $n\pi$ and $n(1-\pi)$ are both greater than or equal to 10,

$$P \approx N \left(\pi, \sqrt{\frac{\pi(1-\pi)}{n}} \right) \tag{9.9}$$

provides a reasonable approximation to the sampling distribution of P . Using (9.9), the standardized test statistic under the assumption that $H_0 : \pi = \pi_0$ is true is

$$Z = \frac{P - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} \approx N(0, 1).$$

The formula to calculate the test statistic's observed value as well as the three possible alternative hypotheses and their rejection regions are described in Table 9.16 on the next page.

When $|p - \pi_0| > \frac{1}{2n}$, many statisticians advocate using a continuity correction when calculating confidence intervals and standardized test statistics' values. A continuity correction of $\pm \frac{1}{2n}$ is automatically applied when using the function `prop.test()`; however,

Table 9.16: Summary for testing the proportion of successes in a binomial experiment (normal approximation)

Null Hypothesis — $H_0 : \pi = \pi_0$

$$\begin{array}{ll} \text{Standardized Test} & z_{\text{obs}} = \frac{p - \pi_0}{\sqrt{\frac{\pi_0(1 - \pi_0)}{n}}} \\ \text{Statistic's Value} & \end{array}$$

Alternative Hypothesis	Rejection Region
$H_1 : \pi < \pi_0$	$z_{\text{obs}} < z_\alpha$
$H_1 : \pi > \pi_0$	$z_{\text{obs}} > z_{1-\alpha}$
$H_1 : \pi \neq \pi_0$	$ z_{\text{obs}} > z_{1-\alpha/2}$
When $ p - \pi_0 > \frac{1}{2n}$, use a correction factor as in Table 9.17.	

not all statisticians recommend the use of a continuity correction with this test, and using one does lead to a more conservative test. The continuity corrections that are applied, as well as the standardized test statistic calculations, can be found in Table 9.17.

Table 9.17: Correction factors when $|p - \pi_0| > \frac{1}{2n}$

Condition	Correction Factor	Standardized Test Statistic
$p - \pi_0 > 0$	$-\frac{1}{2n}$	$z_{\text{obs}} = \frac{p - \pi_0 - \frac{1}{2n}}{\sqrt{\frac{\pi_0(1 - \pi_0)}{n}}}$
$p - \pi_0 < 0$	$+\frac{1}{2n}$	$z_{\text{obs}} = \frac{p - \pi_0 + \frac{1}{2n}}{\sqrt{\frac{\pi_0(1 - \pi_0)}{n}}}$

Example 9.17 ▷ **Normal Approximation: Graduates' Jobs** ◷ A recent report claimed that 20% of all college graduates find a job in their chosen field of study. A random sample of 500 graduates found that 90 obtained work in their field. Using a normal approximation to the distribution of P ,

- (a) Is there statistical evidence to refute the claim at the $\alpha = 0.05$ level?
- (b) Compute a 95% confidence interval for the true proportion of college graduates that find work in their chosen field of study using (8.47) on page 497.

Solution: The answers are as follows:

- (a) Use the five-step procedure.

Step 1: **Hypotheses** — The null and alternative hypotheses to test whether or not 20% of college graduates find work in their chosen field are

$$H_0 : \pi = 0.20 \text{ versus } H_1 : \pi \neq 0.20.$$

Step 2: **Test Statistic** — The test statistic chosen is P , where P is the proportion of college graduates finding work in their chosen field. Provided H_0 is true,

$$P \sim N\left(\pi_0, \sqrt{\frac{\pi_0(1-\pi_0)}{n}}\right)$$

and the standardized test statistic is

$$Z = \frac{P - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} \sim N(0, 1).$$

Step 3: **Rejection Region Calculations** — Because the standardized test statistic has an approximate $N(0, 1)$ distribution and H_1 is a two-sided hypothesis, the rejection region is $|z_{\text{obs}}| > z_{0.975} = 1.96$. The value of the standardized test statistic is

Without Continuity Correction	With Continuity Correction
$z_{\text{obs}} = \frac{p - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}}$ $= \frac{\frac{90}{500} - 0.2}{\sqrt{\frac{(0.2)(1-0.2)}{500}}}$ $= -1.1180.$	$z_{\text{obs}} = \frac{p - \pi_0 + \frac{1}{2n}}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}}$ $= \frac{\frac{90}{500} - 0.2 + \frac{1}{1000}}{\sqrt{\frac{(0.2)(1-0.2)}{500}}}$ $= -1.0621.$
OR	

Step 4: **Statistical Conclusion** — The φ -value is $2 \cdot \mathbb{P}(Z \leq -1.118) = 0.2636$ or $2 \cdot \mathbb{P}(Z \leq -1.0621) = 0.2882$ for continuity corrections not used and used, respectively.

- I. From the rejection region, do not reject H_0 because $|z_{\text{obs}}| = |-1.1180|$ (no continuity correction) is not greater than 1.96, nor is $|z_{\text{obs}}| = |-1.0621|$ (continuity correction) greater than 1.96.
- II. From the φ -value, do not reject H_0 because the φ -value = 0.2636 (without continuity correction) or = 0.2882 (with continuity correction) is greater than 0.05.

Fail to reject H_0 .

R Code 9.11 on the following page computes the standardized test statistics with and without continuity corrections and their corresponding φ values.

Step 5: **English Conclusion** — There is not sufficient evidence to suggest the proportion of college graduates finding work in their chosen fields of study is something other than 20%.

R Code 9.11

```
> Y <- 90
> n <- 500
> p <- Y/n
> PI <- 0.2
> zobs <- (p - PI)/sqrt((PI * (1 - PI))/n) # No c.c.
> pval <- 2 * pnorm(zobs)
> zobsC <- (p - PI + 1/(2 * n))/sqrt((PI * (1 - PI))/n) # Yes c.c.
> pvalC <- 2 * pnorm(zobsC)
> c(ZNCC = zobs, PvalueNCC = pval, ZYCC = zobsC, PvalueYCC = pvalC)

ZNCC  PvalueNCC      ZYCC  PvalueYCC
-1.1180340  0.2635525 -1.0621323  0.2881756
```

(b) An approximate 95% confidence interval is [0.1488, 0.2161] without a continuity correction and [0.1479, 0.2171] with a continuity correction. One is 95% confident that the true proportion of college graduates finding work in their chosen fields of study lies in [0.1488, 0.2161]. Note that this confidence interval contains the hypothesized value of 0.20, corroborating the decision to fail to reject the null hypothesis.

The calculation of 95% confidence intervals as well as verifications of the calculated p -values from step 4 are computed with `prop.test()`, both without and with continuity corrections, in R Code 9.12.

R Code 9.12

```
> prop.test(x = 90, n = 500, p = 0.2, correct = FALSE)

1-sample proportions test without continuity correction

data: 90 out of 500, null probability 0.2
X-squared = 1.25, df = 1, p-value = 0.2636
alternative hypothesis: true p is not equal to 0.2
95 percent confidence interval:
 0.1488049 0.2160747
sample estimates:
 p
0.18

> prop.test(x = 90, n = 500, p = 0.2, correct = TRUE)

1-sample proportions test with continuity correction

data: 90 out of 500, null probability 0.2
X-squared = 1.1281, df = 1, p-value = 0.2882
alternative hypothesis: true p is not equal to 0.2
95 percent confidence interval:
 0.1478847 0.2171388
sample estimates:
 p
0.18
```



Note that the output for `prop.test()` does not give a z_{obs} -value; rather it reports a χ^2_{obs} -value, denoted **X-squared** in the R output, with one degree of freedom. Provided one uses the relationship $Z^2 = \chi^2_1$, it is possible to see that the z_{obs} -values reported in step 3 correspond to the X-squared values given in the output from using `prop.test()` without and with a continuity correction. That is, $-1.118034^2 = 1.25$ and $-1.062132^2 = 1.1281$.

Although the approximation procedures presented in this section lead to the same conclusion as the exact test in the previous section when applied to Example 9.16 on page 563, the approximation procedures of this section are only valid when applied to large samples. In contrast, the exact test presented in the last section will work for both large and small samples and is generally preferred over large sample approximation procedures when the user has access to a computer.

9.9.3 Testing Equality of Proportions with Fisher's Exact Test

One of the most common ways to present numerical data is in a table. When presented with a 2×2 table, where 2×2 refers to the dimensions of the number of internal cells, if the sample size is small, equality of proportions should be tested with **Fisher's exact test**. That is, $H_0 : \pi_X = \pi_Y$, where $X \sim \text{Bin}(m, \pi_X)$ and $Y \sim \text{Bin}(n, \pi_Y)$ are the numbers of successes observed from two independent binomial random variables. To compute Fisher's exact test, let $N = m + n$ be the total sample size and $k = x + y$ be the total number of observed successes. Table 9.18 shows the general form of such a table.

Table 9.18: General form of a 2×2 table

	Success	Failure	Total
X Sample	x	$m - x$	m
Y Sample	y	$n - y$	n
	k	$N - k$	N

Fisher's exact test uses the number of successes from the X sample as its test statistic, namely X . The observed value of X is denoted x . In performing the exact test, the total number of successes is considered fixed. That is, $x + y = k$ is a fixed quantity in the derivation of the test. Specifically,

$$\mathbb{P}(X = i | X + Y = k) = \frac{\binom{m}{i} \binom{n}{k-i}}{\binom{N}{k}}, \text{ where } i = \max\{0, k - n\}, \dots, \min\{m, k\}. \quad (9.10)$$

Note that (9.10) is a hypergeometric distribution, $\text{Hyper}(m, n, k)$, where the parameters are m , n , and k . The three possible alternative hypotheses and the respective p -value calculation formulas are presented in Table 9.19 on the following page. Since the distribution of the statistic is obtained by constructing all possible 2×2 tables, the test has historically been used with small samples. With the advent of inexpensive computing power, it is now feasible to use Fisher's exact test on relatively large samples with fixed marginals.

A statistic that measures how associated X and Y are is the **odds ratio**. It is frequently used in biomedical and sociological studies to measure the association between two variables.

Table 9.19: Summary for testing the proportion of successes with Fisher's exact test

Null Hypothesis — $H_0 : \pi_X = \pi_Y$

Test Statistic's Value — x = number of observed successes from X sample

Alternative Hypothesis	φ -Value Formula
$H_1 : \pi_X < \pi_Y$	$\mathbb{P}(X \leq x H_0) = \sum_{i=\max\{0, k-n\}}^x \frac{\binom{m}{i} \binom{n}{k-i}}{\binom{N}{k}}$
$H_1 : \pi_X > \pi_Y$	$\mathbb{P}(X \geq x H_0) = \sum_{i=x}^{\min\{m, k\}} \frac{\binom{m}{i} \binom{n}{k-i}}{\binom{N}{k}}$
$H_1 : \pi_X \neq \pi_Y$	$\sum_{i=\max\{0, k-n\}}^{\min\{m, k\}} I(\mathbb{P}(X = i) \leq \mathbb{P}(X = x)) \frac{\binom{m}{i} \binom{n}{k-i}}{\binom{N}{k}}$

The odds ratio is defined as

$$\theta = \frac{\pi_X / (1 - \pi_X)}{\pi_Y / (1 - \pi_Y)}. \quad (9.11)$$

An odds ratio other than 1 indicates there is a relationship between X and Y , while an odds ratio of exactly 1 indicates that X and Y are independent. If the odds ratio is larger than 1, π_X is greater than π_Y ; and if smaller, π_X is less than π_Y .

The function `fisher.test()` computes a $(1 - \alpha) \cdot 100\%$ confidence interval for the odds ratio using maximum likelihood techniques with the non-central hypergeometric distribution. An explanation of the procedure is beyond the scope of this text. When the $(1 - \alpha) \cdot 100\%$ confidence interval for the odds ratio's upper bound is less than 1, one can be $(1 - \alpha) \cdot 100\%$ confident that π_X is less than π_Y . Likewise, when the lower bound of a $(1 - \alpha) \cdot 100\%$ confidence interval for the odds ratio is greater than 1, one can be $(1 - \alpha) \cdot 100\%$ confident that π_X is greater than π_Y .

Example 9.18 ▷ **Fisher's Exact Test: Delinquents in Glasses** ◷ A researcher wants to discover if the proportion of non-delinquent juveniles who wear glasses is different from that of juvenile delinquents. He collects the information found in Table 9.20 on the next page from juveniles who failed a vision test. Test whether the proportion of non-delinquents who wear glasses is different from the proportion of juvenile delinquents who wear glasses at an α level of 0.05.

Solution: To solve this problem, use Fisher's exact test and the five-step procedure.

Step 1: Hypotheses — The null and alternative hypotheses to test whether the proportion of non-delinquents who wear glasses is different from the proportion of juvenile delinquents who wear glasses are

$$H_0 : \pi_X = \pi_Y \text{ versus } H_1 : \pi_X \neq \pi_Y.$$

In this case, the random variable X will represent the number of juvenile delinquents who wear glasses, and the random variable Y will represent the number of non-delinquents who wear glasses.

Table 9.20: Juveniles who failed a vision test classified by delinquency and glasses-wearing (Weindling et al., 1986)

	Wear Glasses	Do Not Wear Glasses	Totals
Juvenile Delinquents	1	8	9
Non-delinquents	5	2	7
Totals	6	10	16

Step 2: **Test Statistic** — The test statistic chosen is X , where X is the number of juvenile delinquents who wear glasses. The observed value of the test statistic is $x = 1$. Provided H_0 is true, and conditioning on the fact that $X + Y = k$, $X \sim \text{Hyper}(m, n, k)$.

Step 3: **Rejection Region Calculations** — Rejection is based on the ϕ -value, so none are required.

Step 4: **Statistical Conclusion** — To compute the ϕ -value, compute

$$\sum_{i=\max\{0,k-n\}}^{\min\{m,k\}} I(\mathbb{P}(X=i) \leq \mathbb{P}(X=x)) \frac{\binom{m}{i} \binom{n}{k-i}}{\binom{N}{k}} =$$

$$\sum_{i=\max\{0,6-7\}}^{\min\{9,6\}} I(\mathbb{P}(X=i) \leq \mathbb{P}(X=1)) \frac{\binom{9}{i} \binom{7}{6-i}}{\binom{16}{6}} = 0.035.$$

For such a small sample, the seven possible 2×2 tables that can be constructed where $k = 6$, $m = 9$, and $n = 7$ and their respective ϕ -values are shown in Table 9.21. Since the ϕ -value is 0.035, one rejects H_0 because 0.035 is less than 0.05.

Reject H_0 .

Step 5: **English Conclusion** — There is sufficient evidence to suggest the proportion of non-delinquents who wear glasses is different from the proportion of juvenile delinquents who wear glasses.

Table 9.21: Seven possible 2×2 tables that can be constructed where $k = 6$, $m = 9$, and $n = 7$, with their associated probabilities

Table	Probability	Table	Probability	Table	Probability	Table	Probability
0 9 6 1	0.00087	1 8 5 2	0.0236	2 7 4 3	0.15734	3 6 3 4	0.36713
4 5 2 5	0.33042	5 4 1 6	0.11014	6 3 0 7	0.0104		

R Code 9.13 on the next page computes the probabilities for the seven 2×2 tables given in Table 9.21.

R Code 9.13

```
> p <- dhyper(0:6, 9, 7, 6) # Probabilities for the 7 possible 2X2 tables
> p
[1] 0.0008741259 0.0236013986 0.1573426573 0.3671328671 0.3304195804
[6] 0.1101398601 0.0104895105

> pobs <- dhyper(1, 9, 7, 6)
> pval <- sum(p[p <= pobs])
> pval
[1] 0.03496503
```

R Code 9.14 enters the information from Table 9.20 on the preceding page and performs Fisher's exact test.

R Code 9.14

```
> JV <- matrix(data = c(1, 5, 8, 2), nrow = 2)
> dimnames(JV) <- list(Youth = c("Delinquent", "Non-delinquent"),
+     Glasses = c("Yes", "No"))
> JV

          Glasses
Youth      Yes No
  Delinquent    1   8
  Non-delinquent 5   2

> fisher.test(JV)
```

Fisher's Exact Test for Count Data

```
data: JV
p-value = 0.03497
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
0.0009525702 0.9912282442
sample estimates:
odds ratio
0.06464255
```

Note that values of θ farther from 1.0 in a given direction represent stronger levels of association ($0 < \theta < \infty$). In this case, $\theta = 0.06$ means that the odds ratio for non-delinquents wearing glasses is $\frac{1}{0.06} = 15.5$ times the odds ratio for delinquents wearing glasses. This is a very strong association. ■

Example 9.19 ▷ Fisher's Exact Test of $\pi_X = \pi_Y$: Heart Attacks ◁ Physicians want to know if taking aspirin will help them avoid heart attacks. A group collects the information found in Table 9.22 on the next page. Help them test to see if taking aspirin is beneficial in the prevention of heart attacks at an α level of 0.05.

Solution: To solve this problem, use Fisher's exact test and the five-step procedure.

Table 9.22: Observed heart attacks for those physicians taking aspirin and a placebo (Hennekens, 1988)

	Heart Attack	No Heart Attack	Totals	
Aspirin	104	10,933	11,037	$= m$
Placebo	189	10,845	11,034	$= n$
Totals	$293 = k$	21,778	22,071	$= N$

Step 1: **Hypotheses** — The null and alternative hypotheses to test whether the proportion of physicians who suffer heart attacks while taking aspirin is less than the proportion of physicians who suffer heart attacks while taking a placebo are

$$H_0 : \pi_X = \pi_Y \text{ versus } H_1 : \pi_X < \pi_Y.$$

In this case, let the random variable X represent the number of physicians who had a heart attack while taking aspirin, and let the random variable Y represent the number of physicians who had a heart attack while taking a placebo.

Step 2: **Test Statistic** — The test statistic chosen is X , where X is the number of physicians who had a heart attack while taking aspirin. Provided H_0 is true, and conditioning on the fact that $X + Y = k$, $X \sim \text{Hyper}(m, n, k)$. The observed value of the test statistic is $x = 104$.

Step 3: **Rejection Region Calculations** — Rejection is based on the ϕ -value, so none are required.

Step 4: **Statistical Conclusion** — To calculate the ϕ -value, compute

$$\mathbb{P}(X \leq x_{\text{obs}} = 104) = \sum_{i=0}^{104} \frac{\binom{104+10933}{i} \binom{189+10845}{293-i}}{\binom{22071}{293}} = 0.$$

Note that the limits on the sum are typically the $\max\{0, x - n\}$ and x . In this case $x - n = 104 - 11,034 = -10,930$, so the lower limit of the sum will be zero. This calculation should be done with a computer and is done in R Code 9.15. Because the ϕ -value is 0, which is less than 0.05, reject H_0 .

Reject H_0 .

Step 5: **English Conclusion** — There is sufficient evidence to suggest taking aspirin is beneficial in the prevention of heart attacks for physicians at an α level of 0.05.

R Code 9.15 enters the data from Table 9.22 into R, computes the ϕ -value for Step 4, and applies the function `fisher.test()` to the HA matrix.

R Code 9.15

```
> HA <- matrix(c(104, 189, 10933, 10845), nrow = 2) # Table
> dimnames(HA) <- list(Treatment=c("Aspirin", "Placebo"),
+                         Outcome=c("Heart attack", "No heart attack"))
> HA
```

```

      Outcome
Treatment Heart attack No heart attack
Aspirin        104        10933
Placebo        189        10845

> pval <- phyper(104, 104 + 10933, 189 + 10845, 104 + 189) # x, m, n, k
> pval

[1] 3.252711e-07

> fisher.test(HA, alternative = "less")

Fisher's Exact Test for Count Data

data: HA
p-value = 3.253e-07
alternative hypothesis: true odds ratio is less than 1
95 percent confidence interval:
0.0000000 0.6721508
sample estimates:
odds ratio
0.5458537

```

Note that the odds ratio for physicians having a heart attack taking a placebo is $\frac{1}{0.546} = 1.83$ times the odds ratio for physicians who take aspirin. ■

9.9.4 Large Sample Approximation for Testing the Difference of Two Proportions

In Section 9.9.3, Fisher's exact test was presented for testing the equality of proportions for two independent random samples taken from Bernoulli populations of sizes m and n , respectively. Once the sample sizes become large for Fisher's exact test, even computers begin to have difficulties. Thus, there exists a procedure for approximating the distribution of $P_X - P_Y$ that will lead to a test that does not have nearly the computational requirements of Fisher's exact test. In Section 8.4.2, it was argued that

$$P_X - P_Y \underset{\text{approx}}{\sim} N\left(\pi_X - \pi_Y, \sqrt{\frac{\pi_X(1-\pi_X)}{m} + \frac{\pi_Y(1-\pi_Y)}{n}}\right) \quad (9.12)$$

when taking independent random samples of sizes m and n from $Bernoulli(\pi_X)$ and $Bernoulli(\pi_Y)$ populations, respectively. Using (9.12),

$$Z = \frac{(P_X - P_Y) - (\pi_X - \pi_Y)}{\sqrt{\frac{\pi_X(1-\pi_X)}{m} + \frac{\pi_Y(1-\pi_Y)}{n}}} \underset{\text{approx}}{\sim} N(0, 1). \quad (9.13)$$

Unfortunately, the values of π_X and π_Y are unknown. In Section 8.4.2, π_X and π_Y were replaced with their maximum likelihood estimators, $\hat{\pi}_X = P_X$ and $\hat{\pi}_Y = P_Y$, respectively, to create the asymptotic confidence interval in (8.52). To create a standardized test statistic

with an approximate normal distribution, the same approach will be used. That is,

$$Z = \frac{(P_X - P_Y) - \delta_0}{\sqrt{\frac{P_X(1-P_X)}{m} + \frac{P_Y(1-P_Y)}{n}}} \stackrel{d}{\sim} N(0, 1) \quad (9.14)$$

can be used to test the null hypothesis $H_0 : \pi_X - \pi_Y = \delta_0$. It is often the case that δ_0 is zero. In this case, it is standard practice to create a pooled estimate of the population proportions such that $\pi_X = \pi_Y = \pi$. The pooled estimate of π , denoted P , is

$$P = \frac{X + Y}{m + n} \quad (9.15)$$

which is simply an estimate of the total proportion of successes. When this estimate is used, the standardized test statistic becomes

$$Z = \frac{(P_X - P_Y)}{\sqrt{P(1-P)\left(\frac{1}{m} + \frac{1}{n}\right)}} \stackrel{d}{\sim} N(0, 1). \quad (9.16)$$

There are advantages and disadvantages to both (9.14) and (9.16) as test statistics. The function `prop.test()` bases its confidence interval construction on (9.14) and uses (9.16) for testing hypotheses. Table 9.23 uses the standardized test statistic in (9.16) and provides the rejection regions for the three possible alternative hypotheses.

Table 9.23: Summary for testing the differences of the proportions of successes in two binomial experiments (large sample approximation)

Null Hypothesis — $H_0 : \pi_X = \pi_Y$

$$\text{Standardized Test Statistic's Value} \quad z_{\text{obs}} = \frac{p_X - p_Y}{\sqrt{p(1-p)\left(\frac{1}{m} + \frac{1}{n}\right)}}$$

Alternative Hypothesis	Rejection Region
$H_1 : \pi_X < \pi_Y$	$z_{\text{obs}} < z_\alpha$
$H_1 : \pi_X > \pi_Y$	$z_{\text{obs}} > z_{1-\alpha}$
$H_1 : \pi_X \neq \pi_Y$	$ z_{\text{obs}} > z_{1-\alpha/2}$
When $ p_X - p_Y > \frac{1}{2} \left(\frac{1}{m} + \frac{1}{n} \right)$, use a correction factor as in Table 9.24 on the following page.	

When $|p_X - p_Y| > \frac{1}{2} \left(\frac{1}{m} + \frac{1}{n} \right)$, some statisticians advocate using a continuity correction when calculating confidence intervals and standardized test statistics' values. A continuity correction of $\pm \frac{1}{2} \left(\frac{1}{m} + \frac{1}{n} \right)$ is automatically applied when using the function `prop.test()` on two samples. The continuity corrections that are applied, as well as the standardized test statistic calculations, can be found in Table 9.24 on the next page. When applying the continuity correction to (8.52) on page 506, recall that the continuity correction is subtracted and added to the lower and upper confidence limits, respectively.

Table 9.24: Correction factors when $|p_X - p_Y| > \frac{1}{2} \left(\frac{1}{m} + \frac{1}{n} \right)$

Condition	Correction Factor	Standardized Test Statistic
$p_X - p_Y > 0$	$-\frac{1}{2} \left(\frac{1}{m} + \frac{1}{n} \right)$	$z_{\text{obs}} = \frac{p_X - p_Y - \frac{1}{2} \left(\frac{1}{m} + \frac{1}{n} \right)}{\sqrt{p(1-p) \left(\frac{1}{m} + \frac{1}{n} \right)}}$
$p_X - p_Y < 0$	$+\frac{1}{2} \left(\frac{1}{m} + \frac{1}{n} \right)$	$z_{\text{obs}} = \frac{p_X - p_Y + \frac{1}{2} \left(\frac{1}{m} + \frac{1}{n} \right)}{\sqrt{p(1-p) \left(\frac{1}{m} + \frac{1}{n} \right)}}$

Example 9.20 \triangleright **Large Sample Test of $\pi_X = \pi_Y$: Heart Attacks** \triangleleft Use the data from Table 9.22 on page 573 to test whether physicians who take aspirin are less likely to suffer heart attacks than those who take a placebo at an α level of 0.05. Base the test on the large sample approximation procedures found in Table 9.23 on the preceding page.

Solution: To solve this problem, use the five-step procedure.

Step 1: **Hypotheses** — The null and alternative hypotheses to test whether the proportion of physicians who suffer heart attacks while taking aspirin is less than the proportion of physicians who suffer heart attacks while taking a placebo are

$$H_0 : \pi_X = \pi_Y \text{ versus } H_1 : \pi_X < \pi_Y.$$

In this case, let the random variable P_X represent the proportion of physicians who had a heart attack while taking aspirin, and let the random variable P_Y represent the number of physicians who had a heart attack while taking a placebo.

Step 2: **Test Statistic** — The test statistic chosen is $P_X - P_Y$ since $E[P_X - P_Y] = \pi_X - \pi_Y$. The standardized test statistic under the assumption that H_0 is true is

$$Z = \frac{P_X - P_Y}{\sqrt{P(1-P) \left(\frac{1}{m} + \frac{1}{n} \right)}}.$$

Step 3: **Rejection Region Calculations** — Because the standardized test statistic has an approximate $N(0, 1)$ distribution and H_1 is a lower one-sided hypothesis, the rejection region is $z_{\text{obs}} < z_{0.05} = -1.6449$. The pooled estimate of π is $p = \frac{x+y}{m+n} = \frac{293}{22071}$. The value of the standardized test statistic is

Without Continuity Correction

$$\begin{aligned} z_{\text{obs}} &= \frac{p_X - p_Y}{\sqrt{p(1-p) \left(\frac{1}{m} + \frac{1}{n}\right)}} \\ &= \frac{\frac{104}{11037} - \frac{189}{11034}}{\sqrt{\frac{293}{22071} \left(1 - \frac{293}{22071}\right) \left(\frac{1}{11037} + \frac{1}{11034}\right)}} \\ &= -5.0014 \end{aligned}$$

OR

With Continuity Correction

$$\begin{aligned} z_{\text{obs}} &= \frac{p_X - p_Y + \frac{1}{2} \left(\frac{1}{m} + \frac{1}{n}\right)}{\sqrt{p(1-p) \left(\frac{1}{m} + \frac{1}{n}\right)}} \\ &= \frac{\frac{104}{11037} - \frac{189}{11034} + \frac{1}{2} \left(\frac{1}{11037} + \frac{1}{11034}\right)}{\sqrt{\frac{293}{22071} \left(1 - \frac{293}{22071}\right) \left(\frac{1}{11037} + \frac{1}{11034}\right)}} \\ &= -4.9426. \end{aligned}$$

Step 4: **Statistical Conclusion** — The φ -value is $\mathbb{P}(Z \leq z_{\text{obs}})$ and is approximately 0 for both cases. This is less than 0.05, so reject H_0 .

Reject H_0 .

Step 5: **English Conclusion** — There is sufficient evidence to suggest taking aspirin is beneficial in the prevention of heart attacks for physicians at an α level of 0.05.

R Code 9.16 performs the required test without and with a continuity correction.

R Code 9.16

```
> prop.test(x = c(104, 189), n = c(11037, 11034), correct = FALSE,
+   alternative = "less")

2-sample test for equality of proportions without continuity
correction

data: c(104, 189) out of c(11037, 11034)
X-squared = 25.014, df = 1, p-value = 2.846e-07
alternative hypothesis: less
95 percent confidence interval:
-1.000000000 -0.005173009
sample estimates:
prop 1     prop 2
0.00942285 0.01712887
```

```

> zobsNCC <- prop.test(x = c(104, 189), n = c(11037, 11034),
+   correct = FALSE, alternative = "less")$stat^0.5
> names(zobsNCC) <- NULL
> zobsNCC # Z obs no continuity correction

[1] 5.001388

> prop.test(x = c(104, 189), n = c(11037, 11034), correct = TRUE,
+   alternative = "less")

2-sample test for equality of proportions with continuity
correction

data: c(104, 189) out of c(11037, 11034)
X-squared = 24.429, df = 1, p-value = 3.855e-07
alternative hypothesis: less
95 percent confidence interval:
-1.000000000 -0.005082393
sample estimates:
prop 1     prop 2
0.00942285 0.01712887

> zobsYCC <- prop.test(x = c(104, 189), n = c(11037, 11034),
+   correct = TRUE, alternative = "less")$stat^0.5
> names(zobsYCC) <- NULL
> zobsYCC

[1] 4.942576

```

Notice that if z_{obs} is squared, it will be equal to the values of **X-squared** output from the `prop.test()` function. ■

9.10 Problems

1. Define α and β for a test of hypothesis. What is the quantity $1 - \beta$ called?
2. How can β be made small in a given hypothesis test with fixed α ?
3. Using a 5% significance level, what is the power of the test $H_0 : \mu = 100$ versus $H_1 : \mu \neq 100$ if a sample of size 36 is taken from a $N(120, 50)$?
4. An experiment was conducted to investigate how the resistance of rubber to abrasion is affected by the hardness of the rubber and its tensile strength. The data come from Hand et al. (1994, Data Set #6, Abrasion Loss) and are stored in the data frame **Rubber** of the MASS package. The abrasion loss is measured in grams/hour; the hardness, in degrees shore; and the tensile strength, in kg/cm². Use the five-step procedure to test whether $H_0 : \mu = 170$ versus $H_1 : \mu < 170$ for abrasion loss (**loss**).
5. An apartment appraiser in Vitoria, Spain, feels confident in his appraisals of 90m² or larger pisos (apartments) provided his variability is less than $60,000^2\text{€}^2$. Due to constant movement in the housing market, the regional housing authority suspects the appraiser's variability may be greater than $60,000^2\text{€}^2$. Is there evidence to support the suspicions of the regional housing authority? Test the appropriate hypothesis at the 5% significance level using the five-step procedure. The appraised values of apartments in Vitoria are stored in the variable **totalprice** of the **VIT2005** data frame.
6. The Hubble Space Telescope was put into orbit on April 25, 1990. Unfortunately, on June 25, 1990, a spherical aberration was discovered in Hubble's primary mirror. To correct this, astronauts had to work in space. To prepare for the mission, two teams of astronauts practiced making repairs under simulated space conditions. Each team of astronauts went through 15 identical scenarios. The times to complete each scenario were recorded in days. Is one team better than the other? If not, can both teams complete the mission in less than 3 days? Use a 5% significance level for all tests. The data are stored in the data frame **HUBBLE**.
7. The research and development department of an appliance company suspects the energy consumption required of their 18-cubic-foot refrigerator can be reduced by a slight modification to the current motor. Sixty 18-cubic-foot refrigerators were randomly selected from the company's warehouse. The first 30 had their motors modified while the last 30 were left intact. The energy consumption (kilowatts) for a 24-hour period for each refrigerator was recorded and stored in the data frame **REFRIGERATOR**. Is there evidence that the design modification reduces the refrigerators' average energy consumption?
8. The Yonalasee tennis club has two systems to measure the speed of a tennis ball. The local tennis pros suspects one system (**speed1**) consistently records faster speeds. To test her suspicions, she sets up both systems and records the speeds of 12 serves (three serves from each side of the court). The values are stored in the data frame **TENNIS** in the variables **speed1** and **speed2**. The recorded speeds are in kilometers per hour. Does the evidence support the tennis pro's suspicion? Use $\alpha = 0.10$.
9. An advertising agency is interested in targeting the appropriate gender for a new "low-fat" yogurt. In a national survey of 1200 women, 825 picked the "low-fat" yogurt over a regular yogurt. Meanwhile, 525 out of 1150 men picked the "low-fat" yogurt over the

regular yogurt. Given these results, should the advertisements be targeted at a specific gender? Test the appropriate hypothesis at the $\alpha = 0.01$ level.

10. A plastics manufacturer makes two sizes of milk containers: half-gallon and gallon sizes. The time required for each size to dry is recorded in seconds in the data frame **MILKCARTON**. Test to see if there are differences in average drying times between the container sizes.

11. A multinational conglomerate has two textile centers in two different cities. In order to make a profit, each location must produce more than 1000 kilograms of refined wool per day. A random sample of the wool production in kilograms on five different days over the last year for the two locations was taken. The results are stored in the data frame **WOOL**. Based on the collected data, does the evidence suggest the locations are profitable? Is one location superior to the other?

12. Use the data frame **FERTILIZE**, which contains the height in inches for plants in the variable **height** and the fertilization type in the variable **fertilization** to

- (a) Test if the data suggest that the average height of self-fertilized plants is more than 17 inches. (Use $\alpha = 0.05$.)
- (b) Compute a one-sided 95% confidence interval for the average height of self-fertilized plants ($H_1 : \mu > 17$).
- (c) Compute the required sample size to obtain a power of 0.90 if $\mu_1 = 18$ inches assuming that $\sigma = s$.
- (d) What is the power of the test in part (a) if $\sigma = s$ and $\mu_1 = 18$?

13. A manufacturer of lithium batteries has two production facilities. One facility (A) manufactures a battery with an advertised life of 180 hours, while the second facility (B) manufactures a battery with an advertised life of 200 hours. Both facilities are trying to reduce the variance in their products' lifetimes. Is the variability in battery life equivalent, or does the evidence suggest the facility producing 200-hour batteries has smaller variability than the facility producing 180-hour batteries? Use the data frame **BATTERY** with $\alpha = 0.05$ to test the appropriate hypothesis.

14. In the construction of a safety strobe, a particular manufacturer can purchase LED diodes from one of two suppliers. It is critical that the purchased diodes conform to their stated specifications with respect to diameter since they must be mated with a fixed width cable. The diameter in millimeters for a random sample of 15 diodes from each of the two suppliers is stored in the data frame **LEDDIODE**. Based on the data, is there evidence to suggest a difference in variabilities between the two suppliers? Use an α level of 0.01.

15. The technology at a certain computer manufacturing plant allows silicon sheets to be split into chips using two different techniques. In an effort to decide which technique is superior, 28 silicon sheets are randomly selected from the warehouse. The two techniques of splitting the chips are randomly assigned to the 28 sheets so that each technique is applied to 14 sheets. The results from the experiment are stored in the data frame **CHIPS**. Use $\alpha = 0.05$, and test the appropriate hypothesis to see if there are differences between the two techniques. The values recorded in **CHIPS** are the number of usable chips from each silicon sheet.

16. Phenylketonuria (PKU) is a genetic disorder that is characterized by an inability of the body to utilize an essential amino acid, phenylalanine. Research suggests patients with

phenylketonuria have deficiencies in coenzyme Q10. The data frame **PHENYL** records the level of Q10 at four different times for 46 patients diagnosed with PKU. The variable **Q10.1** contains the level of Q10 measured in μM for the 46 patients. **Q10.2**, **Q10.3**, and **Q10.4** record the values recorded at later times, respectively, for the 46 patients (Artuch et al., 2004).

- (a) Normal patients have a Q10 reading of $0.69 \mu\text{M}$. Using the variable **Q10.2**, is there evidence that the mean value of Q10 in patients diagnosed with PKU is less than $0.69 \mu\text{M}$? (Use $\alpha = 0.01$.)
 - (b) Patients diagnosed with PKU are placed on strict vegetarian diets. Some have speculated that patients diagnosed with PKU have low Q10 readings because meats are rich in Q10. Is there evidence that the patients' Q10 level decreases over time? Construct a 99% confidence interval for the mean difference of the Q10 levels using **Q10.1** and **Q10.4**.
17. According to the Pamplona, Spain, registration, 0.4% of immigrants in 2002 were from Bolivia. In June of 2005, a sample of 3740 registered foreigners was randomly selected. Of these, 87 were Bolivians. Is there evidence to suggest immigration from Bolivia has increased? (Use $\alpha = 0.05$.)
18. Find the power for the hypothesis $H_0 : \mu = 65$ versus $H_1 : \mu > 65$ if $\mu_1 = 70$ at the $\alpha = 0.01$ level assuming $\sigma = s$ for the variable **hard** in the data frame **Rubber** of the MASS package.
19. The director of urban housing in Vitoria, Spain, claims that at least 50% of all apartments have more than one bathroom and that at least 75% of all apartments have an elevator.
- (a) Can the director's claim about bathrooms be contradicted? Test the appropriate hypothesis using $\alpha = 0.10$. Note that the number of bathrooms is stored in the variable **toilets** in the data frame **VIT2005**.
 - (b) Can the director's claim about elevators be substantiated using an α level of 0.10? Use both an approximate method as well as an exact method to reach a conclusion. Are the methods in agreement?
 - (c) Test whether the proportion of apartments built prior to 1980 without garages have a smaller proportion with elevators than without elevators.
20. A rule of thumb used by realtors in Vitoria, Spain, is that each square meter will cost roughly €3000; however, there is some suspicion that this figure is high for apartments in the 55 to 66 m^2 range. Use a 5 m^2 bracket, that is, $[55, 60)$ (small) and $[60, 65)$ (medium), to see if evidence exists that the average between the medium and small apartment sizes is less than €15,000.
- (a) Use the data frame **VIT2005** and the variables **totalprice** and **area** to test the appropriate hypothesis at a 5% significance level.
 - (b) Are the assumptions for using a *t*-test satisfied? Explain.
 - (c) Does the answer for (a) differ if the variances are assumed to be equal? Can the hypothesis of equal variances be rejected?

21. A survey to determine unemployment demographics was administered during the first trimester of 2005 in the Spanish province of Navarra. The numbers of unemployed people according to urban and rural areas and gender follow.

Unemployment in Navarra, Spain, in 2005

	Male	Female	Totals
Urban	4734	6161	10895
Rural	3259	4033	7292
Totals	7993	10194	18187

- (a) Test to see if there is evidence to suggest that $\pi_{\text{male}|\text{urban}} < \pi_{\text{female}|\text{urban}}$ at $\alpha = 0.05$.
- (b) Use an exact test to see if the evidence suggests $\pi_{\text{female}|\text{urban}} > 0.55$.
- (c) Is there evidence to suggest the unemployment rate for females given that they live in a rural area is greater than 50%? Use $\alpha = 0.05$ with an exact test to reach a conclusion.
- (d) Does evidence suggest that $\pi_{\text{female}|\text{urban}} > \pi_{\text{female}|\text{rural}}$?

22. The owner of a transportation fleet is evaluating insurance policies for transporting hazardous waste. The owner has narrowed his possibility of insurers to two companies (A and B). Insurance company A claims to have the least expensive policies on the market while insurer B disputes the claim. To evaluate company A's claim, the owner requests the last 100 policies issued by each insurer. The means and standard deviations are €325 and €85 for company A and €340 and €75 for company B. Based on these summary statistics, the owner was not convinced that company A actually had less expensive rates. Consequently, a representative from company A was sent to speak to the owner. The representative from company A convinced the owner to take another look at the numbers. This time, insurance quotes were sought from both insurers for the next 15 jobs transporting hazardous waste. Results are given in the data frame **INSURQUOTES**. Analyze these data. How is it possible the owner changed his mind with a sample of size 15 versus the results based on a sample of size 100?

23. Environmental monitoring is done in many fashions, including tracking levels of different chemicals in the air, underground water, soil, fish, milk, and so on. It is believed that milk cows eating in pastures where gamma radiation from iodine exceeds $0.3 \mu\text{Gy}/\text{h}$ in turn leads to milk with iodine concentrations in excess of $3.7 \text{ MBq}/\text{m}^3$. Assuming the distribution of iodine in pastures follows a normal distribution with a standard deviation of $0.015 \mu\text{Gy}/\text{h}$, determine the required sample size to detect a 2% increase in baseline gamma radiation ($0.3\mu\text{Gy}/\text{h}$) using an $\alpha = 0.05$ significance level with probability 0.99 or more.

24. A local farmer packages and freezes his spinach. He claims that the packages weigh 340 grams and have a standard deviation of no more than $\sigma = 15$ grams. The manager of a local organic supermarket is somewhat skeptical of the farmer's claim and decides to test the claim using a random sample of 10 frozen spinach packages.

- (a) Find the critical region of the test if $\alpha = 0.05$.
- (b) Find the power of the test if $\sigma = 10$.

25. A cell phone provider has estimated that it needs revenues of €2 million per day in order to make a profit and remain in the market. If revenues are less than €2 million

per day, the company will go bankrupt. Likewise, revenues greater than €2 million per day cannot be handled without increasing staff. Assume that revenues follow a normal distribution with $\sigma = €0.5$ million and a mean of μ .

- (a) Graphically depict the power function for testing $H_0 : \mu = 2$ versus $H_1 : \mu \neq 2$ if $n = 150$ and $\alpha = 0.05$ for values of μ ranging from 1.8 to 2.2.
 - (b) Graphically depict the power for testing $H_0 : \mu = 2$ versus $H_1 : \mu \neq 2$ when $\mu_1 = 2.1$ and $n = 150$ for values of α ranging from 0.001 to 0.999.
 - (c) Graphically depict the power for testing $H_0 : \mu = 2$ versus $H_1 : \mu \neq 2$ when $\mu_1 = 2.1$ and $\alpha = 0.05$ for values of n ranging from 1 to 500.
 - (d) Generalize what is seen in the graphs for (a), (b), and (c).
26. Use simulation to compute the empirical significance level by generating 10,000 samples of size n from a $N(100, 28)$ population using $\alpha = 0.05$ to test the alternative hypothesis $H_1 : \mu \neq 100$. Use the command `set.seed(3)` so the answers can be reproduced.
- (a) Use samples of size $n = 49$.
 - (b) Use samples of size $n = 196$.
 - (c) Use samples of size $n = 1936$.
 - (d) Does increasing the sample size affect the significance level?
27. Use simulation to compute the empirical power for testing $H_0 : \mu = 100$ versus $H_1 : \mu > 100$ when $\mu = 108$ and sampling from a $N(108, 28)$ distribution. Use 10,000 samples with $n = 49$ in the simulation and `set.seed(21)` so that the results will be reproducible.
- (a) Use a significance level of $\alpha = 0.05$.
 - (b) Use a significance level of $\alpha = 0.20$.
 - (c) Compute the theoretical power for scenarios (a) and (b). How do these values compare to those from the simulations?
 - (d) What happens to the empirical power as α increases?
28. Test the null hypothesis that the mean life for a certain brand of 19-mm tubular tires is 1000 miles against the alternative hypothesis that it is less than 1000 miles using $\alpha = 0.05$. Assume that tubular tire life follows a normal distribution with $\sigma = 100$ miles.
- (a) Find the probability of a type I error for $n = 16$ if the null hypothesis is rejected when the sample mean is less than or equal to 960 miles.
 - (b) Plot the power function for $n = 16$ for values of μ between 900 and 1000 miles.
29. Given a normal population with unknown mean and a known variance of $\sigma^2 = 2.5^2$, test the hypothesis $H_0 : \mu = 10$ versus $H_1 : \mu < 10$ at the $\alpha = 0.05$ significance level.
- (a) Use the command `set.seed(42)` to generate 10,000 samples of size $n = 25$ from a $N(10, 2.5)$ population. Is the empirical significance level close to 5%?

- (b) Compute a 95% confidence interval for α when simulating 10,000 samples of size $n = 25$ from a $N(10, 2.5)$ population.
- (c) What is the theoretical power if $\mu_1 = 9.5$ for the given hypothesis test?
- (d) Graphically represent the power for testing $H_0 : \mu = 10$ versus $H_1 : \mu < 10$ for samples of size $n = 25$ from a $N(10, 2.5)$ population when $\alpha = 0.05$ for values of μ from 8 to 10.
- (e) Graphically represent the power for testing $H_0 : \mu = 10$ versus $H_1 : \mu < 10$ for samples of size $n = 25$ when $\mu_1 = 9.5$ for values of α ranging from 0.001 to 0.999.
30. Generate 10,000 samples of size $n_X = 20$ from $X \sim N(8, 2)$ and 10,000 samples of size $n_Y = 20$ from $Y \sim N(6, 2)$. Use `set.seed(9)` so that the answers are reproducible. Assuming X and Y are independent and a 5% significance level,
- What type of distribution does the statistic S_X^2/S_Y^2 follow?
 - Create a density histogram of the 10,000 values of s_X^2/s_Y^2 . Superimpose the theoretical sampling distribution of S_X^2/S_Y^2 on the density histogram.
 - Compute the empirical significance level for testing $H_0 : \sigma_X^2/\sigma_Y^2 = 1$ versus $H_1 : \sigma_X^2/\sigma_Y^2 \neq 1$.
 - Plot the power function for testing $H_0 : \sigma_X^2/\sigma_Y^2 = 1$ versus $H_1 : \sigma_X^2/\sigma_Y^2 \neq 1$ for ratios of σ_X^2/σ_Y^2 from 0.1 to 10.
 - Repeat (d) for $n_X = n_Y = 40$.
 - Repeat (d) for $n_X = n_Y = 80$.
 - Put the graphs from (d), (e), and (f) on the same graph.
 - Plot the power function for testing $H_0 : \sigma_X^2/\sigma_Y^2 = 1$ versus $H_1 : \sigma_X^2/\sigma_Y^2 \neq 1$ for α values between 0.001 and 0.999 if $\sigma_X^2/\sigma_Y^2 = 2$ and
 - $n_X = n_Y = 20$,
 - $n_X = n_Y = 40$, and
 - $n_X = n_Y = 80$.
 - Simulate the power for testing $H_0 : \sigma_X^2/\sigma_Y^2 = 1$ versus $H_1 : \sigma_X^2/\sigma_Y^2 \neq 1$ at the $\alpha = 0.05$ level when $\sigma_X^2/\sigma_Y^2 = 2$ and
 - $n_X = n_Y = 20$,
 - $n_X = n_Y = 40$, and
 - $n_X = n_Y = 80$.
 - Compute the theoretical power for the three previous scenarios.
31. Cleveland (1993) suggested that the yield recorded for the year 1932 was actually 1931's yield and vice versa for the Morris site in the `barley` data set after examining the data using trellis graphics.
- Use the package `lattice` to recreate the trellis graph that would have given Cleveland his insight. (Hint: Load the lattice package; type `?barley`; and read the example given.) Change the default layout so the graph has two columns and three rows of plots.

- (b) Recreate the graph from (a) using `ggplot2`.
- (c) If the years were actually switched as described when the data was originally recorded, is there evidence to suggest the average barley yield in 1932 (recorded as 1931's yield) for the Morris site is greater than the average barley yield in 1932 for the Crookston site? Use the five-step procedure to test the appropriate hypotheses using an $\alpha = 0.05$ significance level.

Chapter 10

Nonparametric Methods

10.1 Introduction

The statistical inference techniques presented in Chapter 8 and Chapter 9 are based on complete satisfaction of all of the assumptions made in the derivations of their sampling distributions. Indeed, many of the techniques in Chapters 8 and 9 are commonly referred to as parametric techniques since not only was the form of the underlying population (generally normal) stated, but so was one or more of the underlying distribution's parameters. This chapter introduces both distribution-free tests as well as nonparametric tests. The collection of inferential techniques known as **distribution-free** is based on functions of the sample observations whose corresponding random variable has a distribution that is independent of the specific distribution function from which the sample was drawn. Consequently, assumptions with respect to the underlying population are not required. **Nonparametric tests** involve tests of a hypothesis where there is no statement about the distribution's parameters; however, it is common practice to refer collectively to both distribution-free tests and nonparametric tests simply as **nonparametric methods**.

When there are analogous parametric and nonparametric tests, comparisons between tests can be made based on power. The **power efficiency** of a test A relative to a test B is the ratio of n_b/n_a , where n_a is the number of observations required by test A for the power of test A to equal the power of test B when n_b observations are used. Since the power value is conditional on the type of alternative hypothesis and on the significance level, power efficiency can be difficult to interpret. One way to avoid this problem is to use the **asymptotic relative efficiency (ARE)** (a limiting power efficiency) of consistent tests. A test is consistent for a specified alternative if the power of the test when that alternative is true approaches 1 as the sample size approaches infinity (Gibbons, 1997). Provided that A and B are consistent tests of a null hypothesis H_0 and alternative hypothesis H_1 at significance level α , the asymptotic relative efficiency of test A to test B is the limiting ratio n_b/n_a , where n_a is the number of observations required by test A for the power of test A to equal the power of test B based on n_b observations while simultaneously $n_b \rightarrow \infty$ and $H_1 \rightarrow H_0$ (Gibbons and Chakraborti, 2003). Although the *ARE* considers infinite (hence not obtainable) sample sizes, the *ARE* provides a good approximation to the relative efficiency for many situations of practical interest. When a nonparametric technique has a parametric analog, the *ARE* will be used to compare the two techniques.

10.2 Sign Test

When the parent distribution is skewed or has long tails, the median is generally a better measure of the distribution's center than is the mean. In this section, a procedure for testing hypotheses concerning the population median is introduced. This procedure is known as the **sign test**. A corresponding confidence interval formula for the median will be derived as well.

To use the sign test, assume X_1, X_2, \dots, X_n is a random sample of n observations drawn from a continuous population with unknown median ψ . The sign test statistic, S , is defined as the number of positive differences among the $X_1 - \psi_0, X_2 - \psi_0, \dots, X_n - \psi_0$, where ψ_0 is the median from the null hypothesis $H_0 : \psi = \psi_0$. The distribution of S when H_0 is true is $S \sim \text{Bin}(n, \pi = 0.5)$.

The sign test may also be used for testing whether the median difference (ψ_D) between two dependent populations (X and Y) is equal to some value, $H_0 : \psi_D = \psi_0$. It is important to point out that ψ_D is not usually equal to $\psi_X - \psi_Y$. The only instance where ψ_D is equal to $\psi_X - \psi_Y$ is when X and Y are symmetric populations. For dependent samples, S is defined as the number of positive differences among the $X_1 - Y_1 - \psi_0, X_2 - Y_2 - \psi_0, \dots, X_n - Y_n - \psi_0$.

Since the assumption of a continuous population is a requirement for using the sign test, theoretically, there should not be any values that are exactly equal to the parameter being tested in the sample; however, due to rounding or crude measurements, it is not uncommon to observe sample values equal to ψ_0 , the value of ψ or ψ_D under the null hypothesis. There are several strategies one can pursue in dealing with values that are equal to the parameter being tested. Some of these include randomization, midranks, average statistic, average probability, least favorable statistic, range of probability, and omission of tied observations. The book by Pratt and Gibbons (1981) gives a more complete discussion of these various techniques. The final approach, **omission of tied observations**, consists of eliminating the value(s) in the sample that are equal to the parameter being tested. This is the approach that will be used in this text. This method leads to some loss of information; however, if the number of observations to be omitted is small compared to the sample size, this loss is usually acceptable. Generally, omission of tied observations decreases the power of the test.

Due to the discrete nature of S , it is generally not possible to define a rejection region that results in a test whose size is exactly equal to a prescribed α . Consequently, the approach presented for this test relies on p -values rather than on defining rejection regions for the statistical conclusion. The three possible alternative hypotheses and their associated p -value calculation formulas are presented in Table 10.1 on the facing page. If one assumes a normal population, the asymptotic relative efficiency (*ARE*) of the sign test relative to the *t*-test is $\frac{2}{\pi} \approx 0.637$. The *ARE* of the sign test relative to the *t*-test is also quite poor (1/3) for the uniform distribution (short tails). For the Laplace distribution (long tails) though, the *ARE* of the sign test in relation to the *t*-test is 2.

10.2.1 Confidence Interval Based on the Sign Test

A corresponding confidence interval for the median is also based on the binomial distribution by using the same assumptions as those for the one-sample sign test; however, the full sample is always used in the calculations. Assume X_1, X_2, \dots, X_n is a random sample of n observations drawn from a continuous population with an unknown median ψ . A confidence interval with a confidence level of at least $(1 - \alpha) \cdot 100\%$ can be constructed by using the k^{th} and $(n - k + 1)^{\text{st}}$ order statistics of the sample, where k is the largest value such that $\mathbb{P}(S < k) \leq \alpha/2$. For a one-sided confidence interval, k is the largest

Table 10.1: Summary for testing the median — sign test

Null Hypothesis — $H_0 : \psi = \psi_0$ Test Statistic's Value — s = number of observed positive differences

Alternative Hypothesis	φ -Value Formula
$H_1 : \psi < \psi_0$	$\mathbb{P}(S \leq s H_0) = \sum_{i=0}^s \binom{n}{i} \left(\frac{1}{2}\right)^n$
$H_1 : \psi > \psi_0$	$\mathbb{P}(S \geq s H_0) = \sum_{i=s}^n \binom{n}{i} \left(\frac{1}{2}\right)^n$
$H_1 : \psi \neq \psi_0$	$\sum_{i=0}^n I(\mathbb{P}(S = i) \leq \mathbb{P}(S = s)) \cdot \binom{n}{i} \left(\frac{1}{2}\right)^n$

Recall that $I(condition) = 1$ if *condition* is true and 0 if *condition* is false.

value such that $\mathbb{P}(S < k) \leq \alpha$. Order statistics are the random variables X_1, X_2, \dots, X_n rearranged in order of relative magnitude and are denoted $X_{(1)}, X_{(2)}, \dots, X_{(n)}$. That is, $X_{(1)} < X_{(2)} < \dots < X_{(n)}$. The actual confidence level is given by

$$1 - 2 \times \mathbb{P}(S < k) \text{ where } \mathbb{P}(S = x) = \binom{n}{x} \left(\frac{1}{2}\right)^n. \quad (10.1)$$

Clearly, k must be a positive integer since it is the subscript of an order statistic. Using (10.1) will seldom produce typical confidence levels such as 90%, 95%, or 99% exactly. Many texts provide charts to find k for the construction of confidence intervals at these typical confidence levels that are based either on always attaining a level of at least $(1 - \alpha) \times 100\%$ confidence or by providing the value of k such that the achieved confidence level is as close to $(1 - \alpha) \times 100\%$ as possible. The first approach will always return confidence intervals with a confidence level of $(1 - \alpha) \times 100\%$ or more. Roughly 50% of the confidence intervals computed with the second approach will return confidence intervals of less than the reported confidence.

The function **SIGN.test()** provided in the **PASWR2** package returns two confidence intervals with exact confidence levels closest to the $(1 - \alpha) \times 100\%$ level specified by the user. One of these intervals has a confidence level lower than the specified level and the other has a higher confidence level than the specified level. Finally, the function uses linear interpolation between these first two intervals to give a confidence interval with the level specified by the user.

10.2.2 Normal Approximation to the Sign Test

For moderately sized samples ($n > 20$), the binomial distribution with $\pi = 0.5$ can be reasonably approximated with the normal distribution. Since $S \sim \text{Bin}(n, 0.5)$, it follows that $\mu_S = n(0.5)$ and $\sigma_S = \sqrt{n(0.5)^2}$. That is, $S \stackrel{\text{d}}{\sim} N(\mu_S, \sigma_S)$. The standardized test statistic under the assumption that $H_0 : \psi = \psi_0$ is true is

$$Z = \frac{S - n(0.5)}{\sqrt{n(0.5)^2}} \sim N(0, 1), \quad (10.2)$$

where S is defined as the number of positive differences among the $X_1 - \psi_0, X_2 - \psi_0, \dots, X_n - \psi_0$. See Figure 10.1 for a graph of a $\text{Bin}(20, 0.5)$ superimposed with a normal distribution with $\mu_S = 20(0.5) = 10$ and $\sigma_S = \sqrt{20(0.5)^2} = 2.2361$.

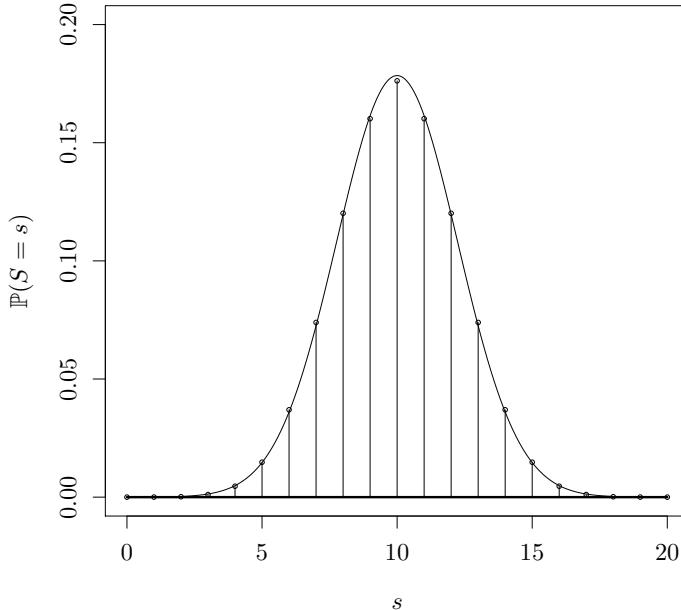


FIGURE 10.1: Graphical representation of a $\text{Bin}(20, 0.5)$ distribution and a superimposed normal distribution with $\mu_S = 20(0.5) = 10$ and $\sigma_S = \sqrt{20(0.5)^2} = 2.2361$

The formula for calculating the observed value of the standardized test statistic as well as the three possible alternative hypotheses and their rejection regions are described in Table 10.2 on the facing page.

A corresponding confidence interval for the median based on (10.2) is formed with the k^{th} and $(n - k + 1)^{\text{st}}$ order statistics of the sample, where

$$k = \frac{n + 1 + z_{\alpha/2} \times \sqrt{n}}{2}. \quad (10.3)$$

For a one-sided confidence interval, replace $z_{\alpha/2}$ with z_α and solve for k . Since k is generally not an integer, it can be either rounded or truncated. To obtain a conservative estimate, one should truncate.

Example 10.1 ▷ Sign Test: Telephone Call Times ◁ Table 10.3 on the next page and the variable `call.time` in the data frame `PHONE` contain the times in minutes of long-distance telephone calls during a one-month period for a small business.

- (a) Use an exact test with $\alpha = 0.05$ to see if 2.1 minutes is a representative measure of center for the telephone call lengths.

Table 10.2: Summary for testing the median — approximation to the sign test

Null Hypothesis — $H_0 : \psi = \psi_0$

$$\text{Standardized Test Statistic's Value} \quad z_{obs} = \frac{s \pm 0.5 - n(0.5)}{\sqrt{n(0.5)^2}}$$

Alternative Hypothesis	Rejection Region
$H_1 : \psi < \psi_0$	$z_{obs} < z_\alpha$
$H_1 : \psi > \psi_0$	$z_{obs} > z_{1-\alpha}$
$H_1 : \psi \neq \psi_0$	$ z_{obs} > z_{1-\alpha/2}$

Note: The quantity ± 0.5 in the numerator of z_{obs} is the continuity correction.
When $H_1 : \psi < \psi_0$, the quantity $+0.5$ is used. When $H_1 : \psi > \psi_0$, the quantity -0.5 is used. When $H_1 : \psi \neq \psi_0$, use $+0.5$ if $s < n(0.5)$ and -0.5 if $s > n(0.5)$.

Table 10.3: Long-distance telephone call times in minutes (**PHONE**)

i	$X_{(i)}$										
1	0.2	5	0.7	9	1.3	13	2.7	17	5.6	21	9.7
2	0.2	6	0.7	10	1.7	14	4.0	18	6.1	22	9.7
3	0.2	7	0.7	11	2.1	15	4.3	19	6.7	23	12.9
4	0.2	8	0.8	12	2.1	16	5.2	20	7.0		

- (b) Construct a 95% confidence interval for the population median.
- (c) Use a normal approximation for the test used in part (a) to test if 2.1 minutes is a representative measure of center for the telephone call lengths.
- (d) Construct a 95% confidence interval for the population median using (10.3).

Solution: First, use the function `eda()` to assess the general shape of telephone call times. The four graphs in Figure 10.2 on the following page all lead one to the conclusion that the distribution of the long-distance telephone call times is positively skewed (skewed right). Consequently, the median is a more representative measure of center than is the mean for these data.

- (a) Use the five-step procedure to test if 2.1 minutes is a representative measure of center.

Step 1: **Hypotheses** — The null and alternative hypotheses to test whether or not 2.1 minutes is a representative measure of the center of telephone call times are

$$H_0 : \psi = 2.1 \text{ versus } H_1 : \psi \neq 2.1.$$

Step 2: **Test Statistic** — The test statistic chosen is S , where S is the number of positive differences among $X_1 - 2.1, X_2 - 2.1, \dots, X_n - 2.1$. Here, $s = 11$. Also note that since there are two instances where $x_i = \psi_0$, n is reduced from 23 to 21.

EXPLORATORY DATA ANALYSIS

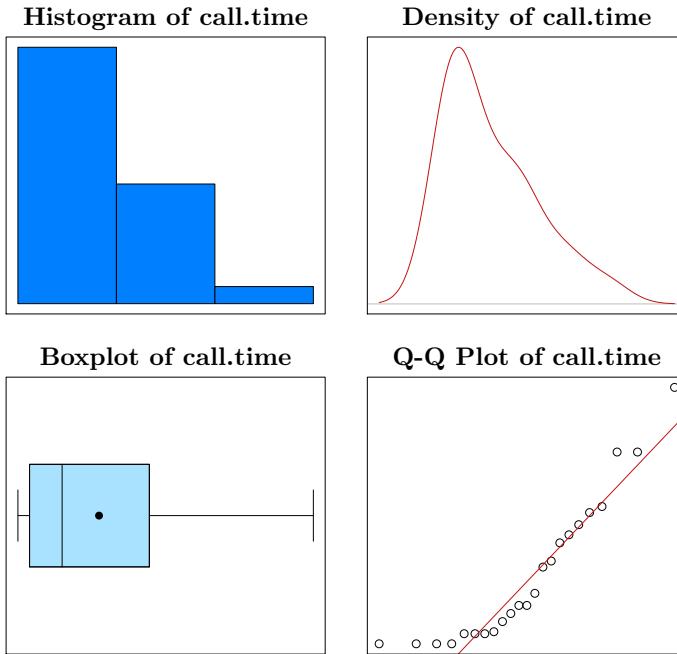


FIGURE 10.2: Graphical representation of the data in `call.time` with the function `eda()`

Step 3: Rejection Region Calculations — Rejection is based on the φ -value, so none are required.

Step 4: Statistical Conclusion — The φ -value is

$$\sum_{i=0}^n I(\mathbb{P}(S = i) \leq \mathbb{P}(S = s)) \cdot \binom{n}{i} \left(\frac{1}{2}\right)^n = 1.$$

See R Code 10.1 for the computation of the φ -value.

Fail to reject H_0 .

Step 5: English Conclusion — There is not sufficient evidence to suggest the median length of long-distance telephone calls is not 2.1 minutes.

R Code 10.1

```
> pvalue <- sum(dbinom(0:21, 21, 0.5)[dbinom(0:21, 21, 0.5) <=
+      dbinom(11, 21, 0.5)])
> pvalue
[1] 1
```

(b) To construct a 95% confidence interval for the population median, start by finding the largest and smallest values of k such that $1 - 2 \times \mathbb{P}(S < k) > 0.95$ and $1 - 2 \times \mathbb{P}(S < k) < 0.95$, respectively. To find these values, use R, and type `1 - 2 * pbisnom(0:23, 23, 0.5)`. To save space, only values 5 through 8 are used.

```
> 1 - 2 * pbinom(5:8, 23, 0.5)
[1] 0.9893780 0.9653103 0.9068604 0.7899604
```

From the R output, it is seen that

$$1 - 2 \times \mathbb{P}(S \leq 6) = 1 - 2 \times \mathbb{P}(S < 7) = 0.9653$$

and

$$1 - 2 \times \mathbb{P}(S \leq 7) = 1 - 2 \times \mathbb{P}(S < 8) = 0.9069.$$

So, use the $k = 7^{\text{th}}$ with the $n - k + 1 = 23 - 7 + 1 = 17^{\text{th}}$ order statistics to form the 96.53% confidence interval, $CI_{0.9653}(\psi) = [0.7, 5.6]$, and the $k = 8^{\text{th}}$ with the $n - k + 1 = 23 - 8 + 1 = 16^{\text{th}}$ order statistics to form the 90.68% confidence interval, $CI_{0.9068}(\psi) = [0.8, 5.2]$. Thus, the 95% interpolated confidence interval, $[L, U]$ is calculated such that

$$\frac{0.9653 - 0.9068}{0.9653 - 0.95} = \frac{0.7 - 0.8}{0.7 - L} \quad \text{AND} \quad \frac{0.9653 - 0.9068}{0.9653 - 0.95} = \frac{5.6 - 5.2}{5.6 - U}$$

$$\Rightarrow L = 0.7262 \quad \Rightarrow U = 5.4952.$$

So, $CI_{0.95}(\psi) = [0.7262, 5.4952]$.

Using the function `SIGN.test()` on the variable (`call.time`) gives the following output:

```
> with(data = PHONE, SIGN.test(call.time, md = 2.1))
```

```
One-sample Sign-Test

data: call.time
s = 11, p-value = 1
alternative hypothesis: true median is not equal to 2.1
95 percent confidence interval:
0.7261939 5.4952244
sample estimates:
median of x
2.1
Conf.Level L.E.pt U.E.pt
Lower Achieved CI      0.9069 0.8000 5.2000
Interpolated CI        0.9500 0.7262 5.4952
Upper Achieved CI      0.9653 0.7000 5.6000
```

(c) Use the five-step procedure using the normal approximation to the sign test to test if 2.1 minutes is a representative measure of center.

Step 1: Hypotheses — The null and alternative hypotheses to test whether or not 2.1 minutes is a representative measure of the center of telephone call times are

$$H_0 : \psi = 2.1 \text{ versus } H_1 : \psi \neq 2.1.$$

Step 2: Test Statistic — The test statistic chosen is S , where S is the number of positive differences among $X_1 - 2.1, X_2 - 2.1, \dots, X_n - 2.1$. Here, $s = 11$. Also note that since there are two instances where $x_i = \psi_0$, n is reduced from 23 to 21.

Step 3: Rejection Region Calculations — Because the standardized test statistic is distributed approximately $N(0, 1)$ and H_1 is a two-sided hypothesis, the rejection region is $|z_{obs}| \geq z_{1-0.05/2} = 1.96$. The value of the standardized test statistic is

$$z_{obs} = \frac{s \pm 0.5 - n(0.5)}{\sqrt{n(0.5)^2}} = \frac{11 - 0.5 - 21(0.5)}{\sqrt{21(0.5)^2}} = 0.$$

Step 4: Statistical Conclusion — The φ -value is $2 \times \mathbb{P}(Z \geq 0) = 1$.

- I. From the rejection region, do not reject H_0 because $|z_{obs}| = 0$ is not larger than 1.96.
- II. From the φ -value, do not reject H_0 because the φ -value = 1 is larger than 0.05.

Fail to reject H_0 .

Step 5: English Conclusion — There is not sufficient evidence to suggest the median length of long-distance telephone calls is not 2.1 minutes.

(d) To construct a confidence interval for the population median using (10.3), solve

$$\begin{aligned} k &= \frac{n + 1 + z_{\alpha/2} \times \sqrt{n}}{2} \\ &= \frac{23 + 1 - 1.96 \times \sqrt{23}}{2} = 7.3. \end{aligned}$$

By truncating k , $k = 7$. The approximate 95% confidence interval for ψ is then

$$[x_{(k)}, x_{(n-k+1)}] = [x_{(7)}, x_{(23-7+1)}] = [0.7, 5.6].$$



10.3 Wilcoxon Signed-Rank Test

In Section 10.2, the sign test was used to test hypotheses concerning the median. The only requirements placed on the data when using the sign test are, first, that the population from which one is sampling be continuous, and, second, that the population has a median. Since the sign test only uses the signs of the differences between each observation and the hypothesized median ψ_0 , a test that incorporates not only the signs of the differences but also their magnitudes might yield a better performance in terms of power. In fact, the test presented in this section uses both the signs of the differences between each observation and the hypothesized median as well as the magnitudes of the differences. In order to use this test, the **Wilcoxon signed-rank test**, one must also assume a symmetric population in addition to the assumptions for the sign test. Consequently, although the Wilcoxon signed-rank test can be used to test hypotheses concerning the median, it can be used equally well to test hypotheses regarding the mean, as the assumption of symmetry in the distribution is equivalent to assuming that the mean and median are equal.

The assumption of symmetry is a rather restrictive assumption compared to those for the sign test. Nevertheless, it is certainly less restrictive than its normal analog, the t -test,

which requires the data not only to come from a population that is symmetric about its median but also to come from a normal distribution. The Wilcoxon signed-rank test does not always perform better than the sign test. For the Laplace distribution (long-tailed), for example, the *ARE* of the sign test relative to the Wilcoxon signed-rank test is 4/3. Table 10.4 lists the *ARE* of the sign test relative to the *t*-test (*ARE*(*S, t*)), the *ARE* of the Wilcoxon signed-rank test relative to the *t*-test (*ARE*(*T⁺, t*)), and the *ARE* of the sign test relative to the Wilcoxon signed-rank test (*ARE*(*S, T⁺*)) for the uniform distribution (short tails), normal distribution, and the Laplace distribution (long tails).

Table 10.4: Asymptotic relative efficiency comparisons

Distribution	<i>ARE</i> (<i>S, t</i>)	<i>ARE</i> (<i>T⁺, t</i>)	<i>ARE</i> (<i>S, T⁺</i>)
Uniform	1/3	1	1/3
Normal	$2/\pi \approx 0.64$	$3/\pi \approx 0.955$	2/3
Laplace	2	1.5	4/3

In summary, for large n , when testing location for symmetric populations, it is generally better to use the sign test with Laplace populations and the Wilcoxon signed-rank test for all other non-normal symmetric distributions. It can be shown that the *ARE* of the Wilcoxon signed-rank test relative to the *t*-test is never less than 0.864 for any continuous distribution and is ∞ for some distributions. For small n , it is not so clear which test will be better.

Given a random sample X_1, X_2, \dots, X_n taken from a continuous population that is symmetric with respect to its median ψ , under the null hypothesis $H_0 : \psi = \psi_0$, the differences, $D_i = X_i - \psi_0$, are symmetrically distributed about zero. Further, positive and negative differences of the same magnitude have the same probability of occurring. As with the sign test, if any of the d_i s are zero, they are removed from the sample before the ranks are computed, and the value of n is reduced accordingly.

To compute the Wilcoxon signed-rank statistic,

Step A: Take the absolute value of the n d_i s.

Step B: Assign the ranks to the n values from step A. If there are ties, use the **midranks**. The midrank is defined as the average rank of the tied observations.

Step C: Multiply the values in step B by the sign of the original d_i s.

Step D: Sum the positive quantities in step C. The result is denoted T^+ . The random variable (test statistic) T^+ is defined as the sum of the positive signed ranks and the random variable T^- is defined as the sum of negative signed ranks.

Provided the null hypothesis is true, $E(T^+) = E(T^-)$. When T^+ is either sufficiently small or sufficiently large, the null hypothesis is rejected. The test statistic T^+ takes on values between 0 and $n(n+1)/2$, and has a mean and variance of $n(n+1)/4$ and $n(n+1)(2n+1)/24$, respectively. The distribution of T^+ is known as the **Wilcoxon signed-rank distribution**. Although conceptually easy to understand, one needs access

to extensive tables or statistical software to compute exact ϕ -values. Further, tabled values for T^+ are generally published only when there are no ties in the absolute values of the d_i s, $|d_i|$ for $i = 1, \dots, n$. When there are ties in the $|d_i|$ s, the function `wilcox.test()` uses a normal approximation to compute the ϕ -values. It is possible to calculate exact ϕ -values when testing hypotheses about the median as well as to construct exact confidence intervals for the median even in the presence of ties using the function `wilcoxe.test()` from the PASWR2 package. The function is rather primitive and should only be used for problems with fewer than 19 observations as the memory requirements are rather large.

Example 10.2 \triangleright **Trivial T^+ Distribution** \triangleleft What is the sampling distribution of T^+ for the trivial case where $n = 3$ and there are no ties among the absolute values of the $n = 3$ d_i s?

Solution: Since there are three values ($n = 3$) that must be ranked and each d_i may have either a positive or negative sign, there are a total of $2^n = 2^3 = 8$ possible sets of signs associated with the three possible ranks (1, 2, 3). Under the null hypothesis, each of the sets of signs is equally likely to occur, and thus each has a probability of $1/8$ of occurring. Table 10.5 lists the eight possible sets of signs and Table 10.6 provides the probability distribution (**pdf**) for T^+ .

Table 10.5: Possible sign and rank combinations for Example 10.2

-1	+1	-1	+1	-1	+1	-1	+1
-2	-2	+2	+2	-2	-2	+2	+2
-3	-3	-3	-3	+3	+3	+3	+3
t^+	0	1	2	3	3	4	5
							6

Table 10.6: PDF of T^+ for Example 10.2

t^+	$\mathbb{P}(T^+ = t^+)$
0	$1/8$
1	$1/8$
2	$1/8$
3	$2/8$
4	$1/8$
5	$1/8$
6	$1/8$



R can compute quantiles (`qsignrank()`), the density function (`dsignrank()`), the distribution function (`psignrank()`), and random numbers (`rsignrank()`) from the Wilcoxon

signed-rank distribution. For example, the probabilities in Table 10.6 on the facing page can be generated with `dsignrank(0:6, 3)`. To obtain further help, type `?dsignrank` at the R prompt.

Due to the discrete nature of T^+ , it is generally not possible to define a rejection region that results in a test whose size is exactly equal to the prescribed α . Consequently, the approach presented for this test relies on ϕ -values rather than on defining rejection regions for the statistical conclusion. The three possible alternative hypotheses and their associated ϕ -value calculation formulas are presented in Table 10.7 on page 599. The ϕ -value formulas given in Table 10.7 can be used to calculate exact ϕ -values with the `psignrank()` function when there are no ties among the non-zero $|d_i|$ s. In the presence of ties, the function `wilcox.test()` uses Table 10.9 on page 604 with a correction factor. The formulas in Table 10.7 on page 599 are still valid when there are ties in the non-zero $|d_i|$ s; however, the exact conditional distribution of T^+ when ties are present is not a base function of R. Example 10.3 shows how R can be used to compute the exact ϕ -value for the conditional distribution of T^+ (the distribution of T^+ with ties in the non-zero $|d_i|$ s).

Example 10.3 ▷ Wilcoxon Signed-Rank Test: Pool pH ◁ A lifeguard is told to maintain the pH of a 50-meter pool at 7.25. He takes pH measurements at each of the four corners of the pool and gets 7.2, 7.3, 7.3, and 7.4. Calculate the ϕ -value for testing the hypothesis that the median pH is greater than 7.25 using the exact conditional distribution for T^+ .

Solution: If the data are symmetric, a Wilcoxon signed-rank test may be appropriate. A visual inspection of the pH measurements reveals they are symmetric around 7.3. The creation of a density plot to verify this assumption is left to the reader. The steps for carrying out the Wilcoxon signed-rank test are, first, to create the d_i values that equal $x_i - \psi_0$ for $i = 1, \dots, n$. Next, take the absolute value of the n d_i s and assign the ranks to the n values. If there are ties, use the **midranks**. Then, multiply the values of the ranks of the $|d_i|$ s by the sign of the original d_i s. Finally, sum the resulting positive quantities to obtain t^+ .

```
> PH <- c(7.2, 7.3, 7.3, 7.4)          # Enter data
> DIFF <- PH - 7.25                     # Create differences (DIFF)
> absD <- abs(DIFF)                    # Absolute value of DIFF (absD)
> rankabsD <- rank(absD)              # Rank the absD values
> signD <- sign(DIFF)                 # Store the signs of DIFF
> signrank <- rankabsD*signD        # Create a vector of signed ranks
> tp <- sum(signrank[signrank>0])    # Calculate t+
> tp
[1] 8
```

After t^+ is calculated, the distribution of T^+ must be enumerated to find the ϕ -value.

```
> n <- length(DIFF)
> signs <- as.matrix(expand.grid(d1 = 0:1, d2 = 0:1, d3 = 0:1, d4 = 0:1))
> signs                                     # 1s represent positive ranks
      d1 d2 d3 d4
[1,]  0  0  0  0
[2,]  1  0  0  0
[3,]  0  1  0  0
[4,]  1  1  0  0
```

```
[5,] 0 0 1 0
[6,] 1 0 1 0
[7,] 0 1 1 0
[8,] 1 1 1 0
[9,] 0 0 0 1
[10,] 1 0 0 1
[11,] 0 1 0 1
[12,] 1 1 0 1
[13,] 0 0 1 1
[14,] 1 0 1 1
[15,] 0 1 1 1
[16,] 1 1 1 1

> mat <- matrix(rankabsD)           # Put rankabsD in matrix form
> mat

[,1]
[1,] 2
[2,] 2
[3,] 2
[4,] 4
```

After the matrix listing the locations of the positive ranks with 1s and the locations of negative ranks with 0s is created (**signs**), matrix multiplication is used to sum the positive ranks to get the distribution of T^+ , where **mat** contains the ranks of the absolute values of the d_i s:

```
> Tp <- signs%*%mat             # (16X4)*(4X1) = 16X1 vector of T+
> Tp <- sort(Tp)               # Sort the distribution of T+
> SampDist <- table(Tp)/2^n    # used for fractions()
> library(MASS)                # Sampling distribution of T+
> fractions(SampDist)

Tp
0      2      4      6      8      10
1/16  3/16  1/4   1/4  3/16  1/16
```

Since H_1 is an upper one-sided hypothesis, the ϕ -value is the sum of the values of the distribution of T^+ that are greater than or equal to the value of our test statistic t^+ . In this case, the t^+ was 8, so the ϕ -value is

$$\phi\text{-value} = \mathbb{P}(T^+ = 8) + \mathbb{P}(T^+ = 10) = 3/16 + 1/16 = 1/4.$$

```
> pvalue <- sum(Tp>=tp)/2^n     # Calculate p-value
> fractions(pvalue)

[1] 1/4
```

This ϕ -value can also be found using the function **wilcoxe.test()** from the PASWR2 package. Note that the function **wilcox.test()** cannot be used because it cannot compute exact ϕ -values when there are ties in the data.

```
> wilcoxon.test(PH, mu = 7.25, alternative = "greater")
```

```
Wilcoxon Signed Rank Test

data: PH
t+ = 8, p-value = 0.25
alternative hypothesis: true median is greater than 7.25
93.75 percent confidence interval:
 7.25 Inf
sample estimates:
(pseudo)median
    7.3
```



The Wilcoxon signed-rank test may also be used for testing whether the median difference (ψ_0) between two dependent populations (X and Y) is equal to some value, $H_0 : \psi_D = \psi_0$. For dependent samples, $D_i = X_i - Y_i - \psi_0$ instead of $D_i = X_i - \psi_0$. The computation of T^+ for dependent samples follows the same steps as those for a single sample.

Table 10.7: Summary for testing the median — Wilcoxon signed-rank test

Null Hypothesis — $H_0 : \psi = \psi_0$

Test Statistic's Value	$t^+ = \text{sum of the positive ranked differences}$
---------------------------	---

Alternative Hypothesis	p-Value Formula
$H_1 : \psi < \psi_0$	$\mathbb{P}(T^+ \leq t^+ H_0)$
$H_1 : \psi > \psi_0$	$\mathbb{P}(T^+ \geq t^+ H_0) = 1 - \mathbb{P}(T^+ \leq t^+ - 1 H_0)$
$H_1 : \psi \neq \psi_0$	$2 \times \min \{\mathbb{P}(T^+ \leq t^+), 1 - \mathbb{P}(T^+ \leq t^+ - 1), 0.5\}$

10.3.1 Confidence Interval for ψ Based on the Wilcoxon Signed-Rank Test

Since $\{X_1, X_2, \dots, X_n\}$ are random variables from a symmetric distribution with median ψ , the pairwise averages $\bar{x}_{ij} = \frac{x_i + x_j}{2}$, where $1 \leq i \leq j \leq n$, are also symmetrically distributed about the median ψ . There are a total of $n(n+1)/2$ of these \bar{x}_{ij} s, frequently called the **Walsh averages**. The $n(n+1)/2$ Walsh averages can be split into $\binom{n}{2}$ means, \bar{x}_{ij} , where $i \neq j$ and n means, \bar{x}_{ii} for $i = 1, \dots, n$. When the Walsh averages are ordered from smallest to largest, the k^{th} and $(\frac{n(n+1)}{2} - k + 1)^{\text{st}}$ order statistics are the lower and upper endpoints of a confidence interval with a confidence level of at least $(1 - \alpha) \cdot 100\%$, where k is the largest value such that $\mathbb{P}(T^+ < k) \leq \alpha/2$. For a one-sided confidence interval,

k is the largest value such that $\mathbb{P}(T^+ < k) \leq \alpha$. Again, k is a positive integer since it is the subscript of an order statistic. The exact confidence level is $1 - 2\mathbb{P}(T^+ < k)$ for a two-sided confidence interval and $1 - \mathbb{P}(T^+ < k)$ for a one-sided confidence interval.

When there are no d_i s ($x_i - \psi_0$) that equal zero, as well as no x_i s that equal zero, testing $H_0 : \psi = \psi_0$ with the procedures described in Section 10.3 yields an equivalent acceptance region to that produced by the confidence interval based on the Walsh averages. If this is not the case, the regions are no longer equivalent.

For the dependent case, use the $n(n+1)/2$ dependent Walsh averages $\overline{x - y}_{ij} = [(x_i - y_i) + (x_j - y_j)]/2$. In this case, $d_i = x_i - y_i - \psi_0$. Here the equivalence between the acceptance region of the hypothesis test and the confidence interval created based on the Walsh averages exists only when $d_i \neq 0$ and $x_i - y_i \neq 0$, $i = 1, \dots, n$.

Example 10.4 ▷ Wilcoxon Signed-Rank Test: Waiting Times ◁ A statistician records how long he must wait for his bus each morning. This information is recorded in Table 10.8 and in the data frame **WAIT**.

- Test to see if his median waiting time is less than 6 minutes.
- Compute a lower 95% confidence interval for the upper bound of the median, ψ .

Table 10.8: Waiting times in minutes (**WAIT**)

x_i	$d_i = x_i - 6$	$ d_i $	sign(d_i)	rank	$ d_i $	signed ranks
8.0	2.0	2.0	+	6		6
2.1	-3.9	3.9	-	12		-12
3.8	-2.2	2.2	-	7		-7
8.6	2.6	2.6	+	8		8
7.3	1.3	1.3	+	4		4
6.1	0.1	0.1	+	1		1
1.4	-4.6	4.6	-	13		-13
2.9	-3.1	3.1	-	10		-10
5.5	-0.5	0.5	-	2		-2
2.7	-3.3	3.3	-	11		-11
4.8	-1.2	1.2	-	3		-3
4.6	-1.4	1.4	-	5		-5
1.0	-5.0	5.0	-	14		-14
8.7	2.7	2.7	+	9		9
0.8	-5.2	5.2	-	15		-15
						$t^+ = \mathbf{28}$

Solution: Before using the Wilcoxon signed-rank test, a check on the assumption of symmetry is made with a density plot in Figure 10.3 on the facing page. Since the density

plot does not appear overly skewed given the relatively small sample size, one may proceed cautiously with a Wilcoxon signed-rank test.

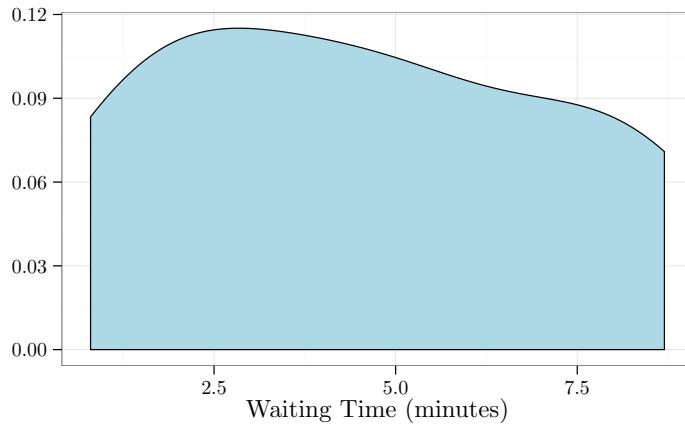


FIGURE 10.3: Density plot of bus waiting times in minutes

(a) Use the five-step procedure to test if the median waiting time is less than 6 minutes.

Step 1: Hypotheses — The null and alternative hypotheses to test if the median waiting time is less than 6 minutes are

$$H_0 : \psi = 6 \text{ versus } H_1 : \psi < 6.$$

Step 2: Test Statistic — The test statistic chosen is T^+ , where T^+ is the Wilcoxon signed-rank statistic. Here, the observed value of T^+ is $t^+ = 28$.

Step 3: Rejection Region Calculations — Rejection is based on the φ -value, so none are required.

Step 4: Statistical Conclusion — The φ -value is $\mathbb{P}(T^+ \leq 28) = 0.0365$, which can be obtained by typing `psignrank(28, 15)`.

Reject H_0 .

Step 5: English Conclusion — There is sufficient evidence to suggest the median waiting time is less than 6 minutes.

The function `wilcox.test()` is used to perform the test in R Code 10.2.

R Code 10.2

```
> with(data = WAIT, wilcox.test(minutes, mu = 6, alternative = "less"))

Wilcoxon signed rank test

data: minutes
V = 28, p-value = 0.0365
alternative hypothesis: true location is less than 6
```

Note that `wilcox.test()` denotes the statistic t^+ with a V .

- (b) To compute an upper 95% confidence interval for ψ , first determine the $n(n + 1)/2 = 15(15 + 1)/2 = 120$ Walsh averages. To ease the drudgery of 120 calculations of means, R is used. Recall that the 120 Walsh averages are composed of $\binom{15}{2} = 105$ means, \bar{x}_{ij} , where $i \neq j$ and 15 means, \bar{x}_{ii} for $i = 1, \dots, 15$.

```
> n2means <- apply(combn(WAIT$minutes, 2), 2, mean) # n choose 2 means
> WalshAverages <- c(WAIT$minutes, n2means)
```

Next, find the largest value k such that $\mathbb{P}(T^+ < k) \leq 0.05$. This can be accomplished in two ways:

- (1) Type `qsignrank(α , n)` into R:

```
> qsignrank(0.05, 15)
[1] 31
```

- (2) Visually inspect `psignrank(0:n*(n + 1)/2, n)` for the largest value k such that $\mathbb{P}(T^+ < k) \leq \alpha$. Note that the first pair $(k - 1, \mathbb{P}(T^+ < k))$ of the output shown is $(28, 0.0365)$, and the pair that gives the answer is $(30, 0.0473)$, which implies $k - 1 = 30$ or $k = 31$.

```
> psi <- psignrank(28:33, 15)
> names(psi) <- 28:33
> psi
```

28	29	30	31	32	33
0.03649902	0.04162598	0.04730225	0.05349731	0.06027222	0.06768799

While either (1) or (2) can be used to determine k , only (1) allows one to compute the exact confidence level ($1 - \text{psignrank}(30, 15) = 0.9527$) for a lower one-sided confidence interval on the upper bound. The 95.27% confidence interval where $k = 31$ is then

$$\left(-\infty, \bar{x}_{\left(\frac{n(n+1)}{2}-k+1\right)}\right] = \left(-\infty, \bar{x}_{(90)}\right] = (-\infty, 5.8].$$

```
> SWA <- sort(WalshAverages) # sort values
> SWA[90]
[1] 5.8
```

This may be done directly with the argument `conf.int = TRUE` in the `wilcox.test()` function if one is using R:

```
> with(data = WAIT,
+       wilcox.test(minutes, mu = 6, alternative = "less", conf.int = TRUE)
+     )
```

```
Wilcoxon signed rank test
```

```

data: minutes
V = 28, p-value = 0.0365
alternative hypothesis: true location is less than 6
95 percent confidence interval:
-Inf 5.8
sample estimates:
(pseudo)median
4.625

```

The answer can also be computed with the function `wilcoxe.test()` from the PASWR2 package:

```

> with(data = WAIT,
+       wilcoxe.test(minutes, mu = 6, alternative = "less")
+     )

Wilcoxon Signed Rank Test

data: minutes
t+ = 28, p-value = 0.0365
alternative hypothesis: true median is less than 6
95.26978 percent confidence interval:
-Inf 5.8
sample estimates:
(pseudo)median
4.625

```



10.3.2 Normal Approximation to the Wilcoxon Signed-Rank Test

For moderately sized samples ($n > 15$), the sampling distribution of T^+ can be reasonably approximated with the normal distribution that has a mean and standard deviation $n(n + 1)/4$ and $\sqrt{n(n + 1)(2n + 1)/24}$, respectively. That is, $T^+ \rightsquigarrow N(n(n + 1)/4, \sqrt{n(n + 1)(2n + 1)/24})$. The standardized test statistic under the assumption that $H_0 : \psi = \psi_0$ is true is

$$Z = \frac{T^+ - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}} \rightsquigarrow N(0, 1). \quad (10.4)$$

See Figure 10.4 on the following page for a graph of the Wilcoxon signed-rank distribution for $n = 15$ superimposed by a normal distribution with $\mu = n(n + 1)/4 = 60$ and $\sigma = \sqrt{n(n + 1)(2n + 1)/24} = 17.6068$.

The formula for calculating the observed value of the standardized test statistic as well as the three possible alternative hypotheses and their rejection regions are described in Table 10.9 on the next page. If there are ties in the $|d_i|$ s, the variance of T^+ is reduced to

$$\frac{n(n+1)(2n+1)}{24} - \frac{\sum_{j=1}^g t_j(t_j - 1)(t_j + 1)}{48} \quad (10.5)$$

where g denotes the number of tied groups of non-zero $|d_i|$ s and t_j is the size of tied group j . In (10.5), an untied observation is considered to be a tied group of size one. In the event

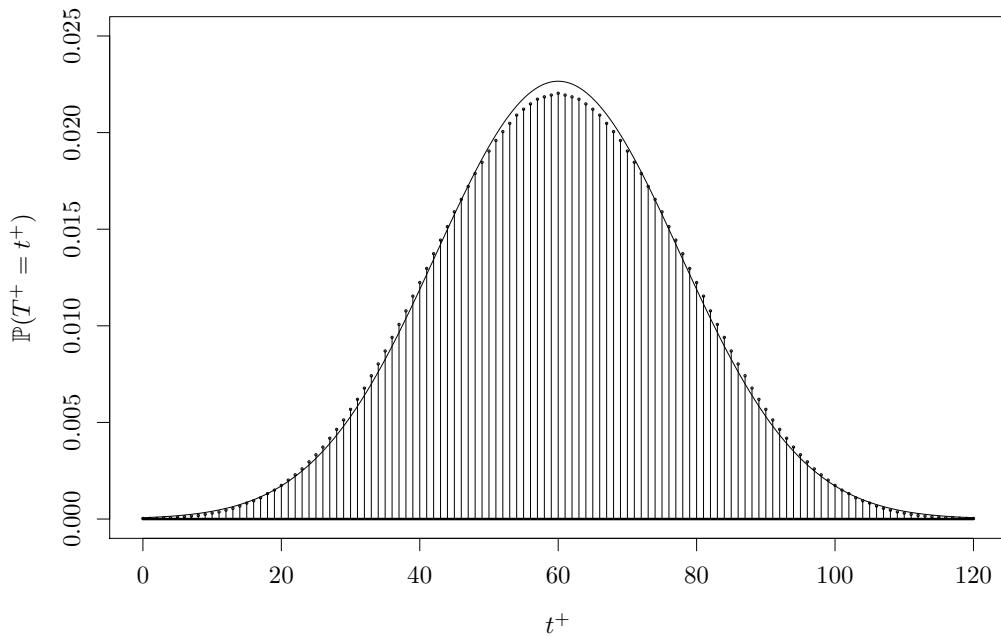


FIGURE 10.4: Graphical representation of the Wilcoxon signed-rank distribution for $n = 15$ superimposed by a normal distribution with $\mu = n(n + 1)/4 = 60$ and $\sigma = \sqrt{n(n + 1)(2n + 1)/24} = 17.6068$

Table 10.9: Summary for testing the median — normal approximation to the Wilcoxon signed-rank test

Null Hypothesis — $H_0 : \psi = \psi_0$

$$\text{Standardized Test Statistic's Value} — z_{obs} = \frac{t^+ \pm 0.5 - n(n+1)/4}{\sqrt{n(n+1)(2n+1)/24 - CF}}$$

$$\text{Correction Factor} — CF = \sum_{j=1}^g t_j(t_j - 1)(t_j + 1)/48$$

Alternative Hypothesis	Rejection Region
$H_1 : \psi < \psi_0$	$z_{obs} < z_\alpha$
$H_1 : \psi > \psi_0$	$z_{obs} > z_{1-\alpha}$
$H_1 : \psi \neq \psi_0$	$ z_{obs} > z_{1-\alpha/2}$

Note: The quantity ± 0.5 in the numerator of z_{obs} is the continuity correction. When $H_1 : \psi < \psi_0$, the quantity $+0.5$ is used. When $H_1 : \psi > \psi_0$, the quantity -0.5 is used. When $H_1 : \psi \neq \psi_0$, use $+0.5$ if $t^+ < n(n + 1)/4$ and -0.5 if $t^+ > n(n + 1)/4$.

that no ties exist, $g = n$ and $t_j = 1$ for $j = 1, \dots, n$, which produces a correction factor of zero.

A corresponding two-sided confidence interval for the median based on (10.4) are the k^{th} and $(n(n+1)/2 - k + 1)^{\text{st}}$ ordered Walsh averages, where

$$k = 0.5 + \frac{n(n+1)}{4} + z_{\alpha/2} \sqrt{\frac{n(n+1)(2n+1)}{24}}. \quad (10.6)$$

For a one-sided confidence interval, replace $z_{\alpha/2}$ with z_{α} . Since k is generally not an integer, it can be either rounded or truncated. To obtain a conservative estimate, one should truncate.

Example 10.5 ▷ Wilcoxon Signed-Rank Test: TV Effects ◁ Gibbons (1997) provides the following data regarding aggressive behavior in relation to exposure to violent television programs with the following exposition:

... a group of children are matched as well as possible as regards home environment, genetic factors, intelligence, parental attitudes, and so forth, in an effort to minimize factors other than TV that might influence a tendency for aggressive behavior. In each of the resulting 16 pairs, one child is randomly selected to view the most violent shows on TV, while the other watches cartoons, situation comedies, and the like. The children are then subjected to a series of tests designed to produce an ordinal measure of their aggression factors. (pages 143–144)

The data that were collected are presented in Table 10.10 on the following page and stored in data frame **AGGRESSION**, where x_i represents aggression test scores for the children who watched violent programming (**violence**) and y_i represents aggression test scores for the children who watched non-violent television programs (**noviolence**).

- (a) Confirm that the distribution is symmetric.
- (b) Test whether the median difference (**violent - noviolence**) for aggression test scores for pairs of children is greater than zero using a significance level of $\alpha = 0.05$ with the normal approximation to the Wilcoxon signed-rank test.
- (c) Use the function **wilcox.test()** to report the exact p -value and the upper one-sided confidence interval for the hypothesis in (b).
- (d) Construct an upper one-sided confidence interval with confidence level of at least 95% using the normal approximation to find k .

Solution: The answers are as follows:

- (a) Before using the Wilcoxon signed-rank test, a quick check on the assumption of symmetry is made with a density plot in Figure 10.5 on the next page. Since the density plot appears relatively symmetric, it is legitimate to proceed with a Wilcoxon signed-rank test.
- (b) Use the five-step procedure to test if the median difference for aggression scores for pairs of children is greater than zero.

Step 1: Hypotheses — The null and alternative hypotheses to test if the median difference for aggression scores for pairs of children is greater than zero are

$$H_0 : \psi_D = 0 \text{ versus } H_1 : \psi_D > 0.$$

Table 10.10: Aggression test scores (**AGGRESSION**)

Pair	x_i	y_i	$d_i = x_i - y_i$	$ d_i $	$\text{sign}(d_i)$	rank $ d_i $	signed ranks
1	35	26	9	9	+	12.5	12.5
2	30	28	2	2	+	4.5	4.5
3	15	16	-1	1	-	1.5	-1.5
4	20	16	4	4	+	8	8
5	25	16	9	9	+	12.5	12.5
6	14	16	-2	2	-	4.5	-4.5
7	37	32	5	5	+	9	9
8	26	24	2	2	+	4.5	4.5
9	36	30	6	6	+	10	10
10	40	33	7	7	+	11	11
11	35	20	15	15	+	16	16
12	20	19	1	1	+	1.5	1.5
13	16	19	-3	3	-	7	-7
14	21	10	11	11	+	15	15
15	17	7	10	10	+	14	14
16	15	17	-2	2	-	4.5	-4.5

$$t^+ = \mathbf{118.5}$$

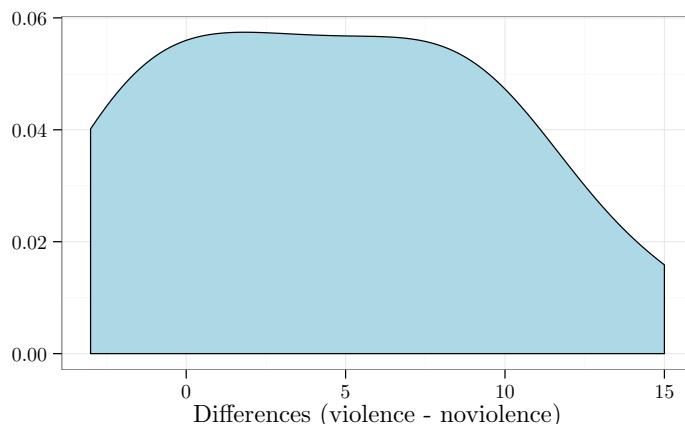


FIGURE 10.5: Density plot of differences of aggression scores

Step 2: **Test Statistic** — The test statistic chosen is T^+ , where T^+ is the Wilcoxon signed-rank statistic. Here, the observed value of T^+ is $t^+ = 118.5$. (See Table 10.10.)

Step 3: **Rejection Region Calculations** — Because the standardized test statistic is distributed approximately $N(0, 1)$ and H_1 is an upper one-sided hypothesis, the rejection region is $z_{obs} > z_{1-0.05} = 1.6449$. Because there are three groups of ties

($g = 3$) where the sizes of the tied groups are 2, 4, and 2, the correction factor is

$$\begin{aligned} CF &= \sum_{j=1}^3 t_j(t_j - 1)(t_j + 1)/48 \\ &= [2(2 - 1)(2 + 1) + 4(4 - 1)(4 + 1) + 2(2 - 1)(2 + 1)] / 48 \\ &= [6 + 60 + 6] / 48 \\ &= 72 / 48 = 3/2. \end{aligned}$$

The value of the standardized test statistic is

$$\begin{aligned} z_{obs} &= \frac{t^+ \pm 0.5 - n(n + 1)/4}{\sqrt{n(n + 1)(2n + 1)/24 - CF}} \\ &= \frac{118.5 - 0.5 - 16(16 + 1)/4}{\sqrt{16(16 + 1)(2(16) + 1)/24 - 3/2}} \\ &= 2.5906. \end{aligned}$$

Step 4: Statistical Conclusion — The ϕ -value is $\mathbb{P}(Z \geq 2.5906) = 0.0048$.

- I. From the rejection region, reject H_0 because $z_{obs} = 2.5906$ is more than 1.6449.
- II. From the ϕ -value, reject H_0 because the ϕ -value = 0.0048 is less than 0.05.

Reject H_0 .

Step 5: English Conclusion — There is sufficient evidence to suggest that children who view violent television programs have higher aggression test scores than children who view non-violent television programs.

R Code 10.3 uses the `wilcox.test()` to compute the requested test. Note that `wilcox.test()` automatically uses a normal approximation to the distribution of T^+ when there are ties in the $|d_i|$ s as well as a correction factor for the variance of T^+ .

R Code 10.3

```
> with(data = AGGRESSION, wilcox.test(violence, noviolence,
+   paired = TRUE, alternative = "greater")
+ )

Wilcoxon signed rank test with continuity correction

data: violence and noviolence
V = 118.5, p-value = 0.00479
alternative hypothesis: true location shift is greater than 0
```

(c) From the output of `wilcoxe.exact()` in R Code 10.4, the ϕ -value is 0.003265 and the upper 95.21% confidence interval for the lower bound on the median is $[2, \infty)$.

R Code 10.4

```
> with(data = AGGRESSION, wilcoxe.test(violence, noviolence,
+   paired = TRUE, alternative = "greater")
+ )
```

```
Wilcoxon Signed Rank Test (Dependent Samples)

data: violence and noviolence
t+ = 118.5, p-value = 0.003265
alternative hypothesis: true median difference is greater than 0
95.20569 percent confidence interval:
 2 Inf
sample estimates:
(pseudo)median
        4.5
```

- (d) The paired differences are stored in `PD` and the sorted Walsh averages are in `SWA`. Using (10.6), k is calculated to be 36, and the k^{th} Walsh average is determined to be 2. Therefore, the 95% confidence interval for ψ_D is $[2, \infty)$.

```
> PD <- AGGRESSION$violence - AGGRESSION$noviolence
> n2means <- apply(combn(PD, 2), 2, mean) # n choose 2 means
> SWA <- sort(c(PD, n2means)) # Sorted Walsh averages
> n <- length(PD)
> k <- 0.5 + n * (n + 1)/4 + qnorm(0.05) * sqrt(n * (n + 1) *
+      (2 * n + 1)/24)
> k <- floor(k)
> k
[1] 36

> SWA[k] # kth Walsh average
[1] 2
```

Another way to achieve the same result is with the function `outer()`, which applies the third argument ("`+`") to the first two vectors in an element-wise manner to create an array, and `!lower.tri`, which returns the values of the upper triangular matrix containing double the Walsh averages. Finally, the upper triangular matrix is divided by two, and then sorted to calculate the values for the sorted Walsh averages.

```
> ADD <- outer(PD, PD, "+")
> SWA2 <- sort(ADD[!lower.tri(ADD)])/2
> SWA2[k] # kth Walsh average
[1] 2
```

10.4 The Wilcoxon Rank-Sum or the Mann-Whitney U -Test

The **Wilcoxon rank-sum** test is due to Wilcoxon (1945). Its widespread use is due in large part to Mann and Whitney, who proposed another test, the **Mann-Whitney U -test**, which is equivalent to the Wilcoxon rank-sum test (Gibbons and Chakraborti, 2003). Be

aware that many combinations of names with either some or all of Mann, Whitney, and Wilcoxon are all typically referring to some variation of the same test. The two-sample Wilcoxon rank-sum test assumes that data come from two independent random samples X_1, X_2, \dots, X_n and Y_1, Y_2, \dots, Y_m of sizes n and m , respectively, where the underlying distributions of X_1, X_2, \dots, X_n and Y_1, Y_2, \dots, Y_m have the same shape. Note that the assumption of identical underlying shapes implies that the variances are also equal. No further assumptions other than continuous data, which is at least on an ordinal scale, are made with the two-sample Wilcoxon rank-sum test. Because the underlying distributions of X and Y are assumed to be identical in the null hypothesis, this test can apply to means, medians, or any other quantile.

If two random samples of size n and m are drawn from two identical populations, all $N = n + m$ observations can be regarded as a single sample from some common population. Further, if the N observations are ordered in a single sequence according to relative magnitude, one expects the X s and Y s to be well mixed in the ordered sequence that represents the sample data. That is, an arrangement of the data where most of the X s are smaller than the Y s, or vice versa, would suggest two distinct populations and not one common population. The Wilcoxon rank-sum statistic, W , is computed by

1. Forming a single sample of all $n + m$ observations.
2. Assigning ranks to the combined sample.
3. Summing the X ranks in the combined sample.

Provided the null hypothesis of identical populations is true, typically denoted $H_0 : F_X(x) = F_Y(x)$ for all x , all $\binom{N}{n}$ assignments of the X ranks are equally likely, each having probability $1/\binom{N}{n}$. Consequently, W values that are either too small or too large will cause the null hypothesis to be rejected. W takes on integer values ranging from $n(n + 1)/2$ to $n(2N - n + 1)/2$ when no ties are present in the ranks. The sampling distribution of W , known as the **Wilcoxon rank-sum** distribution, is symmetric about its mean value $n(N + 1)/2$ and has a variance of $nm(N + 1)/12$.

Due to the discrete nature of W , it is generally not possible to define a rejection region that results in a test whose size is exactly equal to the prescribed α . Consequently, the approach presented for this test relies on ϕ -values rather than on defining rejection regions for the statistical conclusion. The three possible alternative hypotheses and their associated ϕ -value calculation formulas are presented in Table 10.11 on the next page.

A closely related statistic to W proposed by Mann and Whitney, typically denoted by U , is defined as the total number of times the pair (x_i, y_j) contains an x value greater than the y value for all (i, j) . The relationship between W and U can be expressed as $U = W - n(n + 1)/2$, and generalized to include tied ranks by assigning 1/2 to all ties.

The ϕ -value formulas given in Table 10.11 can be used to calculate exact ϕ -values with R's `pwilcox()` function when there are no ties among the ranks. The R definition of the Wilcoxon rank-sum distribution corresponds to the distribution of U . The three possible alternative hypotheses and their associated ϕ -value calculation formulas for the U statistic are presented in Table 10.12 on the following page. In the presence of ties, the function `wilcox.test()` uses Table 10.14 on page 618 with a correction factor. The formulas in Table 10.11 on the following page are still valid when there are ties in the ranks; however, the exact conditional distribution of W when ties are present is not readily available in R. Example 10.7 on page 611 shows how R can be used to compute the exact ϕ -value for the conditional distribution of W (the distribution of W with ties in the ranks).

Example 10.6 ▷ *W and U Sampling Distributions* ◁ Assume the values $\mathbf{x} = \{2, 5\}$

Table 10.11: Summary for testing equality of medians — Wilcoxon rank-sum test

Null Hypothesis — $H_0 : \psi_X - \psi_Y = \delta_0$

Test Statistic's Value — w = sum of the ranked xs in the combined sample

Alternative Hypothesis	ϕ -Value Formula
$H_1 : \psi_X - \psi_Y < \delta_0$	$\mathbb{P}(W \leq w H_0)$
$H_1 : \psi_X - \psi_Y > \delta_0$	$\mathbb{P}(W \geq w H_0) = 1 - \mathbb{P}(W \leq w - 1 H_0)$
$H_1 : \psi_X - \psi_Y \neq \delta_0$	$2 \times \min \{\mathbb{P}(W \leq w), 1 - \mathbb{P}(W \leq w - 1), 0.5\}$

Table 10.12: Summary for testing equality of medians — Mann-Whitney U -test

Null Hypothesis — $H_0 : \psi_X - \psi_Y = \delta_0$

Test Statistic's Value — u = number of times x exceeds y
in the pairs (x_i, y_j) for all (i, j)

Alternative Hypothesis	ϕ -Value Formula
$H_1 : \psi_X - \psi_Y < \delta_0$	$\mathbb{P}(U \leq u H_0)$
$H_1 : \psi_X - \psi_Y > \delta_0$	$\mathbb{P}(U \geq u H_0) = 1 - \mathbb{P}(U \leq u - 1 H_0)$
$H_1 : \psi_X - \psi_Y \neq \delta_0$	$2 \times \min \{\mathbb{P}(U \leq u), 1 - \mathbb{P}(U \leq u - 1), 0.5\}$

and $\mathbf{y} = \{9, 12, 14\}$ are two independent random samples from independent distributions that are assumed to be equal in shape. Enumerate the sampling distributions of W and U .

Solution: Start by reading the values of \mathbf{x} and \mathbf{y} into vectors labeled \mathbf{x} and \mathbf{y} , respectively:

```
> x <- c(2, 5)
> y <- c(9, 12, 14)
> n <- length(x)
> m <- length(y)
> N <- n + m
> r <- rank(c(x, y))
> w <- sum(r[seq(along = x)]) # observed w value
> u <- sum(r[seq(along = x)]) - n*(n + 1)/2 # observed u value
> val <- combn(r, n) # possible rankings for X
> W <- apply(val, 2, sum) # W values
> U <- W - n*(n + 1)/2 # U values
> display <- rbind(val, W, U) # X rankings with W and U
> display
```

[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
1	1	1	1	2	2	2	3	3	4
2	3	4	5	3	4	5	4	5	5
W	3	4	5	6	5	6	7	7	8
U	0	1	2	3	2	3	4	4	5

Note that the values of W are between $n(n+1)/2 = 2(2+1)/2 = 3$ and $n(2N-n+1)/2 = 2[(2)(5)-2+1]/2 = 9$.

```
> xtabs(~W)/choose(5, 2) # Sampling distribution of W

W
 3   4   5   6   7   8   9
0.1 0.1 0.2 0.2 0.2 0.1 0.1
```

Note that the values of U are between 0 and $n \cdot m = 6$.

```
> xtabs(~U)/choose(5, 2) # Sampling distribution of U

U
 0   1   2   3   4   5   6
0.1 0.1 0.2 0.2 0.2 0.1 0.1

> dwilcox(0:6, 2, 3) # Produces distribution of U in R

[1] 0.1 0.1 0.2 0.2 0.2 0.1 0.1
```



Example 10.7 ▷ **Wilcoxon Rank-Sum φ -Value: Pool pH** ◁ Lifeguards are told to maintain the pH of a 50-meter pool at 7.25. The pool manager takes pH measurements at each of the four corners of the pool before the pool opens on two consecutive days. Calculate the φ -value for testing the hypothesis that the difference in median pH readings is zero using the exact conditional distribution of W assuming the pH samples on Day 1 and Day 2 are independent. (The same underlying distribution assumption can be verified graphically.) The pH readings are Day 1 (x): {7.2, 7.2, 7.3, 7.3} and Day 2 (y): {7.3, 7.3, 7.4, 7.4}.

Solution: Use R to calculate the φ -values:

```
> x <- c(7.2, 7.2, 7.3, 7.3)
> y <- c(7.3, 7.3, 7.4, 7.4)
> n <- length(x)
> m <- length(y)
> N <- n + m
> r <- rank(c(x, y))
> w <- sum(r[seq(along = x)]) # observed w value
> w

[1] 12

> val <- combn(r, n) # possible rankings
> W <- apply(val, 2, sum) # W values
> xtabs(~W)/choose(8, 4)
```

```
W
12      15      18      21      24
0.08571429 0.22857143 0.37142857 0.22857143 0.08571429
```

Since H_1 is a two-sided hypothesis, the p -value is

$$2 \times \min\{\mathbb{P}(W \leq w), 1 - \mathbb{P}(W \leq w - 1), 0.5\}.$$

In this case, w was 12, so $\mathbb{P}(W \leq 12) = 0.0857$ and $1 - \mathbb{P}(W \leq 11) = 1$. It follows that the p -value is $2 \times 0.0857 = 0.1714$.

```
> pvalue <- 2 * (sum(W <= w)/choose(N, n))
> pvalue
[1] 0.1714286

> # OR
> pvalue1 <- 2 * mean(W <= w)
> pvalue1
[1] 0.1714286
```

The output from the `wilcoxon.test()` function is

```
> wilcoxon.test(x, y)

Wilcoxon Rank Sum Test

data: x and y
w = 12, p-value = 0.1714
alternative hypothesis: true median is not equal to 0
82.85714 percent confidence interval:
-0.2 0.0
sample estimates:
difference in location
-0.1
```

10.4.1 Confidence Interval Based on the Mann-Whitney U-Test

A confidence interval with a confidence level of at least $(1 - \alpha) \cdot 100\%$ for the shift Δ from population X , from which a sample $x_i, i = 1, \dots, n$ is taken, to population Y , from which a sample $y_j, j = 1, \dots, m$ is taken, can be constructed by using the k^{th} and $(nm - k + 1)^{\text{st}}$ order statistics from the nm differences $x_i - y_j$ where k is the largest value such that $\mathbb{P}(U < k) \leq \alpha/2$. For a one-sided confidence interval, k is the largest value such that $\mathbb{P}(U < k) \leq \alpha$. The exact confidence level is given by $1 - 2 \times \mathbb{P}(U < k)$ for a two-sided confidence interval and $1 - \mathbb{P}(U < k)$ for a one-sided confidence interval. Clearly, k must be a positive integer since it is the subscript of an order statistic.

Example 10.8 ▷ *Confidence Interval for Difference in Medians* ◁ Eight spring piglets are randomly assigned to two different groups and are fed two different diets (A and B). After four weeks, the weight gains in pounds for the piglets eating each diet

are recorded. Find a 90% confidence interval for the median difference in weight gains for the piglets eating each diet. Be sure to verify the assumption of identical underlying distributions except for a shift that is required for constructing the confidence interval.

A:	1.2	1.5	2.3	4.3
B:	4.5	5.7	6.1	8.6

Solution: To verify that the distributions of the piglet weights follow the same distribution other than a shift, density plots as well as side-by-side boxplots are constructed. R Code 10.5 can be used to produce graphs similar to those shown in Figure 10.6 on the next page. Based on the graphs in Figure 10.6, it seems reasonable to assume that the underlying distributions are similar in shape.

R Code 10.5

```
> A <- c(1.2, 1.5, 2.3, 4.3)
> B <- c(4.5, 5.7, 6.1, 8.6)
> DF <- stack(data.frame(A, B))
> ggplot(data = DF, aes(x = values)) +
+   geom_density(aes(fill = ind), alpha = 0.5) +
+   theme_bw() +
+   labs(y = "", x = "weight gain (pounds)") +
+   guides(fill = guide_legend("Diet \nType")) +
+   scale_fill_grey()
> ggplot(data = DF, aes(x = ind, y = values, fill = ind)) +
+   geom_boxplot() +
+   coord_flip() +
+   theme_bw() +
+   labs(x = "", y = "weight gain (pounds)") +
+   guides(fill = guide_legend("Diet \nType")) +
+   scale_fill_grey(start = 0.3, end = 0.7)
```

To find the largest k such that $\mathbb{P}(U < k) \leq \alpha/2$, use the command `pwilcox()` as shown in R Code 10.6.

R Code 10.6

```
> n <- length(A)
> m <- length(B)
> pwilcox(1:(n * m), n, m)

[1] 0.02857143 0.05714286 0.10000000 0.17142857 0.24285714 0.34285714
[7] 0.44285714 0.55714286 0.65714286 0.75714286 0.82857143 0.90000000
[13] 0.94285714 0.97142857 0.98571429 1.00000000
```

By visual inspection, one realizes the largest value k such that $\mathbb{P}(U < k) \leq 0.05$ is $k = 2$. That is, the pair $(2, 0.05714286)$ implies a confidence level of $1 - (2)(0.0286) = 0.9429$. As an alternative to visual inspection, the appropriate value of k can be found using R Code 10.7 on the next page.

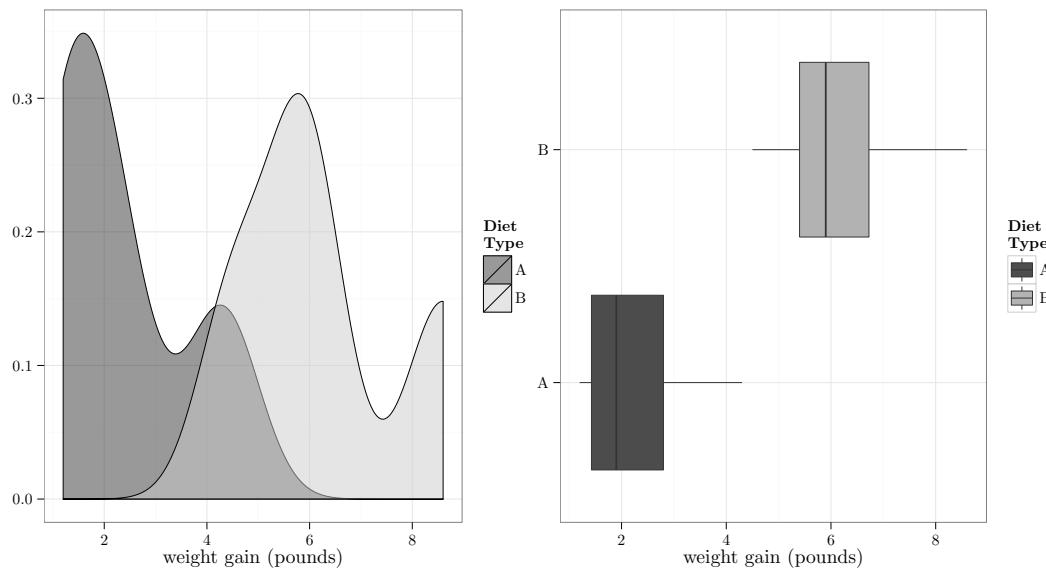


FIGURE 10.6: Density plots as well as side-by-side boxplots for piglet weight gain on diets A and B

R Code 10.7

```
> pwil <- pwilcox(1:(n * m), n, m)
> k <- which(pwil >= 0.05)[1]
> k
[1] 2
```

The nm differences are generated using the R function `outer()` and the k^{th} and $(nm-k+1)^{\text{st}}$ order statistics from the nm differences are identified in R Code 10.8. Consequently, a 94.29% confidence interval for the difference in medians is $CI_{0.9429}(\psi_A - \psi_B) = [-7.1, -1.4]$.

R Code 10.8

```
> diffss <- matrix(sort(outer(A, B, "-")), byrow = FALSE, nrow = 4)
> diffss
[,1] [,2] [,3] [,4]
[1,] -7.4 -4.6 -3.8 -2.2
[2,] -7.1 -4.5 -3.4 -1.8
[3,] -6.3 -4.3 -3.3 -1.4
[4,] -4.9 -4.2 -3.0 -0.2

> CL <- 1 - 2 * pwilcox((k - 1), n, m)
> CL
[1] 0.9428571

> CI <- c(diffss[k], diffss[n * m - k + 1])
> CI
[1] -7.1 -1.4
```

Construction of a confidence interval can be done directly with the argument `conf.int = TRUE` in the `wilcox.test()` function.

```
> wilcox.test(A, B, conf.int = TRUE, conf.level = 0.9)
```

```
Wilcoxon rank sum test

data: A and B
W = 0, p-value = 0.02857
alternative hypothesis: true location shift is not equal to 0
90 percent confidence interval:
-7.1 -1.4
sample estimates:
difference in location
-4
```

Recall that the achieved confidence level is actually 94.29%. The achieved confidence level is reflected in the output for the function `wilcoxe.test()`. Also note that the statistic `w` in `wilcoxe.test()` is the observed Wilcoxon rank-sum statistic not the Mann-Whitney U statistic reported by R's `wilcox.test()`.

```
> wilcoxe.test(A, B)
```

```
Wilcoxon Rank Sum Test

data: A and B
w = 10, p-value = 0.02857
alternative hypothesis: true median is not equal to 0
94.28571 percent confidence interval:
-7.1 -1.4
sample estimates:
difference in location
-4
```



10.4.2 Normal Approximation to the Wilcoxon Rank-Sum and Mann-Whitney U -Tests

For moderately sized samples ($n \geq 10$ and $m \geq 10$), the sampling distribution of W can be reasonably approximated with the normal distribution that has mean and standard deviation $n(N + 1)/2$ and $\sqrt{nm(N + 1)/12}$, respectively. That is, $W \stackrel{\text{d}}{\sim} N(n(N + 1)/2, \sqrt{nm(N + 1)/12})$. The standardized test statistic under the assumption that $H_0 : \psi_X - \psi_Y = \delta_0$ is true is

$$Z = \frac{W - \frac{n(N+1)}{2} - \delta_0}{\sqrt{\frac{nm(N+1)}{12}}} \stackrel{\text{d}}{\sim} N(0, 1). \quad (10.7)$$

See Figure 10.7 on the next page for a graph of the Wilcoxon rank-sum distribution for $n = m = 10$ superimposed by a normal distribution with $\mu = n(N + 1)/2 = 105$ and $\sigma = \sqrt{nm(N + 1)/12} = 13.2288$.

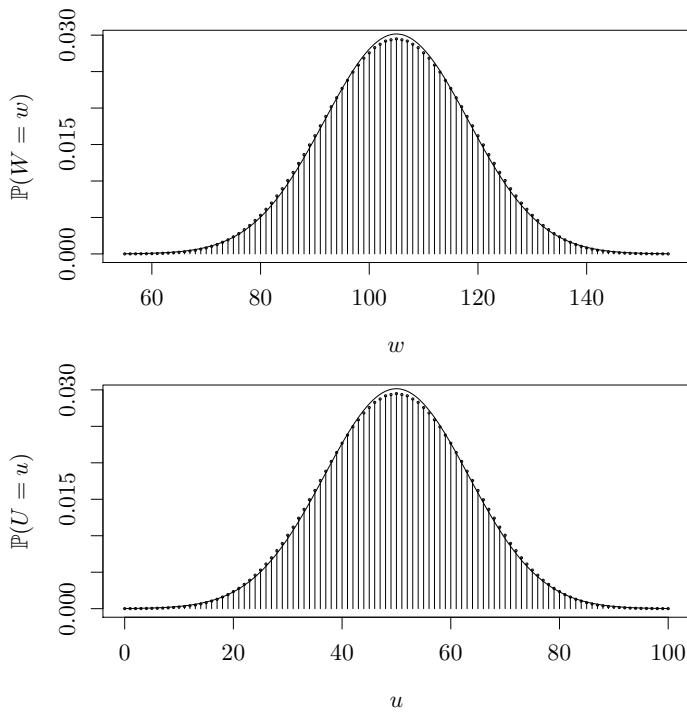


FIGURE 10.7: Graphical representations of the Wilcoxon rank-sum distribution for $n = m = 10$ superimposed by a normal distribution with $\mu = n(N + 1)/2 = 105$ and $\sigma = \sqrt{nm(N + 1)/12} = 13.2288$, and the Mann-Whitney U distribution superimposed by a normal distribution with $\mu = nm/2 = 50$ and $\sigma = \sqrt{nm(N + 1)/12} = 13.2288$

The formula for calculating the observed value of the standardized test statistic as well as the three possible alternative hypotheses and their rejection regions are described in Table 10.13 on the facing page. If there are tied ranks, the variance of W is reduced to

$$\frac{nm(N + 1)}{12} - \frac{nm}{12N(N - 1)} \sum_{j=1}^g t_j(t_j - 1)(t_j + 1) \quad (10.8)$$

where g denotes the number of tied groups and t_j is the size of tied group j . In (10.8), an untied observation is considered to be a tied group of size one. In the event that no ties exist, $g = N$ and $t_j = 1$ for $j = 1, \dots, N$, which produces a correction factor of zero.

The sampling distribution of U can likewise be reasonably approximated with a normal distribution that has a mean of $nm/2$ and a standard deviation of $\sqrt{nm(N + 1)/12}$. The bottom graph of Figure 10.7 shows the sampling distribution of U for $n = m = 10$ superimposed by a normal distribution with $\mu = nm/2 = 50$ and $\sigma = \sqrt{nm(N + 1)/12} = 13.2288$.

The standardized test statistic under the assumption that $H_0 : \psi_X - \psi_Y = \delta_0$ is true is

$$Z = \frac{U - \frac{nm}{2} - \delta_0}{\sqrt{\frac{nm(N+1)}{12}}} \stackrel{d}{\sim} N(0, 1). \quad (10.9)$$

The formula for calculating the observed value of the standardized test statistic as well as the three possible alternative hypotheses and their rejection regions are described in Table 10.14 on page 618.

Table 10.13: Summary for testing the difference in two medians — normal approximation to the Wilcoxon rank-sum test

Null Hypothesis — $H_0 : \psi_X - \psi_Y = \delta_0$

$$\text{Standardized Test Statistic's Value} \quad z_{obs} = \frac{w \pm 0.5 - n(N+1)/2 - \delta_0}{\sqrt{nm(N+1)/12 - CF}}$$

$$\text{Correction Factor} \quad CF = \frac{nm}{12N(N-1)} \sum_{j=1}^g t_j(t_j - 1)(t_j + 1)$$

Alternative Hypothesis	Rejection Region
$H_1 : \psi_X - \psi_Y < \delta_0$	$z_{obs} < z_\alpha$
$H_1 : \psi_X - \psi_Y > \delta_0$	$z_{obs} > z_{1-\alpha}$
$H_1 : \psi_X - \psi_Y \neq \delta_0$	$ z_{obs} > z_{1-\alpha/2}$

Note: The quantity ± 0.5 in the numerator of z_{obs} is the continuity correction. When $H_1 : \psi_X - \psi_Y < \delta_0$, the quantity $+0.5$ is used. When $H_1 : \psi_X - \psi_Y > \delta_0$, the quantity -0.5 is used. When $H_1 : \psi_X - \psi_Y \neq \delta_0$, use $+0.5$ if $w < n(N+1)/2$ and -0.5 if $w > n(N+1)/2$.

A corresponding two-sided confidence interval for the shift in distribution based on (10.9) are the k^{th} and $(nm - k + 1)^{\text{st}}$ ordered differences, where

$$k = 0.5 + \frac{nm}{2} + z_{\alpha/2} \sqrt{\frac{nm(N+1)}{12}}. \quad (10.10)$$

For a one-sided confidence interval, replace $z_{\alpha/2}$ with z_α . Since k is generally not an integer, it can be either rounded or truncated. To obtain a conservative estimate, one should truncate.

Example 10.9 ▷ Wilcoxon Rank-Sum Test: Swim Times ◁ Thirty-two division I swimmers from the same swim team agree to participate in a year-long study to determine whether high (30%) fat diets produce greater improvements in swim times than the standard low (10%) fat diets. Times for the 32 swimmers for the 200-yard individual medley were taken right after the swimmers' conference meet. The swimmers were randomly assigned to follow one of the diets. The group on diet 1 followed a low-fat diet the entire year but lost two swimmers along the way. The group on diet 2 followed the high fat diet the entire year and also lost two swimmers. Times for the 200-yard individual medley were taken one year later for the remaining 28 swimmers. The swimmers' improvements in seconds for both diets are presented in Table 10.15 on page 619 and stored in data frame **SWIMTIMES**, where x_i represents the time improvement in seconds for swimmers on the high fat diet (**highfat**) and y_i represents the time improvement in seconds for swimmers on the low-fat diet (**lowfat**).

- (a) Verify that the time improvement distributions are similar in shape.

Table 10.14: Summary for testing the difference in two medians — normal approximation to the Mann-Whitney U -Test

Null Hypothesis — $H_0 : \psi_X - \psi_Y = \delta_0$

$$\text{Standardized Test Statistic's Value} \quad z_{obs} = \frac{u \pm 0.5 - nm/2 - \delta_0}{\sqrt{nm(N+1)/12 - CF}}$$

$$\text{Correction Factor} \quad CF = \frac{nm}{12N(N-1)} \sum_{j=1}^g t_j(t_j - 1)(t_j + 1)$$

Alternative Hypothesis	Rejection Region
$H_1 : \psi_X - \psi_Y < \delta_0$	$z_{obs} < z_\alpha$
$H_1 : \psi_X - \psi_Y > \delta_0$	$z_{obs} > z_{1-\alpha}$
$H_1 : \psi_X - \psi_Y \neq \delta_0$	$ z_{obs} > z_{1-\alpha/2}$

Note: The quantity ± 0.5 in the numerator of z_{obs} is the continuity correction. When $H_1 : \psi_X - \psi_Y < \delta_0$, the quantity $+0.5$ is used. When $H_1 : \psi_X - \psi_Y > \delta_0$, the quantity -0.5 is used. When $H_1 : \psi_X - \psi_Y \neq \delta_0$, use $+0.5$ if $u < nm/2$ and -0.5 if $u > nm/2$.

- (b) Test whether the median difference for improvements in swim times is different from zero using a significance level of $\alpha = 0.10$ with the normal approximation to the Wilcoxon rank-sum test and the normal approximation to the Mann-Whitney U -test.
- (c) Use the function `wilcox_test()` from the `coin` package, which can be downloaded from your nearest CRAN mirror at <http://cran.r-project.org/mirrors.html> to report the exact p -value and the 90% confidence interval for the hypothesis in (b). According to the documentation of `coin`, this function computes exact conditional (on the data) p -values and quantiles using the shift-algorithm by Streitberg and Röhmel for both tied and untied samples.
- (d) Construct a confidence interval with confidence level of at least 90% using the normal approximation to find k .

Solution: The answers are as follows:

- (a) To use the Wilcoxon rank-sum test, the time improvement distributions must be similar in shape. A comparative boxplot of time improvements for low-fat and high fat diets is found in Figure 10.8 on page 620. Since the comparative boxplot does appear to show the same underlying distribution for time improvements for swimmers eating both diets, it is legitimate to proceed with a Wilcoxon rank-sum test or the Mann-Whitney U -test.
- (b) Use the five-step procedure to test if the median difference for improvements in swim times for high and low-fat diets is different from zero.

Step 1: **Hypotheses** — The null and alternative hypotheses to test if the median difference for improvements in swim times for high and low-fat diets is different from zero are

$$H_0 : \psi_X - \psi_Y = 0 \text{ versus } H_1 : \psi_X - \psi_Y \neq 0.$$

Step 2: **Test Statistic** —

Wilcoxon Rank-Sum Test

The test statistic chosen is W , where the observed value is

$$w = 248.$$

Mann-Whitney Test

The test statistic chosen is U , where the observed value is

$$\begin{aligned} u &= w - n(n + 1)/2 \\ &= 248 - 14(15)/2 = 143. \end{aligned}$$

Step 3: **Rejection Region Calculations** — Because both standardized test statistics are distributed approximately $N(0, 1)$ and H_1 is a two-sided hypothesis, the rejection region is $|z_{obs}| > z_{1-0.10/2} = 1.6449$.

Because there are three groups of ties ($g = 3$), where the sizes of the tied groups

Table 10.15: Sorted improvements in swim times in seconds for high (x) and low (y) fat diets, where rank refers to the rank of the data point in the combined sample of x and y data points (**SWIMTIMES**)

	y_i	rank(y_i)	x_i	rank(x_i)
Tied Rank	0.18	8.5	0.18	8.5
	-0.79	2.0	0.38	10.0
	-0.49	3.0	0.56	11.0
	-0.37	4.0	0.65	12.0
	-0.20	5.0	0.84	13.0
	-0.15	6.0	1.58	20.0
	0.02	7.0	0.89	16.0
	-0.87	1.0	1.18	18.0
Tied Rank	0.87	14.5	0.87	14.5
	0.98	17.0	2.03	22.0
	1.42	19.0	3.53	27.0
	1.71	21.0	4.33	28.0
	3.52	26.0		
Tied Ranks	2.66	24.0	2.66	24.0
			2.66	24.0
				w = 248

are 2, 2, and 3, the correction factor is

$$\begin{aligned}
 CF &= \frac{nm}{12N(N-1)} \sum_{j=1}^3 t_j(t_j-1)(t_j+1) \\
 &= \frac{(14)(14)}{12(28)(28-1)} \times [2(2-1)(2+1) \\
 &\quad + 2(2-1)(2+1) + 3(3-1)(3+1)] \\
 &= \frac{7}{324}[6 + 6 + 24] \\
 &= 7/9.
 \end{aligned}$$

Wilcoxon Rank-Sum Test

The value of the standardized test statistic is

$$\begin{aligned}
 z_{obs} &= \frac{w \pm 0.5 - n(N+1)/2 - \delta_0}{\sqrt{nm(N+1)/12 - CF}} \\
 &= \frac{248 - 0.5 - 14(28+1)/2 - 0}{\sqrt{(14)(14)(28+1)/12 - 7/9}} \\
 &= 2.0464.
 \end{aligned}$$

Mann-Whitney Test

The value of the standardized test statistic is

$$\begin{aligned}
 z_{obs} &= \frac{u \pm 0.5 - nm/2 - \delta_0}{\sqrt{nm(N+1)/12 - CF}} \\
 &= \frac{143 - 0.5 - (14)(14)/2 - 0}{\sqrt{(14)(14)(28+1)/12 - 7/9}} \\
 &= 2.0464.
 \end{aligned}$$

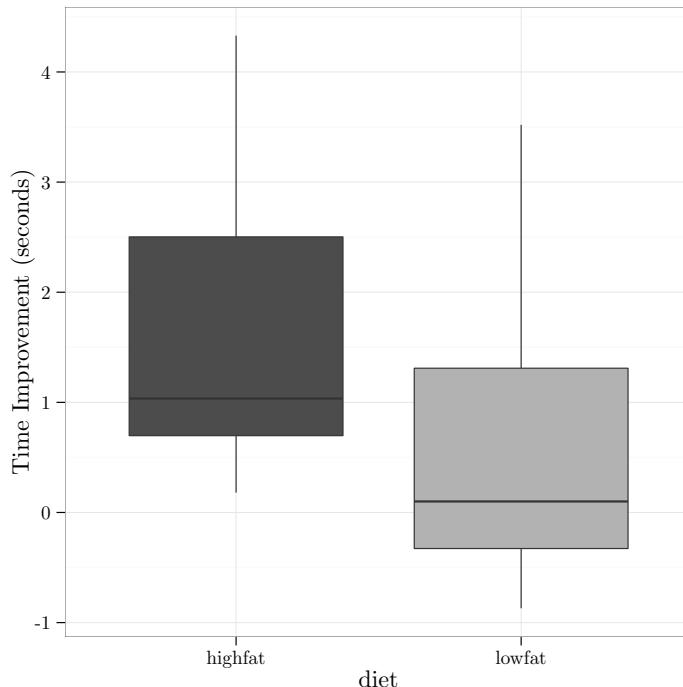


FIGURE 10.8: Comparative boxplot for improvements in swim times for high and low-fat diets

Step 4: **Statistical Conclusion** — The φ -value is $2\mathbb{P}(Z \geq 2.0464) = 0.0407$.

- I. From the rejection region, reject H_0 because $z_{obs} = 2.0464$ is more than 1.6449.
- II. From the φ -value, reject H_0 because the φ -value = 0.0407 is less than 0.10.

Reject H_0 .

Step 5: **English Conclusion** — There is sufficient evidence to suggest that the median time improvements are different for swimmers eating high fat and low-fat diets.

The φ -value can be computed with the R command `wilcox.test(seconds diet, data = SWIMTIMES)`. In the presence of ties, the function `wilcox.test()` automatically uses the normal approximation to the distribution of U , as well as applying a correction factor for the variance of U and an appropriate continuity correction factor to agree with the formula for the standardized test statistic given in Table 10.14 on page 618. R does not report the value of the standardized test statistic but does use the value of the standardized test statistic to compute the φ -value. The φ -value from the output is 0.04072, the exact value found for a z_{obs} value of 2.0464 in step 4.

```
> wilcox.test(seconds ~ diet, data = SWIMTIMES)

Warning in wilcox.test.default(x = c(0.18, 0.38, 0.56, 0.65, 0.84, 0.87, :
cannot compute exact p-value with ties

Wilcoxon rank sum test with continuity correction

data: seconds by diet
W = 143, p-value = 0.04072
alternative hypothesis: true location shift is not equal to 0
```

(c) From the output of `wilcox_test()`, the φ -value is 0.0382 and the 90% confidence interval for the difference in medians is $CI_{0.90}(\psi_X - \psi_Y) = [0.31, 1.68]$.

```
> library(coin)
> wilcox_test(seconds ~ diet, data = SWIMTIMES, distribution = "exact",
+   conf.int = TRUE, conf.level = 0.9)

Exact Wilcoxon Mann-Whitney Rank Sum Test

data: seconds by diet (highfat, lowfat)
Z = 2.0693, p-value = 0.03818
alternative hypothesis: true mu is not equal to 0
90 percent confidence interval:
 0.31 1.68
sample estimates:
difference in location
      1.02
```

Note that the reported standardized test statistic computed with `wilcox_test()` does not use a continuity correction.

(d) The $x_i - y_j$ differences are stored in `diffs`. Using (10.10) on page 617, k is calculated to be 62. The k^{th} difference is determined to be 0.31 and the $nm - k + 1^{\text{st}}$ difference is 1.68. Therefore, the 90% confidence interval, $CI_{0.90}(\psi_X - \psi_Y)$, is [0.31, 1.68].

```
> xtabs(~SWIMTIMES$diet)

SWIMTIMES$diet
highfat  lowfat
    14      14

> n <- xtabs(~SWIMTIMES$diet)[1]
> m <- xtabs(~SWIMTIMES$diet)[2]
> N <- n + m
> highfat <- SWIMTIMES$seconds[SWIMTIMES$diet == "highfat"]
> lowfat <- SWIMTIMES$seconds[SWIMTIMES$diet == "lowfat"]
> diffs <- sort(outer(highfat, lowfat, "-"))
> k <- 0.5 + n * m / 2 + qnorm(0.05) * sqrt(n * m * (N + 1) / 12)
> k <- floor(k)
> names(k) <- NULL
> k

[1] 62

> CI <- c(diffs[k], diffs[n * m - k + 1]) # 90% CI
> CI

[1] 0.31 1.68
```

10.5 The Kruskal-Wallis Test

The Kruskal-Wallis test is an extension of the Wilcoxon rank-sum/Mann-Whitney U -test for two independent samples (covered in Section 10.4) to the situation with a mutually independent samples. As with most statistical procedures, independence is preserved by using random samples. The design structure of this problem is often called a completely randomized design. The null hypothesis is that the a populations are identical. Like the Wilcoxon rank-sum/Mann-Whitney U -test, the only assumption the Kruskal-Wallis test requires is that the a populations be continuous and identical in shape. The null and alternative hypotheses are written

$$\begin{aligned} H_0 : F_1(x) &= F_2(x) = \cdots = F_a(x) \text{ for all } x \quad \text{versus} \\ H_1 : F_i(x) &\neq F_j(x) \text{ for at least one pair } (i, j) \text{ and some } x. \end{aligned} \tag{10.11}$$

Because the underlying distributions of the a populations are assumed to be identical in the null hypothesis, this test can apply to means, medians, or any other quantile and the null and alternative hypotheses are often expressed in terms of the population medians as

$$H_0 : \psi_1 = \psi_2 = \cdots = \psi_a \text{ versus } H_1 : \psi_i \neq \psi_j \text{ for at least one pair } (i, j). \tag{10.12}$$

To test the null hypothesis, all n_1, n_2, \dots, n_a observations are pooled into a single column and ranked from 1 to $N = \sum_{i=1}^a n_i$. The standardized test statistic used with the Kruskal-Wallis test via the function `kruskal.test()` is

$$H = \frac{12 \sum_{i=1}^a n_i (\bar{R}_i - \bar{R}_\bullet)^2}{N(N+1)} \quad (10.13)$$

where n_i is the number of observations in the i^{th} treatment/group, \bar{R}_i is the average of the ranks in the i^{th} treatment/group, and \bar{R}_\bullet is the average of all of the ranks. When ties are present, an adjusted standardized test statistic denoted as H' is also calculated and reported. The adjusted statistic H' is defined as

$$H' = \frac{H}{f_c} = \frac{H}{1 - \frac{\sum_{j=1}^r (t_j^3 - t_j)}{N^3 - N}} \quad (10.14)$$

where t_j is the number of times a given rank was tied in the combined sample of size N and r is the number of ranks in the combined sample of size N that were tied. Provided each $n_i \geq 5$, the sampling distributions of H_{obs} and H' are both approximately chi-square random variables with $a - 1$ degrees of freedom (χ_{a-1}^2). More specifically, when H_0 is true, the statistic H has, as $\min(n_1, \dots, n_a)$ tends to infinity, an asymptotic χ_{a-1}^2 distribution.

Example 10.10 ▷ Kruskal-Wallis Test: Free-Throws ◁ An elementary school gym teacher is interested in evaluating the effectiveness of four free-throw teaching techniques. The gym teacher randomly assigns the 80 students to one of four groups with 20 students per group. After two months, every member of the groups shoots 10 free-throws, and the gym teacher records the results. The number of successful free-throws each student shoots in each of the four groups is presented in Table 10.16. Use the free-throw results to decide if differences exist among teaching methods at the $\alpha = 0.05$ level.

Table 10.16: Number of successful free-throws

Method	Data																			
	6	1	2	0	0	1	1	3	1	2	1	2	4	2	1	1	1	3	7	1
Method1	6	1	2	0	0	1	1	3	1	2	1	2	4	2	1	1	1	3	7	1
Method2	3	2	1	2	1	6	2	1	1	2	1	1	2	3	2	2	3	2	5	2
Method3	2	1	2	3	2	2	4	3	2	3	2	5	1	1	3	7	6	2	2	2
Method4	2	1	1	3	1	2	1	6	1	1	0	1	1	1	1	2	2	1	5	4

Solution: The five-step procedure is used and explained to determine if differences exist among teaching methods. Before proceeding, first examine side-by-side boxplots for free-throws made grouped by teaching method. Based on the boxplots and the density plots in Figure 10.9 on the following page, it seems reasonable to assume that all a populations are similar in shape. R Code 10.9 on the next page can be used to create graphs similar to those in Figure 10.9 on the following page. The axes for the box plots have been switched using `coord_flip()`. When the axes are switched, the order of the methods is also switched. That is, the first method will now appear closest to the origin. To reverse the order of the methods, the function `scale_x_discrete()` is used with the argument `limits = rev(levels(...))`.

R Code 10.9

```

> Method1 <- c(6, 1, 2, 0, 0, 1, 1, 3, 1, 2, 1, 2, 4, 2, 1, 1, 1, 3, 7, 1)
> Method2 <- c(3, 2, 1, 2, 1, 6, 2, 1, 1, 2, 1, 1, 2, 3, 2, 2, 3, 2, 5, 2)
> Method3 <- c(2, 1, 2, 3, 2, 2, 4, 3, 2, 3, 2, 5, 1, 1, 3, 7, 6, 2, 2, 2)
> Method4 <- c(2, 1, 1, 3, 1, 2, 1, 6, 1, 1, 0, 1, 1, 1, 2, 2, 1, 5, 4)
> DF <- stack(data.frame(Method1, Method2, Method3, Method4))
> ggplot(data = DF, aes(x = ind, y = values, fill = ind)) +
+   geom_boxplot() +
+   labs(x = "", y = "Number of successful free-throws") +
+   guides(fill = FALSE) +
+   coord_flip() +
+   scale_x_discrete(limits = rev(levels(DF$ind)))
> ggplot(data = DF, aes(x = values)) +
+   geom_density(aes(fill = ind)) +
+   facet_grid(ind ~ .) +
+   labs(y = "", x = "Number of successful free-throws") +
+   guides(fill = FALSE)

```

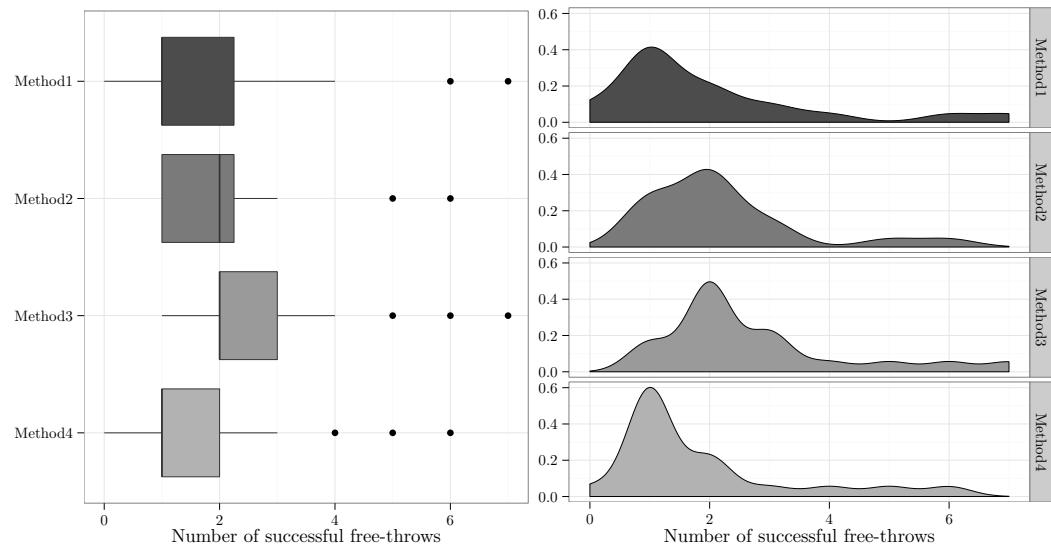


FIGURE 10.9: Boxplots and density plots of free-throw teaching results

Step 1: **Hypotheses** — The hypotheses to test equality of the F_i for $i = 1, \dots, a$ distributions are $H_0 : F_1(x) = F_2(x) = F_3(x) = F_4(x)$ for all x versus $H_1 : F_i(x) \neq F_j(x)$ for at least one pair (i, j) and some x

Step 2: **Test Statistic** — The test statistic R_i is used to evaluate the null hypothesis. Under the assumption that H_0 is true, the standardized test statistic

$$H = \frac{12 \sum_{i=1}^a n_i (\bar{R}_i - \bar{R}_{\bullet})^2}{N(N+1)} \sim \chi_{a-1}^2.$$

Step 3: Rejection Region Calculations — The rejection region is $H_{\text{obs}} > \chi^2_{0.95;3} = 7.8147$. The number of free-throws completed in each of the methods is combined into a single population and ranked among the 80 observations. Table 10.17 shows the actual free-throws with their ranks among the 80 observations.

Table 10.17: Actual free-throws with ranks among all free-throws

Meth1	RankM1	Meth2	RankM2	Meth3	RankM3	Meth4	RankM4	Total
6	76.5	3	63.5	2	45.5	2	45.5	
1	18.0	2	45.5	1	18.0	1	18.0	
2	45.5	1	18.0	2	45.5	1	18.0	
0	2.0	2	45.5	3	63.5	3	63.5	
0	2.0	1	18.0	2	45.5	1	18.0	
1	18.0	6	76.5	2	45.5	2	45.5	
1	18.0	2	45.5	4	70.0	1	18.0	
3	63.5	1	18.0	3	63.5	6	76.5	
1	18.0	1	18.0	2	45.5	1	18.0	
2	45.5	2	45.5	3	63.5	1	18.0	
1	18.0	1	18.0	2	45.5	0	2.0	
2	45.5	1	18.0	5	73.0	1	18.0	
4	70.0	2	45.5	1	18.0	1	18.0	
2	45.5	3	63.5	1	18.0	1	18.0	
1	18.0	2	45.5	3	63.5	1	18.0	
1	18.0	2	45.5	7	79.5	2	45.5	
1	18.0	3	63.5	6	76.5	2	45.5	
3	63.5	2	45.5	2	45.5	1	18.0	
7	79.5	5	73.0	2	45.5	5	73.0	
1	18.0	2	45.5	2	45.5	4	70.0	
Means:	35.050		42.875		50.825		33.250	40.500

The value of H_{obs} is calculated as

$$\begin{aligned}
 H_{\text{obs}} &= \frac{12 \sum_{i=1}^a n_i (\bar{R}_i - \bar{R}_{\bullet})^2}{N(N+1)} \\
 &= \frac{12}{(80 \times 81)} \times \left\{ \begin{array}{l} (20 \times (35.05 - 40.50))^2 + (20 \times (42.875 - 40.50))^2 + \\ (20 \times (50.825 - 40.50))^2 + (20 \times (33.250 - 40.50))^2 \end{array} \right\} \\
 &= 7.20412.
 \end{aligned}$$

The adjusted test statistic H'_{obs} is calculated as

$$\begin{aligned}
 H'_{\text{obs}} &= \frac{H_{\text{obs}}}{f_c} = \frac{H_{\text{obs}}}{1 - \frac{\sum_{j=1}^r (t_j^3 - t_j)}{N^3 - N}} \\
 &= \frac{7.20412}{1 - \left\{ \frac{(3^3 - 3) + (29^3 - 29) + (26^3 - 26) + (10^3 - 10) + (3^3 - 3) + (3^3 - 3) + (4^3 - 4) + (2^3 - 2)}{80^3 - 80} \right\}} \\
 &= \frac{7.204120}{0.9159283} = 7.865376.
 \end{aligned}$$

Step 4: Statistical Conclusion — The ϕ -value for the standardized test statistic without adjustment for ties (H) and the standardized test statistic adjusted for ties (H') are calculated as $\mathbb{P}(\chi_3^2 \geq 7.2041) = 0.0657$ and $\mathbb{P}(\chi_3^2 \geq 7.8654) = 0.0489$, respectively. ϕ -values such as 0.0657 and 0.0489 indicate that observing values as extreme or more than 7.2041 or 7.8654 when the null hypothesis is true are fairly unlikely.

- I. From the rejection region, reject H_0 since $H'_{\text{obs}} = 7.8654 > \chi_{0.95;3}^2 = 7.8147$.
- II. From the ϕ -value, reject H_0 because the ϕ -value = 0.0489 is less than 0.05.

Reject H_0 .

Step 5: English Conclusion — There is statistical evidence to suggest differences exist among the distributions for the four free-throw teaching methods.

To compute the rejection region, the value of the standardized test statistic (`Hobs`), and the value of the standardized test statistic corrected for ties (`Hc`), see R Code 10.10.

R Code 10.10

```

> RR <- qchisq(0.95, 3)                                # Rejection region
> RR

[1] 7.814728

> DF$RKS <- rank(DF$values)                            # Ranks for all
> MRKs <- unstack(DF, RKS ~ ind)                      # Show first five rows
> MRKs[1:5,]

  Method1 Method2 Method3 Method4
1    76.5    63.5    45.5    45.5
2    18.0    45.5    18.0    18.0
3    45.5    18.0    45.5    18.0
4     2.0    45.5    63.5    63.5
5     2.0    18.0    45.5    18.0

> RK <- apply(MRKs, 2, mean)                           # Treatment ranks
> names(RK) <- c("MRKT1", "MRKT2", "MRKT3", "MRKT4")
> RK

  MRKT1  MRKT2  MRKT3  MRKT4
35.050 42.875 50.825 33.250

```

```

> MRK <- mean(RK)                                # Overall mean rank
> MRK
[1] 40.5

> N <- length(DF$RKS)
> n1 <- length(Method1)
> n2 <- length(Method2)
> n3 <- length(Method3)
> n4 <- length(Method4)
> Hobs <- 12*(n1*(RK[1] - MRK)^2 + n2*(RK[2] - MRK)^2 +
+               n3*(RK[3] - MRK)^2 + n4*(RK[4] - MRK)^2)/(N * (N + 1))
> names(Hobs) <- "statistic"
> Hobs

statistic
7.20412

> tj <- xtabs(~RKS, data = DF)
> tj

RKS
 2   18 45.5 63.5   70   73 76.5 79.5
 3   29   26   10     3     3     4     2

> CF <- 1-(sum(tj^3 - tj)/(N^3 - N))          # correction factor
> Hc <- Hobs/CF                                  # corrected statistic
> hs <- c(Hobs, Hc)
> hs

statistic statistic
7.204120 7.865376

> pval <- 1-pchisq(hs, 3)
> names(pval) <- c("pvalueHobs", "pvalueHc")
> pval

pvalueHobs    pvalueHc
0.06566864 0.04887747

```

To find the standardized test statistic corrected for ties and its corresponding φ -value with the function `kruskal.test()`, enter

```

> kruskal.test(values ~ ind, data = DF)

Kruskal-Wallis rank sum test

data: values by ind
Kruskal-Wallis chi-squared = 7.8654, df = 3, p-value = 0.04888

```



Distribution of H The exact distribution of H can be obtained using the fact that under H_0 , all possible $(\sum_{i=1}^a n_i)! / (\prod_{i=1}^a n_i!)$ assignments of n_1 ranks to the treatment 1 observations, n_2 ranks to the treatment 2 observations, ..., n_a ranks to the treatment a observations are equally likely; however, there are practical and computational limits on the range of tables that can be constructed. Consider that Example 10.10 on page 623 has $80!/(20! \cdot 20! \cdot 20! \cdot 20!) = 2.04281602001991e + 45$ possible rank assignments. Consequently, the distribution of H is generally approximated with a χ_{a-1}^2 distribution. Under the null hypothesis, the n_i ranks in sample i are randomly selected from the set $\{1, 2, \dots, \sum_{i=1}^a n_i = N\}$. That is, the ranks in sample i are drawn without replacement from the finite populations of N ranks.

For a finite population, it can be shown that

$$E[\bar{R}_i] = \frac{N+1}{2} \quad \text{and} \quad \text{Var}[\bar{R}_i] = \frac{(N+1)(N-n_i)}{12n_i}.$$

Provided the $\min(n_i)$ is sufficiently large,

$$Z_i = \frac{\bar{R}_i - \frac{N+1}{2}}{\sqrt{\frac{(N+1)(N-n_i)}{12n_i}}} \stackrel{\sim}{\sim} N(0, 1) \quad (10.15)$$

by the Central Limit Theorem. It then follows that $Z_i^2 \stackrel{\sim}{\sim} \chi_1^2$. Although the Z_i s are not independent, when H_0 is true, the statistic

$$H = \sum_{i=1}^a \frac{N-n_i}{N} Z_i^2 = \sum_{i=1}^a \frac{12n_i [\bar{R}_i - \frac{N+1}{2}]^2}{N(N+1)} \quad (10.16)$$

has, as $\min\{n_1, n_2, \dots, n_a\}$ tends to infinity, an asymptotic χ_{a-1}^2 distribution. When the null hypothesis is rejected, one can compare any two groups by calculating

$$Z_{ij\text{obs}} = \frac{|\bar{R}_i - \bar{R}_j|}{\sqrt{\left(\frac{N(N+1)}{12}\right) \left(\frac{1}{n_i} + \frac{1}{n_j}\right)}} \quad (10.17)$$

and declaring treatments i and j significantly different when $Z_{ij\text{obs}} > Z_{1-\alpha/[a(a-1)]}$. By dividing α by $a(a-1)$, the number of pairwise comparisons, the overall significance level is appropriately adjusted.

Example 10.10 on page 623 rejected the null hypothesis of equal distributions and concluded that at least two of the four methods have different distributions. The next step is to decide which one of the four methods the gym instructor should use in teaching students to shoot free-throws. Using (10.17) with an $\alpha = 0.20$, methods 1 and 4 are declared to be significantly different from method 3 since $Z_{13\text{obs}} = 2.1467 > Z_{1-\alpha/[a(a-1)]} = 2.128$ and $Z_{34\text{obs}} = 2.3917 > Z_{1-\alpha/[a(a-1)]} = 2.128$. In this case, the probability that all the statements are correct is $1 - \alpha = 0.8$. The gym instructor should stop using methods 1 and 4. If the instructor had to pick only one method to use, and all other factors were the same, he/she should use method 3 since it is statistically better than method 1 and method 4. Although there is no statistical difference between methods 2 and 3, method 2 is not statistically better than method 1 or method 4. R Code 10.11 on the next page computes the multiple comparisons according to (10.17).

R Code 10.11

```

> a <- 4                                # Four methods
> alpha <- 0.20
> Z12 <- abs(RK[1] - RK[2])/sqrt((N*(N + 1)/12)*(1/n1 + 1/n2))
> Z13 <- abs(RK[1] - RK[3])/sqrt((N*(N + 1)/12)*(1/n1 + 1/n3))
> Z14 <- abs(RK[1] - RK[4])/sqrt((N*(N + 1)/12)*(1/n1 + 1/n4))
> Z23 <- abs(RK[2] - RK[3])/sqrt((N*(N + 1)/12)*(1/n2 + 1/n3))
> Z24 <- abs(RK[2] - RK[4])/sqrt((N*(N + 1)/12)*(1/n2 + 1/n4))
> Z34 <- abs(RK[3] - RK[4])/sqrt((N*(N + 1)/12)*(1/n3 + 1/n4))
> Zij <- c(Z12, Z13, Z14, Z23, Z24, Z34)
> names(Zij) <- c("Z12", "Z13", "Z14", "Z23", "Z24", "Z34")
> CV <- qnorm(1 - alpha/(a*(a - 1)))
> Zij

      Z12      Z13      Z14      Z23      Z24      Z34
1.064848 2.146706 0.244949 1.081858 1.309797 2.391655

> CV
[1] 2.128045

> which(Zij > CV)

Z13 Z34
2   6

```

10.6 Friedman Test for Randomized Block Designs

In Section 10.5, the Kruskal-Wallis rank test for several independent samples was introduced as an extension of the Wilcoxon rank-sum/Mann-Whitney U -test for two independent samples introduced in Section 10.4. In this section, the problem of analyzing related samples is examined. The design structure of the problems addressed in this section is often referred to as a randomized complete block design. In this type of design, there are b blocks and $k \geq 2$ treatments, and the test is designed to detect differences among the k treatments. In this type of scenario, observations are arranged in blocks, which are groups of k experimental units similar to each other in some important characteristic. The rationale behind using a block is to reduce the error of the experiment as much as possible by grouping similar units so that the remaining differences will be largely due to the treatments. The use of “blocks” comes from some of the earliest experimental designs in agriculture where fields were divided in “blocks.”

In a randomized complete block design (RCBD), experimental units are assigned to blocks, and then treatments are randomly assigned to the units within the blocks. To analyze a RCBD with Friedman’s test, ranks are assigned to the observations within each block. The ranked observations are denoted R_{ij} , $i = 1, \dots, b$, $j = 1, \dots, k$. A representation of the ranked data from a RCBD is shown in Table 10.18 on the following page.

The assumptions required to apply Friedman’s test are the same as those required for the Kruskal-Wallis test; namely, all populations sampled are continuous and identical, except

Table 10.18: A representation of the ranked data from a randomized complete block design

	1	2	k	Row Totals		
Blocks	1	R_{11}	R_{12}	\cdots	R_{1k}	$k(k+1)/2$
	2	R_{21}	R_{22}	\cdots	R_{2k}	$k(k+1)/2$
	\vdots	\vdots		\vdots	\vdots	\vdots
	b	R_{b1}	R_{b2}	\cdots	R_{bk}	$k(k+1)/2$
Column Totals:		R_1	R_2	\cdots	R_k	$bk(k+1)/2$

possibly for location. The null hypothesis is that the populations all have the same location. Typically, the null hypothesis of no difference among the k treatments is written in terms of the medians as $H_0 : \psi_1 = \psi_2 = \cdots = \psi_k$. Although the distribution under the null hypothesis could be enumerated, it is not practical to do so as there are a total of $(k!)^b$ distinguishable sets of entries in a $b \times k$ table. The Friedman statistic S is

$$S = \left[\frac{12}{bk(k+1)} \sum_{j=1}^k R_j^2 \right] - 3b(k+1), \quad (10.18)$$

where R_j is the sum of ranks for each treatment, where ranks were assigned within each block. The statistic S has an asymptotic χ_{k-1}^2 distribution as b tends to infinity. For $b > 7$, numerical comparisons have shown χ_{k-1}^2 to be a reasonable approximation to the distribution of S (Gibbons and Chakraborti, 2003). When ties are present in the ranks, S is replaced with the quantity S' :

$$S' = \frac{12 \sum_{j=1}^k R_j^2 - 3b^2 k(k+1)^2}{bk(k+1) - \frac{1}{k-1} \sum_{i=1}^b \left\{ \left(\sum_{j=1}^{g_i} t_{ij}^3 \right) - k \right\}} \quad (10.19)$$

where g_i denotes the number of tied groups in the i^{th} block and t_{ij} is the size of the j^{th} tied group in the i^{th} block. Note that when there are no ties in the blocks, the quantity $\frac{1}{k-1} \sum_{i=1}^b \left\{ \left(\sum_{j=1}^{g_i} t_{ij}^3 \right) - k \right\} = 0$ and S' reduces to S . The null hypothesis is rejected at the α level of significance whenever $S'_{\text{obs}} > \chi_{1-\alpha;k-1}^2$. When the null hypothesis is rejected, one can declare treatments i and j significantly different when $Z_{R_{ij,\text{obs}}} > Z_{1-\alpha/[k(k-1)]}$, where $Z_{R_{ij,\text{obs}}}$ is defined as

$$Z_{R_{ij,\text{obs}}} = \frac{|R_i - R_j|}{\sqrt{\left(\frac{bk(k+1)}{6} \right)}}. \quad (10.20)$$

Typical values of α when performing multiple comparisons are often as large as 0.20 due to the large number of comparisons.

Example 10.11 ▷ Friedman Test: Body Fat ◁ The body fat of 78 high school wrestlers was measured using three separate techniques and the results are stored in the data frame **HSWRESTLER**. The techniques used were hydrostatic weighing (**hwfat**), skin fold measurements (**skfat**), and the Tanita body fat scale (**tanfat**). Do the three methods of recording body fat have equal medians? Use a significance level of $\alpha = 0.05$ to reach your conclusion. If the null hypothesis of equal medians is rejected, determine which treatments are significantly different using an overall experiment-wise error rate of $\alpha = 0.20$.

Solution: Each wrestler in this scenario acts as a block. This particular design structure is also known as a repeated measures design. Before testing the null hypothesis of equal medians, a few graphs are created to verify the assumption of equal-shaped populations. Graphs similar to Figure 10.10 can be created using R Code 10.12.

R Code 10.12

```
> DF <- stack(HSWRESTLER[,c('hwfat', 'skfat', 'tanfat')])  
> head(DF)  
> ggplot(data = DF, aes(x = ind, y = values, fill = ind)) +  
+   geom_boxplot() +  
+   guides(fill = FALSE) +  
+   labs(x = "", y = "Percent body fat") +  
+   coord_flip() +  
+   scale_x_discrete(limits = rev(levels(DF$ind)))  
> ggplot(data = DF, aes(x = values)) +  
+   geom_density(aes(fill = ind)) +  
+   facet_grid(ind ~ .) +  
+   labs(y = "", x = "Percent body fat") +  
+   guides(fill = FALSE)
```

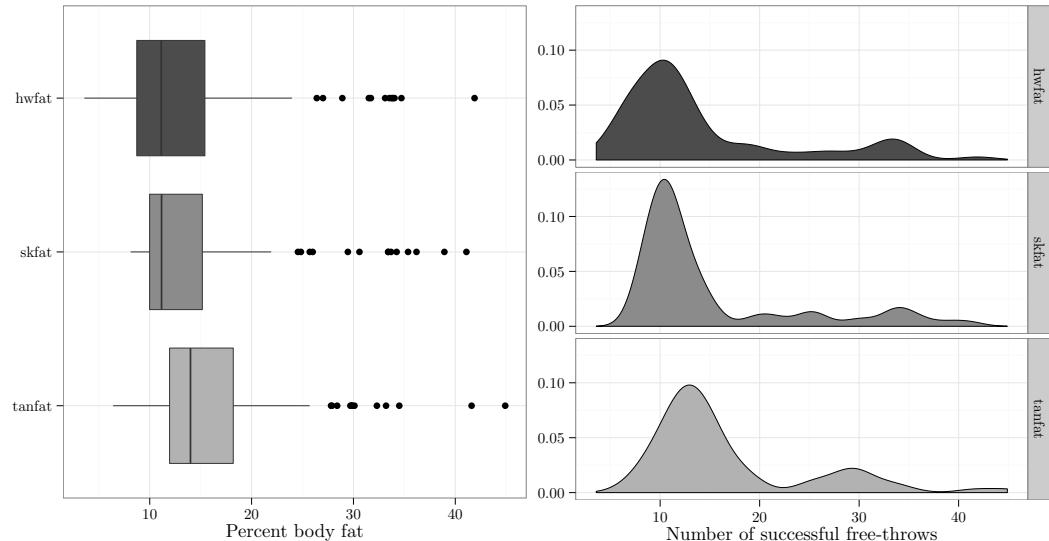


FIGURE 10.10: Comparative boxplots and density plots for hydrostatic weighing (hwfat), skin fold measurements (skfat), and the Tanita body fat scale (tanfat)

Based on the boxplots and the density plots in Figure 10.10, it seems reasonable to assume that the distributions of body fat for the three treatment groups are similar in shape.

Step 1: **Hypotheses** — The hypotheses to test no difference among the k treatments are $H_0 : \psi_1 = \psi_2 = \dots = \psi_k$ versus $H_1 : \psi_i \neq \psi_j$ for at least one pair (i, j) .

Step 2: **Test Statistic** — The test statistic S is used to evaluate the null hypothesis.

Under the assumption that H_0 is true, the standardized test statistic

$$S = \left[\frac{12}{bk(k+1)} \sum_{j=1}^k R_j^2 \right] - 3b(k+1) \sim \chi_{k-1}^2.$$

Step 3: Rejection Region Calculations — The rejection region is $S_{\text{obs}} > \chi_{0.95;2}^2 = 5.9915$. The first six wrestlers' body fat as measured by the three techniques and their corresponding ranks are shown in Table 10.19.

Table 10.19: The first six wrestlers' body fat as measured by the three techniques and their corresponding ranks

Wrestler	Measurement			Rank		
	hwfat	tanfat	skfat	hwfat	tanfat	skfat
1	10.71	11.9	9.80	2	3	1
2	8.53	10.0	10.56	1	2	3
3	6.78	8.3	8.43	1	2	3
4	9.32	8.2	11.77	2	1	3
5	41.89	41.6	41.09	3	2	1
6	34.03	29.9	29.45	3	2	1
:	:	:	:	:	:	:
				$R_1 = 128$	$R_2 = 187$	$R_3 = 153$

The value of S_{obs} is calculated as

$$\begin{aligned} S_{\text{obs}} &= \left[\frac{12}{bk(k+1)} \sum_{j=1}^k R_j^2 \right] - 3b(k+1) \\ &= \left[\frac{12}{78 \cdot 3(3+1)} (128^2 + 187^2 + 153^2) \right] - 3 \cdot 78(3+1) \\ &= 22.48718. \end{aligned}$$

Step 4: Statistical Conclusion — The p -value for the standardized test statistic (S_{obs}) is calculated as $\mathbb{P}(\chi_2^2 \geq 22.4872) = 0$, indicating that observing values as extreme or more than 22.4872 when the null hypothesis is true is very unlikely.

- I. From the rejection region, reject H_0 since $S_{\text{obs}} = 22.4872 > \chi_{0.95;2}^2 = 5.9915$.
- II. From the p -value, reject H_0 because the p -value = 0 is less than 0.05.

Reject H_0 .

Step 5: English Conclusion — There is statistical evidence to suggest differences exist among the three methods used to measure body fat.

To compute the rejection region, the value of the standardized test statistic, and its corresponding value see R Code 10.13 on the next page.

R Code 10.13

```

> cfat <- HSWRESTLER[, c('h wfat', 'tanfat', 'skfat')]
> RK <- t(apply(cfat, 1, rank)) # rankings
> OBSandRK <- cbind(cfat, RK)
> OBSandRK[1:5, ]

   h wfat tanfat skfat h wfat tanfat skfat
1 10.71    11.9  9.80     2      3      1
2  8.53    10.0 10.56     1      2      3
3  6.78     8.3  8.43     1      2      3
4  9.32     8.2 11.77     2      1      3
5 41.89    41.6 41.09     3      2      1

> Rj <- apply(RK, 2, sum)
> b <- dim(HSWRESTLER)[1]
> k <- dim(cfat)[2]
> S <- (12/(b*k*(k + 1)))*sum(Rj^2) - 3*b*(k + 1)
> S

[1] 22.48718

> pval <- pchisq(S, k-1, lower = FALSE)
> pval

[1] 1.309095e-05

```

To find the standardized test statistic and its corresponding p -value with the function `friedman.test()`, see R Code 10.14.

R Code 10.14

```

> DF$Block <- factor(rep(1:78, 3)) # create 78 blocks
> head(DF) # show first 6 rows of DF

  values  ind Block
1 10.71 h wfat    1
2  8.53 h wfat    2
3  6.78 h wfat    3
4  9.32 h wfat    4
5 41.89 h wfat    5
6 34.03 h wfat    6

> friedman.test(values ~ ind | Block, data = DF)

Friedman rank sum test

data: values and ind and Block
Friedman chi-squared = 22.487, df = 2, p-value = 1.309e-05

```

Since the null hypothesis of equal medians is soundly rejected, at least two of the three body fat measuring techniques have different medians. Using (10.20) with an $\alpha = 0.20$, all three of the body fat measuring techniques are declared to be significantly different from each

other since $Z_{R_{12\text{obs}}} = 4.7238 > Z_{1-\alpha/(k(k-1))} = 1.8339$, $Z_{R_{13\text{obs}}} = 2.0016 > Z_{1-\alpha/(k(k-1))} = 1.8339$, and $Z_{R_{23\text{obs}}} = 2.7222 > Z_{1-\alpha/(k(k-1))} = 1.8339$. In this case, the probability that all the statements are correct is $1-\alpha = 0.8$. Since `hwt` is the accepted standard for measuring body fat, neither of the other two methods is an acceptable substitute for measuring body fat for high school wrestlers.

R Code 10.15 computes the multiple comparisons according to (10.20).

R Code 10.15

```
> alpha <- 0.2
> ZR12 <- abs(Rj[1] - Rj[2])/sqrt(b * k * (k + 1)/6)
> ZR13 <- abs(Rj[1] - Rj[3])/sqrt(b * k * (k + 1)/6)
> ZR23 <- abs(Rj[2] - Rj[3])/sqrt(b * k * (k + 1)/6)
> CV <- qnorm(1 - alpha/(k * (k - 1)))
> ZRij <- c(ZR12, ZR13, ZR23)
> names(ZRij) <- c("ZR12", "ZR13", "ZR23")
> ZRij

ZR12      ZR13      ZR23
4.723781  2.001602  2.722179

> CV

[1] 1.833915

> which(ZRij > CV)

ZR12  ZR13  ZR23
1     2     3
```



10.7 Goodness-of-Fit Tests

Many statistical procedures require knowledge of the population from which the sample is taken. For example, using Student's t -distribution for testing a hypothesis or constructing a confidence interval for μ assumes that the parent population is normal. In this section, **goodness-of-fit** (GOF) procedures are presented that will help to identify the distribution of the population from which the sample is drawn. The null hypothesis in a goodness-of-fit test is a statement about the form of the cumulative distribution. When all the parameters in the null hypothesis are specified, the hypothesis is called simple. Recall that in the event the null hypothesis does not completely specify all of the parameters of the distribution, the hypothesis is said to be composite. Goodness-of-fit tests are typically used when the form of the population is in question. In contrast to most of the statistical procedures discussed so far, where the goal has been to reject the null hypothesis, in a GOF test one hopes to retain the null hypothesis. Two general approaches, one designed primarily for discrete distributions (chi-square goodness-of-fit) and one designed primarily for continuous distributions (Kolmogorov-Smirnov), are presented.

10.7.1 The Chi-Square Goodness-of-Fit Test

Given a single random sample of size n from an unknown population F_X , one may wish to test the hypothesis that F_X has some known distribution $F_0(x)$ for all x . For example, using the data frame **SOCWER** from Example 4.4 on page 258, is it reasonable to assume the number of goals scored during regulation time for the 232 soccer matches has a Poisson distribution with $\lambda = 2.5$? Although the problem was previously analyzed, it will be considered again shortly in the context of the chi-square goodness-of-fit test. The chi-square goodness-of-fit test is based on a normalized statistic that examines the vertical deviations between what is observed and what is expected when H_0 is true in k mutually exclusive categories. At times, such as in surveys of brand preferences, where the categories/groups would be the brand names, the sample will lend itself to being divided into k mutually exclusive categories. Other times, the categories/groupings will be more arbitrary. Before applying the chi-square goodness-of-fit test, the data must be grouped according to some scheme to form k mutually exclusive categories. When the null hypothesis completely specifies the population, the probability that a random observation will fall into each of the chosen or fixed categories can be computed. Once the probabilities for a data point to fall into each of the chosen or fixed categories is computed, multiplying the probabilities by n produces the expected counts for each category under the null distribution. If the null hypothesis is true, the differences between the counts observed in the k categories and the counts expected in the k categories should be small. The test criterion for testing $H_0 : F_X(x) = F_0(x)$ for all x against the alternative $H_1 : F_X(x) \neq F_0(x)$ for some x when the null hypothesis is completely specified is

$$\chi_{\text{obs}}^2 = \sum_{i=1}^k \frac{(O_k - E_k)^2}{E_k}, \quad (10.21)$$

where χ_{obs}^2 is the sum of the squared deviations between what is observed (O_k) and what is expected (E_k) in each of the k categories divided by what is expected in each of the k categories. Large values of χ_{obs}^2 occur when the observed data are inconsistent with the null hypothesis and thus lead to rejection of the null hypothesis. The exact distribution of χ_{obs}^2 is very complicated; however, for large n , provided all expected categories are at least 5, χ_{obs}^2 is distributed approximately χ^2 with $k - 1$ degrees of freedom. When the null hypothesis is composite, that is, not all of the parameters are specified, the degrees of freedom for the random variable χ_{obs}^2 are reduced by one for each parameter that must be estimated.

Example 10.12 ▷ Soccer Goodness-of-Fit ◁ Test the hypothesis that the number of goals scored during regulation time for the 232 soccer matches stored in the data frame **SOCWER** has a Poisson **cdf** with $\lambda = 2.5$ with the chi-square goodness-of-fit test and an α level of 0.05. Produce a histogram showing the number of observed goals scored during regulation time and superimpose on the histogram the number of goals that are expected to be made when the distribution of goals follows a Poisson distribution with $\lambda = 2.5$.

Solution: Since the number of categories for a Poisson distribution is theoretically infinite, a table is first constructed of the observed number of goals to get an idea of reasonable categories.

```
> xtabs(~goals, data = SOCCER)
```

goals	0	1	2	3	4	5	6	7	8
19	49	60	47	32	18	3	3	1	

Based on the table, a decision is made to create categories for 0, 1, 2, 3, 4, 5, and 6 or more goals. Under the null hypothesis that $F_0(x)$ is a Poisson distribution with $\lambda = 2.5$, the probabilities of scoring 0, 1, 2, 3, 4, 5, and 6 or more goals are computed with R as follows:

```
> PX <- c(dpois(0:5, 2.5), ppois(5, 2.5, lower = FALSE))
> PX
[1] 0.08208500 0.20521250 0.25651562 0.21376302 0.13360189 0.06680094
[7] 0.04202104
```

Since there were a total of $n = 232$ soccer games, the expected number of goals for the six categories is simply $232 \times \text{PX}$.

```
> EX <- 232*PX
> OB <- c(as.vector(xtabs(~goals, data = SOCCER)[1:6]),
+           sum(xtabs(~goals, data = SOCCER)[7:9]))
> OB
[1] 19 49 60 47 32 18 7

> ans <- cbind(PX, EX, OB)
> row.names(ans) <- c(" X=0", " X=1", " X=2", " X=3", " X=4", " X=5", "X>=6")
> ans

      PX       EX   OB
X=0 0.08208500 19.043720 19
X=1 0.20521250 47.609299 49
X=2 0.25651562 59.511624 60
X=3 0.21376302 49.593020 47
X=4 0.13360189 30.995638 32
X=5 0.06680094 15.497819 18
X>=6 0.04202104  9.748881  7
```

Step 1: **Hypotheses** — The null and alternative hypotheses for using the chi-square goodness-of-fit test to test the hypothesis that the number of goals scored during regulation time for the 232 soccer matches stored in the data frame `SOCCKER` has a Poisson **cdf** with $\lambda = 2.5$ are

$$\begin{aligned} H_0 : F_X(x) = F_0(x) &\sim \text{Pois}(\lambda = 2.5) \text{ for all } x \text{ versus} \\ H_1 : F_X(x) &\neq F_0(x) \text{ for some } x. \end{aligned}$$

Step 2: **Test Statistic** — The test statistic chosen is χ^2_{obs} .

Step 3: **Rejection Region Calculations** — Reject if $\chi^2_{\text{obs}} > \chi^2_{1-\alpha;k-1}$. The χ^2_{obs} is computed with (10.21) in R Code 10.16.

R Code 10.16

```
> chi.obs <- sum((OB - EX)^2/EX)
> chi.obs
[1] 1.39194
```

$$1.3919 = \chi_{\text{obs}}^2 \stackrel{?}{>} \chi_{0.95;6}^2 = 12.5916.$$

Step 4: **Statistical Conclusion** — The p -value is 0.9663.

```
> p.val <- pchisq(chi.obs, 7 - 1, lower = FALSE)
> p.val
[1] 0.9663469
```

- I. Since $\chi_{\text{obs}}^2 = 1.3919$ is not greater than $\chi_{0.95;6}^2 = 12.5916$, fail to reject H_0 .
- II. Since the p -value = 0.9663 is greater than 0.05, fail to reject H_0 .

Fail to reject H_0 .

Step 5: **English Conclusion** — There is no evidence to suggest that the true **cdf** does not equal the Poisson distribution with $\lambda = 2.5$ for at least one x .

To perform a goodness-of-fit test with the function **chisq.test()**, one may specify a vector of observed values for the argument **x=**, and a vector of probabilities of the same length as the vector passed to **x=** to the argument **p=**.

```
> chisq.test(x = OB, p = PX)

Chi-squared test for given probabilities

data: OB
X-squared = 1.3919, df = 6, p-value = 0.9663
```

R Code 10.17 uses base graphics to create a histogram with superimposed expected goals and the result is shown in Figure 10.11 on the next page.

R Code 10.17

```
> hist(SOCCER$goals, breaks = c((-0.5 + 0):(8 + 0.5)), col = "lightblue",
+       xlab = "Goals scored", ylab = "", freq = TRUE, main = "")
> x <- 0:8
> fx <- (dpois(0:8, lambda = 2.5))*232
> lines(x, fx, type = "h")
> lines(x, fx, type = "p", pch = 16)
```

Note that the histogram does not reflect the category ≥ 6 , but rather depicts the observed categories of 6, 7, and 8.

Although the chi-square goodness-of-fit test is primarily designed for discrete distributions, it can also be used with a continuous distribution if appropriate categories are defined.

Example 10.13 \triangleright **Goodness-of-Fit for SAT Scores** \triangleleft Use the chi-square goodness-of-fit test with $\alpha = 0.05$ to test the hypothesis that the SAT scores stored in the data frame **GRADES** have a normal **cdf**. Use categories $(-\infty, \mu - 2\sigma]$, $(\mu - 2\sigma, \mu - \sigma]$, $(\mu - \sigma, \mu]$, $(\mu, \mu + \sigma]$,

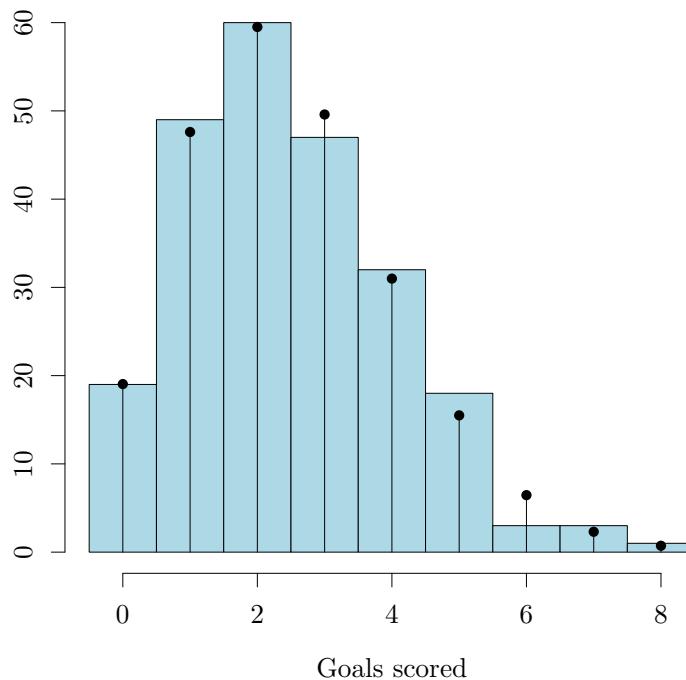


FIGURE 10.11: Histogram of observed goals for **SOCWER** with a superimposed Poisson distribution with $\lambda = 2.5$ (vertical lines)

$(\mu + \sigma, \mu + 2\sigma]$, and $(\mu + 2\sigma, \infty]$. Produce a histogram using the categories specified and superimpose on the histogram the expected number of SAT scores in each category when $F_0(x) \sim N(\mu = \bar{x}, \sigma = s)$.

Solution: The test follows:

Step 1: **Hypotheses** — The null and alternative hypotheses for using the chi-square goodness-of-fit test to test the hypothesis that the SAT scores stored in the data frame **GRADES** have a Normal **cdf** are

$$H_0 : F_X(x) = F_0(x) \sim N(\mu = \bar{x}, \sigma = s) \text{ for all } x \text{ versus}$$

$$H_1 : F_X(x) \neq F_0(x) \text{ for some } x.$$

Step 2: **Test Statistic** — Since the mean and standard deviation are unknown, the first step is to estimate the unknown parameters μ and σ using $\bar{x} = 1134.65$ and $s = 145.6087$.

```
> mu <- mean(GRADES$sat)
> sig <- sd(GRADES$sat)
> c(mu, sig)

[1] 1134.6500 145.6087
```

Because a normal distribution is continuous, it is necessary to create categories that include all the data. The categories $\mu - 3\sigma$ to $\mu - 2\sigma, \dots, \mu + 2\sigma$ to $\mu + 3\sigma$

are 697.824 to 843.4326, 843.4326 to 989.0413, 989.0413 to 1134.65, 1134.65 to 1280.2587, 1280.2587 to 1425.8674, and 1425.8674 to 1571.476. These particular categories include all of the observed SAT scores; however, the probabilities actually computed for the largest and smallest categories will be all of the area to the right and left, respectively, of $\bar{x} \pm 2s$. This is done so that the total area under the distribution in the null hypothesis is one.

```
> bin <- seq(from = mu - 3*sig, to = mu + 3*sig, by = sig)
> round(bin, 0)                                # vector of bin cut points

[1] 698 843 989 1135 1280 1426 1571

> T1 <- table(cut(GRADES$sat, breaks = bin))
> T1                                              # count of observations in bins

(698,843]          (843,989]          (989,1.13e+03]
              4                  27                  65
(1.13e+03,1.28e+03] (1.28e+03,1.43e+03] (1.43e+03,1.57e+03]
              80                 21                  3

> OB <- as.vector(T1)
> OB                                              # vector of observations

[1] 4 27 65 80 21 3

> PR <- c(pnorm(-2), pnorm(-1:2) - pnorm(-2:1),
+         pnorm(2, lower = FALSE)) # area under curve
> EX <- 200*PR                # Expected count in bins
> ans <- cbind(PR, EX, OB)    # column bind values in ans
> ans

      PR          EX  OB
[1,] 0.02275013 4.550026 4
[2,] 0.13590512 27.181024 27
[3,] 0.34134475 68.268949 65
[4,] 0.34134475 68.268949 80
[5,] 0.13590512 27.181024 21
[6,] 0.02275013 4.550026 3
```

Step 3: Rejection Region Calculations — Reject if $\chi^2_{\text{obs}} > \chi^2_{1-\alpha; k-p-1}$.

Now that the expected and observed counts for each of the categories are computed, the χ^2_{obs} value can be computed according to (10.21) and is 4.1737.

```
> chi.obs <- sum((OB - EX)^2/EX)
> chi.obs

[1] 4.173654
```

Step 4: **Statistical Conclusion** — In this problem, two parameters were estimated, and as a consequence, the degrees of freedom are computed as $6 - 2 - 1 = 3$. The p -value is 0.2433.

```
> p.val <- pchisq(chi.obs, 6 - 2 - 1, lower = FALSE)
> p.val
[1] 0.2433129
```

- I. Since $\chi^2_{\text{obs}} = 4.1737$ is not greater than $\chi^2_{0.95;3} = 7.8147$, fail to reject H_0 .
- II. Since the p -value = 0.2433 is greater than 0.05, fail to reject H_0 .

Fail to reject H_0 .

Step 5: **English Conclusion** — There is no evidence to suggest that the true **cdf** of SAT scores is not a normal distribution.

If one uses the R function **chisq.test()**, the degrees of freedom and the subsequent p -value will be incorrect, as illustrated next in R Code 10.18.

R Code 10.18

```
> chisq.test(x = OB, p = PR) # returns incorrect dof and p-value

Chi-squared test for given probabilities

data: OB
X-squared = 4.1737, df = 5, p-value = 0.5247
```

Since it is not feasible to produce a histogram that extends from $-\infty$ to ∞ , a histogram is created where the categories will simply cover the range of observed values. In this problem, the range of the SAT scores is 720 to 1550. The histogram with categories $(\mu - 3\sigma, \mu - 2\sigma]$, $(\mu - 2\sigma, \mu - \sigma]$, $(\mu - \sigma, \mu]$, $(\mu, \mu + \sigma]$, $(\mu + \sigma, \mu + 2\sigma]$, and $(\mu + 2\sigma, \mu + 3\sigma]$, superimposed with the expected number of SAT scores for the categories $(-\infty, \mu - 2\sigma]$, $(\mu - 2\sigma, \mu - \sigma]$, $(\mu - \sigma, \mu]$, $(\mu, \mu + \sigma]$, $(\mu + \sigma, \mu + 2\sigma]$, and $(\mu + 2\sigma, \infty]$ is computed in R Code 10.19 and depicted in Figure 10.12 on the facing page.

R Code 10.19

```
> hist(GRADES$sat, breaks = bin, col = "lightblue", xlab = "SAT scores",
+       ylab = "", freq = TRUE, main = "")
> x <- bin[2:7] - sig/2
> fx <- PR * 200
> lines(x, fx, type = "h")
> lines(x, fx, type = "p", pch = 16)
```



10.7.2 Kolmogorov-Smirnov Goodness-of-Fit Test

In Section 10.7.1, the chi-square goodness-of-fit test worked by measuring the vertical distance between what was observed in a particular category and what was expected in

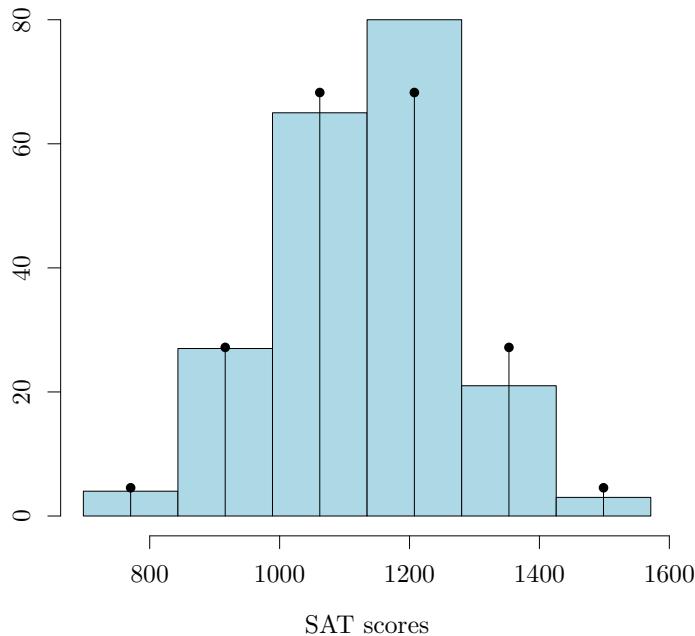


FIGURE 10.12: Histogram of SAT scores in **GRADES** superimposed with the expected number of SAT scores for the categories $(-\infty, \mu - 2\sigma]$, $(\mu - 2\sigma, \mu - \sigma]$, $(\mu - \sigma, \mu]$, $(\mu, \mu + \sigma]$, $(\mu + \sigma, \mu + 2\sigma]$, and $(\mu + 2\sigma, \infty]$ (vertical lines)

that same category under the null hypothesis for each of the k categories. In contrast to the chi-square goodness-of-fit test, the Kolmogorov-Smirnov goodness-of-fit test uses all n observations and measures vertical deviations between the cumulative distribution function (**cdf**), $F_0(x)$ (where all parameters are specified), and the empirical cumulative distribution function (**ecdf**), $\hat{F}_n(x)$, for all x . For large n , the deviations between $F_0(x)$ and $\hat{F}_n(x)$ should be small for all values of x . The statistic D_n , called the Kolmogorov-Smirnov one-sample statistic, is defined as

$$D_n = \sup_x |\hat{F}_n(x) - F_0(x)|. \quad (10.22)$$

The statistic D_n does not depend on $F_0(x)$ as long as $F(x)$ is continuous. The derivation of the sampling distribution of D_n is beyond the scope of this text. The curious reader can refer to Gibbons and Chakraborti (2003), page 114, for the derivation of the sampling distribution of D_n . The statistic and sampling distribution of D_n should only be used with simple hypotheses. When the null hypothesis is composite, the critical values for the Kolmogorov-Smirnov test (based on the sampling distribution of D_n) are extremely conservative. The Kolmogorov-Smirnov test can be used to assess normality provided the distribution is completely specified. In a test of normality where the null hypothesis is not completely specified, the statistic D_n can still be used by estimating the unknown parameters of $F_0(x)$ using maximum likelihood ($\hat{F}_0(x)$) and substituting $\hat{F}_0(x)$ for $F_0(x)$ in (10.22); however, this further complicates the sampling distribution of D_n . When testing a composite normal hypothesis with unknown μ and σ , the test that uses $D_n = \sup_x |\hat{F}_n(x) - \hat{F}_0(x)|$ is called Lilliefors's normality test (explained more fully starting on page 645). Lilliefors used simulation to study the sampling distribution of D_n for composite hypotheses and subsequently to publish critical values for using D_n with composite hypotheses. Simulation will be used to show the differences in the distribution of D_n for a simple null hypothesis

versus the distribution of D_n with a composite null hypothesis.

Recall that the **ecdf** was defined in (3.5) to be:

$$\hat{F}_n(t) = \sum_{i=1}^n \mathbf{I}\{x_i \leq t\}/n.$$

An equivalent expression for the **ecdf** is

$$\hat{F}_n(x) = \begin{cases} 0 & x < X_{(1)} \\ \frac{i}{n} & X_{(i)} \leq x \leq X_{(i+1)} \\ 1 & x > X_{(n)}, \end{cases} \quad (10.23)$$

which will prove useful in computing D_n . When all n observations are distinct, D_n can be computed as

$$D_n = \max_{i=1,\dots,n} M_i \quad (10.24)$$

where

$$M_i = \max \left\{ |\hat{F}_n(X_{(i)}) - F_0(X_{(i)})|, |F_0(X_{(i)}) - \hat{F}_n(X_{(i-1)})| \right\}. \quad (10.25)$$

Since $\hat{F}_n(X_{(i)}) = \frac{i}{n}$ and $\hat{F}_n(X_{(i-1)}) = \frac{i-1}{n}$, (10.25) can be expressed as

$$M_i = \max \left\{ \left| \frac{i}{n} - F_0(X_{(i)}) \right| = D_i^+, \left| F_0(X_{(i)}) - \frac{i-1}{n} \right| = D_i^- \right\}. \quad (10.26)$$

Stated formally, the null and alternative hypotheses for the Kolmogorov-Smirnov test for goodness-of-fit are

$$H_0 : F(x) = F_0(x) \text{ for all } x \text{ versus } H_1 : F(x) \neq F_0(x) \text{ for some } x. \quad (10.27)$$

The null hypothesis is rejected when $D_n > D_{n;1-\alpha}$ or when the test's p -value is less than the largest acceptable α value. Since R will compute the p -value for the Kolmogorov-Smirnov test, critical values for various n and α are not presented. R uses the function **ks.test(x, y, ...)**, where **x** is a numeric vector of observations and **y** is either a numeric vector of data values or a character string naming a cumulative distribution function.

Example 10.14 ▷ **Kolmogorov-Smirnov GOF Test** ◷ Test whether the observations 5, 6, 7, 8, and 9 are from a normal distribution with $\mu = 6.5$ and $\sigma = \sqrt{2}$. That is, the hypothesized distribution is $F_0(x) \sim N(6.5, \sqrt{2})$.

Solution: Since $F_0(x) \sim N(6.5, \sqrt{2})$, it follows that

$$F_0(X_{(i)}) = P(Y \leq X_{(i)}) = P\left(\frac{Y - 6.5}{\sqrt{2}} \leq \frac{X_{(i)} - 6.5}{\sqrt{2}}\right) = P\left(Z \leq \frac{X_{(i)} - 6.5}{\sqrt{2}}\right).$$

To compute $F_0(X_{(i)})$ with R, key in

```
> x <- 5:9
> mu <- 6.5
> sig <- sqrt(2)
> x <- sort(x)
> n <- length(x)
> FoX <- pnorm(x, mean = mu, sd = sig)
> FoX
[1] 0.1444222 0.3618368 0.6381632 0.8555778 0.9614501
```

The quantities $\hat{F}_n(X_{(i)}) = \frac{i}{n}$, $\hat{F}_n(X_{(i-1)}) = \frac{i-1}{n}$, D_i^+ , D_i^- , and M_i are computed and stored in the R objects `FnX`, `Fn1X`, `Dp`, `Dm`, and `Mi`, respectively. The Kolmogorov-Smirnov statistic $D_n = \max_{i=1,\dots,n} M_i$ is 0.25558. The values from the R code are shown in Table 10.20.

```

> FnX <- seq(1:n)/n
> Fn1X <- (seq(1:n) - 1)/n
> DP <- (FnX - FoX)
> DM <- FoX - Fn1X
> Dp <- abs(DP)
> Dm <- abs(DM)
> EXP <- cbind(x, FnX, Fn1X, FoX, Dp, Dm)
> Mi <- apply(EXP[, c(5, 6)], 1, max)
> TOT <- cbind(EXP, Mi)
> TOT

      x FnX Fn1X          FoX          Dp          Dm          Mi
[1,] 5 0.2  0.0 0.1444222 0.05557782 0.1444222 0.1444222
[2,] 6 0.4  0.2 0.3618368 0.03816320 0.1618368 0.1618368
[3,] 7 0.6  0.4 0.6381632 0.03816320 0.2381632 0.2381632
[4,] 8 0.8  0.6 0.8555778 0.05557782 0.2555778 0.2555778
[5,] 9 1.0  0.8 0.9614501 0.03854994 0.1614501 0.1614501

> Dn <- max(Mi)
> Dn

[1] 0.2555778

```

Table 10.20: Calculating D_n

i	$X_{(i)}$	$\frac{i}{n} - F_0(X_{(i)})$	$F_0(X_{(i)}) - \frac{i-1}{n}$	D^+	D^-	M_i
1	5	$\frac{1}{5} - 0.14442$	$0.14442 - 0$	0.055578	0.14442	0.14442
2	6	$\frac{2}{5} - 0.36184$	$0.36184 - \frac{1}{5}$	0.038163	0.16184	0.16184
3	7	$\frac{3}{5} - 0.63816$	$0.63816 - \frac{2}{5}$	0.038163	0.23816	0.23816
4	8	$\frac{4}{5} - 0.85558$	$0.85558 - \frac{3}{5}$	0.055578	0.25558	0.25558
5	9	$\frac{5}{5} - 0.96145$	$0.96145 - \frac{4}{5}$	0.038550	0.16145	0.16145
						$D_n = 0.25558$

The computation of the Kolmogorov-Smirnov statistic D_n and its p -value are shown in R Code 10.20.

R Code 10.20

```
> ks.test(x, y = "pnorm", mean = mu, sd = sig)
```

One-sample Kolmogorov-Smirnov test

```
data: x
D = 0.25558, p-value = 0.8269
alternative hypothesis: two-sided
```

The Komolgorov-Smirnov statistic is labeled `D` in the output produced by `ks.test()`. The value $D_n = 0.2556$ with a corresponding p -value of 0.8269 provides no evidence to reject the null hypothesis that $F_0(x) \sim N(6.5, \sqrt{2})$. Figure 10.13 provides a graphical illustration of the vertical deviations used to compute the statistic D_n for this problem.

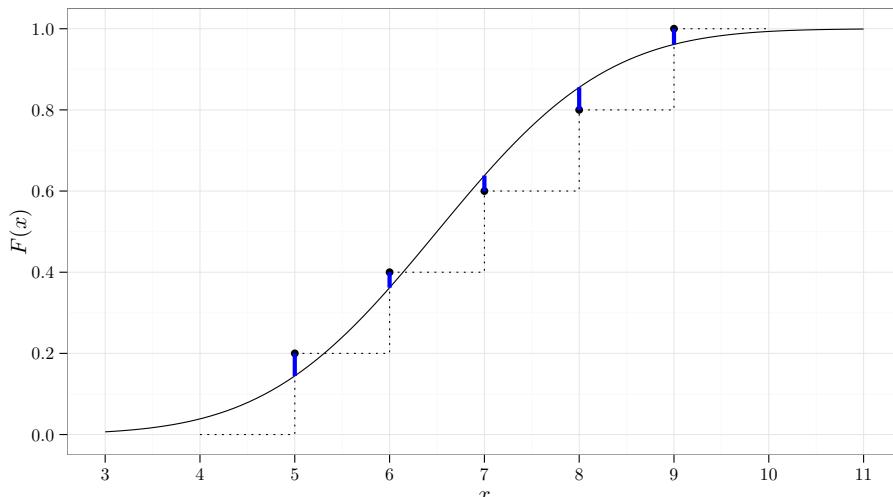


FIGURE 10.13: Graphical illustration of the vertical deviations used to compute the statistic D_n for Example 10.14 on page 642. The solid “S”-shaped line is the hypothesized distribution $F_0(x) \sim N(6.5, \sqrt{2})$. The vertical solid segments between the solid circles and $F_0(x)$ represents the D^+ values. The vertical dotted distance represents the D^- values and the dotted stair shaped values represent the ecdf.

In Example 10.14 on page 642, the statistic $D_n = 0.2556$ returned a p -value of 0.8269. To visualize the sampling distribution of D_n and to find simulated critical values, one can use R Code 10.21.

R Code 10.21

```
> ksdist <- function (n = 10, sims = 10000, alpha = 0.05){
+   Dn <- replicate(sims, ks.test(rnorm(n), pnorm)$statistic)
+   cv <- quantile(Dn, 1 - alpha)
+   plot(density(Dn), col = "blue", lwd = 2, main = "",
+         xlab = paste("Simulated critical value =", round(cv, 3),
+                     "for n =", n, "when the alpha value =", alpha))
+   title(
+     main = list(expression(paste("Simulated Sampling Distribution of ",
+                               D[n]))))
+ }
```

The graph from running `ksdist(n = 5, sims = 10000, alpha = 0.05)` when using a seed of 13 is shown in Figure 10.14. This simulation indicates a value of 0.567 or greater would be required to reject the null hypothesis in Example 10.14 on page 642 at the $\alpha = 0.05$ level. The simulated p -value for the value $D_n = 0.2556$ in Figure 10.14 is 0.8292, very close to the 0.8269 reported from using `ks.test()`.

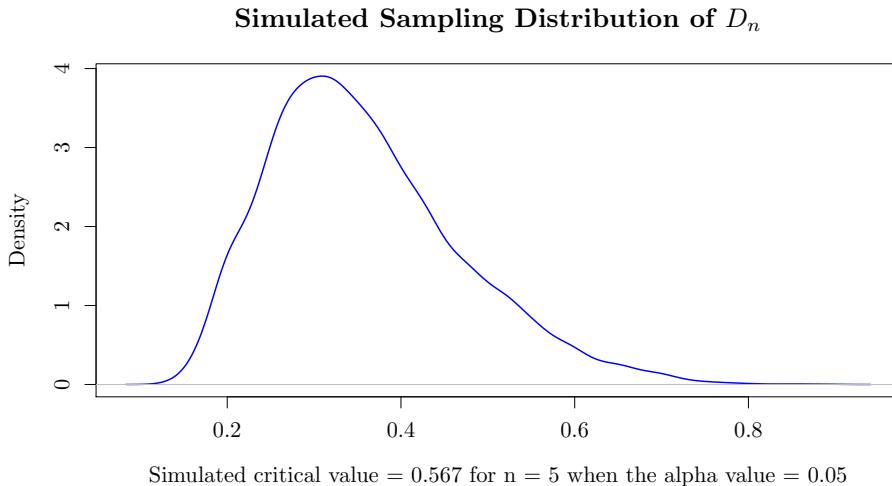


FIGURE 10.14: Graphical illustration of `ksdist(n = 5, sims = 10000, alpha = 0.05)`

Lilliefors's Test of Normality

Expanding on the simulation for the sampling distribution for D_n , consider what happens when the null hypothesis changes from simple to composite using the code for the function `ksLdist()`. Note that the D_n values stored in `D_n[i]` are for a simple null hypothesis of normality while the D_n values stored in `DnL[i]` are for a composite hypothesis of normality. The critical values reported by Lilliefors (1967) were based on simulations using 1000 or more samples using logic similar to the R Code 10.22 used to create the function `ksLdist()`.

R Code 10.22

```
> ksLdist <- function (n = 10, sims = 1000, alpha = 0.05){
+   Dn <- c()
+   DnL <- c()
+   for (i in 1:sims) {
+     x <- rnorm(n)
+     mu <- mean(x)
+     sig <- sd(x)
+     Dn[i] <- ks.test(x, pnorm)$statistic
+     DnL[i] <- ks.test(x, pnorm, mean = mu, sd = sig)$statistic
+   }
+   ys <- range(density(DnL)$y)
+   xs <- range(density(Dn)$x)
+   cv <- quantile(Dn, 1 - alpha)
```

```

+   cvp <- quantile(DnL, 1 - alpha)
+   plot(density(Dn, bw = 0.02), col="blue", lwd=2, ylim=ys, xlim=xs,
+         main = "", , xlab="", sub = paste("Simulated critical value =",,
+         round(cv, 3), "(simple hypothesis) and ", round(cvp, 3),
+             "(composite hypothesis)\n for n =", n,"when the alpha value =",,
+             alpha))
+   title(
+     main = list(expression(paste("Simulated Sampling Distribution of ",,
+       D[n]))))
+   lines(density(DnL, bw = 0.02), col = "red", lwd = 2, lty = 2)
+   legend(mean(xs), max(ys), legend = c("Simple Hypothesis",
+       "Composite Hypothesis"), col = c("blue", "red"), xjust = 0,
+       text.col = c("black", "black"), lty = c(1, 2), bg = "gray95",
+       cex = 1, lwd = 2)
+   box()
+   abline(h = 0)
+
}

```

The function `ksLdist()` allows the user to choose the number of samples with the argument `sims=` and easily to verify the results given by Lilliefors (1967). Dallal and Wilkinson (1986) duplicated the work by Lilliefors (1967) using much larger samples as well as deriving an analytic approximation for the upper tail φ -values for $D_n = \sup_x |\hat{F}_n(x) - \hat{F}_0(x)|$. For φ -values less than 0.100 and sample sizes ranging from 5 to 100, the Dallal-Wilkinson approximation is

$$\widehat{\varphi\text{-value}} = \exp(-7.01256 \cdot D_n^2 \cdot (n + 2.78019) + 2.99587 \cdot D_n \cdot \sqrt{n + 2.78019} - 0.122119 + 0.974598/\sqrt{n} + 1.67997/n) \quad (10.28)$$

The estimated densities from running `ksLdist(sims = 10000, n = 10)` with a seed of 13 are shown in Figure 10.15 on the facing page, which highlights how much less variability is present in the sampling distribution of D_n when the null hypothesis is composite. To test a composite hypothesis of normality correctly, one should use the R function `lillie.test()` available in the R package `nortest`. That is, one should not use the R function `ks.test()`.

Example 10.15 ▷ Long-Distance Phone Calls ◁ Calculate the φ -value and state the English conclusion for testing whether the times spent on long-distance phone calls (`call.time`) in the data frame `PHONE` have a normal distribution using the R function `lillie.test` from the `nortest` package Verify the reported φ -value using (10.28).

Solution: Note that the function `nortest()` labels the statistic D_n with a `D`. The value `nortest()` computes for D_n is 0.191 with a φ -value of 0.0291.

R Code 10.23

```

> library(nortest)
> lillie.test(PHONE$call.time)

Lilliefors (Kolmogorov-Smirnov) normality test

data: PHONE$call.time
D = 0.19102, p-value = 0.0291

```

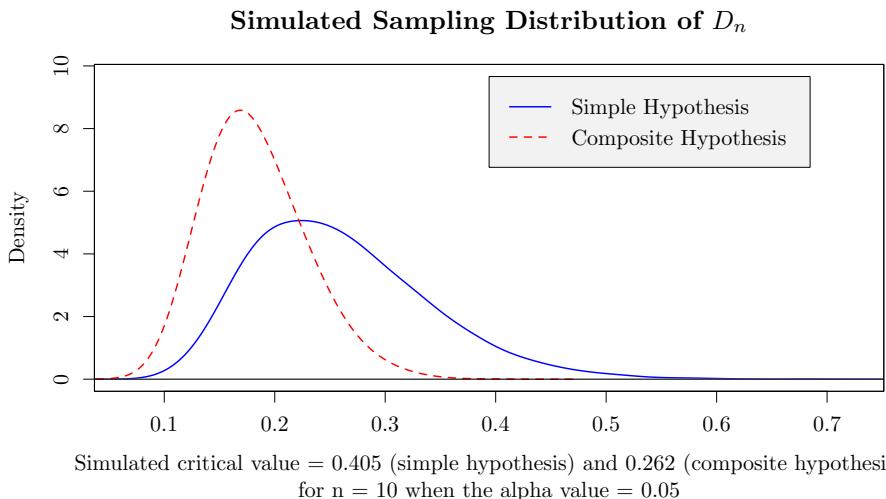


FIGURE 10.15: Estimated densities for simple and composite hypotheses from running `ksLdist(sims = 10000, n = 10)`

To compute the φ -value using (10.28), a small function `DWA()` is written in R Code 10.24. Running the function `DWA()` with the arguments $D_n = 0.191$ and $n = 23$ returns an estimated φ -value of 0.0291.

R Code 10.24

```
> DWA <- function(Dn = 0.3, n = 10) {
+   p.value <- exp(-7.01256 * Dn^2 * (n + 2.78019) + 2.99587 *
+     Dn * (n + 2.78019)^0.5 - 0.122119 + 0.974598/n^0.5 +
+     1.67997/n)
+   names(p.value) <- NULL
+   round(p.value, 4)
+ }
> DWA(Dn = 0.191, n = 23)
[1] 0.0291
```

With a φ -value of 0.0291, the null hypothesis is rejected. There is evidence that phone call length is not normally distributed. ■

10.7.3 Shapiro-Wilk Normality Test

The Shapiro-Wilk test is appropriate for testing normality. More specifically, the test allows for a composite hypothesis of normality. That is, the parameters of the normal distribution do not need to be specified in the null hypothesis of the test (as they must be for the Lilliefors test). Although the test is known to be conservative, it is useful for testing normality with small samples. The test statistic measures how closely the empirical quantiles of the sample follow the corresponding theoretical quantiles of a normal distribution. This means that small values of the test statistic lead to the rejection of the null hypothesis (that the distribution is normal).

To calculate the test statistic for a random sample of size n , x_1, x_2, \dots, x_n , the sample

must be sorted: $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$. The Shapiro-Wilk test statistic takes the form

$$W = \frac{b^2}{nS_u^2}, \quad (10.29)$$

where S_u^2 is the uncorrected sample variance, $b = \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} a_{n-i+1}(x_{(n-i+1)} - x_{(i)})$, and $\lfloor \frac{n}{2} \rfloor$ is the integer part of $\frac{n}{2}$. The coefficients a_{n-i+1} that are calculated automatically by the function `shapiro.test()` are tabulated in Table 6 of Shapiro and Wilk (1965).

The critical region of the test is given by

$$\mathbb{P}(W \leq K | H_0) = \alpha,$$

where α is the significance level. The critical values K can be found in Shapiro and Wilk (1965, Table 5), but they are not displayed in the output for `shapiro.test()`. The vector of weights $\mathbf{a}' = (a_1, \dots, a_n)$, where $a_i = -a_{n-i+1}$, is calculated as

$$\mathbf{a} = \frac{\mathbf{w}' \mathbf{V}^{-1}}{\mathbf{w}' \mathbf{V}^{-1} \mathbf{V}^{-1} \mathbf{w}}, \quad (10.30)$$

where the elements of the vector \mathbf{w} are $w_i = E[x_{(i)}]$ and \mathbf{V} is the covariance matrix of the order statistics $x_{(1)}, x_{(2)}, \dots, x_{(n)}$.

Example 10.16 \blacktriangleright **Shapiro-Wilk Normality Test** \blacktriangleleft Use the Shapiro-Wilk test with the random sample $\{47, 50, 57, 54, 52, 54, 53, 65, 62, 67, 69, 74, 51, 57, 57, 59\}$ to test for normality using $\alpha = 0.05$.

Solution: First, order the data:

$$47 \leq 50 \leq 51 \leq 52 \leq 53 \leq 54 = 54 \leq 57 = 57 = 57 \leq 59 \leq 62 \leq 65 \leq 67 \leq 69 \leq 74.$$

Next, calculate the differences $x_{(n-i+1)} - x_{(i)}$ for $i = 1, 2, \dots, \lfloor \frac{n}{2} \rfloor = 8$:

$$\left| \begin{array}{l} x_{(16)} - x_{(1)} = 74 - 47 = 27 \\ x_{(15)} - x_{(2)} = 69 - 50 = 19 \\ x_{(14)} - x_{(3)} = 67 - 51 = 16 \end{array} \right| \left| \begin{array}{l} x_{(13)} - x_{(4)} = 65 - 52 = 13 \\ x_{(12)} - x_{(5)} = 62 - 53 = 9 \\ x_{(11)} - x_{(6)} = 59 - 54 = 5 \end{array} \right| \left| \begin{array}{l} x_{(10)} - x_{(7)} = 57 - 54 = 3 \\ x_{(9)} - x_{(8)} = 57 - 57 = 0 \end{array} \right|$$

Looking at Table 6 from Shapiro and Wilk (1965) ($n = 16$ and $i = 1, \dots, 8$), one obtains

$$\left| \begin{array}{l} a_{16} = 0.5056 \\ a_{15} = 0.3290 \end{array} \right| \left| \begin{array}{l} a_{14} = 0.2521 \\ a_{13} = 0.1939 \end{array} \right| \left| \begin{array}{l} a_{12} = 0.1447 \\ a_{11} = 0.1005 \end{array} \right| \left| \begin{array}{l} a_{10} = 0.0593 \\ a_9 = 0.0196 \end{array} \right|$$

which means $b = \sum_{i=1}^8 a_{n-i+1}(x_{(n-i+1)} - x_{(i)}) = 28.4392$ and $nS_u^2 = 854$.

The Shapiro-Wilk test statistic value is then

$$W = \frac{b^2}{nS_u^2} = \frac{808.7881}{854} = 0.9471.$$

The critical value K with $\alpha = 0.05$ and $n = 16$ is 0.887. As $W_{obs} = 0.9471 > 0.887$, one fails to reject the null hypothesis of normality.

```
> x <- c(47, 50, 57, 54, 52, 54, 53, 65, 62, 67, 69, 74, 51, 57, 57, 59)
> shapiro.test(x)

Shapiro-Wilk normality test

data: x
W = 0.94705, p-value = 0.4445
```



10.8 Categorical Data Analysis

This section provides an overview of two common scenarios where categorical data are generated and explains how each scenario is analyzed. The basic 2×2 contingency table with fixed row totals was introduced in Section 9.9.3, Testing Equality of Proportions with Fisher's exact test. The 2×2 contingency table can be generalized for I rows and J columns and is referred to as an $I \times J$ contingency table. The sampling scheme employed to acquire the information in the table will determine the type of hypothesis that can be tested. Consider the following two scenarios:

SCENARIO ONE: Is there an association between gender and a person's happiness? To investigate whether happiness depends on gender, one might use information collected from the General Social Survey (GSS) (<http://sda.berkeley.edu/GSS>). In each survey, the GSS asks, "Taken all together, how would you say things are these days — would you say that you are very happy, pretty happy, or not too happy?" Respondents to each survey are coded as either male or female. The information in Table 10.21 shows how a subset of respondents (26-year-olds) were classified with respect to the variables HAPPY and SEX.

Table 10.21: Twenty-six-year-olds' happiness

SEX	HAPPY		
	Very happy	Pretty happy	Not too happy
Male	110	277	50
Female	163	302	63

SCENARIO TWO: In a double blind randomized drug trial (neither the patient nor the physician evaluating the patient knows the treatment, drug or placebo, the patient receives), 400 male patients with mild dementia were randomly divided into two groups of 200. One group was given a placebo over three months while the second group received an experimental drug for three months. At the end of the three months, the physicians (all psychiatrists) classified the 400 patients into one of three categories: improved, no change, or worse. The information in Table 10.22 shows how the psychiatrists classified the patients. Are the proportions in the three status categories the same for the two treatments?

Table 10.22: Mild dementia treatment results

Treatment	Status		
	Improve	No Change	Worse
Drug	67	76	57
Placebo	48	73	79

The two scenarios illustrate two different sampling schemes that both result in $I \times J$ contingency tables. In the first scenario, there is a single population (Americans) and individuals are sampled from this single population and classified into one of the IJ cells of the $I \times J$ contingency table based on the $I = 2$ SEX categories and the $J = 3$ HAPPY categories. The format of an $I \times J$ contingency table when sampling from a single population is shown in Table 10.23. The number of observations from the i^{th} row classified into the j^{th} column is denoted by n_{ij} . It follows that the number of observations in the j^{th} column ($1 \leq j \leq J$) is $n_{\bullet j} = n_{1j} + n_{2j} + \dots + n_{Ij}$, while the number of observations in the i^{th} row ($1 \leq i \leq I$) is $n_{i\bullet} = n_{i1} + n_{i2} + \dots + n_{iJ}$.

The true population proportion of individuals in cell (i, j) will be denoted π_{ij} . Under the assumption of independence between row and column variables (SEX and HAPPY in this example), $\pi_{ij} = \pi_{i\bullet} \times \pi_{\bullet j}$, where $\pi_{i\bullet} = \sum_{j=1}^J \pi_{ij}$ and $\pi_{\bullet j} = \sum_{i=1}^I \pi_{ij}$. That is, $\pi_{i\bullet}$ is the proportion of observations in the population classified in category i of the row variable and $\pi_{\bullet j}$ is the proportion of observations in the population classified in category j of the column variable. Since $\pi_{i\bullet}$ and $\pi_{\bullet j}$ are marginal population proportions, it follows that $\hat{\pi}_{i\bullet} = p_{i\bullet} = \frac{n_{i\bullet}}{n}$ and $\hat{\pi}_{\bullet j} = p_{\bullet j} = \frac{n_{\bullet j}}{n}$, where n is the sample size. Under the assumption of independence the expected count for cell (i, j) is $\mu_{ij} = n\pi_{ij} = n\pi_{i\bullet}\pi_{\bullet j}$ and $\hat{\mu}_{ij} = n\hat{\pi}_{ij} = n\hat{\pi}_{i\bullet}\hat{\pi}_{\bullet j} = n\frac{n_{i\bullet}}{n}\frac{n_{\bullet j}}{n} = \frac{n_{i\bullet}n_{\bullet j}}{n}$.

Table 10.23: Contingency table when sampling from a single population

	Col 1	Col 2	...	Col J	Totals
Row 1	n_{11}	n_{12}	...	n_{1J}	$n_{1\bullet}$
Row 2	n_{21}	n_{22}	...	n_{2J}	$n_{2\bullet}$
:	:	:		:	:
Row I	n_{I1}	n_{I2}	...	n_{IJ}	$n_{I\bullet}$
Totals	$n_{\bullet 1}$	$n_{\bullet 2}$...	$n_{\bullet J}$	n

In the second scenario, there are two distinct populations from which samples are taken. The first population is the group of all patients receiving the experimental drug while the second population is the group of all patients receiving a placebo. In this scenario, there are $I = 2$ separate populations and $J = 3$ categories for the $I = 2$ populations. Individuals sampled from the $I = 2$ distinct populations are classified into one of the $J = 3$ status categories. This scenario has fixed row totals whereas the first scenario does not. In the first scenario, only the total sample size, n , is fixed. That is, neither the row nor the column totals are fixed. This is in contrast to scenario two, where the number of patients in each treatment group (row) was fixed. The notation used for an $I \times J$ contingency table when I samples from I distinct populations differs slightly from the notation used in Table 10.23

on the facing page with a contingency table from a single sample.

Since the sample sizes of the I distinct populations are denoted $n_{i\bullet}$, the total for all I samples is denoted by $n_{\bullet\bullet}$ rather than the notation n used for a single sample in Table 10.23 on the preceding page. Table 10.24 shows the general form and notation used for an $I \times J$ contingency table when sampling from I distinct populations. Each observation in each sample is classified into one of J categories. If $n_{i\bullet}$ denotes the number of observations in the i^{th} sample ($1 \leq i \leq I$) and n_{ij} denotes the number of observations from the i^{th} sample classified into the j^{th} category ($1 \leq j \leq J$), it follows that the number of observations in the j^{th} column is $n_{\bullet j} = n_{1j} + n_{2j} + \dots + n_{Ij}$, while the number of observations in the i^{th} row is $n_{i\bullet} = n_{i1} + n_{i2} + \dots + n_{iJ}$.

Table 10.24: General form and notation used for an $I \times J$ contingency table when sampling from I distinct populations

	Category 1	Category 2	\dots	Category J	Totals
Population 1	n_{11}	n_{12}	\dots	n_{1J}	$n_{1\bullet}$
Population 2	n_{21}	n_{22}	\dots	n_{2J}	$n_{2\bullet}$
\vdots	\vdots	\vdots		\vdots	\vdots
Population I	n_{I1}	n_{I2}	\dots	n_{IJ}	$n_{I\bullet}$
Totals	$n_{\bullet 1}$	$n_{\bullet 2}$	\dots	$n_{\bullet J}$	$n_{\bullet\bullet}$

10.8.1 Test of Independence

Scenario one asks if there is an association between gender and a person's happiness. In Section 3.3.6 on page 213, two events, A and B , were defined as independent when $\mathbb{P}(A \cap B) = \mathbb{P}(A) \times \mathbb{P}(B)$ or, equivalently, when $\mathbb{P}(A|B) = \mathbb{P}(A)$. If, instead of having a random sample from a single population, an $I \times J$ contingency table consisted of entries from the population, association could be mathematically verified by showing that $\mathbb{P}(n_{ij}) \neq \mathbb{P}(n_{i\bullet}) \times \mathbb{P}(n_{\bullet j})$ for some i and j . If by chance $\mathbb{P}(n_{ij}) = \mathbb{P}(n_{i\bullet}) \times \mathbb{P}(n_{\bullet j})$ for all i and j , then one would conclude there is no association between gender and a person's happiness. That is, the variables gender and happiness would be considered mathematically independent. Since the entire population is not given but rather a sample from a population, the values in the $I \times J$ contingency table can be expected to change from sample to sample. The question is, "By how much can the variables deviate from the mathematical definition of independence and still be considered statistically independent?"

The null and alternative hypotheses to test for independence between row and column variables is written $H_0 : \pi_{ij} = \pi_{i\bullet}\pi_{\bullet j}$ versus $H_1 : \pi_{ij} \neq \pi_{i\bullet}\pi_{\bullet j}$. The test statistic is

$$\chi_{\text{obs}}^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(O_{ij} - E_{ij})^2}{E_{ij}}. \quad (10.31)$$

It compares the observed frequencies in the table with the expected frequencies when H_0 is true. Under the assumption of independence, and when the observations in the cells are sufficiently large (usually greater than 5), $\chi_{\text{obs}}^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(n_{ij} - \hat{\mu}_{ij})^2}{\hat{\mu}_{ij}} \sim \chi_{(I-1)(J-1)}^2$, where $\hat{\mu}_{ij} = \frac{n_{i\bullet}n_{\bullet j}}{n} = E_{ij}$ and $n_{ij} = O_{ij}$. The null hypothesis of independence is rejected

when $\chi_{\text{obs}}^2 > \chi_{1-\alpha; (I-1)(J-1)}^2$.

The chi-squared approximation is generally satisfactory if the E_{ij} s ($\hat{\mu}_{ij}$ s) in the test statistic are not too small. Various rules of thumb exist for what might be considered too small. A very conservative rule is to require all E_{ij} s to be 5 or more. This can be accomplished by combining cells with small E_{ij} s and reducing the overall degrees of freedom. At times, it may be permissible to let the E_{ij} of a cell be as low as 0.5.

Test for SCENARIO ONE:

Step 1: **Hypotheses** — $H_0 : \pi_{ij} = \pi_{i\bullet}\pi_{\bullet j}$ (Row and column variables are independent.) versus $H_1 : \pi_{ij} \neq \pi_{i\bullet}\pi_{\bullet j}$ for at least one i, j (Row and column variables are dependent.)

Step 2: **Test Statistic** — The test statistic is

$$\chi_{\text{obs}}^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \sim \chi_{(I-1)(J-1)}^2 = \chi_{(2-1)(3-1)}^2 = \chi_2^2$$

under the assumption of independence. The χ_{obs}^2 value is 4.3215.

Step 3: **Rejection Region Calculations** — The rejection region is

$$\chi_{\text{obs}}^2 > \chi_{1-\alpha; (I-1)(J-1)}^2 = \chi_{0.95; 2}^2 = 5.9915.$$

Before the statistic $\chi_{\text{obs}}^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$ can be computed, the expected counts for each of the ij cells must be calculated. Note that $O_{ij} = n_{ij}$ and $E_{ij} = \frac{n_{i\bullet}n_{\bullet j}}{n}$.

```

> HA <- c(110, 277, 50, 163, 302, 63)
> HAT <- matrix(data = HA, nrow = 2, byrow = TRUE)
> dimnames(HAT) <- list(SEX = c("Male", "Female"),
+   Category = c("Very Happy", "Pretty Happy", "Not To Happy"))
> HAT

      Category
SEX      Very Happy Pretty Happy Not To Happy
  Male        110          277          50
  Female       163          302          63

> E <- outer(rowSums(HAT), colSums(HAT), "*")/sum(HAT)
> E

      Very Happy Pretty Happy Not To Happy
  Male     123.628      262.2      51.17202
  Female    149.372      316.8      61.82798

> # OR
> chisq.test(HAT)$expected

      Category
SEX      Very Happy Pretty Happy Not To Happy
  Male     123.628      262.2      51.17202
  Female    149.372      316.8      61.82798

```

$$\chi_{\text{obs}}^2 = \frac{(110 - 123.6280)^2}{123.6280} + \frac{(277 - 262.2)^2}{262.2} + \cdots + \frac{(63 - 61.828)^2}{61.828} = 4.3215.$$

The value of the test statistic is $\chi_{\text{obs}}^2 = 4.3215$. This can be done with code by entering

```
> chi.obs <- sum((HAT - E)^2/E)
> chi.obs
[1] 4.321482
```

$$4.3215 = \chi_{\text{obs}}^2 > ? \chi_{0.95,2}^2 = 5.9915.$$

Step 4: **Statistical Conclusion** — The p -value is 0.1152.

```
> p.val <- pchisq(chi.obs, 2, lower = FALSE)
> p.val
[1] 0.1152397
```

- I. From the rejection region, since $\chi_{\text{obs}}^2 = 4.3215 < \chi_{0.95,2}^2 = 5.9915$, fail to reject the null hypothesis of independence.
- II. Since the p -value = 0.1152 is greater than 0.05, fail to reject the null hypothesis of independence.

Fail to reject H_0 .

Step 5: **English Conclusion** — There is not sufficient evidence to suggest the variables gender and happiness are statistically dependent.

The function `chisq.test()` can also be used to test the null hypothesis of independence.

```
> chisq.test(HAT)

Pearson's Chi-squared test

data: HAT
X-squared = 4.3215, df = 2, p-value = 0.1152
```

10.8.2 Test of Homogeneity

The question of interest in scenario two is whether the proportions in each of the $j = 3$ categories for the $i = 2$ populations are equivalent. Specifically, is $\pi_{1j} = \pi_{2j}$ for all j ? This question is answered with a test of homogeneity. In general, the null hypothesis for a test of homogeneity with $i = I$ populations is written

$$H_0 : \pi_{1j} = \pi_{2j} = \cdots = \pi_{Ij} \text{ for all } j \text{ versus } H_1 : \pi_{ij} \neq \pi_{i+1,j} \text{ for some } (i, j). \quad (10.32)$$

Expressed in words, the null hypothesis is that the I populations are homogeneous with respect to the J categories versus the I populations are not homogeneous with respect to the J categories. An equivalent interpretation is that for each population $i = 1, 2, \dots, I$, the proportion of people in the j^{th} category is the same. When H_0 is true, $\pi_{1j} = \pi_{2j} = \dots = \pi_{Ij}$ for all j . Under the null hypothesis, $\mu_{ij} = n_{i\bullet}\pi_{ij}$, $\hat{\pi}_{ij} = p_{ij} = \frac{n_{ij}}{n_{\bullet\bullet}}$, and $\hat{\mu}_{ij} = \frac{n_{i\bullet}n_{\bullet j}}{n_{\bullet\bullet}} = E_{ij}$. When H_0 is true, all the probabilities in the j^{th} column are equal, and a pooled estimate of π_{ij} is obtained by adding all the frequencies in the j^{th} column ($n_{\bullet j}$) and dividing the total by $n_{\bullet\bullet}$. The statistic used in this type of problem has the same form as the one used for the test of independence in (10.31). Substituting the homogeneity expressions for O_{ij} and E_{ij} , the statistic is expressed as

$$\chi_{\text{obs}}^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(n_{ij} - n_{i\bullet}n_{\bullet j}/n_{\bullet\bullet})^2}{n_{i\bullet}n_{\bullet j}/n_{\bullet\bullet}} \sim \chi_{(I-1)(J-1)}^2.$$

The null hypothesis of homogeneity is rejected when $\chi_{\text{obs}}^2 > \chi_{1-\alpha;(I-1)(J-1)}^2$.

When row and column totals are not fixed, the numbers in the i, j cells can be used to estimate their corresponding population proportions without assuming the null hypothesis is true. With fixed row or column totals, this estimation cannot be accomplished. That is, $\hat{\pi}_{ij} = p_{ij} \neq \frac{n_{ij}}{n_{\bullet\bullet}}$ when H_0 is false.

Test for SCENARIO TWO:

Step 1: **Hypotheses** — $H_0 : \pi_{1j} = \pi_{2j}$ for all j versus $H_1 : \pi_{i,j} \neq \pi_{i+1,j}$ for some (i, j) . That is, all the probabilities in the same column are equal to each other versus at least two of the probabilities in the same column are not equal to each other.

Step 2: **Test Statistic** — The test statistic is

$$\chi_{\text{obs}}^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \sim \chi_{(I-1)(J-1)}^2 = \chi_{(2-1)(3-1)}^2 = \chi_2^2$$

under the null hypothesis. The χ_{obs}^2 value is 6.7584.

Step 3: **Rejection Region Calculations** — The rejection region is

$$\chi_{\text{obs}}^2 > \chi_{1-\alpha;(I-1)\cdot(J-1)}^2 = \chi_{0.95;2}^2 = 5.9915.$$

Before the statistic $\chi_{\text{obs}}^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$ can be computed, the expected counts for each of the ij cells must be determined. Recall that $O_{ij} = n_{ij}$ and $E_{ij} = \frac{n_{i\bullet}n_{\bullet j}}{n_{\bullet\bullet}}$.

```
> DT <- c(67, 76, 57, 48, 73, 79)
> DTT <- matrix(data = DT, nrow = 2, byrow = TRUE)
> dimnames(DTT) <- list(Treatment = c("Drug", "Placebo"),
+ Category = c("Improve", "No Change", "Worse"))
> DTT
```

	Category		
Treatment	Improve	No Change	Worse
Drug	67	76	57
Placebo	48	73	79

```
> E <- chisq.test(DTT)$expected
> E

      Category
Treatment Improve No Change Worse
  Drug       57.5     74.5    68
  Placebo    57.5     74.5    68
```

$$\chi_{\text{obs}}^2 = \frac{(67 - 57.5)^2}{57.5} + \frac{(76 - 74.5)^2}{74.5} + \dots + \frac{(79 - 68)^2}{68} = 6.7584.$$

The value of the test statistic is $\chi_{\text{obs}}^2 = 6.7584$. This can be done with code by entering

```
> chi.obs <- sum((DTT - E)^2/E)
> chi.obs

[1] 6.758357
```

$$6.7584 = \chi_{\text{obs}}^2 > ? \chi_{.95,2}^2 = 5.9915.$$

Step 4: **Statistical Conclusion** — The p -value is 0.03408.

```
> p.val <- pchisq(chi.obs, 2, lower = FALSE)
> p.val

[1] 0.03407544
```

- I. From the rejection region, since $\chi_{\text{obs}}^2 = 6.7584 > \chi_{0.95;2} = 5.9915$, reject the null hypothesis of homogeneity.
- II. Since the p -value = 0.0341 is less than 0.05, reject the null hypothesis of homogeneity.

Reject H_0 .

Step 5: **English Conclusion** — There is sufficient evidence to suggest that not all of the probabilities for the $i = 2$ populations with respect to each of the J categories are equal.

Using `chisq.test()` directly produces the same results.

```
> chisq.test(DTT)

Pearson's Chi-squared test

data: DTT
X-squared = 6.7584, df = 2, p-value = 0.03408
```



10.9 Nonparametric Bootstrapping

The term “bootstrapping” is due to Efron (1979), and is an allusion to a German legend about a Baron Münchhausen, who was able to lift himself out of a swamp by pulling himself up by his own hair. In later versions he was using his own boot straps to pull himself out of the sea, which gave rise to the term bootstrapping. As improbable as it may seem, taking samples from the original data and using these **resamples** to calculate statistics can actually give more accurate answers than using the single original sample to calculate an estimate of a parameter. In fact, resampling methods require fewer assumptions than the traditional methods found in Chapters 7 and 8 and sometimes give more accurate answers. One of the more common methods of resampling is the bootstrap. The fundamental concept in bootstrapping is the building of a sampling distribution for a particular statistic by resampling from the data that are at hand. Although bootstrap methods are both parametric and nonparametric, attention in this section is focused exclusively on the nonparametric bootstrap. These methods offer the practitioner valuable tools for dealing with complex problems with computationally intensive techniques that rely on today’s computers, which are many times faster than those of a generation ago. Even though resampling procedures rely on the “new” power of the computer to perform simulations, they are based on the “old” statistical principles such as populations, parameters, samples, sampling variation, pivotal quantities, and confidence intervals.

For most students, the idea of a sampling distribution for a particular statistic is completely abstract; however, once work begins with the bootstrap distribution, the bootstrap analog to the sampling distribution, the concreteness of the bootstrap distribution promotes a conceptual understanding of the more abstract sampling distribution. In fact, bootstrap procedures promote statistical thinking by providing concrete analogies to theoretical concepts.

Two R packages for bootstrapping are **bootstrap** by Efron and Tibshirani (1993) (ported to R from S-PLUS by Friedrich Leisch) and **boot** by Canty and Ripley (2015). The package **boot** provides functions and data sets for the book *Bootstrap Methods and Their Applications* by Davison and Hinkley (1997) and is used for several examples in the remainder of this chapter.

10.9.1 Bootstrap Paradigm

Suppose a random sample $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ is taken from an unknown probability distribution, F , and the values $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ are observed. Using \mathbf{x} , the parameter $\theta = t(F)$ is to be estimated. The traditional approach of estimating θ is to make some assumptions about the population structure and to derive the sampling distribution of $\hat{\theta}$ based on these assumptions. This, of course, assumes the derivation of the sampling distribution of the statistic of interest has either been done or that the individual who needs to do the deriving has the mathematical acumen to do so. Often, the use of the bootstrap will be preferable to extensive mathematical calculations.

In the bootstrap paradigm, the original sample, \mathbf{x} , takes the place the population holds in the traditional approach. Subsequently, a random sample of size n is drawn from \mathbf{x} with replacement. The resampled values are called a bootstrap sample and are denoted \mathbf{x}^* . These values are used to calculate an estimate of the statistic of interest, $s(\mathbf{x}) = \hat{\theta}$. This $s(\mathbf{x})$ is not necessarily the plug-in estimate of θ , $\hat{\theta} = t(\hat{F})$, where \hat{F} is the empirical cumulative distribution function as defined in (3.5). It is, however, the function applied to

the bootstrap sample \mathbf{x}^* that creates a bootstrap estimate of θ denoted $\hat{\theta}^*$ or t^* . That is,

$$s(\mathbf{x}^*) = t^* = \hat{\theta}^*. \quad (10.33)$$

The star notation indicates that \mathbf{x}^* is not the original data set \mathbf{x} , but rather, it is a random sample of size n drawn with replacement from \mathbf{x} . That is, given $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$, one possible bootstrap sample \mathbf{x}^* might be $\{x_1^* = x_5, x_2^* = x_3, x_3^* = x_5, \dots, x_n^* = x_7\}$. Some values from the original sample \mathbf{x} may appear once, more than once, or not at all in the bootstrap sample \mathbf{x}^* .

The fundamental bootstrap assumption is that the sampling distribution of the statistic under the unknown probability distribution F may be approximated by the sampling distribution of $\hat{\theta}^*$ under the empirical cumulative distribution function \hat{F} . That is,

$$\begin{aligned} \text{Var}_F(\hat{\theta}) &\approx \text{Var}_{\hat{F}}(\hat{\theta}^*) \\ G_F(a) &\approx G_{\hat{F}}(a) \\ G_F^{-1}(0.95) &\approx G_{\hat{F}}^{-1}(0.95) \end{aligned} \quad (10.34)$$

where G is the cumulative distribution function of the distribution of $\hat{\theta}$. Generally, the bootstrap estimate of the parameter of interest is not computed directly, but it is instead estimated from B bootstrap samples.

The process of creating a bootstrap sample \mathbf{x}^* and a bootstrap estimate $\hat{\theta}^*$ of the parameter of interest is repeated B times (typically $10^4 - 1$ or more). The B bootstrap estimates of θ , the $\hat{\theta}^*$'s, are subsequently used to estimate specific properties of the bootstrap sampling distribution of $\hat{\theta}^*$. There are a total of n^n bootstrap samples where order matters and $\binom{2n-1}{n}$ bootstrap samples where order does not matter (distinct). Yet, a reasonable estimate of the standard error of $\hat{\theta}^*$, $\hat{\sigma}_{\hat{\theta}^*} \equiv \widehat{\text{SE}}_B$, can be achieved with only $B = 200$ bootstrap replications in most problems (Efron and Tibshirani, 1993). For confidence intervals and quantile estimation, B generally should be at least $10^4 - 1$ (Chihara and Hesterberg, 2011). Although one might use the unordered (distinct) bootstrap samples for exhaustive calculations, it is much easier to program the random sampling procedure where order matters.

The general procedure for estimating the standard error of $\hat{\theta}^*$ is

- (1) Generate B independent bootstrap samples $\{\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_B^*\}$, each consisting of n values drawn with replacement from \mathbf{x} .
- (2) Compute the statistic of interest for each bootstrap sample:

$$\hat{\theta}_b^* \equiv t_b^* = s(\mathbf{x}_b^*) \quad b = 1, 2, \dots, B.$$

- (3) Estimate the standard error of $\hat{\theta}$ ($\text{SE}_F(\hat{\theta}) \equiv \sigma_{\hat{\theta}}$) by computing the sample standard deviation of the bootstrap replications of $\hat{\theta}_b^*$, $b = 1, 2, \dots, B$:

$$\sigma_{\hat{\theta}} \approx \widehat{\text{SE}}_B \equiv \hat{\sigma}_{\hat{\theta}^*} = \left[\sum_{b=1}^B \frac{(\hat{\theta}_b^* - \bar{\hat{\theta}}^*)^2}{B-1} \right]^{\frac{1}{2}}, \text{ where } \bar{\hat{\theta}}^* = \sum_{b=1}^B \frac{\hat{\theta}_b^*}{B}. \quad (10.35)$$

The bootstrap algorithm for estimating the standard error of a statistic $\hat{\theta} = s(\mathbf{x})$ is graphically depicted in Figure 10.16.

For most statistics, bootstrap distributions approximate the shape, spread, and bias of the actual sampling distribution; however, bootstrap distributions differ from the actual

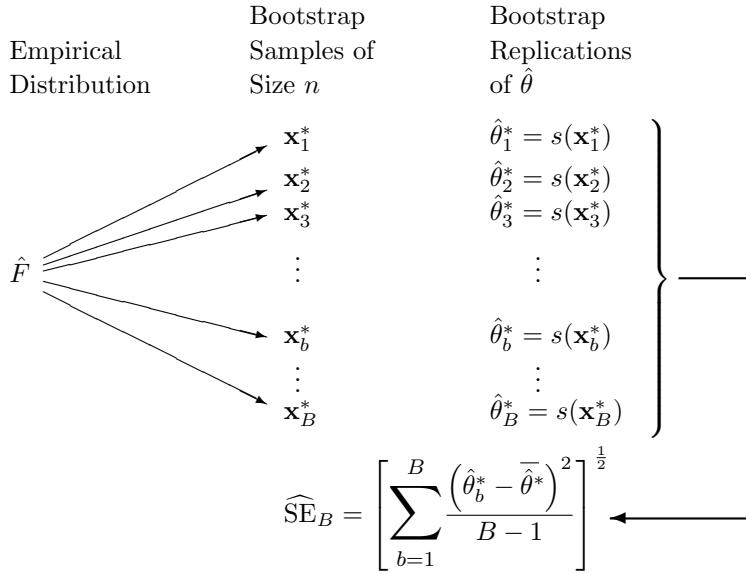


FIGURE 10.16: Graphical representation of the bootstrap based on Efron and Tibshirani (1993, Figure 6.1)

sampling distribution in the locations of their centers. The bootstrap distribution having a similar shape is clear. A similar spread means

$$\text{Var}[s(\mathbf{X})|F] \approx \text{Var}[s(\mathbf{X})|\hat{F}].$$

That is, the variance of the estimator $s(\mathbf{X})$ under the unknown distribution F is approximately the same as the variance of $s(\mathbf{X})$ under the bootstrap distribution obtained by replacing F with \hat{F} . The variance of $s(\mathbf{X})$ under \hat{F} is not computed, but rather estimated from the B bootstrap samples and is

$$\text{Var}[s(\mathbf{X})|\hat{F}] \approx \widehat{\text{Var}}_B[s(\mathbf{X})|\hat{F}] = \sum_{b=1}^B \frac{(\hat{\theta}_b^* - \bar{\theta}^*)^2}{B-1}.$$

The sampling distribution of a statistic $s(\mathbf{X})$ used to estimate the parameter $\theta = t(F)$ is centered at the parameter θ plus any bias, while the bootstrap distribution is centered at $\hat{\theta}$ plus any bias. Recall that the bias of a statistic $\hat{\theta}$ is $E(\hat{\theta}) - \theta$. Consequently, the bias of $s(\mathbf{X}) = \hat{\theta}$ is expressed as

$$\text{Bias}[s(\mathbf{X})|F] = E_F[s(\mathbf{X})] - t(F),$$

while the bias of the bootstrap distribution of $\hat{\theta}$ is

$$\text{Bias}[s(\mathbf{X})|\hat{F}] = E_{\hat{F}}[s(\mathbf{X}^*)] - t(\hat{F}).$$

Generally, the bootstrap bias of $s(\mathbf{X})$ is not computed directly but is instead estimated from B bootstrap samples. That is, $E_{\hat{F}}[s(\mathbf{X}^*)]$ is not computed directly but is estimated by

$$\hat{E}_{\hat{F}}[s(\mathbf{X}^*)] = \sum_{b=1}^B \frac{\hat{\theta}_b^*}{B} = \bar{\theta}^*.$$

The result is an estimated bootstrap bias of $s(\mathbf{X})$ based on B bootstrap samples denoted

$$\widehat{Bias}_B[s(\mathbf{X})] = \bar{\theta}^* - \hat{\theta}. \quad (10.36)$$

Consider Figure 10.17 on the following page where three samples each of size $n = 225$ are taken from a $N(100, 15)$ population and represented with density estimates in the second, third, and fourth rows of the first column of graphs. The solid line in all graphs represents the theoretical mean of 100. Since one is sampling from a normal distribution, it is known that the resulting sampling distribution of the sample mean will have a $N(100, 1)$ distribution, which is represented in the top right graph of Figure 10.17 on the next page. Repeated samples ($B = 10^4 - 1$) of size $n = 225$ were taken from each of the graphs depicted in rows two, three, and four of the first column of Figure 10.17 on the following page using the R function `sample()`. The mean was applied to each bootstrap sample and the resulting bootstrap sampling distribution of the sample mean is depicted directly to the right of the sample from which it was taken.

The shape of the bootstrapped distribution of \bar{X} in rows two, three, and four of the second column of Figure 10.17 on the next page resembles the theoretical shape of the sampling distribution of \bar{X} , shown in the top right of Figure 10.17 on the following page. The standard deviation of the sampling distribution of \bar{X} , $\sigma_{\bar{X}} = 1$, while the standard deviations of the bootstrapped distributions ($\hat{\sigma}_{\bar{x}^*}$) for the graphs depicted in rows two, three, and four of the second column of Figure 10.17 on the next page are 0.9688, 0.9663, and 0.9421, respectively. Note that the $Bias[\bar{X}] = E(\bar{X}) - \mu = 0$, and that the bootstrap biases for the graphs depicted in rows two, three, and four of the second column of Figure 10.17 on the following page are -8e-04, -0.0021, and -0.0044, respectively. Although the bootstrap distributions are not centered at $\mu = 100$, they do resemble the shape of the theoretical sampling distribution for \bar{X} , have similar spreads when compared to the theoretical sampling distribution of \bar{X} , and have biases very close to zero.

Example 10.17  **Bootstrap Distributions**  Given a random variable X , which is $\Gamma(\alpha = 1, \lambda = 1/3)$, generate three random samples each of size $n = 100$ from X . Store the random samples in objects named `rsg1`, `rsg2`, and `rsg3`, respectively. Generate $B = 10^4 - 1$ bootstrap samples from `rsg1`, `rsg2`, and `rsg3`. (Recall a bootstrap sample is a sample of the same size as the original sample taken with replacement.) Apply the mean to each bootstrap sample and store the means of the bootstrapped samples from `rsg1`, `rsg2`, and `rsg3` in the objects `ThetaHatStar1`, `ThetaHatStar2`, and `ThetaHatStar3`, respectively.

- (a) What is the exact sampling distribution of \bar{X} ?
- (b) Compute the estimated bootstrap bias of \bar{X} , $\widehat{Bias}_B[\bar{X}] = \bar{X}^* - \bar{X}$ using the values in `ThetaHatStar1`, `ThetaHatStar2`, and `ThetaHatStar3`.
- (c) Compute the estimated bootstrap standard error of \bar{X} using the values in `ThetaHatStar1`, `ThetaHatStar2`, and `ThetaHatStar3`.
- (d) Graph the distributions of X , \bar{X} , `rsg1`, `rsg2`, `rsg3`, `ThetaHatStar1`, `ThetaHatStar2`, and `ThetaHatStar3`. Specifically, place density estimates of `rsg1`, `rsg2`, and `rsg3` directly beneath the distribution of X . Use a solid vertical line to denote the $E[X]$ for all four graphs. Place a dashed vertical line in the density estimates of `rsg1`, `rsg2`, and `rsg3` at the mean of the values in the respective objects. Place density estimates of `ThetaHatStar1`, `ThetaHatStar2`, and `ThetaHatStar3` directly beneath the distribution of \bar{X} . Use a solid vertical line to denote the $E[\bar{X}]$ for all four graphs. Place a dashed vertical line in the density estimates of `ThetaHatStar1`, `ThetaHatStar2`, and `ThetaHatStar3` at the mean of the values in the respective objects.

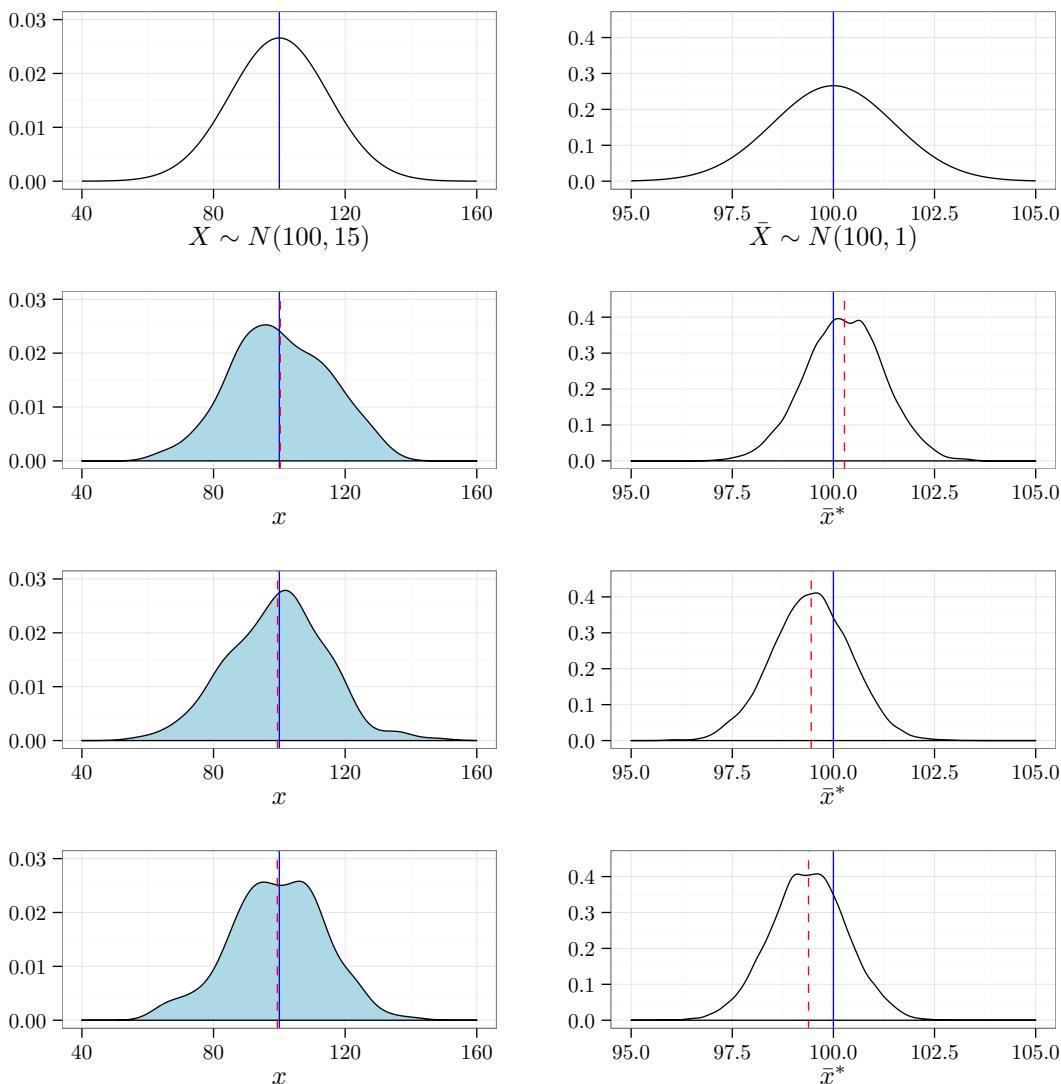


FIGURE 10.17: The top left graph shows a $N(100, 15)$ distribution with the solid vertical line marking the mean (100) of the distribution. The graphs in the first column in the second, third, and fourth rows are density estimates based on taking random samples of size $n = 225$ from the $N(100, 15)$ distribution. The top graph in the second column shows the theoretical sampling distribution of \bar{X} for samples of size $n = 225$ when sampling from a $N(100, 1)$ distribution ($N(100, 1)$). The graphs in the second, third, and fourth rows of the second column are the bootstrap distributions of \bar{X} created from bootstrapping the samples immediately to each graph's left. The dashed vertical lines in all graphs are drawn at the mean of the values used to create that graph.

(e) Discuss the bias, shape, and spread of the bootstrap distributions of \bar{X} .

Solution: The bootstrap distribution of the mean is approximated using the function `sample()` inside a `for()` loop. R Code 10.25 on the next page performs the requested tasks. R Code 10.26 on the facing page performs the requested bootstrapping using the package `boot`. To use the function `boot()`, one must first define a function that takes at

least the arguments `data=`, and `i=`. Although the same seed is used in both R Code 10.25 and R Code 10.26, there are slight differences in the programming of how the bootstrap samples are generated, which produce different but not significantly different output.

R Code 10.25

```
> set.seed(14)
> alpha <- 1
> lambda = 1/3
> MX <- alpha/lambda
> VX <- alpha/lambda^2
> n <- 100
> rsg1 <- rgamma(n, alpha, lambda)
> rsg2 <- rgamma(n, alpha, lambda)
> rsg3 <- rgamma(n, alpha, lambda)
> B <- 10^4 - 1
> ThetaHatStar1 <- numeric(B)
> ThetaHatStar2 <- numeric(B)
> ThetaHatStar3 <- numeric(B)
> for (i in 1:B) {
+   bootsample1 <- sample(rsg1, size = n, replace = TRUE)
+   bootsample2 <- sample(rsg2, size = n, replace = TRUE)
+   bootsample3 <- sample(rsg3, size = n, replace = TRUE)
+   ThetaHatStar1[i] <- mean(bootsample1)
+   ThetaHatStar2[i] <- mean(bootsample2)
+   ThetaHatStar3[i] <- mean(bootsample3)
+ }
```

R Code 10.26

```
> library(boot)
> GammaMean <- function(data, i) {
+   d <- data[i]
+   M <- mean(d)
+ }
> set.seed(14)
> B <- 10^4 - 1
> b.obj1 <- boot(data = rsg1, statistic = GammaMean, R = B)
> b.obj2 <- boot(data = rsg2, statistic = GammaMean, R = B)
> b.obj3 <- boot(data = rsg3, statistic = GammaMean, R = B)
```

(a) Recall property 4 from from 285, which states that the sampling distribution for \bar{X} when sampling from $a\Gamma(\alpha, \lambda)$ is $\Gamma(n\alpha, n\lambda)$. In this problem since $n = 100$, the exact sampling distribution of \bar{X} is $\Gamma(n\alpha = 100 \cdot 1 = 100, n\lambda = 100 \cdot 1/3 = 100/3)$.

(b) The estimated bootstrap bias of \bar{X} , $\widehat{Bias}_B[\bar{X}] = \bar{X}^* - \bar{X}$ using the values in `ThetaHatStar1`, `ThetaHatStar2`, and `ThetaHatStar3` is computed in R Code 10.27.

R Code 10.27

```
> BootBias1 <- mean(ThetaHatStar1) - mean(rsg1)
> BootBias2 <- mean(ThetaHatStar2) - mean(rsg2)
```

```
> BootBias3 <- mean(ThetaHatStar3) - mean(rsg3)
> c(BootBias1, BootBias2, BootBias3)

[1] -0.0051374958 0.0017221531 -0.0009370108
```

$$\begin{aligned}_1\widehat{Bias}_B[\bar{X}] &= \bar{X}^* - \bar{X} = 2.8681 - 2.8733 = -0.0051 _2\widehat{Bias}_B[\bar{X}] &= \bar{X}^* - \bar{X} = 3.22 - 3.2183 = 0.0017 _3\widehat{Bias}_B[\bar{X}] &= \bar{X}^* - \bar{X} = 2.9766 - 2.9775 = -9e-04\end{aligned}$$

(c) The estimated bootstrap standard error of \bar{X} using the values in `ThetaHatStar1`, `ThetaHatStar2`, and `ThetaHatStar3` is computed in R Code 10.28. From part (a), \bar{X} is $\Gamma(n\alpha = 100 \cdot 1 = 100, n\lambda = 100 \cdot 1/3 = 100/3)$, and from (4.16) it is known that if $X \sim \Gamma(\alpha, \lambda)$, the $Var[X] = \frac{\alpha}{\lambda^2}$; therefore $\sigma_{\bar{X}} = \sqrt{100/(\frac{100}{3})^2} = 0.3$.

R Code 10.28

```
> sigxbar <- sqrt(alpha/(n * lambda^2))
> sigxbar

[1] 0.3

> c(sd(ThetaHatStar1), sd(ThetaHatStar2), sd(ThetaHatStar3))

[1] 0.3314342 0.2705354 0.2895893
```

The estimated standard errors for each of the three bootstrapped samples are

$$\begin{aligned}_1\widehat{SE}_B &= 0.3314, _2\widehat{SE}_B &= 0.2705, \text{ and} _3\widehat{SE}_B &= 0.2896.\end{aligned}$$

The function `boot()` stores the bootstrapped statistic in the object `t`, and the observed value of the user-defined statistic applied to the values in `data` in the object `t0`. The estimated bootstrap bias of \bar{X} , $\widehat{Bias}_B[\bar{X}] = \bar{X}^* - \bar{X}$ using the values in `b.obj1`, `b.obj2`, and `b.obj3` is computed in R Code 10.29. The estimated bootstrap standard error of \bar{X} using the values in `b.obj1`, `b.obj2`, and `b.obj3` is also computed in R Code 10.29.

R Code 10.29

```
> BootBias1b <- mean(b.obj1$t) - b.obj1$t0
> BootBias2b <- mean(b.obj2$t) - b.obj2$t0
> BootBias3b <- mean(b.obj3$t) - b.obj3$t0
> c(BootBias1b, BootBias2b, BootBias3b)

[1] 0.001601065 -0.001924439 -0.002356920

> c(sd(b.obj1$t), sd(b.obj2$t), sd(b.obj3$t))

[1] 0.3308901 0.2711390 0.2887591
```

Note: The results computed in R Code 10.29 on the preceding page are automatically computed and displayed when using the function `boot()` as shown in R Code 10.30.

R Code 10.30

```
> b.obj1
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP
```

```
Call:
```

```
boot(data = rsg1, statistic = GammaMean, R = B)
```

```
Bootstrap Statistics :  
    original      bias      std. error  
t1* 2.873255  0.001601065   0.3308901
```

```
> b.obj2
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP
```

```
Call:
```

```
boot(data = rsg2, statistic = GammaMean, R = B)
```

```
Bootstrap Statistics :  
    original      bias      std. error  
t1* 3.21828 -0.001924439   0.271139
```

```
> b.obj3
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP
```

```
Call:
```

```
boot(data = rsg3, statistic = GammaMean, R = B)
```

```
Bootstrap Statistics :  
    original      bias      std. error  
t1* 2.977524 -0.00235692   0.2887591
```

(d) Figure 10.18 on page 665 was created with code similar to that in R Code 10.31 on the following page. The package `gridExtra` is used to arrange the `ggplot2` graphs into two columns on the same graphics device. The range on the x -axis for all graphs in the first column is 0 to $\mu_X + 4\sigma_X$ while the range for all graphs in the second column is $\mu_{\bar{X}} \pm 5\sigma_{\bar{X}}$.

R Code 10.31

```

> library(ggplot2)
> p1 <- ggplot(data = data.frame(x = c(0, MX + 4*sqrt(VX))), aes(x = x)) +
+   stat_function(fun = dgamma, args = list(alpha, lambda)) +
+   geom_vline(xintercept = MX, lty = "solid", col = "blue")
> p2 <- ggplot(data = data.frame(x = c(MX - 5*sqrt(VX/n),
+                                         MX + 5*sqrt(VX/n))), aes(x = x)) +
+   stat_function(fun = dgamma, args = list(n*alpha, n*lambda)) +
+   geom_vline(xintercept = MX, lty = "solid", col = "blue")
> p3 <- ggplot(data = data.frame(x = rsg1), aes(x = x)) +
+   geom_density(fill = "lightblue") +
+   geom_vline(xintercept = c(MX, mean(rsg1)), lty = c("solid", "dashed"),
+              col = c("blue", "red"))
> p4 <- ggplot(data = data.frame(x = ThetaHatStar1), aes(x = x)) +
+   geom_density() +
+   xlim(c(MX - 5*sqrt(VX/n), MX + 5*sqrt(VX/n))) +
+   geom_vline(xintercept = c(MX, mean(ThetaHatStar1)),
+              lty = c("solid", "dashed"), col = c("blue", "red"))
> p5 <- ggplot(data = data.frame(x = rsg2), aes(x = x)) +
+   geom_density(fill = "lightblue") +
+   geom_vline(xintercept = c(MX, mean(rsg2)), lty = c("solid", "dashed"),
+              col = c("blue", "red"))
> p6 <- ggplot(data = data.frame(x = ThetaHatStar2), aes(x = x)) +
+   geom_density() +
+   xlim(c(MX - 5*sqrt(VX/n), MX + 5*sqrt(VX/n))) +
+   geom_vline(xintercept = c(MX, mean(ThetaHatStar2)),
+              lty = c("solid", "dashed"), col = c("blue", "red"))
> p7 <- ggplot(data = data.frame(x = rsg3), aes(x = x)) +
+   geom_density(fill = "lightblue") +
+   geom_vline(xintercept = c(MX, mean(rsg3)), lty = c("solid", "dashed"),
+              col = c("blue", "red"))
> p8 <- ggplot(data = data.frame(x = ThetaHatStar3), aes(x = x)) +
+   geom_density() +
+   xlim(c(MX - 5*sqrt(VX/n), MX + 5*sqrt(VX/n))) +
+   geom_vline(xintercept = c(MX, mean(ThetaHatStar3)),
+              lty = c("solid", "dashed"), col = c("blue", "red"))
> library(gridExtra)
> grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8, ncol = 2)

```

- (e) Example 5.14 on page 330 showed that $E[\bar{X}] = \mu_{\bar{X}} = \mu$, and that $\sigma_{\bar{X}}^2 = \frac{\sigma_X^2}{n}$, from which it follows that the $Bias[\bar{X}] = E[\bar{X}] - \mu = \mu - \mu = 0$. The bootstrap bias, $\widehat{Bias}_B[\bar{X}] = \bar{X}^* - \bar{X}$, for each of the three samples (-0.0051, 0.0017, -9e-04) is close to zero. The bootstrap bias for each sample is computed by subtracting the value for the dashed vertical lines in column two from the dashed lines in column one of Figure 10.18 on the facing page. The shapes of the three bootstrapped distributions of \bar{X} resemble the theoretical shape of the sampling distribution of \bar{X} shown in the top right of Figure 10.18 on the next page since all are unimodal with a slight positive skew. The spreads of the bootstrap distributions as measured by their standard errors (0.3314, 0.2705, 0.2896) are all close to the theoretical standard deviation of \bar{X} , $\sigma_{\bar{X}} = 0.3$.

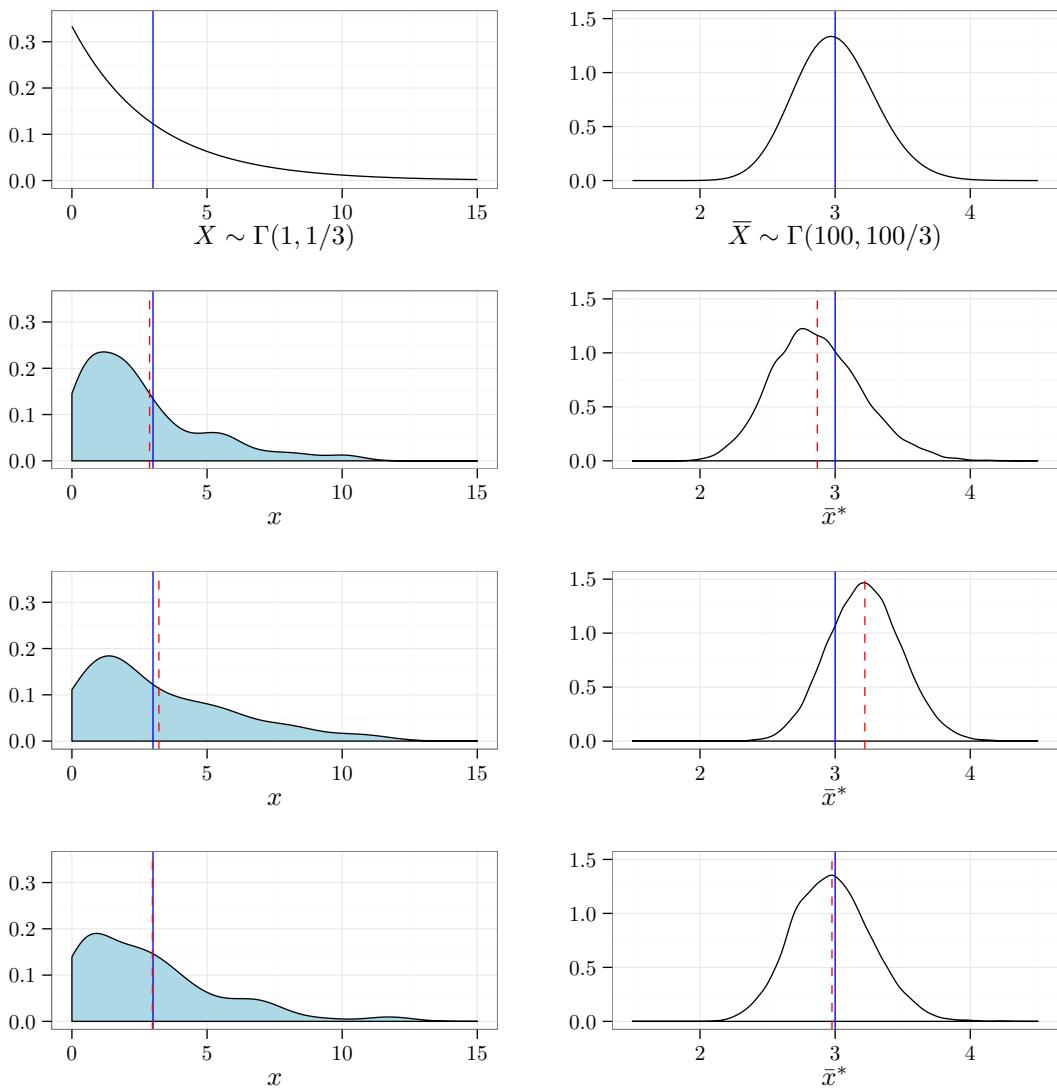


FIGURE 10.18: The top right graph shows $\text{a}\Gamma(1, 1/3)$ distribution with the solid vertical line marking the mean (3) of the distribution. The graphs in the first column in the second, third, and fourth rows are density estimates based on taking random samples of size $n = 100$ from the $\text{a}\Gamma(1, 1/3)$ distribution. The top graph in the second column shows the theoretical sampling distribution of \bar{X} when sampling from $\text{a}\Gamma(1, 1/3)$ distribution. The graphs in the second, third, and fourth rows of the second column are the bootstrap distributions of \bar{X} created from bootstrapping the samples immediately to each graph's left. The dashed vertical lines in all graphs are drawn at the mean of the values used to create that graph.

10.9.2 Confidence Intervals

With estimates of the standard error (standard deviation) and bias of some statistic of interest, various types of confidence intervals for the parameter θ can be constructed.

Although exact confidence intervals for specific problems can be computed, most confidence intervals are approximate. The most common confidence interval for a parameter θ when $\hat{\theta}$ follows either a normal or approximately normal distribution is

$$\hat{\theta} \pm z_{1-\alpha/2} \hat{\sigma}_{\hat{\theta}}. \quad (10.37)$$

The function `boot.ci()` in the package `boot` creates five types of confidence intervals.

The **normal** confidence interval in `boot.ci()` is a slight modification to (10.37) that incorporates both a bootstrap adjustment for bias and a bootstrap estimate of the standard error. The normal confidence interval is calculated as

$$CI_{1-\alpha}(\theta) = \left[\hat{\theta} - \widehat{Bias}_B(\hat{\theta}) - z_{1-\alpha/2} \cdot \widehat{SE}_B, \hat{\theta} - \widehat{Bias}_B(\hat{\theta}) + z_{1-\alpha/2} \cdot \widehat{SE}_B \right]. \quad (10.38)$$

The **basic bootstrap** confidence interval is based on the idea that the quantity $\hat{\theta}^* - \hat{\theta}$ has roughly the same distribution as $\hat{\theta} - \theta$. Since (10.39) has $(\hat{\theta}^* - \hat{\theta}) \approx (\hat{\theta} - \theta)$, (10.40) follows. To get (10.41) from (10.40), subtract $\hat{\theta}$ inside the probability statement and divide by -1 :

$$\mathbb{P} \left[\hat{\theta}_{((B+1)\cdot\alpha/2)}^* - \hat{\theta} \leq \hat{\theta}^* - \hat{\theta} \leq \hat{\theta}_{((B+1)\cdot(1-\alpha/2))}^* - \hat{\theta} \right] \approx 1 - \alpha, \quad (10.39)$$

$$\mathbb{P} \left[\hat{\theta}_{((B+1)\cdot\alpha/2)}^* - \hat{\theta} \leq \hat{\theta} - \theta \leq \hat{\theta}_{((B+1)\cdot(1-\alpha/2))}^* - \hat{\theta} \right] \approx 1 - \alpha, \quad (10.40)$$

$$\mathbb{P} \left[2\hat{\theta} - \hat{\theta}_{((B+1)\cdot(1-\alpha/2))}^* \leq \theta \leq 2\hat{\theta} - \hat{\theta}_{((B+1)\cdot\alpha/2)}^* \right] \approx 1 - \alpha. \quad (10.41)$$

Equations (10.39) to (10.41) lead to the basic bootstrap confidence interval given in (10.42):

$$CI_{1-\alpha}(\theta) = \left[2\hat{\theta} - \hat{\theta}_{((B+1)\cdot(1-\alpha/2))}^*, 2\hat{\theta} - \hat{\theta}_{((B+1)\cdot\alpha/2)}^* \right]. \quad (10.42)$$

The **percentile** confidence interval is based on the quantiles of the B bootstrap replications of $s(\mathbf{X})$. Specifically, the $(1 - \alpha)$ percentile confidence interval of θ uses the $\alpha/2$ and the $1 - \alpha/2$ quantiles of the $\hat{\theta}^*$ values to create a $(1 - \alpha) \cdot 100\%$ confidence interval for θ :

$$CI_{1-\alpha}(\theta) = \left[\hat{\theta}_{((B+1)\cdot\alpha/2)}^*, \hat{\theta}_{((B+1)\cdot(1-\alpha/2))}^* \right]. \quad (10.43)$$

The notation $\hat{\theta}_{(\text{Integer})}^*$ is used to denote the (Integer)th $\hat{\theta}^*$ of the B sorted $\hat{\theta}^*$ values. The values of B and α are generally chosen so that $(B+1)\cdot\alpha/2$ is an integer. In cases where $(B+1)\cdot\alpha/2$ is not an integer, interpolation can be used. Note that different programs/functions use different interpolation techniques. In particular, the function `quantile()` allows the user a choice of nine different algorithms to compute quantiles. One may have noticed that the percentile confidence interval uses $\hat{\theta}_{((B+1)\cdot\alpha/2)}^*$ to construct the *lower* endpoint of the confidence interval while the basic bootstrap interval uses $\hat{\theta}_{((B+1)\cdot\alpha/2)}^*$ in the construction of the *upper* endpoint of its confidence interval. Is one of the methods backwards? If not, does one method work better than the other? In fact, neither method is backward and neither method is uniformly superior to the other; however, the bootstrap percentile confidence interval is transformation respecting and Efron and Tibshirani (1993) show it has better coverage than the normal bootstrap confidence interval, which is not transformation respecting. Transformation respecting means that if $[\hat{\theta}_{\text{lower}}^*, \hat{\theta}_{\text{upper}}^*]$ is a confidence interval for θ , and $t(\theta)$ is any monotone transformation of θ , then the corresponding confidence interval for $t(\theta)$ is $[t(\hat{\theta}_{\text{lower}}^*), t(\hat{\theta}_{\text{upper}}^*)]$.

At this point, a reasonable question might be which confidence interval is recommended for general usage since the normal confidence interval is based on large sample properties and the percentile and basic bootstrap confidence interval formulas give different answers when the distribution of $\hat{\theta}^*$ is skewed. In fact, the answer is to use *none* of the confidence intervals discussed thus far. The bootstrap confidence interval procedure recommended for general usage is the BC_a method, which stands for bias-corrected and accelerated. The first three methods discussed (normal, percentile, and basic bootstrap) have first-order accuracy, while the BC_a method is second-order accurate. Accuracy in this context simply refers to the coverage errors. A confidence interval is first-order accurate if the coverage error tends to zero at a rate of $1/\sqrt{n}$ for a sample of size n , and is second-order accurate if the coverage error tends to zero at a rate of $1/n$ for a sample of size n . The bottom line is that there are theoretical reasons to prefer the BC_a confidence interval over the normal, percentile, and basic bootstrap confidence intervals in addition to it being second-order accurate and transformation respecting.

To compute a BC_a interval for θ , $CI_{1-\alpha}(\theta) = [\hat{\theta}_{\text{lower}}^*, \hat{\theta}_{\text{upper}}^*]$, first compute the bias factor, z , where

$$z = \Phi^{-1} \left[\frac{\sum_{b=1}^B I\{\hat{\theta}_b^* < \hat{\theta}\}}{B} \right]. \quad (10.44)$$

Recall the definition of Φ^{-1} on page 297. Provided the estimated bootstrap distribution, $s(\mathbf{x}^*) = \hat{\theta}^*$, is symmetric with respect to $\hat{\theta}$, and if $\hat{\theta}$ is unbiased, then $\frac{\sum_{b=1}^B I\{\hat{\theta}_b^* < \hat{\theta}\}}{B}$ will be close to 0.5, and the bias correction factor z will be close to zero since $\Phi^{-1}(0.5) = 0$, with R, `qnorm(0.5) = 0`. Next, compute the skewness correction factor:

$$a = \frac{\sum_{i=1}^n (\bar{\hat{\theta}}_{(-i)} - \hat{\theta}_{(-i)})^3}{6 \left[\sum_{i=1}^n (\bar{\hat{\theta}}_{(-i)} - \hat{\theta}_{(-i)})^2 \right]^{\frac{3}{2}}} \quad (10.45)$$

where $\hat{\theta}_{(-i)}$ is the value of $\hat{\theta} = s(\mathbf{X})$ when the i^{th} value is deleted from the sample of n values and $\bar{\hat{\theta}}_{(-i)} = \sum_{i=1}^n \frac{\hat{\theta}_{(-i)}}{n}$. Using z and a , compute

$$a_1 = \Phi \left[z + \frac{z + z_{\alpha/2}}{1 - a(z + z_{\alpha/2})} \right] \text{ and } a_2 = \Phi \left[z + \frac{z + z_{1-\alpha/2}}{1 - a(z + z_{1-\alpha/2})} \right]. \quad (10.46)$$

Now, lower = $(B + 1) \cdot a_1$ and upper = $(B + 1) \cdot a_2$. When either lower or upper is not an integer, interpolation can be used to obtain the lower and upper endpoints of the BC_a confidence interval:

$$CI_{1-\alpha}(\theta) = [\hat{\theta}_{\text{lower}}^*, \hat{\theta}_{\text{upper}}^*]. \quad (10.47)$$

Let $k = \lfloor (B + 1) \cdot a_i \rfloor$ for $i = 1, 2$. Then the appropriate interpolation is

$$\hat{\theta}_{((B+1) \cdot a_i)}^* = \hat{\theta}_{(k)}^* + \frac{\Phi^{-1}(a_i) - \Phi^{-1}\left(\frac{k}{B+1}\right)}{\Phi^{-1}\left(\frac{k+1}{B+1}\right) - \Phi^{-1}\left(\frac{k}{B+1}\right)} \cdot \left(\hat{\theta}_{(k+1)}^* - \hat{\theta}_{(k)}^* \right) \text{ for } i = 1, 2. \quad (10.48)$$

The **studentized** bootstrap confidence interval is based on estimating the actual distribution of the t statistic from the data. The estimated distribution of T^* is denoted

$$T^* = \frac{(\hat{\theta}^* - \hat{\theta})}{\sqrt{\widehat{Var}(\hat{\theta}^*)}} \quad (10.49)$$

where the quantities $\hat{\theta}^*$ and $\sqrt{\widehat{Var}(\hat{\theta}^*)}$ are statistics computed from a bootstrap sample. The studentized confidence interval is

$$CI_{1-\alpha}(\theta) = \left[\hat{\theta} - T_{((B+1)\cdot(1-\alpha/2))}^* \cdot \hat{\sigma}_{\hat{\theta}}, \hat{\theta} - T_{((B+1)\cdot\alpha/2)}^* \cdot \hat{\sigma}_{\hat{\theta}} \right]. \quad (10.50)$$

Equation (10.50) is similar to (8.10) with two differences. One, since the theoretical t distribution is symmetric, (8.10) is generally written

$$CI_{1-\alpha}(\mu) = \left[\bar{x} - t_{1-\alpha/2;n-1} \frac{s}{\sqrt{n}}, \bar{x} + t_{1-\alpha/2;n-1} \frac{s}{\sqrt{n}} \right]$$

instead of the equivalent expression

$$CI_{1-\alpha}(\mu) = \left[\bar{x} - t_{1-\alpha/2;n-1} \frac{s}{\sqrt{n}}, \bar{x} - t_{\alpha/2;n-1} \frac{s}{\sqrt{n}} \right].$$

The second difference between (8.10) and (10.50) is that the theoretical t quantile in (8.10) is replaced by the bootstrapped T^* in (10.50). The studentized bootstrap confidence interval is second-order accurate but is not transformation respecting.

Example 10.18 ▷ Bootstrap CIs with M1 Motorspeedway Times ◁ The times recorded are those for 41 successive vehicles traveling northwards along the M1 motorway in England when passing a fixed point near Junction 13 in Bedfordshire on Saturday, March 23, 1985. After subtracting the times, the following 40 interarrival times, reported to the nearest second, are stored in **SDS4** under the variable **times**.

- (a) Determine the distribution of the interarrival times.
- (b) Create a density estimate and quantile-quantile plot of the bootstrap distribution of \bar{X} .
- (c) Calculate bootstrap confidence intervals for the mean of interarrival times using the function **boot.ci()** from the **boot** package. Specifically, use the arguments **norm**, **basic**, **perc**, **bca**, and **stud** with the function **boot.ci()** to create normal approximation, basic, percentile, BC_a , and studentized 95% bootstrap confidence intervals.
- (d) Verify the confidence intervals returned for the mean from **boot.ci()** using the appropriate equations.

Solution: The answers are as follows:

- (a) It appears that the conditions for an approximate Poisson process are satisfied. The estimated parameter, λ , for the Poisson process, $\hat{\lambda}$, is 0.1282 cars per second. Consequently, the waiting time until the next car follows an exponential distribution with mean $1/\lambda$. In this case, the estimated mean of the exponential distribution (waiting time) is 7.8 seconds/car. An estimated density of the interarrival times with a superimposed $Exp(\lambda = 0.1282)$ shown in Figure 10.19 on the facing page suggests this distribution is reasonable. In addition, both the mean and the standard deviation of **times** are roughly equal, as they should be with an exponential distribution. R Code 10.33 on the next page can be used to create a graph similar to Figure 10.19 on the facing page.

R Code 10.32

```
> TotalTime <- max(cumsum(SDS4$times)) # Total time in seconds
> n <- 40 # Number of interarrival times
```

```

> est.lambda <- n/TotalTime           # Estimated lambda for Poisson
> est.mean <- 1/est.lambda          # Est. waiting time for next car
> ans <- c(TotalTime, n, est.lambda, est.mean)
> names(ans) <- c("Total Time", "Number of Cars", "Est Lambda", "Est Mean")
> ans

  Total Time Number of Cars      Est Lambda      Est Mean
 312.0000000     40.0000000      0.1282051     7.8000000

> c(mean(SDS4$times), sd(SDS4$times))  # Exponential Check
[1] 7.800000 7.871402

```

R Code 10.33

```

> ggplot(data = SDS4, aes(x = times)) +
+   geom_density(fill = "pink", alpha = 0.3) +
+   stat_function(fun = dexp, arg = list(est.lambda), lty = "dashed") +
+   labs(x = "Interarrival Times (seconds)", y = "") +
+   theme_bw()

```

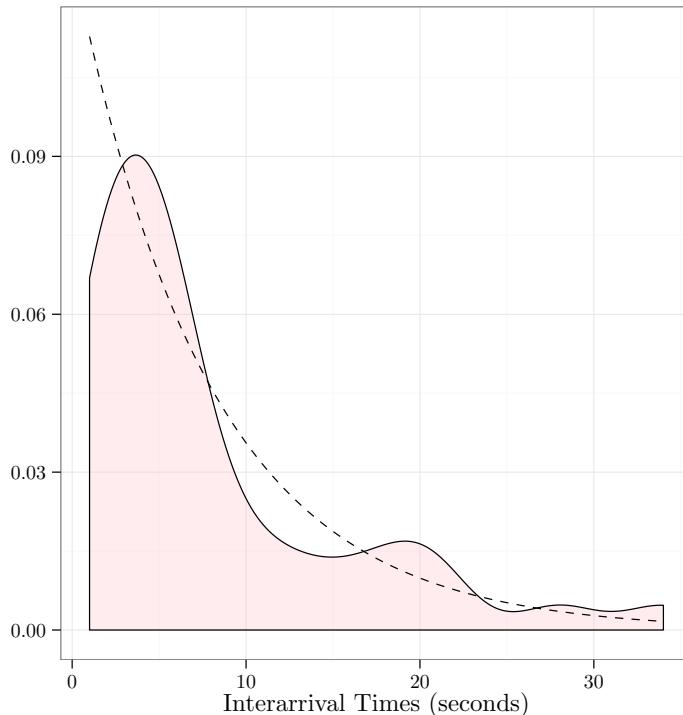


FIGURE 10.19: Estimated density of interarrival times at the M1 motorspeedway with a superimposed dashed line of an $Exp(\lambda = 0.1282)$

(b) R Code 10.34 on the next page defines the function `times.mean()`, which computes

the mean and an estimate of the variance of the sample mean for a bootstrap sample. A variance estimate for the statistic of interest is not a requirement for constructing normal, basic, percentile or BC_a confidence intervals; however, an estimate of the variance for the bootstrapped statistic must be computed for studentized confidence intervals. In this problem, it is known that the variance of \bar{X} is σ^2/n . Estimating the variance for other statistics may not be easy. The number of bootstrap replications B is set equal to $10^4 - 1$, and the bootstrapped distribution of \bar{X} , denoted by t^* , when using the `boot` package, is created with a call to the function `boot()`. Results from applying `boot()` to the interarrival times are stored in the object `b.obj`. Note that `R` in the function `boot()` is the number of bootstrap replications, which is denoted B in this text, so `R` is set equal to `B`. A random seed value of 10 is used so the reader can reproduce the results from R Code 10.34.

R Code 10.34

```
> library(boot)
> times.mean <- function(data, i){
+   d <- data[i]
+   M <- mean(d)
+   V <- var(d)/40 # sigma^2/n
+   c(M, V)
+ }
> B <- 10^4 - 1
> set.seed(10)
> b.obj <- boot(data = SDS4$times, statistic = times.mean, R = B)
> b.obj
```

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

```
boot(data = SDS4$times, statistic = times.mean, R = B)
```

```
Bootstrap Statistics :
      original     bias    std. error
t1* 7.800000 -0.01249375  1.2149888
t2* 1.548974 -0.04247343  0.4833002
```

It is possible to create a histogram and quantile-quantile plot of the $\hat{\theta}^*$ values using the function `plot()` on the bootstrapped object (`b.obj`), but the authors prefer `ggplot2` graphs. R Code 10.35 creates a density estimate and a quantile-quantile plot of the $\hat{\theta}^*$ values with `ggplot2` graphics and displays the results in Figure 10.20 on the facing page. Two bootstrapped statistics are returned with the function `times.mean()` and stored in the `t` component, a $B \times 2$ matrix, of the returned boot object (`b.obj`). R Code 10.35 uses the $\hat{\theta}^*$ values stored in the first column of `t`. Note that the $\widehat{Var}(\hat{\theta}^*)$ values are stored in the second column of `t`.

R Code 10.35

```
> DF <- data.frame(ths = b.obj$t)
> p <- ggplot(data = DF, aes(x = ths.1)) +
+   geom_density(fill = "lightblue") +
```

```
+   labs(x = "$\hat{\theta}^*$", y = "") +
+   theme_bw()
> p
> p1 <- ggplot(data = DF, aes(sample = ths.1)) +
+   stat_qq() +
+   theme_bw() +
+   coord_fixed() # 1:1 scaling between axes
> p1
```

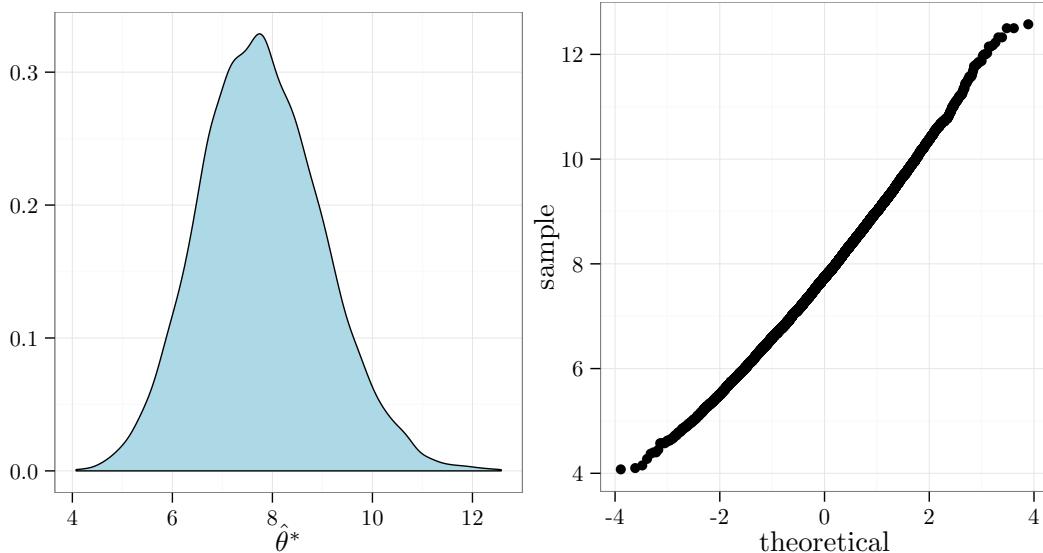


FIGURE 10.20: Density estimate and quantile-quantile plot of $\hat{\theta}^*$

(c) The five bootstrap confidence intervals for the mean are created using the function `boot.ci()`. Specifically, the function `boot.ci()` is applied to the object `b.obj` in R Code 10.36.

R Code 10.36

```
> boot.ci(b.obj, conf = 0.95, type = c("norm", "basic", "perc",
+                                         "bca", "stud"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 9999 bootstrap replicates

CALL :
boot.ci(boot.out = b.obj, conf = 0.95, type = c("norm", "basic",
  "perc", "bca", "stud"))

Intervals :
Level      Normal              Basic              Studentized
95%    ( 5.431, 10.194 )    ( 5.275, 10.050 )    ( 5.681, 11.070 )
```

Level	Percentile	BCa
95%	(5.550, 10.325)	(5.800, 10.700)
Calculations and Intervals on Original Scale		

(d) A 95% normal bootstrap confidence interval using (10.38) is created in R Code 10.37. Note that the returned confidence interval agrees with the interval returned from the `boot.ci()` function in R Code 10.36.

R Code 10.37

```
> ThetaHat <- b.obj$t0[1]
> ThetaHat

[1] 7.8

> Bias <- mean(b.obj$t[,1]) - b.obj$t0[1]
> Bias

[1] -0.01249375

> alpha <- 0.05
> NBCI <- (ThetaHat - Bias) + c(-1, 1)*qnorm(1 - alpha/2)*sd(b.obj$t[,1])
> NBCI

[1] 5.43116 10.19383
```

A 95% basic bootstrap confidence interval using (10.42) is created in R Code 10.38. Note that the returned confidence interval agrees with the interval returned from the `boot.ci()` function in R Code 10.36 on the previous page.

R Code 10.38

```
> BBCI <- c(2*ThetaHat - sort(b.obj$t[,1])[((B + 1)*(1 - alpha/2))],
+          2*ThetaHat - sort(b.obj$t[,1])[((B + 1)*(alpha/2))])
> BBCI

[1] 5.275 10.050
```

A 95% percentile bootstrap confidence interval using (10.43) is created in R Code 10.39. Note that the returned confidence interval agrees with the interval returned from the `boot.ci()` function in R Code 10.36 on the previous page.

R Code 10.39

```
> PBCI <- c(sort(b.obj$t[,1])[((B + 1)*(alpha/2))],
+             sort(b.obj$t[,1])[((B + 1)*(1 - alpha/2))])
> PBCI

[1] 5.550 10.325
```

A 95% BC_a confidence interval using (10.47) is created in R Code 10.40 on the facing page, including the bias factor z from (10.44), the a value (skewness correction factor) from (10.45), and the a_1 and a_2 values from (10.46). Since $(B + 1) \cdot a_1$ is not an integer (423.5), the values for $\hat{\theta}_{\text{lower}}^*$ and $\hat{\theta}_{\text{upper}}^*$ are computed with the interpolation (10.48). Note that the

returned confidence interval agrees with the interval returned from the `boot.ci()` function in R Code 10.36 on page 671.

R Code 10.40

```
> z <- qnorm(sum(b.obj$t[,1] < ThetaHat)/B) # bias factor
> z
[1] 0.04501367

> n <- length(SDS4$times)
> u <- numeric(n)
> for(i in 1:n){
+   u[i] <- mean(SDS4$times[-i])
+ }
>ubar <- mean(u)
> numa <- sum((ubar - u)^3)
> dena <- 6*sum((ubar - u)^2)^(3/2)
> a <- numa/dena # skewness correction factor
> a
[1] 0.04308169

> a1 <- pnorm(z + (z + qnorm(alpha/2))/(1 - a*(z + qnorm(alpha/2))))
> a2 <- pnorm(z + (z + qnorm(1 - alpha/2))/(1 - a*(z + qnorm(1 - alpha/2))))
> (B + 1)*a1
[1] 423.544

> (B + 1)*a2
[1] 9874.399

> ### Interpolation
> kLower <- floor((B + 1)*a1)
> kUpper <- floor((B + 1)*a2)
> c(kLower, kUpper)
[1] 423 9874

> ll <- sort(b.obj$t[,1])[kLower] + (qnorm(a1) - qnorm(kLower/(B + 1))) /
+   (qnorm((kLower + 1)/(B + 1)) - qnorm(kLower/(B + 1))) *
+   (sort(b.obj$t[,1])[kLower + 1] - sort(b.obj$t[,1])[kLower])
> ul <- sort(b.obj$t[,1])[kUpper] + (qnorm(a2) - qnorm(kUpper/(B + 1))) /
+   (qnorm((kUpper + 1)/(B + 1)) - qnorm(kUpper/(B + 1))) *
+   (sort(b.obj$t[,1])[kUpper + 1] - sort(b.obj$t[,1])[kUpper])
> BCaCI <- c(ll, ul)
> BCaCI
[1] 5.8 10.7
```

A 95% studentized bootstrap confidence interval using (10.50) is created in R Code 10.41 on the following page. Note that the returned confidence interval agrees with the interval returned from the `boot.ci()` function in R Code 10.36 on page 671.

R Code 10.41

```
> TS <- (b.obj$t[,1] - b.obj$t0[1])/sqrt(b.obj$t[,2])
> CT <- sort(TS)[c((B + 1)*0.025, (B + 1)*0.975)]
> CT # critical t values

[1] -2.627383  1.702412

> BTCI <- c(b.obj$t0[1] - CT[2]*sqrt(b.obj$t0[2]),
+             b.obj$t0[1] - CT[1]*sqrt(b.obj$t0[2]))
> BTCI

[1] 5.681215 11.069984
```



Computationally intensive methods such as the bootstrap have no problem computing standard errors for even the most complex statistics. The key when creating bootstrap statistics is to mimic how the original data were obtained. For example, one could derive the standard error of the correlation coefficient, r , by assuming the sample comes from a bivariate normal distribution. Such derivations are arduous and require mathematical diligence, which is why bootstrapping is useful in this case. If the statistic is complex, it may not be possible to derive a closed form formula for the standard error.

Example 10.19 The data frame `Animals` from the MASS package is used in this example.

- Construct a 95% bootstrap percentile confidence interval for the correlation coefficient (ρ) between `log(brain)` and `log(body)`.
- Create a density estimate of the bootstrapped correlation coefficients and shade the plot to show the 95% bootstrapped percentile confidence interval.

Solution: The answers are as follows:

- Two different solutions are presented for the construction of a 95% bootstrap percentile confidence interval; but, the key idea with both approaches is that the sampling is performed on paired values. This type of sampling scheme is often referred to as *case resampling* and is frequently used to compute standard errors for the parameters of linear models. The first solution given in R Code 10.42 computes the bootstrapped statistic (r^*) inside a `for` loop, then uses both the function `quantile()` as well as (10.43) to compute the confidence interval for ρ .

R Code 10.42

```
> library(MASS)
> n <- dim(Animals)[1]
> B <- 10^4 - 1
> R <- numeric(B)
> set.seed(2)
> for(b in 1:B){
+   i <- sample(1:n, size = n, replace = TRUE)
+   BRAIN <- Animals$brain[i]
+   BODY <- Animals$body[i]
+   R[b] <- cor(log(BRAIN), log(BODY))
+ }
```

```
> ## Using quantile function
> quantile(R, probs = c(0.025, 0.975), type = 2)

 2.5%    97.5%
0.5433646 0.9587591

> ## Using formula
> RS <- sort(R)
> c(RS[(B + 1)*0.025], RS[(B + 1)*0.975])

[1] 0.5433646 0.9587591
```

The 95% bootstrap percentile confidence interval for ρ is [0.5434, 0.9588]. The second solution given in R Code 10.43 computes the requested confidence interval with calls to `boot()` and `boot.ci()` from the package `boot` after creating the function `cor.boot()`, the statistic passed to the function `boot()`. Note that `cor.boot()` performs case resampling of the values in its `data` object.

R Code 10.43

```
> B <- 10^4 - 1
> cor.boot <- function(data, i){
+   d <- data[i, ]
+   RS <- cor(log(d[, 1]), log(d[, 2]))
+ }
> set.seed(1)
> b.obj <- boot(data = Animals, statistic = cor.boot, R = B)
> b.obj
```

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

```
boot(data = Animals, statistic = cor.boot, R = B)
```

Bootstrap Statistics :

	original	bias	std. error
t1*	0.7794935	-0.003215607	0.09960159

```
> boot.ci(b.obj, type = "perc")
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 9999 bootstrap replicates

CALL :

```
boot.ci(boot.out = b.obj, type = "perc")
```

Intervals :

Level Percentile

95% (0.5545, 0.9559)

Calculations and Intervals on Original Scale

(b) R Code 10.44 is used to create the estimated density for r^* shown in Figure 10.21.

R Code 10.44

```
> library(ggplot2)
> DF <- data.frame(x = b.obj$t)
> p <- ggplot(data = DF, aes(x = x)) +
+   geom_density(fill = "pink", alpha = 0.3) +
+   labs(x = "$r^*$", y = "") +
+   theme_bw()
> x.dens <- density(b.obj$t)
> df.dens <- data.frame(x = x.dens$x, y = x.dens$y)
> # shade left tail
> p + geom_area(data = subset(df.dens, x <= 0.5545 & x >= min(DF$x)),
+                 aes(x = x, y = y), fill = "blue", alpha = 1) +
+   # shade right tail
+   geom_area(data = subset(df.dens, x >= 0.9559 & x <= max(DF$x)),
+             aes(x = x, y = y), fill = "blue", alpha = 1)
```

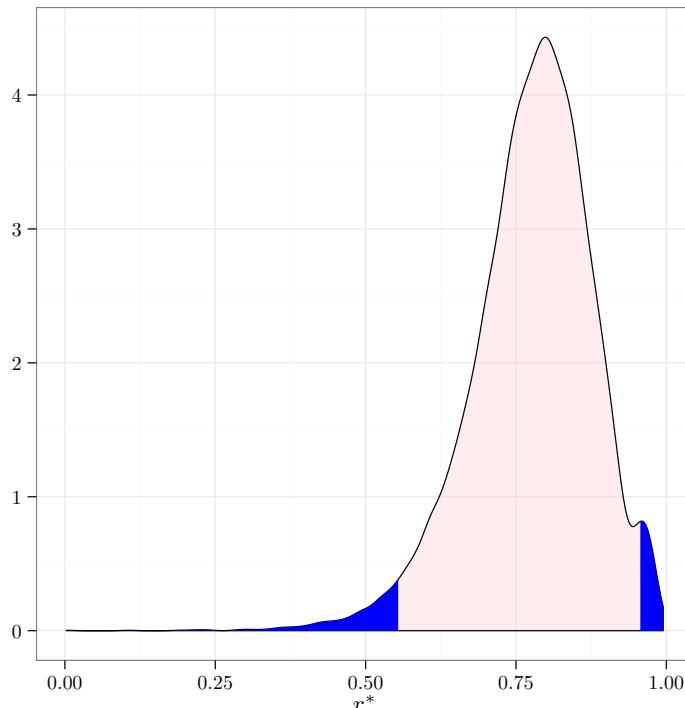


FIGURE 10.21: Density estimate of r^* with shaded 95% bootstrap percentile confidence interval



10.9.3 Bootstrapping and Regression

There are two basic resampling schemes for regression models. The first scheme is referred to as *resampling cases* and was briefly discussed in Example 10.19 on page 674. Consider a data frame with $i = n$ observations on $j = p$ predictors $x_{i1}, x_{i2}, \dots, x_{ip}$ and one response variable Y_i such that the data frame has dimensions $n \times (p + 1)$. The basic idea behind resampling cases is to repeat the following process B times:

1. Take a sample of size n with replacement from the n rows (cases).
2. Compute statistics on the resampled data frame such as estimated coefficients for a regression model.

The second sampling scheme is *resampling residuals*. Recall that each observation Y_i can be expressed as $\hat{Y}_i + \hat{\varepsilon}_i$. In resampling residuals, the \hat{Y}_i s are fixed, and the n residuals are sampled with replacement to obtain the new observations $Y_i^* = \hat{Y}_i + \hat{\varepsilon}_i^*$ for $i = 1, 2, \dots, n$. The idea behind resampling residuals is to repeat the following process B times:

1. Resample the n residuals with replacement and create $Y_i^* = \hat{Y}_i + \hat{\varepsilon}_i^*$ for $i = 1, 2, \dots, n$.
2. Regress the Y_i^* values on the fixed predictors and compute statistics such as estimated coefficients for a regression model.

Davison and Hinkley (1997) suggest using scaled and recentered residuals when resampling residuals where the i^{th} scaled and recentered residual is defined as

$$r_i = \frac{\hat{\varepsilon}_i}{\sqrt{1 - h_i}} - \bar{r} \quad (10.51)$$

and h_i is the i^{th} leverage value for the fitted regression model. The package **car** has the function **Boot()**, which provides a simple front-end to the **boot()** function. The argument **method=** allows the user to specify either "case" or "residuals" for the bootstrap method. In addition, **car** has the generic function **confint()** that returns confidence intervals for objects of class **boot**.

Resampling cases makes no assumptions about the homogeneity of variance, which provides a possible advantage over resampling residuals when the assumption of homogeneity of variance is not satisfied. When the assumption of homogeneity of variance is satisfied, resampling residuals provides a potential advantage in efficiency over resampling cases (Davison and Hinkley, 1997). Although regression assumes the levels of the predictor variables are fixed, the estimated coefficients and their standard errors are quite robust when the predictors are random. In a designed experiment where the levels of the predictors are established by the experimenter, one would prefer to use residual resampling over case resampling. In general, the bootstrap does not offer much improvement in accuracy over classical confidence intervals in the regression setting unless there are gross violations of the regression assumptions.

Example 10.20 ▷ Estimating Regression Coefficients with the Bootstrap ◁
The body fat percentage of 78 high school wrestlers was measured using three separate techniques, and the results are stored in the data frame **HSWRESTLER**. The techniques used were hydrostatic weighing (**hwfat**), skin fold measurements (**skfat**), and the Tanita body fat scale (**tanfat**). Construct a 95% BC_a confidence interval for β_1 by regressing **hwfat** on **tanfat**.

- (a) Use case resampling.
- (b) Use residual resampling with the scaled and recentered residuals defined in (10.51).

- (c) Use the `Boot` function from the package `car` to construct 95% BC_a confidence intervals for β_1 using both case resampling and residual resampling.

Solution: The answers are as follows:

(a) Two solutions are presented for case resampling. The first solution given in R Code 10.45 takes advantage of the `subset` function when creating the function `caseresamp`, which is used as the statistic in the function `boot()`. Specifically, the argument `subset = i` performs case resampling of the `data` object passed to the function `casesamp()`.

R Code 10.45

```
> caseresamp <- function(data, i){
+   mod <- lm(hwfat ~ tanfat, data = data, subset = i)
+   coef(mod)
+ }
> set.seed(1)
> b.obj <- boot(HSWRESTLER, statistic = caseresamp, R = 10000)
> boot.ci(b.obj, type = "bca", index = 2, conf = 0.95)
```

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS

Based on 10000 bootstrap replicates

CALL :

```
boot.ci(boot.out = b.obj, conf = 0.95, type = "bca", index = 2)
```

Intervals :

Level BCa

95% (0.8264, 1.1145)

Calculations and Intervals on Original Scale

```
> confint(object = lm(hwfat ~ tanfat, data = HSWRESTLER),
+           parm = "tanfat", level = 0.95)

      2.5 %    97.5 %
tanfat 0.8658364 1.089223
```

The second solution given in R Code 10.46 creates a function to fit the regression, `hwfit()`, and a function to perform the case resampling, `hwcase()`. Note that the returned bootstrapped BC_a confidence interval for the slope of the regression line, β_1 , is identical for both functions.

R Code 10.46

```
> hwfit <- function(data){
+   coef(lm(data$hwfat ~ data$tanfat))
+ }
> hwcase <- function(data, i){
+   hwfit(data[i, ])
+ }
> set.seed(1)
> hwboot <- boot(HSWRESTLER, statistic = hwcase, R = 10000)
> boot.ci(hwboot, type = "bca", index = 2, conf = 0.95)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
```

```
Based on 10000 bootstrap replicates
```

```
CALL :
```

```
boot.ci(boot.out = hwboot, conf = 0.95, type = "bca", index = 2)
```

```
Intervals :
```

```
Level      BCa
```

```
95%  ( 0.8264,  1.1145 )
```

```
Calculations and Intervals on Original Scale
```

```
> confint(object = lm(hwfat ~ tanfat, data = HSWRESTLER),
+           parm = "tanfat", level = 0.95)
```

```
        2.5 % 97.5 %
```

```
tanfat 0.8658364 1.089223
```

(b) R Code 10.47 creates a new data frame **FAT** containing all of the variables in **HSWRESTLER** and two new variables **RESIDr2**, and **fit**. **RESIDr2** contains residuals defined according to (10.51), and **fit** contains the \hat{Y}_i s from regressing **hwfat** on **tanfat** using the original values in **HSWRESTLER**. The function **rresamp()** takes samples of size n with replacement from the values in **RESIDr2** and adds the result to the values in **fit** to create new values in the variable **hwfat**. The new **hwfat** values are then regressed on the fixed values in **tanfat**. The function **hatvalues()** is used to extract the leverage values from the fitted regression model **hw.lm**.

R Code 10.47

```
> hw.lm <- lm(hwfat ~ tanfat, data = HSWRESTLER)
> hwmf <- function(data){
+   coef(lm(data$hwfat ~ data$tanfat))
+ }
> FAT <- data.frame(HSWRESTLER, RESIDr2 = resid(hw.lm) /
+                      sqrt(1 - hatvalues(hw.lm)) -
+                      mean(resid(hw.lm)/sqrt(1 - hatvalues(hw.lm))),
+                      fit = fitted(hw.lm))
> FAT[1:6, 7:11]

  hwfat tanfat skfat  RESIDr2       fit
1 10.71   11.9   9.80 1.2084304  9.510531
2  8.53   10.0  10.56 0.8848525  7.653225
3  6.78    8.3   8.43 0.7977684  5.991425
4  9.32    8.2  11.77 3.4723857  5.893672
5 41.89   41.6  41.09 3.5980840 38.543156
6 34.03   29.9  29.45 7.0918448 27.106061

> rresamp <- function(data, i){
+   d <- data
+   d$hwfat <- d$fit + d$RESIDr2[i]
+   hwmf(d)
+ }
> set.seed(1)
```

```
> rboot <- boot(FAT, statistic = rresamp, R = 10000)
> boot.ci(rboot, type = "bca", index = 2)

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = rboot, type = "bca", index = 2)

Intervals :
Level      BCa
95%   ( 0.8607,  1.0820 )
Calculations and Intervals on Original Scale

> confint(hw.lm, parm = "tanfat")

              2.5 %    97.5 %
tanfat  0.8658364 1.089223
```

(c) R Code 10.48 computes a 95% bootstrap BC_a confidence interval for β_1 using the `Boot()` function from `car` package using case resampling.

R Code 10.48

```
> library(car)
> set.seed(1)
> summary(RbootC <- Boot(hw.lm, R = 10000, method = "case"))

          R original  bootBias  bootSE  bootMed
(Intercept) 10000 -2.12207 -0.051323 1.136418 -2.14285
tanfat       10000  0.97753  0.002723 0.073805  0.97986

> confint(RbootC, level= 0.95, type = "bca", parm = "tanfat")

Bootstrap quantiles, type = bca

              2.5 %    97.5 %
tanfat  0.8263641 1.114512
```

R Code 10.49 computes a 95% bootstrap BC_a confidence interval for β_1 using the `Boot()` function from `car` package using residual resampling.

R Code 10.49

```
> library(car)
> set.seed(1)
> summary(RbootR <- Boot(hw.lm, R = 10000, method = "residual"))

          R original  bootBias  bootSE  bootMed
(Intercept) 10000 -2.12207  0.0056516 1.051212 -2.11730
tanfat       10000  0.97753 -0.0004084 0.056633  0.97745

> confint(RbootR, level= 0.95, type = "bca", parm = "tanfat")
```

```
Bootstrap quantiles, type = bca
2.5 %    97.5 %
tanfat 0.8649886 1.085407

> boot.ci(RbootR, level= 0.95, type = "bca", index = 2)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = RbootR, type = "bca", index = 2, level = 0.95)

Intervals :
Level      BCa
95%   ( 0.8650,  1.0854 )
Calculations and Intervals on Original Scale
```

Using the `Boot()` function can reduce the amount of programming needed to construct bootstrap confidence intervals for regression coefficients. From R Code 10.48 on the facing page and R Code 10.49 on the preceding page one can see that the function `boot.ci()` from the package `boot` works on objects of class `Boot`. Other functions from the package `boot` that work on objects of class `Boot` include `boot.array()`, `plot.boot()`, and `empinf()`. Functions from the `car` package that work on objects of class `Boot` include `summary.boot()`, `confint.boot()`, and `hist.boot()`. For further details on any of the functions in the `car` package, see their respective help files.



10.10 Permutation Tests

Permutation tests are computationally intensive techniques that actually predate computers. Until recently, permutation tests were more of a theoretical ideal than a useful technique. With the advent of high-powered computers, permutation tests have moved out of the abstract into the world of the practical. The permutation test is examined in only one context here — the two-sample problem. The fundamental idea behind the permutation test is that if there are no differences between two treatments, all data sets obtained by randomly assigning the data to the two treatments have an equal chance of being observed. Permutation tests are especially advantageous when working with small samples where verification of assumptions required for tests such as the pooled *t*-test are difficult.

To test a hypothesis with a permutation test:

- Step 1: Choose a test statistic $\hat{\theta}$ that measures the effect under study. Note that certain statistics will have more power to detect the effect of interest than others.
- Step 2: Create the sampling distribution that the test statistic in step 1 would have if the effect is not present in the population. In other words, create the sampling distribution for the test statistic of interest under the assumption that the null hypothesis is true.

Step 3: Find the “observed test statistic” in the sampling distribution from step 2. Observed values in the extremes of the sampling distribution suggest that the effect under study is “real.” In contrast, observed values in the main body of the sampling distribution imply that the effect is likely to have occurred by chance.

Step 4: Calculate the ϕ -value based on the observed test statistic. This may be

$$\begin{aligned}\mathbb{P}(|\hat{\theta}| \geq |\hat{\theta}_{\text{obs}}|) \\ \mathbb{P}(\hat{\theta} \geq \hat{\theta}_{\text{obs}}) \\ \mathbb{P}(\hat{\theta} \leq \hat{\theta}_{\text{obs}})\end{aligned}$$

for the inequality in H_A being \neq , $>$, or $<$, respectively.

The Two-Sample Problem

Suppose two independent random samples $\mathbf{z} = \{z_1, z_2, \dots, z_n\}$ and $\mathbf{y} = \{y_1, y_2, \dots, y_m\}$ are drawn from possibly different probability distributions F and G . The question of interest is whether $F = G$. Assuming $H_0 : F = G$ is true, it is possible to create the permutation sampling distribution for some statistic, $\hat{\theta}$, of interest. Let N equal the combined sample size $n + m$, let $\mathbf{v} = \{v_1, v_2, \dots, v_N\}$ be the combined and ordered vector of values, and let $\mathbf{g} = \{g_1, g_2, \dots, g_N\}$ be a vector indicating the group membership for \mathbf{v} . Since there are n z_i s and m y_j s in \mathbf{g} , there are $\binom{N}{n}$ possible ways of partitioning N elements into two subsets of sizes n and m . Consequently, under the null hypothesis that $F = G$, the vector \mathbf{g} has probability $1/\binom{N}{n}$ of equaling any one of its possible values. That is, all combinations of z_i s and y_j s are equally likely if $F = G$.

Suppose $H_A : F > G$ and $\hat{\theta} = \bar{z} - \bar{y}$. Then the exact ϕ -value is found by computing $\#\{\hat{\theta} \geq \hat{\theta}_{\text{obs}}\}/\binom{N}{n}$. For all but relatively trivial sized samples n and m , the number $\binom{N}{n}$ will be huge, making the enumeration of all possible samples of the statistic of interest a monumental task. Consider that, if $n = 15$ and $m = 10$, complete enumeration would require listing $\binom{25}{10} = 3,268,760$ possible outcomes.

Consequently, an approximation to the exact ϕ -value is often obtained by resampling without replacement the original data some “large” number of times, B , which is usually at least 999, and approximating the ϕ -value with

$$\phi\text{-value} \approx \frac{\left[1 + \#\left\{\hat{\theta}_b^* \geq \hat{\theta}_{\text{obs}}\right\}\right]}{(B+1)} = \frac{\left[1 + \sum_{i=1}^B \mathbf{I}\left\{\hat{\theta}_b^* \geq \hat{\theta}_{\text{obs}}\right\}\right]}{(B+1)}. \quad (10.52)$$

The resampling is done without replacement to approximate the ϕ -value from a permutation test. When the sampling is done with replacement, a bootstrap test is performed. Bootstrap tests are somewhat more general than permutation tests since they apply to a wider class of problems; however, they do not return “exact” ϕ -values.

Example 10.21  **Permutation Test**  The data set used in this problem (**RATBP**) is originally from Ott and Mendenhall (1985, problem 8.17). Researchers wanted to know whether a drug was able to reduce the blood pressure of rats. Twelve rats were chosen and the drug was administered to six rats, the treatment group, chosen at random. The other six rats, the control group, received a placebo. The drops in blood pressure (mmHg) for the treatment group (with probability distribution G) and the control group (with probability distribution F) are stored in the variables **Treat**(y) and **Contr**(z), respectively. Note that positive numbers indicate blood pressure decreased while negative numbers indicate that it rose. Under the null hypothesis, $H_0 : F = G$, the data come from a single population. The

question of interest is, “How likely are differences as extreme as those observed between the treatment and control groups to be seen if the null hypothesis is true?” Use $\hat{\theta} = \bar{z} - \bar{y}$ as the statistic of interest to test the alternative hypothesis $H_1 : F < G$ and compute:

- (a) The exact permutation ϕ -value,
- (b) An estimated permutation ϕ -value based on 9,999 permutation replications, and
- (c) An estimated density curve for the exact permutation distribution from part (a) and an estimated density curve for the estimated permutation distribution from part (b).

Solution: The test statistic of interest (step 1) has been specified to be $\hat{\theta} = \bar{z} - \bar{y}$. Finding the ϕ -values requires the creation of sampling distributions with different methods. To start the process, create individual vectors `Contr` and `Treat` that contain the drops in blood pressure for the control and treatment groups, respectively. Then, create a vector that combines the individual vectors into a combined vector. R Code 10.50 creates the three vectors. Note that the indexed values 1 through 6 of the combined vector (`mmHg[1:6]`) correspond to the values stored in the object `Contr`, and that indexed values 7 through 12 of the combined vector (`mmHg[7:12]`) correspond to the values stored in the object `Treat`.

R Code 10.50

```
> Contr <- RATBP$mmHg[RATBP$group == "control"]
> Treat <- RATBP$mmHg[RATBP$group == "treatment"]
> mmHg <- RATBP$mmHg                                # combined values
> mmHg

[1] 9.0 12.0 36.0 77.5 -7.5 32.5 69.0 24.0 63.0 87.5 77.5 40.0

> Contr

[1] 9.0 12.0 36.0 77.5 -7.5 32.5

> Treat

[1] 69.0 24.0 63.0 87.5 77.5 40.0
```

- (a) **Exact Permutation Method** The matrix `COMB` contains all of the possible combinations, $\binom{12}{6} = 924$ of them, for how the combined vector of 12 values can be split into a vector of size six. Each row in `COMB` has indices that are used to extract values from the combined vector `mmHg` to create the first sample. The second sample consists of the values in `mmHg` corresponding to the indices not present in a given row of `COMB`.

```
> nx <- length(Contr)
> ny <- length(Treat)
> COMB <- t(combn(nx + ny, nx))
> dim(COMB)

[1] 924    6

> nn <- dim(COMB)[1]
> tail(COMB)

[,1] [,2] [,3] [,4] [,5] [,6]
```

```
[919,]   6   7   8   9   10  12
[920,]   6   7   8   9   11  12
[921,]   6   7   8   10  11  12
[922,]   6   7   9   10  11  12
[923,]   6   8   9   10  11  12
[924,]   7   8   9   10  11  12
```

Using the indices for the 924th possible combination given in row 924 of the `COMB` matrix which are 7, 8, 9, 10, 11, 12, one obtains the values 69.0, 24.0, 63.0, 87.5, 77.5, 40.0 for the first sample. The complement of the values in the first sample are the values for the second sample. To obtain the complement one can use negative indices.

```
> mmHg[COMB[924, ]]
[1] 69.0 24.0 63.0 87.5 77.5 40.0

> mmHg[-COMB[924, ]]
[1] 9.0 12.0 36.0 77.5 -7.5 32.5
```

R Code 10.51 uses a `for` loop to store all of the possible values for $\hat{\theta} = \bar{z} - \bar{y}$ in the object `theta.hat`. The object `theta.obs` is the actual observed value of $\hat{\theta} = 26.5833 - 60.1667 = -33.5833$, the mean blood pressure drop of the control group minus the mean blood pressure drop of the treatment group.

R Code 10.51

```
> theta.obs <- mean(Contr) - mean(Treat)
> theta.obs
[1] -33.58333

> theta.hat <- numeric(nn)
> for(i in 1:nn){
+   theta.hat[i] <- mean(mmHg[COMB[i, ]]) - mean(mmHg[-COMB[i, ]])
+ }
```

The exact ϕ -value is the number of values in `theta.hat` less than or equal to -33.5833 divided by the total number of possible combinations, $\binom{12}{6} = 924$. This value is 0.0314, the exact permutation ϕ -value.

```
> pvalue <- sum(theta.hat <= theta.obs)/choose(12, 6)
> pvalue
[1] 0.03138528

> # Or
> mean(theta.hat <= theta.obs)
[1] 0.03138528
```

The R function `oneway_test()` from the `coin` package can also be used to compute an exact ϕ -value as illustrated in R Code 10.52 on the facing page.

R Code 10.52

```
> library(coin)
> oneway_test(mmHg ~ group, data = RATBP, alternative = "less",
+               distribution = "exact")

Exact 2-Sample Permutation Test

data: mmHg by group (control, treatment)
Z = -1.871, p-value = 0.03139
alternative hypothesis: true mu is less than 0
```

The $Z = -1.871$ in the output of R Code 10.52 is computed by dividing `theta.obs` (-33.5833) by the standard deviation of the permutation distribution (17.949). If the argument `distribution = "exact"` is changed to `distribution = "asymptotic"`, the ϕ -value is estimated based on the standard normal distribution.

(b) **Estimated Permutation Method** based on $B = 9999$ resamples of `mmHg` without replacement: Two solutions are provided. The first solution given in R Code 10.53 uses a `for()` loop where the resampling is done with the function `sample()`. The second solution illustrated in R Code 10.54 uses the `boot()` function from the `boot` package.

R Code 10.53

```
> B <- 10^4 - 1
> theta.hatE <- numeric(B)
> set.seed(1)
> for(i in 1:B){
+   index <- sample(12, 6, replace = FALSE)
+   theta.hatE[i] <- mean(mmHg[index]) - mean(mmHg[-index])
+ }
> pvalue <- (sum(theta.hatE <= theta.obs) + 1)/(B + 1)
> pvalue
[1] 0.0323
```

The ϕ -value is computed according to (10.52) and the estimated permutation ϕ -value based on $B = 9999$ permutation replications is 0.0323. The second approach creates the function `blood.fun()` which computes the mean difference between the first six values and the last six values in a vector of length twelve. Note that the object `data` is what will be resampled. To resample without replacement, the argument `sim = "permutation"` is used with `boot()`.

R Code 10.54

```
> library(boot)
> blood.fun <- function(data, i){
+   d <- data[i]
+   M <- mean(d[1:6]) - mean(d[7:12])
+   M
+ }
> set.seed(13)
> perm.obj <- boot(mmHg, statistic = blood.fun, R = B, sim = "permutation")
```

```
> perm.obj

DATA PERMUTATION

Call:
boot(data = mmHg, statistic = blood.fun, R = B, sim = "permutation")

Bootstrap Statistics :
      original    bias    std. error
t1* -33.58333 33.59821   17.85792

> pval.perm <- (sum(perm.obj$t <= perm.obj$t0) + 1)/(B + 1)
> pval.perm

[1] 0.0315
```

The p -value using the second approach in R Code 10.54 on the preceding page is also computed according to (10.52) and the estimated permutation p -value based on $B = 9999$ permutation replications stored in the object `pval.perm` is 0.0315.

(c) **Estimated Density of Permutation Distribution** The estimated density of the exact permutation distribution is represented with a solid line while the estimated density based on $B = 9999$ permutation replications is represented with a dashed line in Figure 10.22 on the next page. It is left to the reader to verify that the standard deviation of the exact permutation distribution is 17.949. The dotted line in Figure 10.22 on the facing page represents $aN(0, 17.949)$ distribution. The vertical line labeled $\hat{\theta}_{obs}$ represents the observed value of the statistic (-33.5833).

R Code 10.55

```
> p <- ggplot(data = data.frame(x = theta.hat), aes(x = x)) +
+   geom_density(fill = "pink", alpha = 0.2) +
+   theme_bw() +
+   labs(x = "$\\hat{\\theta} = \\bar{z} - \\bar{y}$", y = "") +
+   geom_density(data = data.frame(x = theta.hatE), aes(x = x),
+                color = "red", linetype = "dashed") +
+   geom_segment(aes(x = theta.obs, y = 0, xend = theta.obs,
+                    yend = 0.010), size = 0.1) +
+   stat_function(fun = dnorm, args = list(mean = 0, 17.94905),
+                linetype = "dotted") +
+   geom_text(data = NULL, x = theta.obs, y = 0.011,
+             label = "$\\hat{\\theta}_{obs}$", size = 4)
> p
```

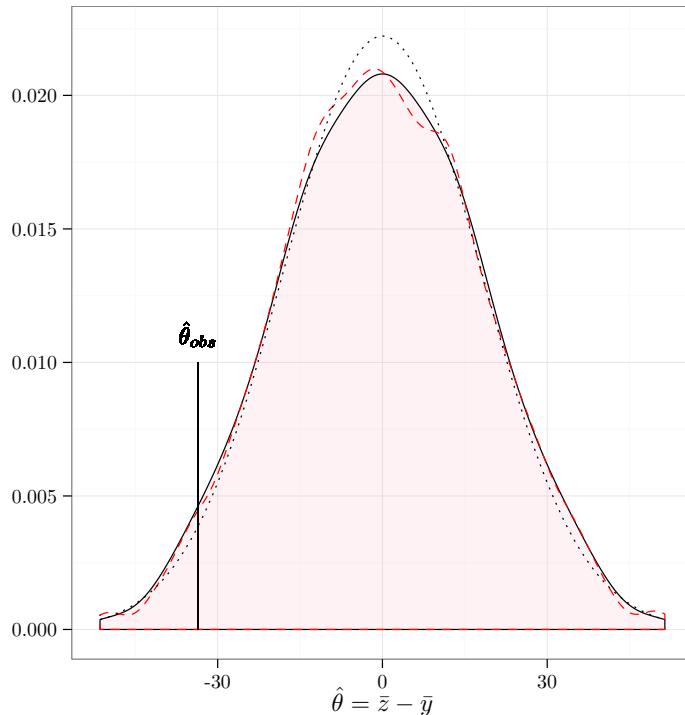


FIGURE 10.22: Density estimate for the permutation distribution of $\hat{\theta} = \bar{z} - \bar{y}$. The solid line is the density estimate for the exact permutation distribution of $\hat{\theta} = \bar{z} - \bar{y}$. The dashed line is the density estimate based on simulating the permutation distribution of $\hat{\theta} = \bar{z} - \bar{y}$. The dotted line is a normal distribution with a mean of zero and standard deviation equal to the standard deviation of the exact permutation distribution of $\hat{\theta} = \bar{z} - \bar{y}$.

10.11 Problems

1. Provide a brief explanation of the pros and cons of using a nonparametric test.
2. What are the assumptions made with respect to the distribution from which the data come when
 - (a) using a sign test?
 - (b) using the Wilcoxon signed-rank test?
3. When testing the median difference (ψ_D) of two dependent samples, does the sign test or the Wilcoxon signed-rank test have more power? For the recommended test, what assumption(s) must be made?
4. Explain when it might be appropriate to use
 - (a) a Kruskal-Wallis test.
 - (b) a Friedman Test.
5. Explain the concept “goodness-of-fit” as used for tests in this chapter.
6. The service department of an automobile dealer is being evaluated on the quality of their service. The parent company has randomly chosen 11 clients who have had their vehicles serviced in the last six months. These clients have been asked to evaluate their satisfaction level with the service they received. The following satisfaction scores were obtained:

Scores: 6 10 8 3 6 2 8 9 10 10 2

Service departments where the company has evidence that the median rating is more than 7 will receive a bonus. Perform the appropriate hypothesis test to determine if this service department should be awarded a bonus.

7. A Mendebaldea real estate agent claims Mendebaldea, Spain, has larger apartments than those in San Jorge, Spain. A San Jorge real estate agent disputes this claim. To resolve the issue, two random samples of the total area of several apartments (given in m^2) are taken from each community in 2002 and stored in the data frame **APTSIZE**.

Mendebaldea	90	92	90	83	85	105	136
San Jorge	75	75	53	78	52	90	78

- (a) Is there evidence to support the Mendebaldea agent's claim?
 - (i) Use an exact procedure.
 - (ii) Use an approximate procedure.
- (b) Find a confidence interval for the median of Mendebaldea minus the median of San Jorge with a confidence level of at least 0.90.

8. To study the retained carbon of trees, a random sample of 41 plots has been drawn in different mountainous regions of Navarra (Spain). In these plots, the carbon retained by leaves has been measured in kg/ha, depending on the forest classification: Areas with 90% or more beech trees (*Fagus Sylvatica*) are labeled monospecific, while areas with many species of trees are labeled multispecific. The data are stored in the data frame **FAGUS**. Is there evidence that leaves from different forest classifications retain the same amount of carbon?

9. The R data set **USJudgeRatings** provides 43 lawyers' ratings of state judges serving in the U.S. Superior Court. Use `help(USJudgeRatings)` to obtain a detailed view of the file. Suppose the variables integrity (INTG) and demeanor (DMNR) are chosen.

- (a) Test whether lawyers are more likely to give a judge high integrity ratings rather than high demeanor ratings.
- (b) Find a confidence interval for the median difference of integrity and demeanor with a confidence level of 0.90.

10. A company manager is studying the possibility of giving 20 minutes of rest to her employees in a resting room. To check the viability of this proposal, she analyzed 12 random days of productivity where employees took 20 minutes of rest and 12 random days where they did not. The groups' productivity scores are given in the following table where higher scores represent greater productivity.

With Rest	9	8	8	7	6	7	8	9	7	7	7	6
Without Rest	7	9	5	6	7	3	9	9	4	5	6	4

Is there evidence to suggest that taking a rest produces an increase in group productivity? Answer based on the results from a

- (a) Wilcoxon signed-rank test,
- (b) *t*-test, and a
- (c) Permutation test.

11. A Japanese company and an American company claim that they have developed a new technology to increase network transmission speeds. The marketing managers of both companies simultaneously announce that they can transmit 1 terabyte per second. To substantiate their claims, each company submits trial data (in seconds) of transmitting one terabyte with the new technologies:

Japanese company	0.98	0.95	0.91	0.93	0.94
American company	0.94	0.89	0.88	0.90	0.93

Is there evidence to suggest the transmission speed using the technology developed by the American company is superior to the transmission speed using the technology developed by the Japanese company? Compute the ϕ -value to answer the question with the following techniques:

- (a) Enumerate all possible combinations with the function `combn()` to find the ϕ -value for an exact permutation test.
- (b) Use the function `oneway_test()` from the `coin` package to calculate an appropriate ϕ -value. Does this ϕ -value match the one in part (a)?
- (c) Obtain an estimated permutation ϕ -value using the `boot()` function from the `boot` package.
- (d) What conclusion do the ϕ -values support?

12. The R data frame `sleep` shows the increase or decrease in hours of sleep for one group of patients when compared with a control group. The group was provided with two different soporific drugs. Is there evidence to suggest that one drug is superior (induces more sleep) to the other drug? Answer based on the results from a

- (a) Wilcoxon signed-rank test, and a
- (b) Permutation test.

13. In 1876, Charles Darwin had his book *The Effect of Cross and Self-Fertilization in the Vegetable Kingdom* published. Darwin planted two seeds, one obtained by cross-fertilization and the other by auto-fertilization in two opposite but separate locations of a pot. He planted seeds in this fashion in a total of 15 pots. Self-fertilization, also called autogamy or selfing, is the fertilization of a plant with its own pollen. Cross-fertilization, or allogamy, is the fertilization with pollen of another plant, usually of the same species. Darwin recorded the plants' heights in inches. The data frame `FERTILIZE` from the `PASWR2` package contains the data from this experiment.

- (a) Are the samples independent or paired?
- (b) Should normality be assumed? Regardless of how you answer the question, use a t -test to test if the mean difference (cross fertilized plant height – self fertilized plant height) is zero.
- (c) Use a Wilcoxon signed-rank test to test if the mean difference (cross fertilized plant height – self fertilized plant height) is zero.
- (d) Use a permutation test to test if the mean difference (cross fertilized plant height – self fertilized plant height) is zero.
- (e) Are the ϕ -values for the tests performed in (b), (c), and (d) similar?

14. Salaries for graduates of three engineering universities ten years after graduation are provided in the data frame `ENGINEER` of the `PASWR2` package. Seventeen graduates were randomly selected from each university, and their salaries in thousands of dollars were recorded. Is there any evidence to suggest graduates earn different salaries based on the university from which they graduated?

15. An engineering team is studying four different circuits that regulate the light intensity of a conference room. An accelerated life test was used to estimate the lifetime of each circuit. The results (lifetimes in thousands of hours) are stored in the data frame `CIRCUIT` and are

Design 1	3.07	1.20	0.95	1.38	5.48	1.19
Design 2	0.33	0.60	0.39	2.05	0.25	1.71
Design 3	1.75	2.41	2.02	2.24	1.69	1.24
Design 4	2.03	3.50	1.95	3.09	2.90	2.37

- (a) Do an exploratory analysis of the data and decide if normality can be assumed.
- (b) Use the Kruskal-Wallis test to decide if significant differences exist among the mean lifetimes of different circuit designs. Use $\alpha = 0.05$.
- (c) Use a permutation test to decide if significant differences exist among the mean lifetimes of different circuit designs. Use $\alpha = 0.05$. Hint: resample the `design` values inside an ANOVA.
16. The R data frame `airquality` shows daily air quality measurements in New York City, from May to September 1973.
- (a) Read the definition of the variables with the command `help(airquality)`. What does the symbol `NA` mean?
- (b) Create boxplots and density plots of `Month` versus `ozone`. Do the graphs for each month exhibit similar shapes?
- (c) Is it reasonable to assume that each month has a normally distributed ozone level?
- (d) Use a Kruskal-Wallis test to determine if there is evidence to suggest the mean ozone level is not the same for all the months.
- (e) Use a permutation test to determine if the mean ozone level is not the same for all the months. Hint: resample the `Month` values inside an ANOVA.
17. The R data frame `warpbreaks` gives the number of warp breaks per loom, where a loom corresponds to a fixed length of yarn.
- (a) Use the function `xtabs()` to create a contingency table containing the number of warp breaks classified by `wool` and `tension` from the data in `warpbreaks`.
- (b) Is there an association between wool type and tension level?
18. The music industry wants to know if the musical style on a CD influences how many illegal copies of it are sold. To achieve this purpose, the company chooses six cities randomly and writes down the number of illegal CDs available on the street categorized by music type: classical music, flamenco, heavy metal, and pop-rock. The data are shown in the following table.

City	Musical Style			
	Classical	Flamenco	Heavy Metal	Pop-Rock
City 1	4	1	6	9
City 2	3	4	5	10
City 3	2	1	8	14
City 4	5	3	2	7
City 5	2	3	6	14
City 6	9	1	2	6

- (a) Create boxplots and density plots of the number of illegal CDs available for each music style.
- (b) Are the distribution shapes similar?
- (c) Are there significant differences in the numbers of CDs available according to musical style?

19. A regulatory commission is investigating whether an association exists between certain banks and the regions where they are located as measured by the number of branches in each region. The branches belong to the following banks: Bilbao-Vizcaya (BBVA), Caja Madrid (CM), La Caixa (LC), and Banco Santander (BS). The regions are Navarra, Álava, Guipuzcoa, and Vizcaya. The number of branches classified by banks and regions follow.

Region	Bank			
	BBVA	CM	LC	BS
Navarra	47	8	54	43
Álava	31	5	21	17
Guipuzcoa	64	4	43	43
Vizcaya	134	11	104	66

- (a) Read the data into a data frame named **DF**, and create a frequency table of the number of bank branches classified by region and bank type using the function **xtabs()**.
- (b) Perform a test of independence to see if there is evidence of an association between certain banks and the regions where they are located as measured by the number of branches in each region. If any of the cells have less than five observations, combine those cells with other rows/columns before performing the test.
- (c) Perform a permutation test of independence to see if there is evidence of an association between certain banks and the regions where they are located as measured by the number of branches in each region. Hint: Create a data frame with one observation per row using the function **expand.dft()** from the package **vcdExtra**. Permute either the **bank** or the **region** column of the resulting data frame using the function **sample()** inside a call to **chisq.test()**.

20. The data frame **DEPEND** from the **PASWR2** package shows the number of dependent children (**number**) for 50 families (**count**). Use a goodness-of-fit test to see if a Poisson distribution with $\lambda = 2$ can reasonably be used to model the number of dependent children.

21. Is it reasonable to assume that the **time** variable from the data frame **PHONE** in the **PASWR2** package follows an exponential distribution with a mean equal to 3.7?

22. The data frame **TESTSCORES** in the **PASWR2** package gives the test grades of 29 students taking a basic statistics course.

- (a) Use the function **eda()** on the data. Can normality be assumed?
- (b) Use a Kolmogorov-Smirnov (Lilliefors) test to assess normality.

23. A government grant is funding a study to calculate how long it takes for the average consumer to establish an Internet connection. A random sample of 20 Internet users' connection times in seconds are 0.03, 0.48, 0.49, 0.52, 0.66, 0.69, 0.70, 0.76, 0.82, 1.20, 1.22, 1.39, 1.62, 1.85, 1.97, 2.25, 2.84, 3.44, 3.48, and 4.02.
- Use a Kolmogorov-Smirnov test to see if it is reasonable to assume that Internet connection time follows an exponential distribution with a mean of 1.5 seconds.
 - Use a chi-square test to see if it is reasonable to assume that Internet connection time follows an exponential distribution with a mean of 1.5 seconds.
 - Is there evidence to suggest that the median connection time is greater than 1 second?
24. Perform a simulation study to determine the power of both Kolmogorov-Smirnov's and Shapiro-Wilk's normality tests.
- Set the seed equal to 897, and simulate $m = 10,000$ samples of sizes $n = 10, 20, 30$, and 40 from a χ_1^2 distribution. Compute the simulated power for both tests using $\alpha = 0.05$.
 - Set the seed equal to 897, and simulate $m = 10,000$ samples of sizes $n = 10, 20, 30$, and 40 from a $Unif(0, 1)$ distribution. Compute the simulated power for both tests using $\alpha = 0.05$.
 - Set the seed equal to 897, and simulate $m = 10,000$ samples of sizes $n = 10, 20, 30$, and 40 from a $\beta(8, 3)$ distribution. Compute the simulated power for both tests using $\alpha = 0.05$.
 - Set the seed equal to 897, and simulate $m = 10,000$ samples of sizes $n = 10, 20, 30$, and 40 from a $N(0, 1)$ distribution. Compute the simulated power for both tests using $\alpha = 0.05$.
 - Generalize your findings from (a) through (d).
25. The R data frame `HairEyeColor` contains classifications of 592 students by gender, hair color, and eye color.
- Is hair color independent of eye color for men? If the expected count for any cell is less than five, the function `chisq.test()` will issue a warning stating that the Chi-squared approximation may be incorrect. Instead of collapsing categories, perform a permutation test of independence. Do you reach the same conclusion using a permutation test that you reached using `chisq.test()` to test for independence? Hint: Create a data frame with one observation per row using the function `expand.dft()` from the package `vcdExtra`. Permute either the `Hair` or the `Eye` column of the resulting data frame using the function `sample()` inside a call to `chisq.test()`.
 - Is hair color independent of eye color for women? Answer the question using a test of independence as well as a permutation test of independence. Do both approaches reach the same conclusion?
26. The sinking of the *Titanic* occurred on the 15th of April in 1912. The data frame `TITANIC3` contains information regarding class, gender, and survival as well as several other variables.

- (a) Create contingency tables of
- passenger class versus survival,
 - male passengers' class versus survival, and
 - female passengers' class versus survival.
- (b) Is there an association between class and survival for all passengers, men, and/or women?

27. Mental inpatients in the Virgen del Camino Hospital (Pamplona, Spain) are interviewed by expert psychiatrists to diagnose their illnesses. An important aspect in diagnosis is determining the severity of any delusions a patient might suffer. A new questioning technique has been developed to detect the presence of delusions. The technique assigns a score from 0 to 5, where 5 indicates the presence of strong delusions and 0 indicates no delusions. The psychiatrists wish to know if the new technique actually results in high scores for patients who have previously been diagnosed as suffering from severe delusions. The scores that follow were obtained from randomly selected patients who were known to suffer from delusions and those who were known not to suffer with delusions:

Delusions	Score						
Present	5	5	4	5	4	5	5
Absent	1	0	5	0	4	4	0

Do the data provide evidence that the new test yields higher scores for those patients who are known to suffer from delusions than for those who do not suffer from delusions?

28. It is believed by conservative psychiatrists that the use of illegal drugs can produce persistent hallucinations, even after drug use stops. Some more liberal psychiatrists dispute this assertion. The following data rate the severity of hallucinations suffered by randomly selected mental inpatients from the Virgen del Camino Hospital (Pamplona, Spain), where 5 indicates severe hallucinations and 0 indicates no hallucinations. The patients are divided by whether or not they consumed illegal drugs before being admitted to the hospital.

Illegal Drugs	Score							
Not Consumed	2	0	0	5	5	2	4	0
Consumed	0	4	5	5	4	5	5	2

Use an exact permutation test to determine if there is evidence that the severity of hallucinations in patients who have consumed illegal drugs is greater than the severity of hallucinations in patients who have not consumed illegal drugs.

29. Generate 10 values from a $N(0, 1)$ distribution with the seed set at 10. Calculate a bootstrap estimation of the standard error of \bar{X} using $B = 10,000$ replications. Repeat the experiment generating samples of sizes 100 and 1000 from the standard normal. Use both a `for` loop and the function `boot()` to generate the bootstrap distribution of \bar{X} . What conclusions can be drawn?

30. The “Wisconsin Card Sorting Test” is widely used by psychiatrists, neurologists, and neuropsychologists with patients who have a brain injury, neurodegenerative disease, or a mental illness such as schizophrenia. Patients with any sort of frontal lobe lesion generally

do poorly on the test. The data frame **WCST** and the following table contain the test scores from a group of 50 patients from the *Virgen del Camino* Hospital (Pamplona, Spain).

23	12	31	8	19	11	36	94	6	10	22	7	18	26	35	78	11
7	28	25	17	8	20	47	5	13	28	19	7	19	38	8	15	40
19	42	17	6	8	6	11	10	19	65	13	17	5	26	15	4	

- (a) Use the function `eda()` from the **PASWR2** package to explore the data and decide if normality can be assumed.
 - (b) What assumption(s) must be made to compute a 95% confidence interval for the population mean?
 - (c) Compute the confidence interval from (b).
 - (d) Compute a 95% BC_a bootstrap confidence interval for the mean test score.
 - (e) Should you use the confidence interval reported in (c) or the confidence interval reported in (d)?
31. A school psychologist administered the Stanford-Binet intelligence quotient (IQ) test in two counties. Forty randomly selected gifted and talented students were selected from each county. The Stanford-Binet IQ test is said to follow a normal distribution with a mean of 100 and standard deviation of 16. The data collected are stored in the data frame **SBIQ**.

County1							County2						
130	126	139	126	124	149	124	127	125	127	132	139	132	125
138	138	140	127	140	124	124	130	131	140	130	132	134	128
121	125	134	121	125	126	122	137	121	121	141	141	137	126
137	146	127	124	142	122	126	124	124	128	145	123	126	132
124	126	121	138	124	126	137	135	126	128	144	121	135	125
122	131	128	122	144			125	136	122	130	130		

- (a) Although the standard deviation for the Stanford-Binet IQ test is known, should it be used? Justify.
- (b) Construct a 96% confidence interval for the true average IQ difference for gifted and talented students between the two counties.
- (c) Construct a 96% bootstrap percentile confidence interval for the true average IQ difference for gifted and talented students between the two counties.

Chapter 11

Experimental Design

11.1 Introduction

This chapter deals with designed experiments where the experimenter follows a specific protocol established before the experiment starts. This protocol should dictate how randomization is performed and how measurements are taken. As a consequence of adhering to an established protocol, designed experiments allow the user to make strong inferences about the nature of observed differences.

Experiments are generally conducted to compare groups in terms of some response of interest. The methods considered in this chapter assume the response variable is continuous. The factors, independent variables whose levels are set by the experimenter, are categories or continuous variables that have been categorized into a fixed number of discrete levels. The treatments of an experiment are applied to **experimental units**, and measurements on the response variable are taken where the objective of the experiment is to compare the observed responses. When the combinations of the levels of two or more factors form the treatments of interest, the experiment is known as a **factorial design**.

For example, an agricultural researcher may be interested in determining which of three different fertilizers produces the greatest soybean yield. In this example, the three fertilizers correspond to three treatments the experimenter wants to compare, and the three fertilizers collectively constitute a factor. In the event a second factor, such as two different methods of watering the soybeans, is of interest, the experiment will consist of $3 \times 2 = 6$ different treatment combinations and is called a factorial design.

Suppose an agronomist is interested in determining which of three types of wheat (*Triticum aestivum*, *Triticum durum*, or *Triticum spelta*) produces the greatest yield for a particular geographical location. Available to the agronomist are six plots of equal size, all in the same geographical area. In this setting, the factor of interest is wheat, and the treatments are the three types of wheat. When the plots are homogeneous in their physical characteristics, distinguishing differences in treatments becomes easier if differences exist. Since there are likely to be some differences in the plots, the researcher will want to assign the wheat types to the six plots randomly in order to minimize any possible bias due to plots. By randomizing the assignment of treatments, the possibility of confounding differences due to types with differences due to plots is minimized. When the assignment of treatments is done in a completely random fashion, the design is known as a completely randomized design (CRD). When experimental units are similar (homogeneous) with respect to some characteristic, they can be grouped together into **blocks**. In the wheat study, some of the plots may be exposed to more sun than other plots, or some plots may receive more water than other plots. When the experimental units are more homogeneous within a block than they are between blocks, treatments are assigned to experimental units within each block to reduce variability. Such a design is known as a randomized complete block design (RCBD). See Figures 11.1 and 11.2 on the next page for possible assignments of treatments for a

CRD and RCBD, respectively.

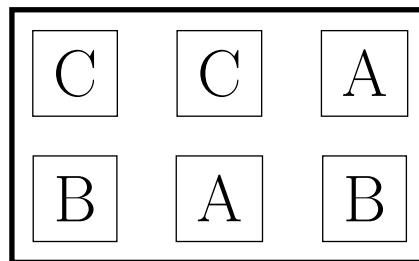


FIGURE 11.1: Representation of a completely randomized design where treatments A, B, and C are assigned at random to six experimental units

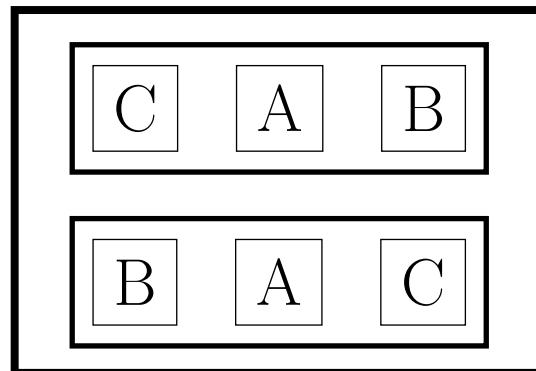


FIGURE 11.2: Representation of a randomized complete block design where treatments A, B, and C are assigned at random to three experimental units in each block

Before proceeding further, some of the more important experimental design concepts are explained:

- **Treatments** are levels of a factor or combinations of factor levels the experimenter wants to compare.
- **Experimental units** are anything to which treatments are applied, for example, animals, plots, plants, or people.
- **Responses** are outcomes observed after the application of a treatment to an experimental unit.
- **Experimental error** is random variation present in the experiment not under the control of the experimenter. Experimental error can happen because of measurement error defined as obtaining different responses from measuring the same quantity in different trials, and the natural variation of realizing different responses from experimental units given the same treatment.

- **Treatment structure** specifies the set of factors the experimenter has selected to study or compare.
- **Design structure** defines how experimental units are assigned to treatment groups.
- **Randomization** is the use of some well-defined probabilistic mechanism to assign treatments to experimental units. Randomization reduces the possibility of bias and confounding. Randomization should also be used, if possible, with any variable not under the direct control of the experimenter that may influence the measured response.
- **Replication** is the independent assignment of several experimental units to each treatment (factor combination) resulting in independent observations. Replication shows the results are reproducible and allows the experimenter to estimate the experimental error. When the number of experimental units is the same for all treatments, the design is referred to as a balanced design. Unbalanced designs do not have an equal number of experimental units for all treatments.

Understanding both the treatment structure and the design structure is essential for conducting proper data analysis. Different statistical models will be introduced throughout the chapter, and analysis of variance (ANOVA) will be used to perform tests on those models. ANOVA measures the differences in means due to the factors that are fixed effects. When the effects are random, variance components are used to determine the variability due to the factors. All of the fixed effects models assume

1. The measured responses are independent of one another;
2. The model errors are independent of one another and follow a normal distribution; and
3. The variance is homogeneous across treatments.

When using statistical models, it is important to keep in mind that a model is simply a mathematical expression of how the researcher believes the response is explained using the independent variables of the experiment (predictors). Models are expressed in R with the syntax `response ~ predictors`, where `~` means that the `response` is modeled by the `predictors`. There may be several plausible models for a particular experiment. Finding an adequate model is an iterative process that starts with

1. Identifying an appropriate model based on the treatment and design structure of the experiment;
2. Validating the model's assumptions using diagnostic plots; and
3. Selecting a different model or transforming the response variable when the model's assumptions are not satisfied until a plausible model is found.

Once a model has been validated, formal inference to test for no treatment effects (equality of treatment means) and estimation of the model's parameters can be undertaken. In the event formal inference suggests differences in treatments, multiple comparisons are used to determine which treatments are significantly different from one another.

Motivational Example: Tires A tire manufacturer is interested in investigating the handling properties for different tread patterns. The data frame `TIRE` has the stopping distances measured to the nearest foot for a standard-sized car to come to a complete stop from a speed of 60 miles per hour. There are six measurements of the stopping distance for four different tread patterns labeled A, B, C, and D. The same driver and car were used for

all 24 measurements. While the numbers in **TIRE** do not reveal the randomization scheme used for the experiment, the order of treatments was assigned at random.

One way to ensure treatments are randomly assigned to the 24 runs is to use a random number generator. This can be accomplished with R by typing

```
> population <- rep(LETTERS[1:4], 6)
> set.seed(4)
> Treatment <- sample(population, size = 24, replace = FALSE)
> DF <- data.frame(Run = 1:24, Treatment)
> head(DF)    # Show first six rows of DF
```

Run	Treatment
1	C
2	A
3	C
4	B
5	A
6	A

In particular, this randomization would assign tire tread C to the first run, A to the second run, and so on. It is always a good idea to examine experimental data graphically before initiating any formal inferential procedure. Side-by-side boxplots are often a good starting point when comparing several treatments. When the number of observations in each treatment group is relatively small, dotplots will often prove more helpful than boxplots. The function `oneway.plots()` from the **PASWR2** package is used to create Figure 11.3 on the facing page.

From the boxplots and dotplots shown in Figure 11.3 on the next page, it appears that there are differences in stopping distances based on different tire treads. At this point, it would be nice to formalize the last sentence with an inferential procedure. It is initially tempting to many to perform pairwise *t*-tests on all six ($\binom{4}{2} = 6$) of the pairwise differences; however, this should not be done! If the probability of correctly accepting the null hypothesis is $1 - \alpha = 0.95$, then the probability of correctly accepting the null hypothesis for all six pairwise tests assuming independence among tests would be $(0.95)^6 = 0.7351$. The type I error rate is not 5% but 26.4908% in this case. Of course, the more treatments that are compared, the more likely one is to make a type I error. What would the type I error rate be if the individual error rate for a single comparison is 5% and seven treatments were compared? (Answer: 65.9438%) The appropriate procedure for testing the equality of several means is analysis of variance, which is introduced in the context of a completely randomized design.

Completely Randomized Design The simplest randomized design for comparing several treatments is the completely randomized design (CRD). CRDs have $a \geq 2$ treatments to compare and N experimental units. Each treatment is applied to n_i ($i = 1, 2, \dots, a$) experimental units, where $n_1 + n_2 + \dots + n_a = N$. In order to conduct the experiment, the researcher randomly assigns treatments to the experimental units (design structure). Although the sizes of the a samples need not be identical, the power of the test is maximized when $n_1 = n_2 = \dots = n_a$ for the a treatments. On each experimental unit, a response variable Y is measured. In Example 11.1 on the preceding page, Y represents the distance to the nearest foot required to stop a particular model of car traveling at 60 miles per hour using four different brands of tires. The CRD, when there is one factor with a levels (treatments) and no assumed relationships among the a levels, is called a **one-way treatment** structure. The layout for such a design is shown in Table 11.1 on the next page.

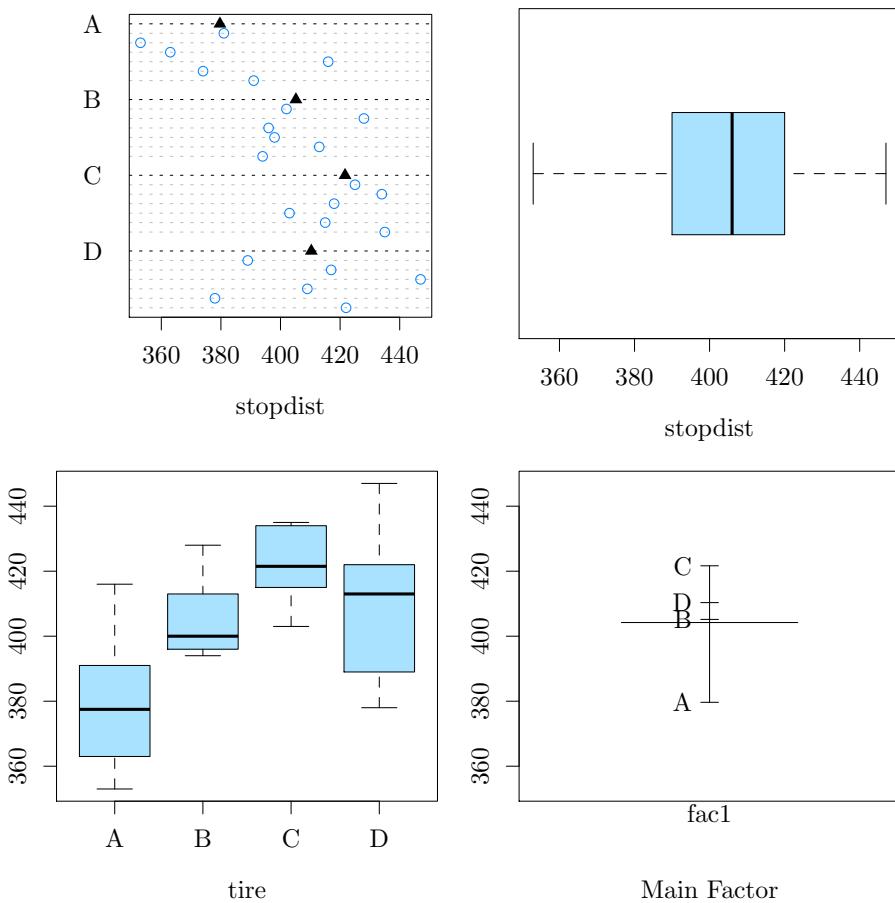


FIGURE 11.3: Output from the function `oneway.plots(stopdist, tire)` where the top left graph depicts a dotplot where the open circles below a letter are the individual stopping distances for a particular tire and the solid triangles are the mean stopping distances for a particular tire; the top right boxplot depicts the stopping distance for all tires, while the bottom left boxplot depicts side-by-side boxplots for the stopping distances by tire; and the bottom right graph is a design plot showing means with short horizontal lines for each tire using the data frame `TIRE`.

Table 11.1: One-way design

Treatment	Responses				Totals	Means
1	Y_{11}	Y_{12}	\dots	Y_{1n_1}	$Y_{1\bullet}$	$\bar{Y}_{1\bullet} = \sum Y_{1j}/n_1$
2	Y_{21}	Y_{22}	\dots	Y_{2n_2}	$Y_{2\bullet}$	$\bar{Y}_{2\bullet} = \sum Y_{2j}/n_2$
\vdots	\vdots	\vdots		\vdots	\vdots	\vdots
a	Y_{a1}	Y_{a2}	\dots	Y_{an_a}	$Y_{a\bullet}$	$\bar{Y}_{a\bullet} = \sum Y_{aj}/n_a$
					$Y_{\bullet\bullet}$	$\bar{Y}_{\bullet\bullet}$

Notation is critical, and the following conventions are used throughout the chapter. The

sum of the observations in the i^{th} treatment group is $Y_{i\bullet} = \sum_{j=1}^{n_i} Y_{ij}$, and the mean of the observations in the i^{th} treatment group is $\bar{Y}_{i\bullet} = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} = \frac{Y_{i\bullet}}{n_i}$. The bar indicates a mean while the dot (\bullet) indicates that values have been added over the indicated subscript. The sum of all observations is $Y_{\bullet\bullet} = \sum_{i=1}^a \sum_{j=1}^{n_i} Y_{ij}$. The grand mean of all observations is denoted $\bar{Y}_{\bullet\bullet} = \frac{1}{N} \sum_{i=1}^a \sum_{j=1}^{n_i} Y_{ij} = \frac{Y_{\bullet\bullet}}{N}$.

To describe the observations, the linear statistical model

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij} \text{ for } i = 1, 2, \dots, a \text{ and } j = 1, 2, \dots, n_a \quad (11.1)$$

is used, where Y_{ij} is the j^{th} observation of the i^{th} treatment, μ is a parameter common to all treatments called the overall mean, τ_i is a parameter unique to the i^{th} treatment called the i^{th} treatment effect, and ϵ_{ij} is a random error component. For hypothesis testing, the model errors are assumed to be normally and independently distributed with mean zero and constant standard deviation ($NID(0, \sigma)$). The careful reader will realize that this implies the variance is assumed to be constant for all a treatments.

The model given in (11.1) can be used for two different scenarios with respect to the treatment effects. When the treatments are specifically chosen by the experimenter and there is no desire to extend the results to other treatments, the model is referred to as a **fixed effects model**. On the other hand, when the treatments are selected at random from a larger population of possible treatments and the experimenter would like to extend the conclusions of the experiment of all treatments in the population, the model is called a **random effects model**. What follows deals with the fixed effects model.

11.2 Fixed Effects Model

Although there are a means μ_i , one for each of the a treatments, Model (11.1) uses $a + 1$ parameters (μ and the $a \tau_i$ s) to describe the a means. This implies that μ and τ_i are not uniquely determined. A frequently used solution is to impose the constraint $\sum_{i=1}^a n_i \tau_i = 0$. When the n_i s are equal, the constraint can be written as $\sum_{i=1}^a \tau_i = 0$. Although other solutions to the overparameterized model exist, estimators for Model (11.1) in this text will only consider the sum to zero constraint on the τ_i s as a solution for the overparameterized model. Different software packages often impose differing constraints on the overparameterized model, and the user should pay close attention to how the software computes estimates for the model. The natural and unbiased estimator for μ_i is $\bar{Y}_{i\bullet}$, the average of the observations in that treatment group. Likewise, the natural and unbiased estimator of μ is $\bar{Y}_{\bullet\bullet}$, the grand mean of all of the responses. Using either least squares or maximum likelihood to derive estimators of μ and τ_i for Model (11.1) results in the aforementioned quantities. Verifying the least squares estimators of the parameters of Model (11.1) (using the sum to zero constraint) as well as the maximum likelihood estimators of Model (11.1) (using the sum to zero constraint) is left as an exercise for the reader.

Using maximum likelihood techniques, an estimator of σ^2 , $\tilde{\sigma}^2$, is found to be

$$\sum_{i=1}^a \sum_{j=1}^{n_i} \frac{(Y_{ij} - \bar{Y}_{i\bullet})^2}{N}.$$

Unfortunately, the expected value of $\tilde{\sigma}^2$ is $\frac{(N-a)\sigma^2}{N}$, which is a biased estimator of σ^2 . Since,

$$E(aX) = a \cdot E(X),$$

$$\frac{N}{N-a} \cdot \sum_{i=1}^a \sum_{j=1}^{n_i} \frac{(Y_{ij} - \bar{Y}_{i\bullet})^2}{N}$$

will yield an unbiased estimator of σ^2 :

$$\hat{\sigma}^2 = \sum_{i=1}^a \sum_{j=1}^{n_i} \frac{(Y_{ij} - \bar{Y}_{i\bullet})^2}{N-a}.$$

These facts are summarized in Table 11.2.

Table 11.2: Parameters and estimators for fixed effects, one-way CRD Model (11.1)

Parameter	Estimator
μ	$\bar{Y}_{\bullet\bullet}$
μ_i	$\bar{Y}_{i\bullet}$
τ_i	$\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet}$
ε_{ij}	$Y_{ij} - \bar{Y}_{i\bullet}$

Although estimating the parameters for Model (11.1) is important, the goal of the experimenter is generally to discern whether or not the a treatment means are equal, and if they are not equal, which treatments are better (for example, have a higher mean). Specifically, the null hypotheses of interest are

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_a \quad \text{versus} \quad H_1 : \mu_i \neq \mu_j \text{ for some } (i, j).$$

When the null hypothesis is true, all treatments have a common mean μ and an equivalent statement of the null hypothesis can be written in terms of the treatment effects as

$$H_0 : \tau_1 = \tau_2 = \cdots = \tau_a = 0 \quad \text{versus} \quad H_1 : \tau_i \neq 0 \text{ for some } i.$$

Consequently, testing the equality of treatment means is equivalent to testing that the treatment effects are all zero. As mentioned earlier, the appropriate procedure for testing the null hypothesis of equal treatment means is the analysis of variance, which is a decomposition of the total variability into its component parts as shown next.

11.3 Analysis of Variance (ANOVA) for the One-Way Fixed Effects Model

Consider the identity

$$Y_{ij} - \bar{Y}_{\bullet\bullet} = (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet}) + (Y_{ij} - \bar{Y}_{i\bullet}), \quad (11.2)$$

which partitions the deviation of any observation from the grand mean into two parts. The first part, $(\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})$, is the deviation of the i^{th} treatment mean from the grand mean.

The second part is the deviation of the observation from the i^{th} treatment mean. Squaring and summing both sides of (11.2) produces

$$\begin{aligned} \sum_{i=1}^a \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{\bullet\bullet})^2 &= \sum_{i=1}^a \sum_{j=1}^{n_i} [(\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet}) + (Y_{ij} - \bar{Y}_{i\bullet})]^2 \\ &= \sum_{i=1}^a n_i (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2 + \sum_{i=1}^a \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\bullet})^2 \\ &\quad + 2 \sum_{i=1}^a \sum_{j=1}^{n_i} (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})(Y_{ij} - \bar{Y}_{i\bullet}). \end{aligned} \quad (11.3)$$

However, the cross product in (11.3) is zero since

$$\sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\bullet}) = Y_{i\bullet} - n_i \bar{Y}_{i\bullet} = Y_{i\bullet} - n_i \cdot \frac{Y_{i\bullet}}{n_i} = 0.$$

Consequently,

$$\sum_{i=1}^a \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{\bullet\bullet})^2 = \sum_{i=1}^a n_i (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2 + \sum_{i=1}^a \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\bullet})^2 \quad (11.4)$$

which says the total variability in the data can be partitioned into two parts. The quantity $\sum_{i=1}^a n_i (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2$ measures the difference between the observed treatment means and the grand mean. Specifically, it is a measure of variability due to the treatments and is denoted $SS_{\text{Treatment}}$ (sum of squares due to treatments). The quantity $\sum_{i=1}^a \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\bullet})^2$ measures the differences of observations within a treatment from the treatment mean, which must be due to error and is referred to as SS_{Error} (sum of squares due to error). The quantity on the left-hand side of the equal sign in (11.4) is called the total sum of squares corrected for the mean and is denoted SS_{Total} . The symbolic representation of (11.4) is

$$SS_{\text{Total}} = SS_{\text{Treatment}} + SS_{\text{Error}}. \quad (11.5)$$

Since there are a total of $\sum_{i=1}^a n_i = N$ observations, SS_{Total} has $N - 1$ degrees of freedom. One degree of freedom is lost for estimating μ with the grand mean, $\bar{Y}_{\bullet\bullet}$. $SS_{\text{Treatment}}$ has $a - 1$ degrees of freedom since there are a treatment means and SS_{Error} has $N - a$ degrees of freedom. To adjust for the number of treatments, $SS_{\text{Treatment}}$ is divided by its degrees of freedom, $a - 1$. The resulting quantity is known as the **mean square treatment** $MS_{\text{Treatment}} = \frac{SS_{\text{Treatment}}}{df_{\text{Treatment}}}$ and is also called the between-treatments error variance. In order to know whether the $MS_{\text{Treatment}}$ value is large, it is compared to an estimate of σ^2 , namely, MS_{Error} , where $MS_{\text{Error}} = \frac{SS_{\text{Error}}}{df_{\text{Error}}}$, which is also called the within-treatments error variance. Note that MS_{Error} can be expressed as

$$\hat{\sigma}^2 = MS_{\text{Error}} = \frac{SS_{\text{Error}}}{df_{\text{Error}}} = \frac{\sum_{i=1}^a \left[\sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\bullet})^2 \right]}{df_{\text{Error}}}. \quad (11.6)$$

If the term within the square braces is divided by its degrees of freedom ($n_i - 1$), it is easy to recognize that quantity as the sample variance for the i^{th} treatment:

$$S_i^2 = \sum_{j=1}^{n_i} \frac{(Y_{ij} - \bar{Y}_{i\bullet})^2}{n_i - 1}, \quad i = 1, 2, \dots, a \quad (11.7)$$

Combining the sample variances, a single estimate of the population variance emerges as

$$\begin{aligned} \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2 + \cdots + (n_a - 1)S_a^2}{(n_1 - 1) + (n_2 - 1) + \cdots + (n_a - 1)} &= \frac{\sum_{i=1}^a \left[\sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\bullet})^2 \right]}{\sum_{i=1}^a (n_i - 1)} \\ &= \frac{SS_{\text{Error}}}{N - a} = MS_{\text{Error}}. \end{aligned}$$

The pooled estimate of the variance from the two-sample t -test in Section 9.7.4 has now been generalized for a different samples.

If there are no differences among the a treatment means, $MS_{\text{Treatment}}$ is an unbiased estimate of σ^2 , and the ratio of $MS_{\text{Treatment}}/MS_{\text{Error}}$ will be close to 1. If differences actually exist among the a treatment means, then the ratio, $MS_{\text{Treatment}}/MS_{\text{Error}}$ should be larger than 1. In fact, it can be shown that

$$E(MS_{\text{Error}}) = \sigma^2 \quad \text{and} \quad E(MS_{\text{Treatment}}) = \sigma^2 + \sum_{i=1}^a \frac{n_i \tau_i^2}{a - 1},$$

implying that when H_0 is false, $E(MS_{\text{Treatment}}) > E(MS_{\text{Error}})$ since some $\tau_i \neq 0$. When H_0 is true, $\tau_i = 0$ for all i and $E(MS_{\text{Treatment}}) = E(MS_{\text{Error}}) = \sigma^2$. With a little effort, it can be shown that

$$\frac{MS_{\text{Error}}}{\sigma^2} \sim \frac{\chi_{df_{\text{Error}}}^2}{df_{\text{Error}}} = \frac{\chi_{N-a}^2}{N - a}$$

regardless of whether H_0 is true or not, and that

$$\frac{MS_{\text{Treatment}}}{\sigma^2} \sim \frac{\chi_{df_{\text{Treatment}}}^2}{df_{\text{Treatment}}} = \frac{\chi_{a-1}^2}{a - 1}$$

when H_0 is true independently of MS_{Error} . Consequently, using Definition 6.2 on page 397, when H_0 is true, the ratio $MS_{\text{Treatment}}/MS_{\text{Error}} \sim F_{a-1; N-a}$. Thus, H_0 is rejected in an α -level test if $F_{\text{obs}} > f_{1-\alpha; a-1, N-a}$, where $F_{\text{obs}} = MS_{\text{Treatment}}/MS_{\text{Error}}$. The R function `aov()` used with `summary()` returns a table similar to Table 11.3 on the following page.

Example 11.1 \triangleright **Tire ANOVA Table** \triangleleft Use the data frame `TIRE` and compute the values for the ANOVA table using both the formulas and the R function `summary(aov())` to test the null hypothesis that all the tire treads have identical mean stopping distances versus the alternative hypothesis that there is at least one mean that is different.

Solution: The hypotheses being tested are

$$H_0 : \tau_i = 0 \text{ for all } i \quad \text{versus} \quad H_1 : \tau_i \neq 0 \text{ for some } i.$$

Table 11.3: ANOVA table for one-way completely randomized design

Source of Variation (Source)	Degrees of Freedom (df)	Sum of Squares (SS)	Mean Square (MS)	F
Treatments	$a - 1$	$SS_{\text{Treatment}} = \sum_{i=1}^a n_i (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2$	$MS_{\text{Treatment}} = \frac{SS_{\text{Treatment}}}{a - 1}$	$\frac{MS_{\text{Treatment}}}{MS_{\text{Error}}}$
Error	$N - a$	$SS_{\text{Error}} = \sum_{i=1}^a \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\bullet})^2$	$MS_{\text{Error}} = \frac{SS_{\text{Error}}}{N - a}$	
Total	$N - 1$	$SS_{\text{Total}} = \sum_{i=1}^a \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{\bullet\bullet})^2$		

$$\begin{aligned}
 SS_{\text{Treatment}} &= \sum_{i=1}^a n_i (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2 \\
 &= 6 \cdot (379.6667 - 404.2083)^2 + 6 \cdot (405.1667 - 404.2083)^2 \\
 &\quad + 6 \cdot (421.6667 - 404.2083)^2 + 6 \cdot (410.3333 - 404.2083)^2 = 5673.125.
 \end{aligned}$$

$$\begin{aligned}
 SS_{\text{Total}} &= \sum_{i=1}^a \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{\bullet\bullet})^2 \\
 &= (391 - 404.2083)^2 + (374 - 404.2083)^2 + (416 - 404.2083)^2 \\
 &\quad + \dots + (389 - 404.2083)^2 = 12771.9583.
 \end{aligned}$$

$$SS_{\text{Error}} = SS_{\text{Total}} - SS_{\text{Treatment}} = 12771.96 - 5673.12 = 7098.8333.$$

$$MS_{\text{Treatment}} = SS_{\text{Treatment}}/df_{\text{Treatment}} = 5673.125/3 = 1891.0417.$$

$$MS_{\text{Error}} = SS_{\text{Error}}/df_{\text{Error}} = 7098.8333/20 = 354.9417.$$

$$F = MS_{\text{Treatment}}/MS_{\text{Error}} = 5673.125/7098.8333 = 5.3278.$$

The p -value for the test is $P(F_{3, 20} \geq 5.3278) = 0.0073$. Based on the small p -value, the null hypothesis of no tire tread effect ($\tau_i = 0$ for all i) is rejected. This suggests at least one tire tread effect is not zero. Thus, the question then becomes, “Which tire tread has the shortest mean stopping distance?” The statistical conclusion as well as the validity of any multiple comparison procedures used to detect individual differences between tire treads assume the one-way model (11.1) is sound. Checking model assumptions for the one-way

CRD is discussed in Section 11.5, followed by multiple comparison procedures in Section 11.7.

The ANOVA values are computed with R Code 11.1 and are shown in Table 11.4 on the following page.

R Code 11.1

```
> TreatmentMeans <- tapply(TIRE$stopdist, TIRE$tire, mean)
> TreatmentMeans                               #  $Y_{\{i \text{ dot}\}}$ 

      A          B          C          D
379.6667 405.1667 421.6667 410.3333

> a <- length(TreatmentMeans)
> N <- length(TIRE$stopdist)
> xtabs(~TIRE$tire)

TIRE$tire
A B C D
6 6 6 6

> n <- xtabs(~TIRE$tire)[1]                  # note all have 6
> names(n) <- NULL                           # remove name
> df.treat <- a - 1                          # df for treatments
> df.error <- N - a                          # df for error
> GrandMean <- mean(TIRE$stopdist)           #  $Y_{\{\text{dot dot}\}}$ 
> GrandMean

[1] 404.2083

> SStreat <- n*sum((TreatmentMeans - GrandMean)^2)
> SStreat

[1] 5673.125

> SStotal <- sum((TIRE$stopdist - GrandMean)^2)
> SStotal

[1] 12771.96

> SSerror <- sum((TIRE$stopdist - rep(TreatmentMeans, each = n))^2)
> SSerror

[1] 7098.833

> MStreat <- SStreat/df.treat
> MStreat

[1] 1891.042

> MSerror <- SSerror/df.error
> MSerror

[1] 354.9417
```

```
> Fobs <- MSTreat/MSerror
> Fobs

[1] 5.327753

> pvalue <- pf(Fobs, df.treat, df.error, lower = FALSE)
> pvalue

[1] 0.007315521
```

Using the two functions `summary()` and `aov()` together is illustrated in R Code 11.2.

R Code 11.2

```
> mod <- aov(stopdist ~ tire, data = TIRE)
> summary(mod)           # Show ANOVA table for mod

Df Sum Sq Mean Sq F value    Pr(>F)
tire      3   5673   1891.0    5.328 0.00732 **
Residuals 20   7099    354.9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The treatment means and the grand mean can also be computed using the function `model.tables()`:

```
> model.tables(mod, type = "means")

Tables of means
Grand mean

404.2083

tire
tire
  A     B     C     D
379.7 405.2 421.7 410.3
```

Table 11.4: Tire ANOVA table

Source of Variation (Source)	Degrees of Freedom (<i>df</i>)	Sum of Squares (<i>SS</i>)	Mean Square (<i>MS</i>)	<i>F</i>
Treatments	$4 - 1 = 3$	5673.125	$\frac{5673.125}{3} = 1891.0417$	$\frac{1891.0417}{354.9417} = 5.3278$
Error	$24 - 4 = 20$	7098.8333	$\frac{7098.8333}{20} = 354.9417$	
Total	$24 - 1 = 23$	12771.9583		

11.4 Power and the Non-Central F Distribution

The concept of power and the non-central t -distribution for one-sample and two-sample problems was discussed in Section 9.7. In this section, computing power is extended to the $a \geq 2$ samples problem. Specifically, the problem of determining the required sample size to detect a given difference is addressed. Consider a slightly different but equivalent expression for the $MS_{\text{Treatment}}$ and MS_{Error} given in Table 11.3 on page 706 when $a = 2$.

$$F = \frac{MS_{\text{Treatment}}}{MS_{\text{Error}}} = \frac{\frac{\sum_{i=1}^2 \sum_{j=1}^{n_i} (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2}{df_{\text{Treatment}}}}{\frac{\sum_{i=1}^2 \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_{i\bullet})^2}{df_{\text{Error}}}} \equiv \frac{(\bar{Y}_{1\bullet} - \bar{Y}_{2\bullet})^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1}}{S_p^2} \quad (11.8)$$

The reader should verify that the right-hand side of (11.8) is the same as the expression on the left of the \equiv . Two facts that should be kept in mind during the verification are $\sum_{j=1}^{n_i} Y_{ij} = \bar{Y}_{i\bullet} n_i$ for $i = 1, 2$ and $(n_1 + n_2)\bar{Y}_{\bullet\bullet} = \sum_{i=1}^2 \sum_{j=1}^{n_i} Y_{ij}$. Rewriting the right side of (11.8) gives

$$F = \frac{(\bar{Y}_{1\bullet} - \bar{Y}_{2\bullet})^2}{S_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)} = \left[\frac{(\bar{Y}_{1\bullet} - \bar{Y}_{2\bullet})}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \right]^2 = [t]^2. \quad (11.9)$$

One verifies that the pooled t -test from Section 9.7 is simply a special case of the F -test used in ANOVA when $a = 2$. It is important to emphasize that the equivalence of the pooled t -test and the F -test used in ANOVA for $a = 2$ groups applies only to the non-directional hypothesis $H_1 : \mu_1 \neq \mu_2$ because

$$[t_{1-\alpha/2; df_{\text{Error}}}]^2 = f_{1-\alpha; 1, df_{\text{Error}}}, \text{ but} \quad (11.10)$$

$[t_{1-\alpha; df_{\text{Error}}}]^2 \neq f_{1-\alpha; 1, df_{\text{Error}}}$, as would be required for a directional hypothesis. In Section 9.7, the non-centrality parameter γ for the pooled t -test was defined as

$$\gamma = \frac{\mu_1(X, Y) - \mu_0(X, Y)}{\sigma_{\bar{X}-\bar{Y}}}.$$

An equivalent expression for defining the non-centrality parameter is

$$\gamma = \frac{(\mu_1 - \mu_2) \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1/2}}{\sigma}. \quad (11.11)$$

One should take note of the similarities between

$$t = \frac{(\bar{Y}_{1\bullet} - \bar{Y}_{2\bullet}) \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1/2}}{S_p} \quad (11.12)$$

and (11.11). Specifically, the quantity in (11.12) is used to measure the statistical differences between the **sample** means. In a similar fashion, (11.11) is used to measure the statistical differences between the population means. Rewriting (11.12) and (11.11), one notes

$$F = t^2 = \frac{(\bar{Y}_{1\bullet} - \bar{Y}_{2\bullet})^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1}}{S_p^2} = \frac{MS_{\text{Treatment}}}{MS_{\text{Error}}}$$

and

$$\lambda = \gamma^2 = \frac{(\mu_1 - \mu_2)^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)^{-1}}{\sigma^2} = \frac{SS_{\text{Hypothesis}}(\text{population})}{\sigma^2}$$

where $SS_{\text{Hypothesis}}(\text{population})$ is the sum of squares for treatments obtained by replacing $\bar{Y}_{1\bullet}$ with μ_1 , $\bar{Y}_{2\bullet}$ with μ_2 , and $\bar{Y}_{\bullet\bullet}$ with $\frac{n_1\mu_1+n_2\mu_2}{n_1+n_2}$. By defining the non-centrality parameter λ as the ratio of $SS_{\text{Hypothesis}}(\text{population})$ to σ^2 , it becomes easy to calculate λ using statistical software. The $SS_{\text{Hypothesis}}(\text{population})$ will always be the sum of squares formula for the H_0 being tested, thus this method of computing λ extends to whatever hypothesis the user would like to test. It is not limited merely to the equality of treatment means. For any completely randomized design, $SS_{\text{Hypothesis}}(\text{population}) = \sum_{i=1}^a n_i (\mu_{i\bullet} - \bar{\mu}_{\bullet\bullet})^2$, where $\bar{\mu}_{\bullet\bullet} = (\sum_{i=1}^a n_i \mu_{i\bullet}) / (\sum_{i=1}^a n_i)$. Recall that power is the probability that the null hypothesis will be rejected when it is false. In this case,

$$\text{Power}(\lambda) = \mathbb{P}[F_{a-1; N-a, \lambda}^* > f_{1-\alpha; a-1; N-a; \lambda=0}]. \quad (11.13)$$

$\text{Power}(\lambda)$ is maximized when all a groups have an equal number of observations; however, using $SS_{\text{Hypothesis}}$ to compute the non-centrality parameter adjusts for experiments with different sample sizes. R has the function `power.anova.test()`, which can be used to determine the sample size for the a samples when resources are allocated such that each group has the same size.

Example 11.2 \triangleright **Tires' Stopping Distance** \triangleleft Suppose the tire manufacturer believes the true mean stopping distance for tread patterns A, B, C, and D to be 390, 405, 415, and 410 feet, respectively, with a common standard deviation that could be as high as 20 feet or as small as 10 feet. Assume sets of tires are put on the car (a single car is used for all tests to reduce variability) in random order.

- (a) Suppose the manufacturer wants to test $H_0 : \mu_B - \mu_A = 0$ versus $H_1 : \mu_B - \mu_A > 0$ using $\alpha = 0.05$, assuming $\sigma = 10$. Determine the power of the test if six sets of tires with each tread are available.
- (b) Determine the probability that differences among the means will be detected using $\alpha = 0.05$ assuming $\sigma = 20$ feet if six sets of tires with each tread are available. Simulate the non-central F distribution and compute the power by simulation. How does the simulation compare to the theoretical answer?
- (c) Determine the probability that differences among the means will be detected using $\alpha = 0.05$ if six sets of tires with each tread are available and assuming $\sigma = 10$ feet.
- (d) Assuming the stopping distance standard deviation for all tire sets is $\sigma = 20$ feet, what is the minimum number of tire sets that need to be used to ensure the probability of detecting tire tread differences is at least 80%?
- (e) Given 6 sets of tires with tread A, 6 sets of tires with tread B, 12 sets of tires with tread C, and 12 sets of tires with tread D, what is the probability of detecting tire tread differences if the true stopping standard deviation for all tire tread sets is $\sigma = 14$ feet?

Solution: The answers are as follows:

- (a) To find the theoretical power for detecting differences between the two means, first find the non-centrality parameter λ :

$$\lambda = \frac{SS_{\text{Hypothesis}}}{\sigma^2} = \frac{\sum_{i=1}^a n_i (\mu_{i\bullet} - \bar{\mu}_{\bullet\bullet})^2}{\sigma^2} = \frac{6 \cdot (405 - 397.5)^2 + 6 \cdot (390 - 397.5)^2}{10^2} = 6.75.$$

$\text{Power}(\gamma = \sqrt{\lambda} = 2.5981) = \mathbb{P}(t_{10; \gamma=2.5981}^* > t_{0.95; 10} = 1.8125) = 0.7799$. The values are verified with R Code 11.3.

R Code 11.3

```
> HypMeans <- c(405, 390)
> a <- length(HypMeans) # Number of groups
> n <- 6 # Number in each group
> N <- a*n # Total number of expt. units
> df.error <- N - a # DOF for error
> Sigma <- 10
> alpha <- 0.05
> Y <- rep(HypMeans, each = n) # Responses
> Treat <- factor(rep(LETTERS[c(2, 1)], each = 6)) # Treatment factor
> SStreat <- summary(aov(Y ~ Treat))[[1]][1, 2] # SS for treatment
> lambda <- SStreat/Sigma^2
> lambda
[1] 6.75

> Gamma <- sqrt(lambda)
> Gamma
[1] 2.598076

> CritT <- qt(1 - alpha, df.error)
> Power <- pt(CritT, df.error, ncp = Gamma, lower = FALSE)
> Power
[1] 0.7798662
```

The function `power.t.test()` also returns the answer $\text{Power}(\gamma = \sqrt{\lambda} = 2.5981) = \mathbb{P}(t_{10; \gamma=2.5981}^* > t_{0.95; 10} = 1.8125) = 0.7799$.

```
> power.t.test(n = 6, delta = 15, sd = 10, alternative = "one.sided")
```

Two-sample t test power calculation

```
n = 6
delta = 15
sd = 10
sig.level = 0.05
power = 0.7798662
alternative = one.sided
```

NOTE: n is number in *each* group

Note that the alternative hypothesis is directional and the F distribution cannot be used to answer the question. The answer is obtained with a non-central t -distribution. A graphical representation of the power is given in Figure 11.4 on the following page.

- (b) To find the theoretical power for detecting differences among the means, first find the non-centrality parameter λ :

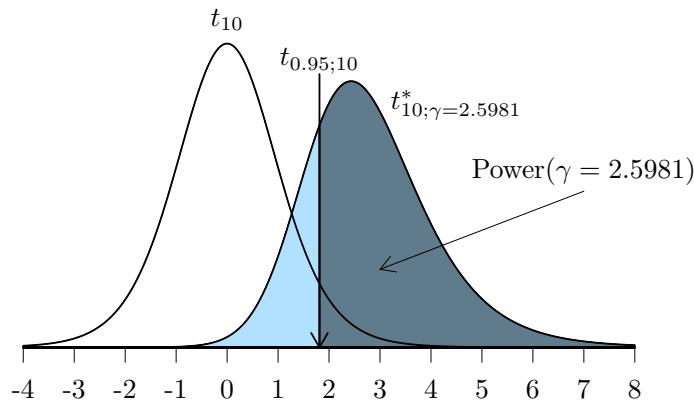


FIGURE 11.4: Power for the directional alternative hypothesis $H_1 : \mu_B - \mu_A > 0$ when $\gamma = 2.5981$ at the $\alpha = 0.05$ level

$$\begin{aligned}\lambda &= \frac{SS_{\text{Hypothesis}}}{\sigma^2} = \frac{\sum_{i=1}^a n_i (\mu_{i\bullet} - \bar{\mu}_{\bullet\bullet})^2}{\sigma^2} \\ &= \frac{6 \cdot (390 - 405)^2 + 6 \cdot (405 - 405)^2 + 6 \cdot (415 - 405)^2 + 6 \cdot (410 - 405)^2}{20^2} = \frac{2100}{20^2} = 5.25.\end{aligned}$$

The probability of detecting differences among the means given $\lambda = 5.25$ is $\text{Power}(\lambda = 5.25) = \mathbb{P}(F_{3, 20; \lambda=5.25}^* > f_{0.95; 3, 20} = 3.0984) = 0.3862$. R Code 11.4 shows the code to compute the power.

R Code 11.4

```
> alpha <- 0.05
> n <- 6
> HypMeans <- c(390, 405, 415, 410)      # Hypothesized means
> a <- length(HypMeans)                      # Number of groups
> N <- a*n                                     # Total number of expt. units
> df.error <- N - a                           # DOF error
> Sigma <- 20
> Y <- rep(HypMeans, each = n)                 # Responses
> Treat <- factor(rep(LETTERS[1:4], each = 6)) # Treatment factor
> SStreat <- summary(aov(Y ~ Treat))[[1]][1, 2] # SS treatment
> lambda <- SStreat/Sigma^2
> lambda
[1] 5.25

> CritF <- qf(1 - alpha, a - 1, N - a)
> CritF
[1] 3.098391

> TheoPower <- pf(CritF, a - 1, N - a, lambda, lower = FALSE)
> TheoPower
[1] 0.3862415
```

A graphical representation of the central and non-central F distributions along with a shaded region for the power at $\lambda = 5.25$ is shown in Figure 11.5. R Code 11.5 is used

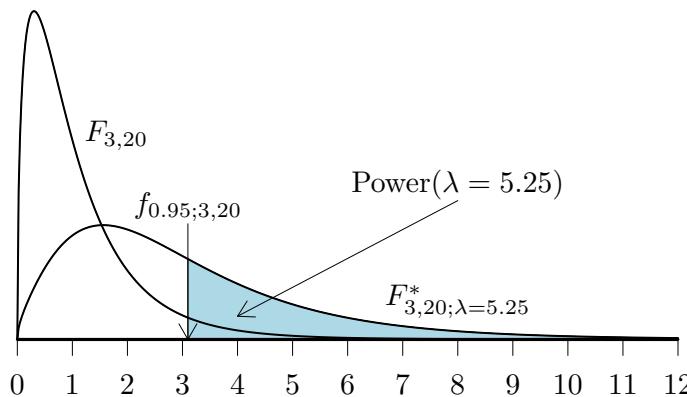


FIGURE 11.5: Power for detecting treatment differences when $\lambda = 5.25$ at the $\alpha = 0.05$ level

to simulate a non-central F distribution and to compute a simulated power of detecting differences among the means.

R Code 11.5

```
> set.seed(10)
> a <- 4          # Number of groups
> n <- 6          # Number in each group
> alpha <- 0.05    # Alpha level
> N <- a*n        # Total number of expt. units
> CritF <- qf(1 - alpha, a - 1, N - a)  # Critical F value
> mu1 <- 390; mu2 <- 405; mu3 <- 415; mu4 <- 410  # True means
> sigma <- 20      # Assumed sigma
> SIMS <- 10^4     # Number of simulations
> FS <- numeric(SIMS)    # Storage for FS
> for(i in 1:SIMS){
+   y1 <- rnorm(n, mu1, sigma)  # Values from mu1, sigma
+   y2 <- rnorm(n, mu2, sigma)  # Values from mu2, sigma
+   y3 <- rnorm(n, mu3, sigma)  # Values from mu3, sigma
+   y4 <- rnorm(n, mu4, sigma)  # Values from mu4, sigma
+   Y <- c(y1, y2, y3, y4)    # Combined responses
+   treat <- factor(rep(LETTERS[1:4], each = n))  # Treatment factor
+   FS[i] <- summary(aov(Y ~ treat))[[1]][1, 4]  # F values
+ }
> SimPower <- mean(FS > CritF)    # Simulated power
> SimPower

[1] 0.3834
```

R Code 11.6 on the following page can be used to create Figure 11.6 on the next page which

shows a central $F_{3,20}$ distribution and the simulated non-central $F_{3,20;\lambda=5.25}^*$ distribution created from R Code 11.5.

R Code 11.6

```
> DF <- data.frame(x = FS)
> x.dens <- density(FS)
> df.dens <- data.frame(x = x.dens$x, y = x.dens$y)
> p <- ggplot(data = DF)
> p + geom_density(aes(x=x, y=..density..), fill="skyblue1", alpha=0.2) +
+   stat_function(fun = df, args = list(3, 20), n = 500) +
+   geom_area(data = subset(df.dens, x >= CritF & x <= 15),
+             aes(x = x, y = y), fill = "skyblue4", alpha = 0.6) +
+   labs(x = "", y = "") + theme_bw() + coord_cartesian(xlim = c(0, 12))
```

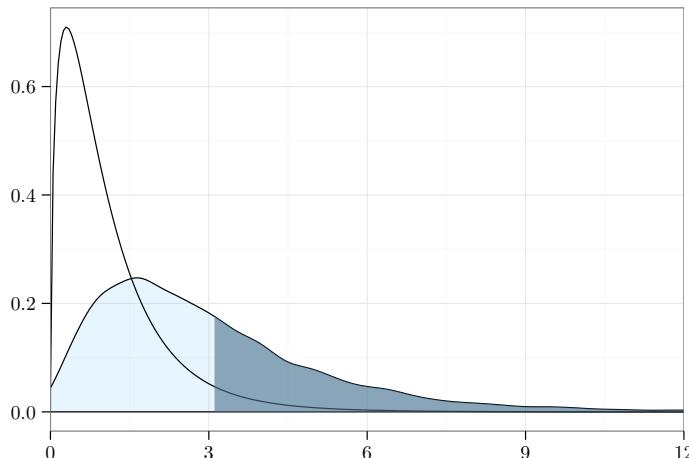


FIGURE 11.6: Histogram of simulated $F_{3,20;\lambda=5.25}^*$ superimposed by a central $F_{3,20}$ distribution

R Code 11.7 indicates there is less than a 1% difference between the simulated power computed in R Code 11.5 on the previous page and the actual power. The code also computes the theoretical quantiles (0.01, 0.05, 0.10, 0.20, 0.80, 0.90, 0.95, 0.99) for an $F_{3,20;\lambda=5.25}^*$ distribution, labeled TQ, as well as the quantiles for the simulated $F_{3,20;\lambda=5.25}^*$ from R Code 11.5 on the preceding page and stores the results in SQ. The percent differences are stored in the variable PD, and the reader can verify that all but one of the simulated quantiles are within 1.8% of their theoretical values. The shape of the simulated $F_{3,20;\lambda=5.25}^*$ distribution shown in Figure 11.6 closely resembles the shape of the theoretical $F_{3,20;\lambda=5.25}^*$ distribution shown in Figure 11.5 on the preceding page.

R Code 11.7

```
> TheoPower
[1] 0.3862415
> SimPower
```

```
[1] 0.3834

> PerDiff <- (abs(TheoPower - SimPower)/TheoPower)*100
> PerDiff      # less than 1% different

[1] 0.7356877

> values <- c(0.01, 0.05, 0.10, 0.20, 0.80, 0.90, 0.95, 0.99)
> TQ <- qf(values, 3, 20, lambda)      # theoretical quantile
> SQ <- quantile(FS, probs = values)    # simulated quantile
> PD <- (abs(TQ - SQ)/TQ)*100        # percent difference
> round(rbind(TQ, SQ, PD), 5)

      1%      5%     10%    20%    80%    90%    95%    99%
TQ 0.19332 0.51909 0.79431 1.23986 4.50804 5.95773 7.43227 11.09788
SQ 0.17686 0.51224 0.78759 1.22996 4.46730 5.91439 7.29868 11.07146
PD 8.51606 1.31967 0.84562 0.79845 0.90354 0.72740 1.79747 0.23802
```

- (c) To find the theoretical power for detecting differences among the means, first find the non-centrality parameter λ :

$$\begin{aligned}\lambda &= \frac{SS_{Hypothesis}}{\sigma^2} \\ &= \frac{\sum_{i=1}^a n_i(\mu_{i\bullet} - \bar{\mu}_{\bullet\bullet})^2}{\sigma^2} \\ &= \frac{6 \cdot (390 - 405)^2 + 6 \cdot (405 - 405)^2 + 6 \cdot (415 - 405)^2 + 6 \cdot (410 - 405)^2}{10^2} \\ &= \frac{2100}{10^2} = 21.\end{aligned}$$

The probability of detecting differences among the means at the $\alpha = 0.05$ level with a given $\lambda = 21$ is $\text{Power}(\lambda = 21) = \mathbb{P}(F_{3, 20; \lambda=21}^* > f_{0.95; 3, 20} = 3.0984) = 0.9502$. R Code 11.8 shows the commands to compute the power.

R Code 11.8

```
> alpha <- 0.05
> n <- 6                      # Number per group
> HypMeans <- c(390, 405, 415, 410)  # Hypothesized means
> a <- length(HypMeans)         # Number of groups
> N <- a*n                     # Total number of expt. units
> df.error <- N - a            # DOF error
> Sigma <- 10                  # Assumed sigma
> Y <- rep(HypMeans, each = n)   # Responses
> Treat <- factor(rep(LETTERS[1:4], each = 6))  # Treatment factor
> SStreat <- summary(aov(Y ~ Treat))[[1]][1, 2] # SS treatment
> lambda <- SStreat/Sigma^2       # Non-centrality parameter
> lambda

[1] 21

> CritF <- qf(1 - alpha, a - 1, N - a) # Critical F value
> CritF
```

```
[1] 3.098391

> TheoPower <- pf(CritF, a - 1, N - a, lambda, lower = FALSE)
> TheoPower

[1] 0.9501649
```

Since sample sizes are equal in the a groups, the R function `power.anova.test()` can be used to solve the problem as well.

```
> power.anova.test(groups = a, n = n, between.var = var(HypMeans),
+   within.var = 10^2)
```

Balanced one-way analysis of variance power calculation

```
groups = 4
n = 6
between.var = 116.6667
within.var = 100
sig.level = 0.05
power = 0.9501649
```

NOTE: n is number in each group

(d) Since λ is a function of sample size, one solution is to find n such that $\mathbb{P}(F_{a-1, a \cdot n - a, \lambda}^* > f_{0.95, a-1, a \cdot n - a}) \geq 0.80$. R Code 11.9 uses a loop to find the value of n such that the power is at least 80%. Power is maximized with a total of N sets of tires when each of the a treatments receives n sets of tires such that $N = a \cdot n$. That is, power is maximized with equal treatment sizes.

R Code 11.9

```
> Sigma <- 20                                # Assumed sigma
> Power <- 0                                  # Initialize Power to 0
> npg <- 1                                    # Initial number per group
> HypMeans <- c(390, 405, 415, 410)          # Hypothesized means
> a <- length(HypMeans)                        # Number of groups
> while(Power < 0.80){                         # Loop until Power is at least 0.80
+   npg <- npg + 1                              # Increment npg by one
+   N <- a*npg                                   # Total number of expt. units
+   alpha <- 0.05                                 # Alpha level
+   Y <- rep(HypMeans, each = npg)               # Responses
+   treat <- factor(rep(LETTERS[1:a], each = npg)) # Treatment factor
+   SStreat <- summary(aov(Y ~ treat))[[1]][1, 2] # SS treatment
+   lambda <- SStreat/Sigma^2                   # Non-centrality parameter
+   CritF <- qf(1 - alpha, a - 1, N - a)        # Critical F value
+   Power <- pf(CritF, a - 1, N - a, ncp = lambda, lower = FALSE)
+ }
> c(npg, lambda, Power)
```

```
[1] 14.0000000 12.2500000 0.8176811
```

From the output, note that when $n = 14$, $\lambda = 12.25$, which returns a power of 81.7681%. Since the problem permits equal n per treatment group, the R function `power.anova.test()` may also be used to solve the problem.

R Code 11.10

```
> power.anova.test(groups = a, between.var = var(HypMeans),
+   within.var = 20^2, power = 0.8)
```

Balanced one-way analysis of variance power calculation

```
groups = 4
n = 13.47806
between.var = 116.6667
within.var = 400
sig.level = 0.05
power = 0.8
```

NOTE: n is number in each group

```
> npg <- ceiling(power.anova.test(groups = a, between.var = var(HypMeans),
+   within.var = 20^2, power = 0.8)$n)
> npg
[1] 14
```

(e) To find the theoretical power for detecting differences among the means, first find the non-centrality parameter λ :

$$\begin{aligned}\lambda &= \frac{SS_{Hypothesis}}{\sigma^2} \\ &= \frac{\sum_{i=1}^a n_i (\mu_{i\bullet} - \bar{\mu}_{\bullet\bullet})^2}{\sigma^2} \\ &= \frac{6 \cdot (390 - 405)^2 + 6 \cdot (405 - 405)^2 + 12 \cdot (415 - 405)^2 + 12 \cdot (410 - 405)^2}{14^2} \\ &= \frac{2625}{14^2} = 13.3929.\end{aligned}$$

The probability of detecting differences among the means at the $\alpha = 0.05$ level with a given $\lambda = 13.3929$ is $\text{Power}(\lambda = 13.3929) = \mathbb{P}(F_{3, 32; \lambda=13.3929}^* > f_{0.95; 3, 32} = 2.9011) = 0.8349$. R Code 11.11 shows the commands to compute the power. Figure 11.7 on the following page shows a graphical representation of the central and non-central F distributions with the area representing $\text{Power}(\lambda = 13.3929)$ lightly shaded.

R Code 11.11

```
> alpha <- 0.05
> n1 <- 6; n2 <- 6; n3 <- 12; n4 <- 12 # Numbers per group
> HypMeans <- c(390, 405, 415, 410) # Hypothesized means
> a <- length(HypMeans) # Number of groups
> N <- n1 + n2 + n3 + n4 # Total number of expt. units
> df.error <- N - a # DOF error
```

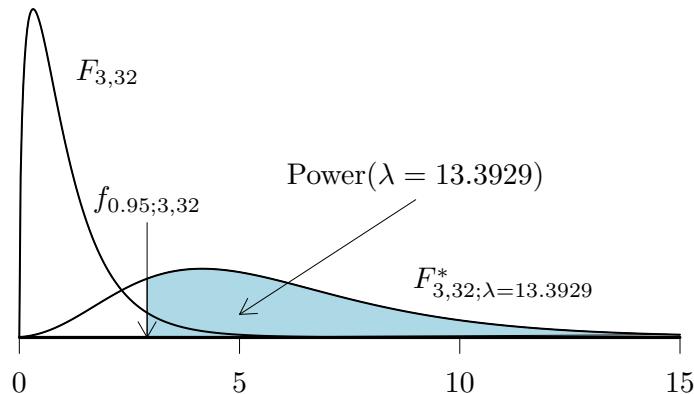
```

> Sigma <- 14                                # Assumed sigma
> Y <- rep(HypMeans, times = c(n1, n2, n3, n4)) # Responses
> Treat <- factor(rep(LETTERS[1:4], times = c(n1, n2, n3, n4)))
> SStreat <- summary(aov(Y ~ Treat))[[1]][1, 2] # SS treatment
> lambda <- SStreat/Sigma^2                  # Non-centrality parameter
> lambda
[1] 13.39286

> CritF <- qf(1 - alpha, a - 1, N - a)    # Critical F value
> CritF
[1] 2.90112

> TheoPower <- pf(CritF, a - 1, N - a, lambda, lower = FALSE)
> TheoPower
[1] 0.8349338

```

FIGURE 11.7: Central and non-central F distributions

11.5 Checking Assumptions

The values in the ANOVA table and the subsequent inferences made from those values are based on the assumption that the data follow the model

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij} \quad (11.14)$$

where the τ_i s are fixed but unknown numbers and the ε_{ij} s are independent normals with a mean of zero and constant variance. Consequently, the three basic assumptions concerning the errors:

- 1) independence,
- 2) normal distribution, and
- 3) constant variance

should be investigated. Since the actual errors are unknown quantities, they will never be observed; however, it is reasonable to use estimates (or predictors) of the errors, the residuals, to assess the three basic assumptions concerning errors. Recall from Chapter 2 that a residual is the difference between what is observed and what is predicted ($\hat{\varepsilon}_{ij} = Y_{ij} - \hat{Y}_{ij}$). For Model (11.14), $\hat{Y}_{ij} = \bar{Y}_{\bullet\bullet} + \bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet} = \bar{Y}_{i\bullet}$. While (11.14) may be a reasonable approximation to some real-life phenomena, real-life data are never exactly normal. The real question is whether the assumptions have been violated to such an extent that the inferences based on the particular model in question would be invalidated. Although a few formal tests are presented, most of the material that follows deals with visual diagnostics for the three basic assumptions concerning errors.

11.5.1 Checking for Independence of Errors

The most important assumption for (11.14) to be valid and the most challenging assumption to correct if it fails is the assumption of independence. The material in this text will not address how to deal with dependent data, which is the topic of a more advanced course. One of the easier dependencies to detect is a dependence in time. When values are either very similar (positive dependence) or very different (negative dependence) to each other in time, the assumption of independence becomes untenable. An easy way visually to inspect data for dependence is to plot the residuals on the vertical axis versus a time sequence on the horizontal axis. Naturally, if there is no time component to the data, this graph will not reveal any useful information not found in other residual plots.

It is often helpful to standardize the residuals so they have unit variance. Many books define standardized residuals as

$$r_{ij} = \frac{\hat{\varepsilon}_{ij}}{\sqrt{MS_{\text{Error}}}};$$

however, the standard deviation of the ij^{th} residual is actually $\sigma \cdot \sqrt{1 - h_{ii}}$, where the h_{ii} s are the diagonal elements of the hat matrix (discussed in more detail in Chapter 12: Regression). For Model (11.14), the h_{ii} values are simply $1/n_i$. By estimating σ with the $\sqrt{MS_{\text{Error}}}$, the standardized residuals (r_{ij}) are computed as

$$r_{ij} = \frac{\hat{\varepsilon}_{ij}}{\sqrt{\widehat{Var}(\hat{\varepsilon}_{ij})}} = \frac{\hat{\varepsilon}_{ij}}{\sqrt{MS_{\text{Error}} \cdot \sqrt{1 - h_{ii}}}}. \quad (11.15)$$

The function `rstandard()` computes standardized residuals according to (11.15).

Modifications to R Code 11.12, which uses the `TIRE` data set from the motivational problem (Example 11.1) at the beginning of the chapter, can be used to help the user assess the assumption of independence among the errors. Based on Figure 11.8 on the following page, no discernible pattern is seen that might threaten the assumption of independent errors.

R Code 11.12

```
> mod.aov <- aov(stopdist ~ tire, data = TIRE)
> r <- rstandard(mod.aov)
> DF <- data.frame(TIRE, r = r)
```

```
> head(DF)
> p <- ggplot(data = DF, aes(x = order, y = r)) +
+   geom_point() +
+   geom_line() +
+   geom_hline(yintercept = 0, linetype = "dashed") +
+   labs(x = "Ordered Values", y = "Standardized Residuals") +
+   theme_bw()
> p
```

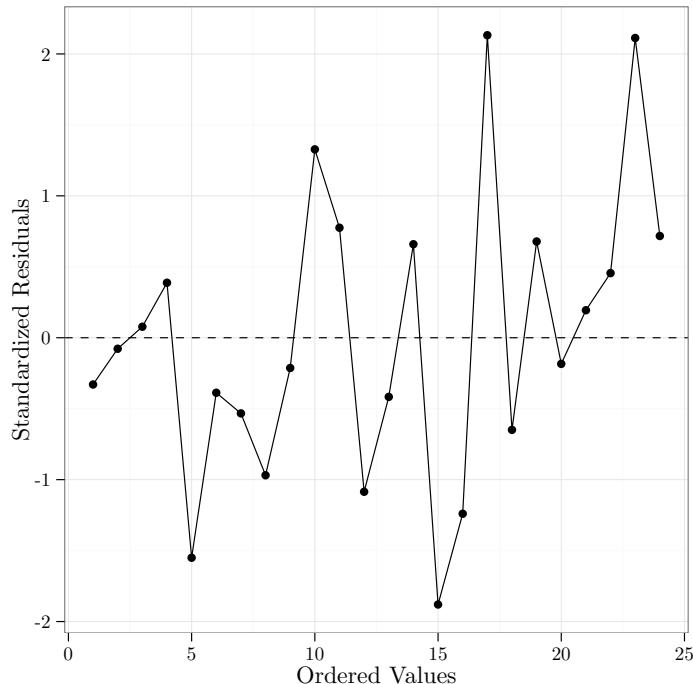


FIGURE 11.8: Standardized residuals versus order for `mod.aov` using the **TIRE** data set

11.5.2 Checking for Normality of Errors

The quantile-quantile plot is a graphical procedure for assessing normality. The quantile-quantile plot can be performed on either the residuals or the standardized residuals. If standardized residuals are used, the plotted observations should follow a straight line with an intercept of zero and a slope of one. Reading quantile-quantile plots, especially when the total number of residuals ($N = \sum_{i=1}^a n_i$) is small, requires a high degree of skill. A formal test of normality can be obtained with the function `shapiro.test()`. Modifications to R Code 11.13 on the next page, which uses the **TIRE** data set from the motivational problem (Example 11.1), can be used to help the user assess the normality of errors assumption. Figure 11.9 on the facing page shows a quantile-quantile plot of the standardized residuals with a superimposed line with a zero intercept and a slope of one indicating the assumption of normal errors is reasonable. The p -value (0.7584) from the Shapiro-Wilk normality test shown in R Code 11.14 on the next page provides further corroboration that the normality

assumption of the errors is reasonable.

R Code 11.13

```
> p <- ggplot(data = DF, aes(sample = r)) +  
+   stat_qq() +  
+   geom_abline(intercept = 0, slope = 1, linetype = "dotted")  
> p
```

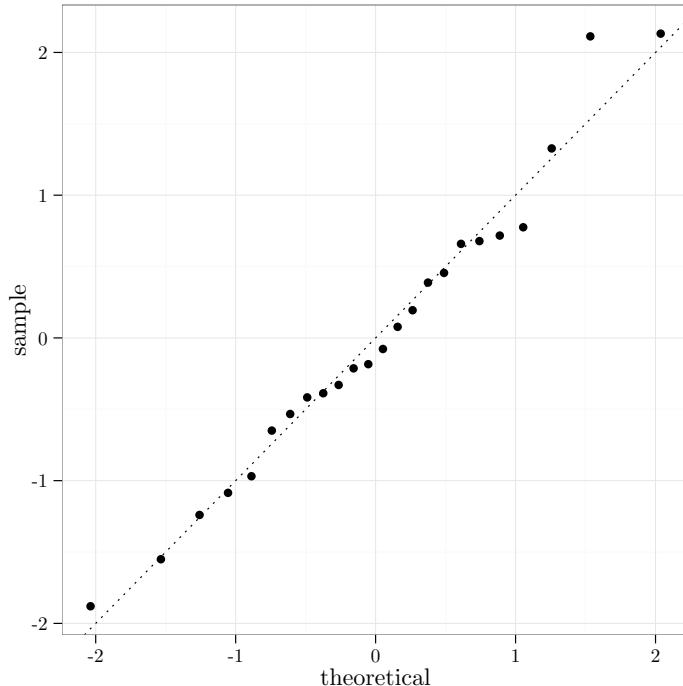


FIGURE 11.9: Quantile-quantile plot of the standardized residuals with a superimposed line with a zero intercept and a slope of one for the model `mod.aov` using the **TIRE** data frame

R Code 11.14

```
> shapiro.test(r)
```



```
Shapiro-Wilk normality test
```



```
data: r  
W = 0.97372, p-value = 0.7584
```

11.5.3 Checking for Constant Variance

Many formal tests for equality of variance exist. Most of these tests are very sensitive to normality assumptions and will not give reliable results if normality is violated. As with independence and normality of errors assumptions, the constant variance assumption should be checked with graphical procedures. Specifically, to assess constant variance, a plot of the residuals ($\hat{\varepsilon}_{ij}$) or the standardized residuals r_{ij} on the vertical axis should be plotted against the fitted values (\hat{Y}_{ij}) on the horizontal axis. Recall that for (11.14), the fitted values are simply $\bar{Y}_{i\bullet}$. This plot will look like several vertical stripes of points, one for each treatment group. If the variance is constant, the vertical lengths of the stripes for each of the i groups will be similar. Figure 11.10 shows a plot of the standardized residuals versus the fitted values of (11.14), suggesting there are no serious departures from homogeneity of variance.

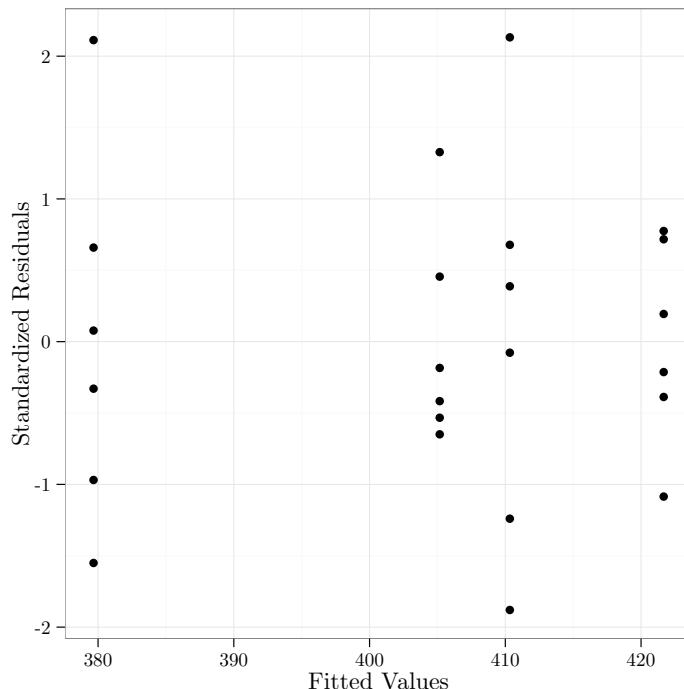


FIGURE 11.10: Plot of the standardized residuals versus the fitted values for `mod.aov` using the **TIRE** data set

If one insists on testing equality of variance, a modified version of Levene's test is recommended. Specifically, compute the quantity $Z_{ij} = |Y_{ij} - \tilde{Y}_{i\bullet}|$, the absolute deviations from the group medians. Treat the Z_{ij} values as the data, and use the standard ANOVA formulas presented in Table 11.2 on page 703 on the Z_{ij} values. A significant finding with the standard F -test on the Z_{ij} values indicates non-constant variance. This particular modification to Levene's test, which uses the absolute deviations from the group medians, is relatively insensitive to non-normality and is easily implemented with R. It is also a preprogrammed function `leveneTest()` in the `car` package. R Code 11.15 on the facing page uses the data frame **TIRE** from Example 11.1 to assess the homogeneity of variance

with respect to the errors assumption by performing a modified Levene's test. The p -value (0.4224) from the modified Levene test provides further corroboration that the homogeneity of variance assumption of the errors is not unreasonable.

R Code 11.15

```
> med <- tapply(TIRE$stopdist, TIRE$tire, median)
> Zij <- abs(TIRE$stopdist - med[TIRE$tire])
> TIREA <- data.frame(TIRE, Zij)
> summary(aov(Zij ~ tire, data = TIREA))

      Df Sum Sq Mean Sq F value Pr(>F)
tire      3   388.8   129.6   0.979  0.422
Residuals 20  2647.8   132.4

> # Or using the leveneTest()
> car::leveneTest(mod.aov)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  0.9789 0.4224
20
```

The function `checking.plots()` from the `PASWR2` package creates the three graphs discussed in Sections 11.5.1, 11.5.2, and 11.5.3 that assess independence, normality, and constant variance, respectively, plus a density plot of the standardized residuals. The function assumes the values used to construct the `aov` object are provided in the order the experiment was conducted. If this is not the case, the user must first reorder the values used to construct the `aov` object or the graph will not properly show the ordered residuals. If the order in which the values were collected is not known from the data, the user should ignore the ordered residuals plot. R Code 11.16 reorders the data frame `TIRE` according to the order the experiment was conducted and stores the results in `TIREO`.

R Code 11.16

```
> TIREO <- TIRE[order(TIRE$order),] # reordering the DF
> head(TIREO)

  stopdist tire order
2       374    A     1
21      409    D     2
6       381    A     3
23      417    D     4
5       353    A     5
14      415    C     6

> mod.aov0 <- aov(stopdist ~ tire, data = TIREO)
```

The graphs from using `checking.plots()` with `mod.aov0` are shown in Figure 11.11 on the next page. By default, the function `checking.plots()` labels the three largest (in absolute value) standardized residuals using the argument `n.id = 3`.

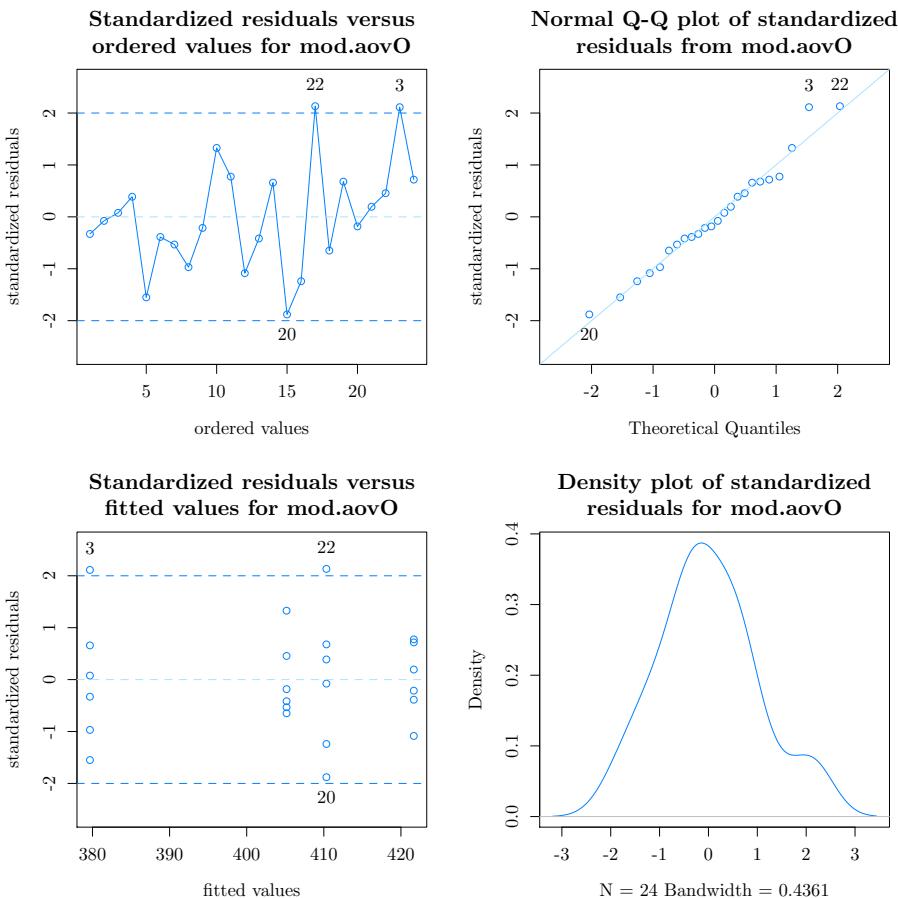


FIGURE 11.11: Graphs to assess independence, normality, and constant variance created with `checking.plots(mod.aov0)` using the data frame `TIREO`

11.6 Fixing Problems

When diagnostics indicate that the assumptions for a particular model are not satisfied, either the data must be modified or the method of analysis must be changed to be less sensitive to the assumptions. The three assumptions for error terms of (11.14) are that they are 1) independent, 2) have a normal distribution, and 3) have homogeneity of variance. Working with dependent errors is quite challenging and will not be discussed other than to say that proper randomization should always be used in the collection of data to reduce the possibility of dependence among errors. In the event an analysis indicates dependent errors, the original design should be re-evaluated. The normal errors assumption can often be violated without affecting the estimation and inferences associated with the chosen model provided the errors' departures from normality are not severe. Non-constant variance in contrast to the normality assumption will impact estimation and inferences associated with the chosen model and needs to be evaluated closely. The balance of fixing problems will center on how to deal with 1) non-normal errors and 2) non-constant variance.

11.6.1 Non-Normality

When a quantile-quantile plot of the residuals indicates skewness (typically to the right) a transformation on the response variable will often alleviate the problem of non-normal errors. Finding a meaningful and appropriate transformation is often challenging. One technique that searches computationally for an appropriate transformation of the response variable that directly addresses normality is the Box-Cox method. The Box-Cox method estimates the parameter λ for the transformation $Y' = Y^\lambda$, where

$$Y' = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & \text{for } \lambda \neq 0, \\ \ln Y & \text{for } \lambda = 0, \end{cases} \quad (11.16)$$

by the method of maximum likelihood. Figure 11.12 shows transformations in common use.

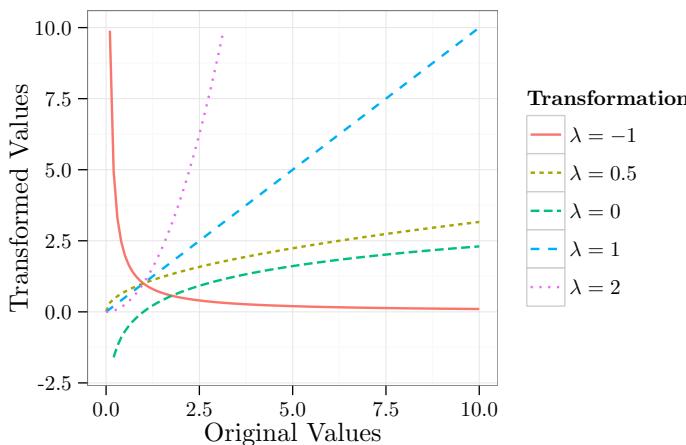


FIGURE 11.12: Transformations in common use with the Box-Cox method

The function `boxCox()` of the `car` package produces a plot of the log-likelihood against the transformation parameter λ for a particular model. By default, the range of λ is from -2 to 2 ; however, once the value of λ that maximizes the log-likelihood is known, the range of the plot in `boxCox()` can be tightened to highlight the area where the function is maximized with the argument `lambda=`. For more details, see the `boxCox()` help file. The `boxCox()` function is generally used to approximate an appropriate transformation. The value of λ that maximizes the log-likelihood function may turn out to be 0.53 ; but if there is a possible explanation for taking the square root of the response, the transformation applied should be $\lambda = 0.5$ and not the value that maximizes the log-likelihood function.

Observations that do not fit the pattern of the rest of the data in the quantile-quantile plot (outliers) can distort an analysis, and one should consider removing the outlier(s) and performing the analysis without the offending point(s). Oftentimes, outliers are simply poorly transcribed experimental results such as an incorrectly placed decimal or a misplaced label; these can be legitimately removed. On the other hand, just because a value is an outlier does not mean it should always be eliminated from the data; rather, outliers may imply that the model being used is incorrect. Does this mean that if the values in a quantile-quantile plot are not exactly linear, then there are problems? Fortunately not! With equal treatment sizes, the F -test used with ANOVA is quite robust to non-normal

errors when the homogeneity of variance assumption is satisfied. The reader should perform their own simulations to verify that sampling distribution for $MS_{\text{Treatment}}/MS_{\text{Error}}$ when sampling from non-normal distributions is quite close to the F distribution. Unfortunately, subsequent inference on individual parameters using one-sided confidence intervals is sensitive to the normality assumption and can result in poor conclusions when the errors do not follow a normal distribution. In general, the F -test will detect differences for most distributions, but violations of normality invalidate post-hoc conclusions.

11.6.2 Non-Constant Variance

Non-constant variance is typically fixed by transforming the response variable. The Box-Cox method discussed to fix the problem of non-normal errors will oftentimes alleviate both the problem of unequal variances as well as non-normal errors. The implications for the F -test when the variances among the a groups are different depends to a large extent on whether the groups have equal sample sizes. When the a groups have equal sample sizes, unequal variance only slightly alters the φ -value for an F -test. The situation is very different, however, when sample sizes among the a groups are unequal. When larger variances are associated with the larger sample sizes, the F -test will be conservative, and when the larger variances are associated with smaller sample sizes, the F -test is liberal. Welch (1951) derived a method of testing several means that does not require the assumption of equal variance and is implemented in R using the function `oneway.test()`. Welch's statistic (W) for testing several means is defined as

$$W = \frac{\sum_{i=1}^a w_i (\bar{Y}_i - \tilde{Y})^2 / (a-1)}{\left[1 + \frac{2}{3}(a-2)\Lambda\right]} \sim F_{a-1; 1/\Lambda} \quad (11.17)$$

where

$$w_i = \frac{n_i}{s_i^2}, \quad \tilde{Y} = \frac{\sum_{i=1}^a w_i \bar{Y}_i}{\sum_{i=1}^a w_i}, \quad \text{and} \quad \Lambda = \frac{3 \sum_{i=1}^a \left\{ \left[1 - \left(\frac{w_i}{\sum_{i=1}^a w_i} \right) \right]^2 / (n_i - 1) \right\}}{a^2 - 1}.$$

Example 11.3 ▷ Fat Cats ◁ In a weight loss study on obese cats, overweight cats were randomly assigned to one of three groups and boarded in a kennel. In each of the three groups, the cats' total caloric intake was strictly controlled (1 cup of generic cat food) and monitored for 10 days. The difference between the groups was that group A was given $\frac{1}{4}$ of a cup of cat food every 6 hours, group B was given $\frac{1}{3}$ a cup of cat food every 8 hours, and group C was given $\frac{1}{2}$ a cup of cat food every 12 hours. The weights of the cats at the beginning and end of the study were recorded and the differences in weights (grams) are stored in the variable `weight` of the data frame `FCD`. Are there mean weight differences among the three treatments?

Solution: The hypothesis of interest is $H_0 : \tau_i = 0$ for all i versus $H_1 : \tau_i \neq 0$ for some i given the model $Y_{ij} = \mu + \tau_i + \varepsilon_{ij}$, where $\varepsilon_{ij} \sim N(0, \sigma)$. To see if the assumption of NID errors is reasonable, the function `checking.plots()` is applied to the model `FCD.aov <- aov(weight~diet)`, and the graphical output is displayed in Figure 11.13 on the next page. Since there is no time/order component to the data, the ordered residuals plot should be ignored. The assumption of equal variance seems tenuous. Increasing variance as the mean increases is seen in the standardized residuals versus fitted graph (bottom left graph). By using the function `boxCox()` from the `car` package applied to `FCD.aov`, a log transformation is suggested (type `car::boxCox(FCD.aov)` and view the result); however, the log transformation does not fix the unequal variance assumption (see

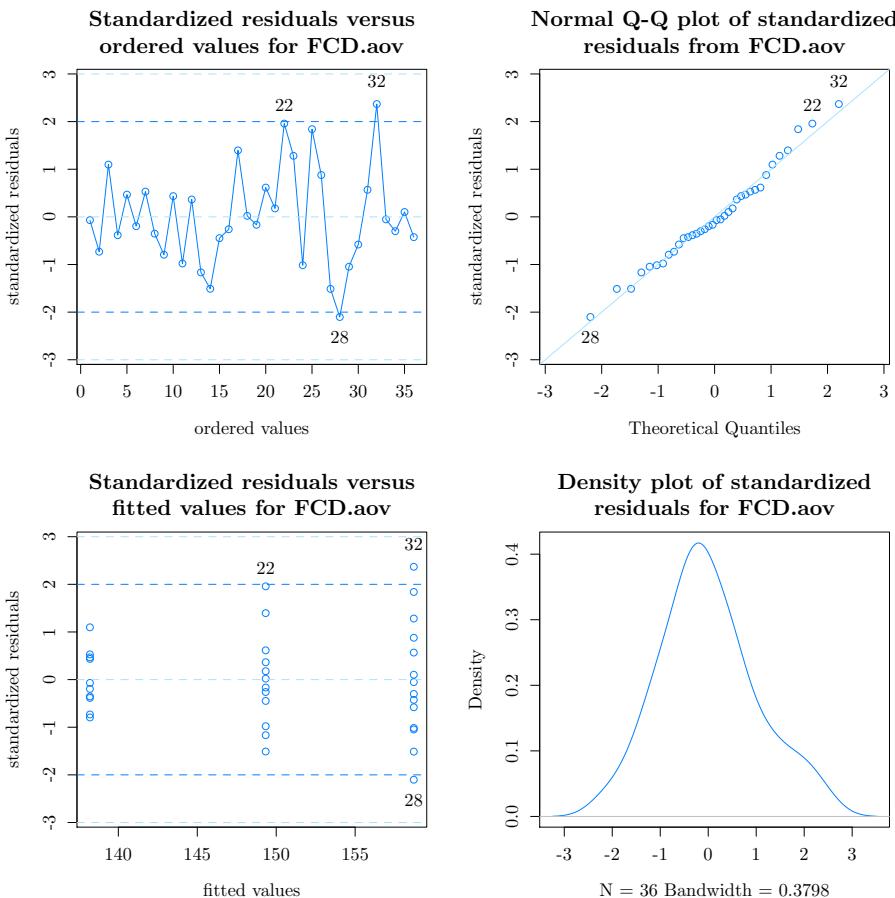


FIGURE 11.13: `checking.plots()` applied to the model `FCD.aov` (`aov(weight ~ diet)`) with the **FCD** data frame

Figure 11.14 on the following page). It is interesting to point out in this particular case that, despite the logarithmic transformation, variance is still increasing, yet the modified Levene test returns a p -value of 0.2347, indicating no evidence of unequal variance (see R Code 11.17).

R Code 11.17

```
> med <- tapply(FCD$weight, FCD$diet, median)
> Zij <- abs(FCD$weight - med[FCD$diet])
> FCDA <- cbind(FCD, Zij)
> ANOVAa <- summary(aov(Zij ~ diet, data = FCDA))
> ANOVAa

Df Sum Sq Mean Sq F value Pr(>F)
diet      2    1606    802.9    1.939   0.16
Residuals 33  13664    414.1

> medL <- tapply(log(FCD$weight), FCD$diet, median)
> ZijL <- abs(log(FCD$weight) - log(med[FCD$diet]))
> FCDAL <- cbind(FCD, ZijL)
```

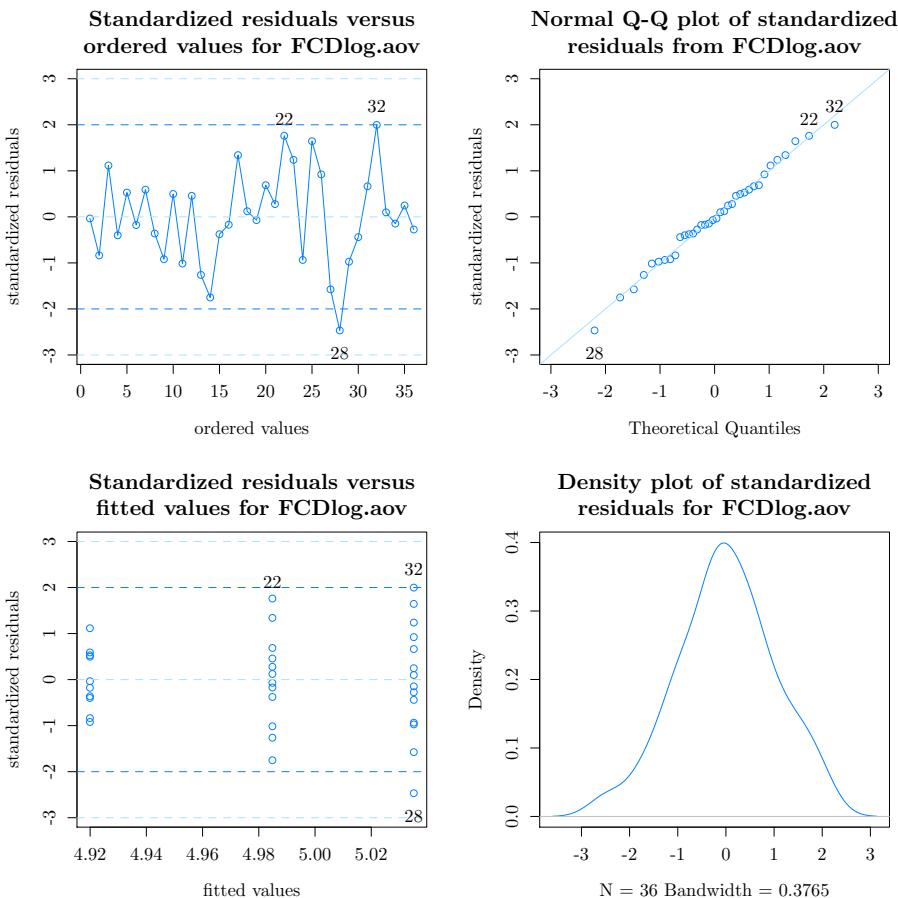


FIGURE 11.14: `checking.plots()` applied to the model `FCDlog.aov` (`aov(log(weight) ~ diet)`) with the `FCD` data frame

```
> ANOVAb <- summary(aov(ZijL ~ diet, data = FCDAL))
> ANOVAb

Df Sum Sq Mean Sq F value Pr(>F)
diet      2 0.0517 0.02584   1.515  0.235
Residuals 33 0.5628 0.01706
```

Since the transformation does not remedy the increasing variance problem and there were no normality problems with the original data, Welch's test for equal means with unequal variance is used on the original measurements. The large p -value of 0.2562 from Welch's test indicates there is no reason to believe the three methods of feeding obese cats result in different weight losses. R Code 11.18 computes the Welch test statistic and its p -value according to (11.17) as well as using the R function `oneway.test()`.

R Code 11.18

```
> ni <- with(data = FCD, tapply(weight, diet, length)) # Number per group
> ni

A   B   C
```

```

10 12 14

> a <- length(ni)
> si2 <- with(data = FCD, tapply(weight, diet, var)) # Variance per group
> si2

      A          B          C
377.0667 1044.7879 1691.1429

> ybar <- with(data = FCD, tapply(weight, diet, mean)) # Mean per group
> ybar

      A          B          C
138.2000 149.3333 158.7143

> wi <- ni/si2
> ytild <- sum(wi*ybar)/sum(wi)
> ytild

[1] 144.6319

> wlamb <- 3*sum((1 - (wi/sum(wi)))^2/(ni - 1))/(a^2 - 1)
> wlamb

[1] 0.04631835

> dfn <- (a - 1)           # Degrees of freedom numerator
> dfn

[1] 2

> dfd <- 1/wlamb           # Degrees of freedom denominator
> dfd

[1] 21.58971

> W <- sum(wi*(ybar - ytild)^2/(a - 1))/(1 + 2/3*(a - 2)*wlamb)
> W                         # Computed W value

[1] 1.451544

> pvalue <- pf(W, dfn, dfd, lower = FALSE)
> pvalue

[1] 0.2561727

> # Or
> oneway.test(weight ~ diet, data = FCD)

One-way analysis of means (not assuming equal variances)

data: weight and diet
F = 1.4515, num df = 2.00, denom df = 21.59, p-value = 0.2562

```

11.7 Multiple Comparisons of Means

When the null hypothesis for the completely randomized design, $H_0 : \mu_1 = \mu_2 = \dots = \mu_a$, is rejected with an F -test, the test does not indicate which means are different or how they differ. To do this, several tests are required; however, as noted earlier, repeated application of a test drastically increases type I errors.

Suppose a set of K null hypotheses $H_{01}, H_{02}, \dots, H_{0K}$ are to be tested where the overall hypothesis H_0 is true if all of the H_{0i} 's for $i = 1, 2, \dots, K$ are true:

$$H_0 : H_{01} \cap H_{02} \cap \dots \cap H_{0K} \quad (11.18)$$

Note that H_0 is rejected if any of the H_{0i} 's are rejected. The **comparison-wise error rate** is the probability of rejecting a particular H_{0i} in a single test when H_{0i} is true. Controlling the comparison-wise error rate at the α_c level means that the expected proportion of individual tests that reject H_{0i} when H_{0i} is true is α_c . This is the only error rate considered thus far and has previously been denoted as merely α . It is simply the risk one is willing to take of making a type I error in a single test. In contrast to the comparison-wise error rate, the **experiment-wise error rate** is the probability of rejecting at least one of the H_{0i} 's in a series of tests when all of the H_{0i} 's are true, and is denoted α_e . It is the risk of making at least one type I error among the family of comparisons in (11.18). The experiment-wise error rate, α_e , can be evaluated for a family of *independent* tests. Although a set of tests that might be of interest, such as all pairwise differences of a means, are not independent tests, an upper limit on α_e can be established by assuming the tests are independent. There are a total of $m_a = \binom{a}{2} = a(a-1)/2$ tests needed to evaluate all pairwise differences among a means.

The probability of a type I error for any single test is α_c and the probability of a correct decision is $1 - \alpha_c$. If it is assumed that the m_a tests are independent, then the random variable $X = \text{number of type I errors}$ has a binomial distribution:

$$X \sim \text{Bin}(n = m_a, \pi = \alpha_c).$$

Since α_e is the probability of making at least one type I error in the family of tests (m_a),

$$\alpha_e = \mathbb{P}(X \geq 1) = 1 - \mathbb{P}(X = 0) = 1 - \binom{m_a}{0} \alpha_c^0 (1 - \alpha_c)^{m_a} = 1 - (1 - \alpha_c)^{m_a}$$

Table 11.5: α_e values for given α_c and various numbers of comparisons

		$K = \text{number of independent comparisons}$				
		2	5	10	20	50
α_c	0.01	0.0199	0.0490	0.0956	0.1821	0.3950
	0.05	0.0975	0.2262	0.4013	0.6415	0.9231
	0.10	0.1900	0.4095	0.6513	0.8784	0.9948

Glancing at Table 11.5, one sees very clearly that for fixed α_c , as K increases, α_e tends to 1. In other words, the probability of making at least one type I error in a series of tests

approaches 1 as the number of tests increases. Consequently, multiple comparisons will generally attempt to control α_e , the experiment-wise error rate. To obtain a rough idea of the value of α_e , one can use the Bonferroni inequality $\alpha_e \leq K \cdot \alpha_c$. Likewise, a rough estimate of α_c is α_e/K .

11.7.1 Fisher's Least Significant Difference

Fisher's least significant difference (protected LSD) requires an overall F -test of H_0 . If H_0 is rejected, t -tests are used with a common variance estimator (MS_{Error}) for comparisons of interest. This procedure, despite its appearance, controls neither α_c nor α_e and is not a recommended testing procedure. It is included here for pedagogical reasons only. For pairwise comparisons, group means are considered different if

$$|\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}| > \underbrace{t_{1-\frac{\alpha_c}{2}; df_{\text{Error}}} \cdot \sqrt{MS_{\text{Error}}} \cdot \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}}_{\text{LSD}}. \quad (11.19)$$

The $(1 - \alpha_c) \cdot 100\%$ confidence interval on the difference of means based on the LSD is

$$CI_{1-\alpha_c}(\mu_i - \mu_j) = \left[(\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}) - t_{1-\alpha_c/2; df_{\text{Error}}} \sqrt{MS_{\text{Error}}} \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}, (\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}) + t_{1-\alpha_c/2; df_{\text{Error}}} \sqrt{MS_{\text{Error}}} \sqrt{\frac{1}{n_i} + \frac{1}{n_j}} \right]. \quad (11.20)$$

When the number of comparisons is small ($K \leq 5$), the problem of an increasing α_e for using Fisher's LSD can be addressed with the **Bonferroni** method.

The Bonferroni method divides α_c by the total number (K) of comparisons. Means are considered different if the difference of sample means is greater than Bonferroni's significant difference (BSD):

$$|\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}| > \underbrace{t_{1-\frac{\alpha_c}{2K}; df_{\text{Error}}} \cdot \sqrt{MS_{\text{Error}}} \cdot \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}}_{\text{BSD}}. \quad (11.21)$$

The $(1 - \alpha_e) \cdot 100\%$ confidence interval on the difference of means based on the BSD is

$$CI_{1-\alpha_e}(\mu_i - \mu_j) = \left[(\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}) - t_{1-\frac{\alpha_c}{2K}; df_{\text{Error}}} \sqrt{MS_{\text{Error}}} \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}, (\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}) + t_{1-\frac{\alpha_c}{2K}; df_{\text{Error}}} \sqrt{MS_{\text{Error}}} \sqrt{\frac{1}{n_i} + \frac{1}{n_j}} \right]. \quad (11.22)$$

The experiment-wise error rate using α_c/K can be much less than α_e , thus this method is very conservative and has correspondingly low power.

11.7.2 The Tukey's Honestly Significant Difference

The Tukey's honestly significant difference (HSD) was designed to control α_e . As such, it does a much better job of keeping α_e close to its nominal level than does the Bonferroni procedure. The HSD procedure is based on the studentized range statistic. The studentized range statistic, Q , for a set of treatment means is

$$Q = \frac{\bar{Y}_{\max} - \bar{Y}_{\min}}{\hat{\sigma}/\sqrt{n}}. \quad (11.23)$$

The distribution of Q depends on the number of treatments (a) and the degrees of freedom for $\hat{\sigma}$ (MS_{Error}), denoted by ν . In the one-way CRD, $df_{\text{Error}} = N - a$. The notation $q_{1-\alpha; a, \nu}$ denotes the studentized range value with $1 - \alpha$ area to the left with a and ν degrees of freedom, respectively. The R function `qtukey()` returns values from the studentized range distribution. For example, $q_{0.95; 4, 20} = 3.9583$ is obtained by entering

```
> qtukey(0.95, 4, 20)
```

```
[1] 3.958293
```

The HSD method rejects any pairwise null hypothesis $H_0 : \mu_i = \mu_j$ at the α_e level if

$$\left| \bar{Y}_{i\bullet} - \bar{Y}_{j\bullet} \right| > \underbrace{q_{1-\alpha_e; a, \nu} \cdot \frac{\sqrt{MS_{\text{Error}}}}{\sqrt{n}}}_{HSD}. \quad (11.24)$$

Note that

$$\frac{\left| \bar{Y}_{i\bullet} - \bar{Y}_{j\bullet} \right|}{\sqrt{MS_{\text{Error}}} \sqrt{\frac{1}{n} + \frac{1}{n}}} = |t| > \frac{q_{1-\alpha_e; a, \nu}}{\sqrt{2}},$$

which implies a confidence interval for $\mu_i - \mu_j$ at the $1 - \alpha_e$ level using the studentized range statistic is written as

$$CI_{1-\alpha_e}(\mu_i - \mu_j) = \left[(\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}) - \frac{q_{1-\alpha_e; a, \nu}}{\sqrt{2}} \sqrt{MS_{\text{Error}}} \sqrt{\frac{1}{n} + \frac{1}{n}}, (\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}) + \frac{q_{1-\alpha_e; a, \nu}}{\sqrt{2}} \sqrt{MS_{\text{Error}}} \sqrt{\frac{1}{n} + \frac{1}{n}} \right]. \quad (11.25)$$

Strictly speaking, HSD is only applicable to the equal sample size problem. For unequal sample sizes, HSD can be approximated as

$$\text{HSD} \approx \frac{q_{1-\alpha_e; a, \nu}}{\sqrt{2}} \cdot \sqrt{MS_{\text{Error}}} \cdot \sqrt{\frac{1}{n_i} + \frac{1}{n_j}},$$

which is typically conservative compared to the case of equal n_i and n_j .

Note that the critical values used by the LSD, HSD, and BSD procedures for detecting pairwise differences are

$$t_{1-\frac{\alpha}{2}; \nu} \leq \frac{q_{1-\alpha; a, \nu}}{\sqrt{2}} \leq t_{1-\frac{\alpha}{2K}; \nu}.$$

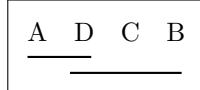
This inequality implies that LSD has the most power for detecting differences, followed by HSD and then BSD. Unfortunately, LSD does not control α_e , while HSD and BSD both do. Consequently, of the three methods used to compare pairwise means, HSD is the one recommended because it controls α_e with equal n and is only slightly conservative when n_i and n_j are unequal.

11.7.3 Displaying Pairwise Comparisons

Pairwise comparisons for K means generate $\binom{K}{2} = K(K - 1)/2$ tests. A compact method for displaying the results is to

1. Sort the K means in increasing order.
2. Place the labels of those sorted means on a horizontal axis.
3. Draw lines under groups that are not significantly different.

If consecutive groups are not significantly different, use a single line segment under all of such groups. Suppose there are four treatments being studied, which are labeled A, B, C, and D. The diagram



indicates that A and D are not distinguishable from each other, nor are D, C, and B distinguishable from each other. Only A can be distinguished from C and B.

11.8 Other Comparisons among the Means

At times, comparisons other than pairwise are of interest. For example, suppose tires with tread A and tread B are made in South Carolina and tires with tread C and tread D are made in Florida. In this scenario, assuming the stopping distance for a car traveling 60 miles per hour was being measured, one may want to know if there are differences due to tire manufacturing location and would want to test

$$H_0 : \frac{\mu_A + \mu_B}{2} = \frac{\mu_C + \mu_D}{2}.$$

Any linear combination of means $C = \sum_{i=1}^a c_i \mu_i$, where $\sum_{i=1}^a c_i = 0$, is called a **contrast**. An estimate of the contrast $C = \sum_{i=1}^a c_i \mu_i$ can be obtained from the observed data and expressed as $\hat{C} = \sum_{i=1}^a c_i \bar{Y}_{i\bullet}$. A contrast of observed means is an unbiased estimate of the corresponding true treatment means:

$$E(\hat{C} = \sum_{i=1}^a c_i \bar{Y}_{i\bullet}) = \sum_{i=1}^a c_i \mu_i. \quad (11.26)$$

Since the treatment means are independent, the variance of the observed contrast is

$$\text{Var}(\hat{C} = \sum_{i=1}^a c_i \bar{Y}_{i\bullet}) = \sigma^2 \sum_{i=1}^a \frac{c_i^2}{n_i}. \quad (11.27)$$

Using the standard form of a t -statistic,

$$\frac{\text{unbiased estimator} - \text{hypothesized value}}{\text{standard error of estimator}},$$

a test statistic for testing $H_0 : \sum_{i=1}^a c_i \mu_i = \delta$ is written as

$$t = \frac{\sum_{i=1}^a c_i \bar{Y}_{i\bullet} - \delta}{\sqrt{MS_{\text{Error}}} \cdot \sqrt{\sum_{i=1}^a \frac{c_i^2}{n_i}}}, \quad (11.28)$$

which is distributed as a t -distribution with $N - a$ degrees of freedom when H_0 is true. A confidence interval for any contrast is then

$$\boxed{CI_{1-\alpha} \left(\sum_{i=1}^a c_i \mu_i \right) = \left[\sum_{i=1}^a c_i \bar{Y}_{i\bullet} - t_{1-\frac{\alpha}{2}; N-a} \cdot \sqrt{MS_{\text{Error}}} \cdot \sqrt{\sum_{i=1}^a \frac{c_i^2}{n_i}}, \sum_{i=1}^a c_i \bar{Y}_{i\bullet} + t_{1-\frac{\alpha}{2}; N-a} \cdot \sqrt{MS_{\text{Error}}} \cdot \sqrt{\sum_{i=1}^a \frac{c_i^2}{n_i}} \right].} \quad (11.29)$$

The sum of squares can also be computed for a contrast. In particular, the sum of squares for $\sum_{i=1}^a c_i \bar{Y}_{i\bullet}$ is

$$SS_{\hat{C}} = \frac{\left(\sum_{i=1}^a c_i \bar{Y}_{i\bullet} \right)^2}{\sum_{i=1}^a \frac{c_i^2}{n_i}}, \quad (11.30)$$

which has 1 degree of freedom. To test if the contrast $C = \sum_{i=1}^a c_i \mu_i$ is zero the ratio $SS_{\hat{C}} / MS_{\text{Error}}$ is formed, which follows an $F_{1, df_{\text{Error}}}$ when H_0 is true.

11.8.1 Orthogonal Contrasts

The contrasts C and D are said to be orthogonal if

$$\sum_{i=1}^a \frac{c_i d_i}{n_i} = 0.$$

Orthogonal contrasts are independent of one another and partition the treatment sum of squares. That is, if one computes the sum of squares for a full set of orthogonal contrasts ($a - 1$ contrasts for a treatments), adding up the $a - 1$ orthogonal contrasts will equal the treatment sum of squares ($SS_{\text{Treatment}}$). Unfortunately, the construction of a complete set of meaningful contrasts is not an easy proposition. Contrasts should be used to answer scientific questions of interest rather than because a complete set of orthogonal contrasts can be computed.

Example 11.4 \triangleright **Drosophila** \triangleleft The data set **DROSOPHILA** contains per diem fecundity (number of eggs laid per female per day for the first 14 days of life) for 25 females from each of three lines of *Drosophila melanogaster*. The three lines are Nonselected (control), Resistant, and Susceptible. The original measurements are from an experiment conducted by R. R. Sokal (Sokal and Rohlf, 1994, p. 237). Test if there are

- (a) Differences between the three genetic lines,
- (b) Differences in fecundity between the Resistant and the Susceptible lines versus the Nonselected line, and
- (c) Fecundity differences between the Resistant and the Susceptible lines.

Solution: The first question (a) seeks to answer if there are differences in the treatment means. In this case, the hypothesis of interest is $H_0 : \mu_{\text{Nonselected}} = \mu_{\text{Resistant}} = \mu_{\text{Susceptible}}$. The second question is typical of experiments with two new treatments and a control. The

null hypothesis for question (b) is equality between the Nonselected line (control) and the Resistant and the Susceptible lines (the two new treatments), written

$$H_0 : \mu_{\text{Nonselected}} = \frac{\mu_{\text{Resistant}} + \mu_{\text{Susceptible}}}{2}.$$

The hypothesis needed to answer question (c) of whether the two treatments (Resistant and Susceptible) are different is written

$$H_0 : \mu_{\text{Resistant}} = \mu_{\text{Susceptible}}.$$

(a) To test $H_0 : \mu_{\text{Nonselected}} = \mu_{\text{Resistant}} = \mu_{\text{Susceptible}}$ versus $H_1 : \mu_i \neq \mu_j$ for some $i \neq j$, an F -test is formed from the ratio of $MS_{\text{Treatment}}/MS_{\text{Error}} = 8.67$ that yields a p -value of 0.0004. Based on the small p -value, the null hypothesis of equal means is rejected. The evidence suggests mean fecundity between lines is different. The values for the ANOVA table needed to test the null hypothesis are provided Table 11.6. R Code 11.19 creates the ANOVA table.

R Code 11.19

```
> mod.dro <- aov(fecundity ~ line, data = DROSOPHILA)
> summary(mod.dro) # ANOVA

      Df Sum Sq Mean Sq F value    Pr(>F)
line       2   1362   681.1   8.666 0.000424 ***
Residuals 72   5659    78.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 11.6: ANOVA table for model `fecundity~line` using `DROSOPHILA` data

Source	<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	<i>p</i> -value
Treatments	$3 - 1 = 2$	1362	$\frac{1362}{2} = 681$	$\frac{681}{79} = 8.67$	$4e - 04$
Error	$75 - 3 = 72$	5659	$\frac{5659}{72} = 79$		
Total	74	7021			

Before answering (b) and (c), the residuals are examined (not shown) for the model `fecundity~line` with the function `checking.plots()`. No problems are noted, so the second and third questions can be answered using the orthogonal contrasts

$$C_1 = \mu_{\text{Nonselected}} - \frac{\mu_{\text{Resistant}} + \mu_{\text{Susceptible}}}{2},$$

which has coefficients $c_i = (1, -0.5, -0.5)$, and

$$C_2 = \mu_{\text{Resistant}} - \mu_{\text{Susceptible}},$$

which has coefficients $d_i = (0, 1, -1)$. Contrasts C_1 and C_2 are orthogonal because

$$\sum_{i=1}^a \frac{c_i \cdot d_i}{n_i} = \frac{1 \times 0}{25} + \frac{-0.5 \times 1}{25} + \frac{-0.5 \times -1}{25} = 0.$$

Since there are $a = 3$ treatments, there are two degrees of freedom for a set of orthogonal contrasts. The sum of squares for the first contrast is 1329.0817 and the sum of squares for the second contrast is 33.1298. The sum of squares for treatments is 1362.2115, which equals the sum of the sum of squares for the two orthogonal contrasts: $1329.0817 + 33.1298$. The ϕ -value for the first contrast (ϕ -value = $1e - 04$) provides strong evidence to suggest $\mu_{\text{Nonselected}} \neq \frac{\mu_{\text{Resistant}} + \mu_{\text{Susceptible}}}{2}$. The ϕ -value for the second contrast (ϕ -value = 0.5182) provides insufficient evidence to reject the null hypothesis $\mu_{\text{Resistant}} = \mu_{\text{Susceptible}}$.

The sums of squares for \hat{C}_1 and \hat{C}_2 using (11.30) are computed as

$$SS_{\hat{C}_1} = \frac{[(1 \times 33.372) + (-0.5 \times 25.256) + (-0.5 \times 23.628)]^2}{\frac{1^2}{25} + \frac{(-.5)^2}{25} + \frac{(-.5)^2}{25}} = 1329.0817$$

and

$$SS_{\hat{C}_2} = \frac{[(0 \times 33.372) + (1 \times 25.256) + (-1 \times 23.628)]^2}{\frac{0^2}{25} + \frac{1^2}{25} + \frac{(-1)^2}{25}} = 33.1298.$$

Note the ϕ -values in Table 11.7 are individual ϕ -values. That is, they are not simultaneously correct ϕ -values. To obtain ϕ -values adjusted for simultaneous inference or simultaneous confidence intervals, one should use the R package `multcomp`.

Table 11.7: ANOVA table for orthogonal contrasts with **DROSOPHILA**

Source	df	SS	MS	F	ϕ -value
\hat{C}_1	1	1329.08	1329.08	16.91	$1e - 04$
\hat{C}_2	1	33.13	33.13	0.42	0.5182
Treatments	2	1362.21	681.11	8.67	$4e - 04$
Error	72	5659.00	78.60		
Total	74	7021.21			

The values used in the ANOVA table for the contrasts were computed using R Code 11.20.

R Code 11.20

```
> MSerror <- summary(mod.dro)[[1]][2, 3] # MS error
> SSTreat <- summary(mod.dro)[[1]][1, 2] # SS treatments
> DFerror <- summary(mod.dro)[[1]][2, 1] # DDF error
> ybar <- with(data = DROSOPHILA, tapply(fecundity, line, mean))
> ni <- xtabs(~ line, data = DROSOPHILA) # Number per group
> ci <- c(1, -0.5, -0.5) # Coefficients for contrast 1
> di <- c(0, 1, -1) # Coefficients for contrast 2
> ortho <- sum(ci*di/ni) # Verify orthogonality
> ortho
```

```
[1] 0

> SSC1 <- (sum(ci*ybar))^2/sum(ci^2/ni)      # SS contrast 1
> SSC2 <- (sum(di*ybar))^2/sum(di^2/ni)        # SS contrast 2
> FC1 <- SSC1/MSerror                          # F value contrast 1
> FC2 <- SSC2/MSerror                          # F value contrast 2
> Fs <- c(FC1, FC2)
> pvalue <- pf(Fs, 1, DFerror, lower = FALSE)
> res <- cbind(SS = c(SSC1, SSC2), Fs, pvalue)
> rownames(res) <- c("C1", "C2")
> res

  SS          Fs      pvalue
C1 1329.0817 16.909967 0.0001027371
C2   33.1298  0.421512 0.5182493283
```

There are several ways to obtain contrasts with R by changing the type of contrasts R uses. Contrast settings for R include `contr.helmert`, `contr.poly`, `contr.sum`, `contr.treatment`, and `contr.SAS`. The interested reader should refer to the help documentation by typing `?contr.helmert` for more explanation. R uses `contr.treatment` for unordered factors which is not strictly a contrast in its default options. The option `contr.helmert` produces Helmert contrasts, which are orthogonal contrasts when there are an equal number of observations at each of the factor levels. For example, the default contrasts for `line` are `contr.treatment`, which contrast each level with the baseline level (`Nonselected`).

```
> contrasts(DROSOPHILA$line)

             Resistant Susceptible
Nonselected      0          0
Resistant        1          0
Susceptible      0          1
```

To compute Helmert contrasts, type

```
> contrasts(DROSOPHILA$line) <- contr.helmert(levels(DROSOPHILA$line))
> contrasts(DROSOPHILA$line)

 [,1] [,2]
Nonselected    -1    -1
Resistant       1    -1
Susceptible     0     2
```

To compute the sum of squares for the contrasts used in parts (b) and (c), see R Code 11.21. Note that there are two orthogonal contrasts, each with one degree of freedom shown in the output. The first contrast, $C_1 = \mu_{\text{Nonselected}} - \frac{\mu_{\text{Resistant}} + \mu_{\text{Susceptible}}}{2}$, is represented in the output as `C(line, CON, 1)` while the second contrast, $C_2 = \mu_{\text{Resistant}} - \mu_{\text{Susceptible}}$, is represented in the output as `C(line, CON, 2)`.

R Code 11.21

```
> contrasts(DROSOPHILA$line)[, 1] <- ci
> contrasts(DROSOPHILA$line)[, 2] <- di
> CON <- contrasts(DROSOPHILA$line)
```

```
> colnames(CON) <- c("Non vs. Res and Sus", "Res vs. Sus")
> CON

      Non vs. Res and Sus Res vs. Sus
Nonselected           1.0          0
Resistant            -0.5          1
Susceptible          -0.5         -1

> summary(aov(fecundity ~ C(line, CON, 1) + C(line, CON, 2),
+               data = DROSOPHILA))

      Df Sum Sq Mean Sq F value    Pr(>F)
C(line, CON, 1)   1   1329   1329.1  16.910 0.000103 ***
C(line, CON, 2)   1     33     33.1   0.422 0.518249
Residuals        72   5659    78.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The contrasts can also be tested using the `lm()` function. The contrasts C_1 and C_2 are labeled `line1` and `line2` in R Code 11.22.

R Code 11.22

```
> summary(lm(fecundity ~ line, data = DROSOPHILA)) # lm output

Call:
lm(formula = fecundity ~ line, data = DROSOPHILA)

Residuals:
    Min      1Q  Median      3Q      Max 
-18.472  -5.764  -0.728   4.436  24.872 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 27.419     1.024  26.784 < 2e-16 ***
line1       5.953     1.448   4.112 0.000103 ***
line2       0.814     1.254   0.649 0.518249
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.866 on 72 degrees of freedom
Multiple R-squared:  0.194, Adjusted R-squared:  0.1716 
F-statistic: 8.666 on 2 and 72 DF,  p-value: 0.0004244
```

To obtain simultaneous p -values and confidence intervals, R Code 11.23 on the facing page uses the package `multcomp`. The function `glht()` (general linear hypotheses) is used with two arguments: `model`, and `linfct`. The object assigned to `model` is a fitted model (`mod.dro`), and the object assigned to `linfct` is a specification of the linear hypotheses to be tested. Multiple comparisons are further specified by using the `mcp()` function.

R Code 11.23

```
> library(multcomp)                      # needed for glht()
> MC <- glht(model = mod.dro, linfct = mcp(line = t(CON)) )
> summary(MC)                           # Summary of MC object

Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: User-defined Contrasts

Fit: aov(formula = fecundity ~ line, data = DROSOPHILA)

Linear Hypotheses:
Estimate Std. Error t value Pr(>|t|)
Non vs. Res and Sus == 0     8.930      2.172   4.112 0.000205 ***
Res vs. Sus == 0            1.628      2.508   0.649 0.766699
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)

> CI <- confint(MC, level = 0.95)  # 95% CIs for contrasts
> CI
```

Simultaneous Confidence Intervals

Multiple Comparisons of Means: User-defined Contrasts

Fit: aov(formula = fecundity ~ line, data = DROSOPHILA)

Quantile = 2.2827
95% family-wise confidence level

Linear Hypotheses:

	Estimate	lwr	upr
Non vs. Res and Sus == 0	8.9300	3.9728	13.8872
Res vs. Sus == 0	1.6280	-4.0961	7.3521

Figure 11.15 on the next page shows simultaneous 95% family-wise confidence intervals for the two contrasts obtained from R Code 11.24.

R Code 11.24

```
> opar <- par(no.readonly = TRUE)      # Read in graphical parameters
> par(mar = c(5.1, 10.1, 4.1, 2.1))  # Enlarge left margin
> CI <- confint(MC, level = 0.95)    # Compute 95% CIs
> plot(CI)                          # Graph CIs
> par(opar)                         # Reset graphical parameters
```

95% family-wise confidence level

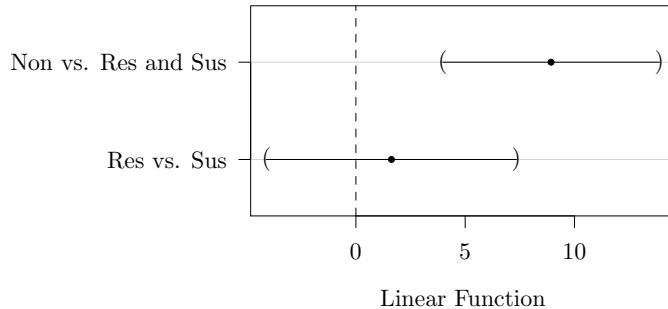


FIGURE 11.15: Simultaneous 95% confidence intervals for the contrasts C_1 and C_2

11.8.2 The Scheffé Method for All Contrasts

The Scheffé method controls the experiment-wise error rate α_e for all possible comparisons, including contrasts, suggested by the data. Consequently, it is the appropriate technique for examining a large number of unplanned comparisons. Its relatively low power limits its legitimate use to data snooping or to investigating contrasts that cannot be handled by other techniques. The Scheffé method is equivalent to the F -test in that the Scheffé method will not find differences in means if the F -test does not reject H_0 . Also, if the F -test does reject H_0 , then there exists at least one comparison that the Scheffé method will declare significant. Unfortunately, finding the comparison(s) that the Scheffé method will declare significant is a process composed entirely of trial and error.

To test the null hypothesis $H_0 : \sum_{i=1}^a c_i \mu_i = 0$ with the Scheffé test statistic S , the ratio $S_{\text{obs}} = \frac{SS_{\hat{C}}/(a-1)}{MS_{\text{Error}}}$ is formed, where $SS_{\hat{C}}$ is as given in (11.30). The null hypothesis is rejected at the α_e level for $S_{\text{obs}} > f_{1-\alpha_e; a-1, \nu}$, where $\nu = df_{\text{Error}}$. For the one-way CRD, $\nu = N - a$. For other models, ν will be different. A confidence interval for an arbitrary contrast, $\sum_{i=1}^a c_i \mu_i$, at the $1 - \alpha_e$ confidence level is

$$CI_{1-\alpha_e} \left(\sum_{i=1}^a c_i \mu_i \right) = \left[\sum_{i=1}^a c_i \bar{Y}_{i\bullet} - \sqrt{(a-1)f_{1-\alpha_e; a-1, \nu}} \cdot \sqrt{MS_{\text{Error}} \sum_{i=1}^a \frac{c_i^2}{n_i}}, \sum_{i=1}^a c_i \bar{Y}_{i\bullet} + \sqrt{(a-1)f_{1-\alpha_e; a-1, \nu}} \cdot \sqrt{MS_{\text{Error}} \sum_{i=1}^a \frac{c_i^2}{n_i}} \right]. \quad (11.31)$$

The Scheffé confidence intervals have simultaneous $1 - \alpha_e$ coverage over any set of contrasts.

11.9 Summary of Comparisons of Means

Let the questions to be answered determine the type of contrast that is tested. If the researcher is only interested in determining differences among the means, Tukey's HSD should be used. Scheffé's method provides a constant α_e protection for any contrast, which makes it ideal for “data snooping.” Tukey's HSD intervals can be obtained using the function

`TukeyHSD()`. For simultaneous inference, the R package `multcomp` should be consulted.

Example 11.5 ▷ Pairwise Mean Comparisons ◁ Compare all treatment means from Example 11.1 to determine which tire treads have the shortest stopping distance. Use $\alpha = 0.05$ with

- (a) Fisher's least significant difference,
- (b) Bonferroni's significant difference,
- (c) Tukey's honestly significant difference,
- (d) Scheffé's method.

Solution: Each of the methods provides a cutoff value for considering a difference of means significant. The estimated means are $\hat{\mu}_A = 379.6667$, $\hat{\mu}_B = 405.1667$, $\hat{\mu}_C = 421.6667$, and $\hat{\mu}_D = 410.3333$. The estimated mean differences with which these values will be compared are

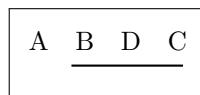
- I. $\hat{\mu}_B - \hat{\mu}_A = \bar{Y}_{2\bullet} - \bar{Y}_{1\bullet} = 25.5$.
- II. $\hat{\mu}_C - \hat{\mu}_A = \bar{Y}_{3\bullet} - \bar{Y}_{1\bullet} = 42$.
- III. $\hat{\mu}_D - \hat{\mu}_A = \bar{Y}_{4\bullet} - \bar{Y}_{1\bullet} = 30.6667$.
- IV. $\hat{\mu}_C - \hat{\mu}_B = \bar{Y}_{3\bullet} - \bar{Y}_{2\bullet} = 16.5$.
- V. $\hat{\mu}_D - \hat{\mu}_B = \bar{Y}_{4\bullet} - \bar{Y}_{2\bullet} = 5.1667$.
- VI. $\hat{\mu}_D - \hat{\mu}_C = \bar{Y}_{4\bullet} - \bar{Y}_{3\bullet} = -11.3333$.

- (a) Fisher's LSD considers group means significantly different if

$$\underbrace{\left| \bar{Y}_{i\bullet} - \bar{Y}_{j\bullet} \right| > t_{1-\frac{\alpha}{2}; df_{\text{Error}}} \cdot \sqrt{MS_{\text{Error}}} \cdot \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}}_{\text{LSD}}$$

$$\text{LSD} = 2.086 \cdot \sqrt{354.9417} \cdot \sqrt{\frac{1}{6} + \frac{1}{6}} = 22.6895.$$

Comparing this value with the statistics from I–VI indicates that μ_A is significantly different from μ_B , μ_D , and μ_C :



The function `pairwise.t.test()` returns a p -value for comparisons between group levels that adjusts for multiple testing with the argument `p.adjust.methods`. Since Fisher's least significant difference does not make any adjustments for multiple testing, the argument used is `p.adjust.method = "none"`. Any differences where the p -value is less than $\alpha = 0.05$ are declared significantly different.

```
> pairwise.t.test(TIRE$stopdist, TIRE$tire, p.adjust.method = "none")
```

```
Pairwise comparisons using t tests with pooled SD
```

```
data: TIRE$stopdist and TIRE$tire
```

A	B	C
B 0.02950	-	-
C 0.00097	0.14493	-
D 0.01059	0.63993	0.30987

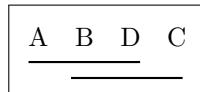
```
P value adjustment method: none
```

(b) Bonferroni's significant difference considers group means significantly different if

$$|\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}| > \underbrace{t_{1-\frac{\alpha_e}{2-K}; df_{\text{Error}}} \cdot \sqrt{MS_{\text{Error}}} \cdot \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}}_{\text{BSD}}$$

$$\text{BSD} = 2.9271 \cdot \sqrt{354.9417} \cdot \sqrt{\frac{1}{6} + \frac{1}{6}} = 31.8389.$$

Comparing this value with the statistics from I–VI indicates that μ_A is significantly different from μ_C . The reader may note that treatment A is not significantly different from treatment B; and, at the same time, treatment B is not significantly different from treatment C. However, treatments A and C are significantly different from each other. When comparing pairwise means, the transitive property does not hold. For orthogonal contrasts, on the other hand, the transitive property will hold.



```
> pairwise.t.test(TIRE$stopdist, TIRE$tire, p.adjust.method = "bonferroni")
```

```
Pairwise comparisons using t tests with pooled SD
```

```
data: TIRE$stopdist and TIRE$tire
```

A	B	C
B 0.1770	-	-
C 0.0058	0.8696	-
D 0.0636	1.0000	1.0000

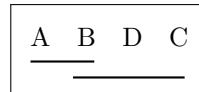
```
P value adjustment method: bonferroni
```

(c) Tukey's honestly significant difference considers group means significantly different if

$$|\bar{Y}_{i\bullet} - \bar{Y}_{j\bullet}| > \underbrace{q_{1-\alpha_e; a, \nu} \cdot \frac{\sqrt{MS_{\text{Error}}}}{\sqrt{n}}}_{HSD}$$

$$HSD = 3.9583 \cdot \frac{\sqrt{354.9417}}{\sqrt{6}} = 30.4446.$$

Comparing this value with the statistics from I–VI indicates that μ_A is significantly different from μ_D and μ_C :



R Code 11.25 computes Tukey's HSD's pairwise confidence intervals, while the function `plot()` can be applied to `CI2` to create a graphical representation of the confidence intervals.

R Code 11.25

```
> mod.tire <- aov(stopdist ~ tire, data = TIRE)
> CI2 <- TukeyHSD(mod.tire, which = "tire")
> CI2

Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = stopdist ~ tire, data = TIRE)

$tire
      diff      lwr      upr      p adj
B-A  25.500000 -4.9446409 55.94464 0.1213153
C-A  42.000000 11.5553591 72.44464 0.0049515
D-A  30.666667  0.2220258 61.11131 0.0479540
C-B  16.500000 -13.9446409 46.94464 0.4464584
D-B   5.166667 -25.2779742 35.61131 0.9637307
D-C -11.333333 -41.7779742 19.11131 0.7273681
```

Figure 11.16 on the following page is a `ggplot2` graph of the six 95% Tukey HSD pairwise confidence intervals. R Code 11.26 can be modified to create a graph similar to Figure 11.16 on the following page. Note that the information in `CI2` is converted into the data frame `tire.hsd`. It may prove helpful to step through each layer in R Code 11.26.

R Code 11.26

```
> tire.hsd <- data.frame(CI2$tire)
> tire.hsd$Comparison <- row.names(tire.hsd)
> ggplot(data = tire.hsd, aes(x = Comparison, y = diff, ymin = lwr,
+                               ymax = upr)) +
+   geom_pointrange() +
+   geom_hline(yintercept = 0, linetype = "dashed") +
+   geom_errorbar(aes(ymin = lwr, ymax = upr), width = 0.2) +
+   labs(y = "\nDifferences in mean levels of tire", x = "",
+        title = "95% family-wise confidence level\n") +
+   coord_flip() +
+   theme_bw()
```

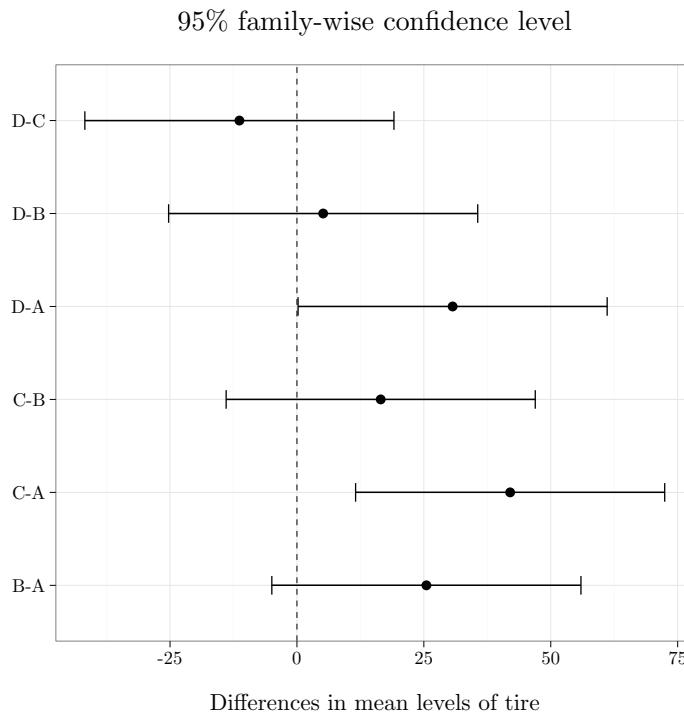


FIGURE 11.16: Graphical representation of confidence intervals based on Tukey's HSD for the model `stopdist ~ tire` using the data frame `TIRE`

(d) If $\sum_{i=1}^a c_i \bar{Y}_{i\bullet}$ is greater than

$$\sqrt{(a-1)f_{1-\alpha_e; a-1, \nu}} \cdot \sqrt{MS_{\text{Error}} \sum_{i=1}^a \frac{c_i^2}{n_i}},$$

Scheffé's method will consider the group means significantly different.

In this case,

$$\sqrt{(a-1)f_{1-\alpha_e; a-1, \nu}} \cdot \sqrt{MS_{\text{Error}} \sum_{i=1}^a \frac{c_i^2}{n_i}} = \sqrt{(4-1) \cdot 3.0984} \cdot \sqrt{354.9417 \cdot \frac{2}{6}} = 33.1625.$$

Comparing this value with the statistics from I–VI indicates that μ_A is significantly different from μ_C :

A	B	D	C
<hr/>			

R Code 11.27 shows how one might compute the LSD, BSD, HSD, and Scheffé statistics as well as the pairwise mean differences.

R Code 11.27

```
> alpha <- 0.05
```

```

> mod.tire <- aov(stopdist ~ tire, data = TIRE)
> a <- length(levels(TIRE$tire))
> N <- length(TIRE$stopdist)
> dfe <- N - a
> MSE <- summary(mod.tire)[[1]][2, 3]
> MEANS <- tapply(TIRE$stopdist, TIRE$tire, mean)
> MM <- outer(MEANS, MEANS, "-")
> ni <- xtabs(~tire, data = TIRE)[1] # ni's are all the same (6)
> names(ni) <- NULL
> K <- choose(a, 2) # number of comparisons (6)
> tLSD <- qt(1 - alpha/2, dfe) # LSD critical value
> tBSD <- qt(1 - alpha/(2*K), dfe) # Bonferroni critical value
> tHSD <- qtukey(1 - alpha, a, dfe) # Tukey critical value
> cSCH <- sqrt((a - 1)*qf(1 - alpha, a - 1, dfe)) # Scheffe CV
> LSD <- tLSD * sqrt(MSE) * sqrt(1/ni + 1/ni)
> BON <- tBSD * sqrt(MSE) * sqrt(1/ni + 1/ni)
> HSD <- tHSD * sqrt(MSE) * sqrt(1/ni)
> # Note: Contrast ci's are (1, -1, 0, 0) ... so sum(ci^2) = 2
> SCH <- cSCH * sqrt(MSE * 2/ni)
> c(LSD = LSD, BON = BON, HSD = HSD, SCH = SCH) # Sig differences

      LSD      BON      HSD      SCH
22.68948 31.83892 30.44464 33.16245

> MM # outer(MEANS, MEANS, "-")

      A          B          C          D
A 0.00000 -25.50000 -42.00000 -30.666667
B 25.50000  0.00000 -16.50000 -5.166667
C 42.00000 16.50000  0.00000 11.333333
D 30.66667  5.166667 -11.33333  0.000000

> MM[lower.tri(MM)]

```

[1] 25.500000 42.000000 30.666667 16.500000 5.166667 -11.333333

11.10 Random Effects Model (Variance Components Model)

In the motivational problem Example 11.1, the levels of the factor (tread type) were considered fixed, since only four tread types were available and a decision was sought for the effect of stopping distance for these four types of treads. If, however, the experimenter is interested in a factor that has a large number of possible values and randomly selects a of the possible levels from the population of factor levels, the experiment is modeled as a random effects model. This model is different from the fixed effects model because the levels of the factor are chosen at random. Consequently, inference will apply to the entire population of factor levels, not merely to the a levels in the model. For example, consider a clothing manufacturer that produces work clothes. The strength of the material used

in the clothes varies depending on the wool supplier. The manufacturer contracts with a few out of many hundreds of wool suppliers (usually on the basis of price). Since the number of suppliers is very large, by randomly selecting a few suppliers, one can estimate the variability in clothing strength due to suppliers. That is, one is not interested in the particular randomly selected supplier per se. Rather, the goal is to learn something about the suppliers' variability as a whole relative to clothing strength. The statistical model for the one-way random effects remains

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}.$$

In contrast, τ_i is now considered a random variable; whereas in the fixed effect model, it was a parameter. That is, in the random effects model, both τ_i and ε_{ij} are random variables. Consequently the constraint $\sum_{i=1}^a \tau_i = 0$ from the fixed effects model does not apply to the random effects model. The assumptions, then, for the one-way random effects model are

- (1) $\varepsilon_{ij} \sim NID(0, \sigma^2)$.
- (2) $\tau_i \sim NID(0, \sigma_\tau^2)$.
- (3) τ_i and ε_{ij} are independent.

Because of assumption number (3), the variance of any observation is $\sigma_{Y_{ij}}^2 = \sigma_{\tau_i}^2 + \sigma^2$. In the random effects model, one is interested in estimating variance components, not in testing treatment means. The reason for this is that the means will vary due to the random nature of selecting the a treatments from the entire population of possible treatments. The partitioning of the sum of squares employed with the fixed effects model is still valid with the random effects model; however, the hypotheses of interest are now

$$H_0 : \sigma_\tau^2 = 0 \quad \text{versus} \quad H_1 : \sigma_\tau^2 > 0,$$

which are tested using the ANOVA procedure outlined for the fixed effects model. If the null hypothesis cannot be rejected, $\sigma_\tau^2 = 0$, it is concluded that there are no treatment differences. On the other hand, if the alternative hypothesis is supported, $\sigma_\tau^2 > 0$, the conclusion is that variability exists among treatments.

The test statistic for testing $\sigma_\tau^2 = 0$ is $MS_{\text{Treatment}}/MS_{\text{Error}}$, which follows an $F_{a-1, N-a}$ distribution when the null hypothesis is true. Although the same ANOVA table is used for fixed effects and random effects models, the interpretations are different. The conclusions from a random effects model are not limited to the a treatments used in the computation of the test statistic but rather apply to the entire population of treatments. Estimators for the two variance components when the a treatments have equal sample size n are

$$\hat{\sigma}^2 = MS_{\text{Error}} \quad \text{and} \quad \hat{\sigma}_\tau^2 = \frac{MS_{\text{Treatment}} - MS_{\text{Error}}}{n}. \quad (11.32)$$

When treatment sample sizes are unequal, the n in (11.32) is replaced with n' , where

$$n' = \frac{1}{a-1} \sum_{i=1}^a n_i - \frac{\sum_{i=1}^a n_i^2}{\sum_{i=1}^a n_i}. \quad (11.33)$$

Example 11.6  A food processing company that uses many hundreds of freezers is studying the variability of its freezers with respect to the texture of frozen carrots. The shear measured in kN on frozen carrots from four randomly selected freezers (labeled A, B, C, and D) is shown in Table 11.8 and is available in the **FOOD** data frame. The company would like all of its freezers to be homogeneous in order to control the taste of the frozen carrots.

- (a) Test the null hypothesis $H_0 : \sigma_\tau^2 = 0$ for freezers.
 (b) Estimate the component of variance for freezers.

Table 11.8: Shear on frozen carrots by freezer

A	1.96	1.94	1.98	1.92
B	1.82	1.80	1.86	1.84
C	1.92	1.90	1.94	1.90
D	1.90	1.92	1.98	1.96

Solution: The answers are as follows:

- (a) The company in this problem is ultimately interested in reducing freezer variability and wants to know if there is more variability in their frozen carrots due to the carrots themselves or due to the numerous freezers used in freezing the carrots.

The hypotheses to be tested are

$$H_0 : \sigma_\tau^2 = 0 \quad \text{versus} \quad H_1 : \sigma_\tau^2 > 0$$

Table 11.9: Frozen carrots ANOVA table

Source of Variation (Source)	Degrees of Freedom (df)	Sum of Squares (SS)	Mean Square (MS)	F	p-value
Treatments	3	0.0357	0.0119	15.6813	2e - 04
Error	12	0.0091	8e - 04		
Total	15	0.0448			

From Table 11.9, one can see that there is strong evidence to suggest $\sigma_\tau^2 > 0$ ($p\text{-value} = 2e - 04$). In other words, there is more variability due to the freezers than variability due to the carrots.

(b) $\hat{\sigma}^2 = MS_{\text{Error}} = 8e - 04$

$$\hat{\sigma}_\tau^2 = \frac{MS_{\text{Treatment}} - MS_{\text{Error}}}{n} = \frac{0.0119 - 8e - 04}{4} = 0.0028.$$

R Code 11.28 can be used to verify the values in Table 11.9 as well as the estimate $\hat{\sigma}_\tau^2$.

R Code 11.28

```
> carrot.mod <- aov(shear ~ freezer, data = FOOD)
> n <- xtabs(~freezer, data = FOOD)[1] # Number per freezer
```

```

> names(n) <- NULL
> ANOVA <- summary(carrot.mod) # ANOVA
> ANOVA

      Df  Sum Sq  Mean Sq F value    Pr(>F)
freezer     3 0.03567 0.011892   15.68 0.000188 ***
Residuals  12 0.00910 0.000758
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> MST <- ANOVA[[1]][1, 3] # MS treatments
> MSE <- ANOVA[[1]][2, 3] # MS error
> sig2tau <- (MST - MSE)/n
> sig2tau

[1] 0.002783333

```

11.11 Randomized Complete Block Design

The t -tests from Sections 9.7.4 and 9.7.6 were used to compare two treatments. Comparisons between the two treatments used the paired t -test when the measurements being compared were related, and consequently more homogeneous. The main idea behind using the paired t -test was to reduce the overall variability of the experiment by pairing observations. When comparing two treatments, whenever the variability within the pairs is smaller than the between-pairs variability, detection of the treatment effect is improved by using a paired design. When observations that are homogeneous in some respect are grouped together, the result is referred to as a **block**. Blocks are used in many settings to reduce variability. Some of these include agricultural studies with different strips of land, different litters of animals, and batches of chemical materials. In this section, the paired t -test is generalized to $a \geq 2$ treatments and the resulting design is referred to as a **randomized complete block design**. (Thus, pairing is a special case of blocking where each block is of size two.) The design is called complete because each treatment is used in every block. Instead of treatments being assigned to experimental units, as was the case in the completely randomized design, the randomized complete block design assigns treatments to an equal number of experimental units (usually one) at random within each block. The statistical model used to represent a randomized complete block design (RCBD) is

$$Y_{ij} = \mu + \tau_i + \beta_j + \varepsilon_{ij} \text{ for } i = 1, 2, \dots, a \text{ and } j = 1, 2, \dots, b \quad (11.34)$$

where μ is the grand mean; τ_i is the i^{th} treatment effect, which is the difference between the mean response of the i^{th} treatment over all blocks and the grand mean; β_j is the j^{th} block effect, which is the difference between the mean response of the j^{th} block over all treatments and the grand mean; and ε_{ij} are the $NID(0, \sigma)$ error terms. Treatment and block effects are considered fixed effects, defined as deviations from the grand mean so that $\sum_{i=1}^a \tau_i = 0$ and $\sum_{j=1}^b \beta_j = 0$. Note that model (11.34) is a completely additive model, which assumes blocks and treatments do not interact. That is, if treatment one causes the expected response to increase by 3 units ($\tau_1 = 3$), and if the first block decreases the

expected response by 1 unit ($\beta_1 = -1$), then the expected response for both treatment and block one is $E(Y_{11}) = \mu + \tau_1 + \beta_1 = \mu + 3 - 1 = \mu + 2$. A RCBD is really a design with two factors, where only one factor (the one measuring the treatment effect) is of interest. The other factor (called a block) is used to reduce the experiment's variability and to enhance its ability to detect treatment differences for the factor of interest. Analysis of the RCBD differs from a two-factor design because the blocking factor is not randomized. This dependence in the blocking factor means there is no theoretical justification for a test of blocks. However, one will often look at the ratio $MS_{\text{Blocks}}/MS_{\text{Error}}$ to get an idea if blocking was beneficial. Just keep in mind that the ratio $MS_{\text{Blocks}}/MS_{\text{Error}}$ does not truly follow an F distribution, as does the ratio $MS_{\text{Treatment}}/MS_{\text{Error}}$. One must remember that blocks should only be used when doing so reduces the overall design variability. To do otherwise reduces the power of the test.

The least squares estimators for the parameters in (11.34) are

$$\hat{\mu} = \bar{Y}_{\bullet\bullet} \quad (11.35)$$

$$\hat{\tau}_i = \bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet} \quad (11.36)$$

$$\hat{\beta}_j = \bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet} \quad (11.37)$$

and the residuals are

$$\hat{\varepsilon}_{ij} = Y_{ij} - \hat{Y}_{ij} = Y_{ij} - \bar{Y}_{i\bullet} - \bar{Y}_{\bullet j} + \bar{Y}_{\bullet\bullet}. \quad (11.38)$$

Each Y_{ij} from (11.34) can be decomposed into four parts by substituting the least squares estimates of μ , τ_i , β_j , and ε_{ij} for the parameters' values:

$$Y_{ij} = \hat{\mu} + \hat{\tau}_i + \hat{\beta}_j + \hat{\varepsilon}_{ij} \quad (11.39)$$

$$Y_{ij} = \bar{Y}_{\bullet\bullet} + (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet}) + (\bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet}) + (Y_{ij} - \bar{Y}_{i\bullet} - \bar{Y}_{\bullet j} + \bar{Y}_{\bullet\bullet})$$

$$(Y_{ij} - \bar{Y}_{\bullet\bullet}) = (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet}) + (\bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet}) + (Y_{ij} - \bar{Y}_{i\bullet} - \bar{Y}_{\bullet j} + \bar{Y}_{\bullet\bullet}).$$

Squaring and summing over $i = 1, 2, \dots, a$ and $j = 1, 2, \dots, b$ gives

$$\sum_{i=1}^a \sum_{j=1}^b (Y_{ij} - \bar{Y}_{\bullet\bullet})^2 = \sum_{i=1}^a \sum_{j=1}^b [(\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet}) + (\bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet}) + (Y_{ij} - \bar{Y}_{i\bullet} - \bar{Y}_{\bullet j} + \bar{Y}_{\bullet\bullet})]^2. \quad (11.40)$$

When the right side of (11.40) is expanded, all three cross products sum to zero (which is left to the reader to verify), giving

$$\begin{aligned} \underbrace{\sum_{i=1}^a \sum_{j=1}^b (Y_{ij} - \bar{Y}_{\bullet\bullet})^2}_{SS_{\text{Total}}} &= \underbrace{\sum_{i=1}^a \sum_{j=1}^b (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2}_{SS_{\text{Treatment}}} + \underbrace{\sum_{i=1}^a \sum_{j=1}^b (\bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet})^2}_{SS_{\text{Block}}} \\ &\quad + \underbrace{\sum_{i=1}^a \sum_{j=1}^b (Y_{ij} - \bar{Y}_{i\bullet} - \bar{Y}_{\bullet j} + \bar{Y}_{\bullet\bullet})^2}_{SS_{\text{Error}}}. \end{aligned} \quad (11.41)$$

The symbolic representation of (11.41) is

$$SS_{\text{Total}} = SS_{\text{Treatment}} + SS_{\text{Block}} + SS_{\text{Error}}.$$

The corresponding degrees of freedom are

$$\underbrace{a \cdot b - 1}_{\text{total } df} = \underbrace{a - 1}_{\text{treatment } df} + \underbrace{b - 1}_{\text{block } df} + \underbrace{(a - 1)(b - 1)}_{\text{error } df}.$$

The mean squares are computed as with the completely randomized design model by dividing each sum of squares by its corresponding degrees of freedom. The expected value of the mean squares, if treatments and blocks are fixed, can be shown to be

$$\begin{aligned} E(MS_{\text{Treatments}}) &= \sigma^2 + \frac{b \cdot \sum_{i=1}^a \tau_i^2}{a - 1} \\ E(MS_{\text{Blocks}}) &= \sigma^2 + \frac{a \cdot \sum_{j=1}^b \beta_j^2}{b - 1} \\ E(MS_{\text{Error}}) &= \sigma^2. \end{aligned}$$

Consequently, to test for no treatment effect, one uses the ratio $MS_{\text{Treatment}} / MS_{\text{Error}}$, which has an F distribution with $(a - 1)$ and $(a - 1)(b - 1)$ degrees of freedom when H_0 is true. There is no formal test for blocks; however, examining the ratio $MS_{\text{Blocks}} / MS_{\text{Error}}$, and comparing it to an F distribution with $(b - 1)$ and $(a - 1)(b - 1)$ degrees of freedom will give an indication of whether blocking is appropriate. If blocking is not appropriate, then it should be eliminated in future experiments. The ANOVA table for the randomized complete block design is given in Table 11.10 on the facing page.

Possible CRBD Treatment Assignments: Tire Wear Suppose a tire manufacturer is interested in determining tire tread loss after 10,000 miles of driving for the company's best-selling four tire models. Four cars and four tires of each tire model are available for the experiment. Let the tire models be denoted with the letters A, B, C, and D and the four cars be denoted as Car1, Car2, Car3, and Car4. One possible design is to assign the four tires of model A, B, C, and D to cars Car1, Car2, Car3, and Car4, respectively. This particular design confounds tire model with car, however. That is, it would not be known whether the differences in tire wear are due to cars or tire model. Another solution might be to use a completely randomized design, but not all tire models will necessarily be used on all cars. Consider the completely random assignment of tire models to cars (CRD) given in R Code 11.29. Note that tire model D is never used with Car1, tire model C is never used with Car2, and tire model A is never used with Car3. Further, any variation in model A may simply be due to Car1, Car2, and Car4. Although the completely randomized design averaged out the car effects, it did not eliminate the variance among cars. The randomized complete block design does remove the variability due to cars. One possible assignment of tire models within cars is given under the variable CRBD.

R Code 11.29

```
> car <- rep(c("Car1", "Car2", "Car3", "Car4"), each = 4)
> tire <- rep(LETTERS[1:4], each = 4)
> set.seed(13)
> CRD <- sample(tire)
> CRBD <- c(sample(LETTERS[1:4]), sample(LETTERS[1:4]),
+            sample(LETTERS[1:4]), sample(LETTERS[1:4]))
> DDF <- cbind(car, tire, CRD, CRBD)
> DDF
```

Table 11.10: ANOVA table for the randomized complete block design

Source of Variation (Source)	Degrees of Freedom (df)	Sum of Squares (SS)	Mean Square (MS)	F
Treatments	$a - 1$	$SS_{\text{Treatment}} = b \cdot \sum_{i=1}^a \hat{\tau}_i^2 \equiv$ $\sum_{i=1}^a \sum_{j=1}^b (\bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet})^2$	$MS_{\text{Treatment}} =$ $\frac{SS_{\text{Treatment}}}{a - 1}$	$\frac{MS_{\text{Treatment}}}{MS_{\text{Error}}}$
Blocks	$b - 1$	$SS_{\text{Blocks}} = a \cdot \sum_{j=1}^b \hat{\beta}_j^2 \equiv$ $\sum_{i=1}^a \sum_{j=1}^b (\bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet})^2$	$MS_{\text{Block}} =$ $\frac{SS_{\text{Block}}}{b - 1}$	
Error	$(a - 1)(b - 1)$	$SS_{\text{Error}} = \sum_{i=1}^a \sum_{j=1}^b \hat{\varepsilon}_{ij}^2 \equiv$ $\sum_{i=1}^a \sum_{j=1}^b (Y_{ij} - \bar{Y}_{i\bullet} - \bar{Y}_{\bullet j} + \bar{Y}_{\bullet\bullet})^2$	$MS_{\text{Error}} =$ $\frac{SS_{\text{Error}}}{(a - 1)(b - 1)}$	
Total	$a \cdot b - 1$	$SS_{\text{Total}} =$ $\sum_{i=1}^a \sum_{j=1}^b (Y_{ij} - \bar{Y}_{\bullet\bullet})^2$		

```

car      tire CRD CRBD
[1,] "Car1" "A"   "C"   "B"
[2,] "Car1" "A"   "A"   "D"
[3,] "Car1" "A"   "B"   "C"
[4,] "Car1" "A"   "A"   "A"
[5,] "Car2"  "B"   "D"   "A"
[6,] "Car2"  "B"   "A"   "B"
[7,] "Car2"  "B"   "D"   "C"
[8,] "Car2"  "B"   "B"   "D"
[9,] "Car3"  "C"   "C"   "A"
[10,] "Car3"  "C"   "C"   "B"
[11,] "Car3"  "C"   "D"   "D"
[12,] "Car3"  "C"   "B"   "C"
[13,] "Car4"  "D"   "C"   "B"
[14,] "Car4"  "D"   "D"   "D"
[15,] "Car4"  "D"   "A"   "A"
[16,] "Car4"  "D"   "B"   "C"

```

Example 11.7 \triangleright **Tire Wear** \triangleleft The data frame **TIREWEAR** contains measurements for the amount of tread loss after 10,000 miles of driving in thousandths of an inch. The tread loss from the **TIREWEAR** data frame is presented in tabular form in Table 11.11 along with the order the tires were assigned to the car in parentheses. The tires were tested first on **Car1**, second on **Car2**, third on **Car3**, and last on **Car4**. Use the values in Table 11.11 to test for treatment (tire model) effects using an additive RCBD. Sums and estimates are shown in Table 11.12.

- Verify that an additive model is appropriate.
- Compute the ANOVA table to test $H_0 : \tau_i = 0$ for all i versus $H_1 : \tau_i \neq 0$ for some i .
- Represent the Y_{ij} values using (11.39).
- Verify graphically that $\varepsilon_{ij} \sim N(0, \sigma)$.
- Determine which tires are different (have the least tread loss) using Tukey's HSD at $\alpha_e = 0.05$.

Table 11.11: The tread loss from the **TIREWEAR** data frame

	Car1	Car2	Car3	Car4
A	10 (4)	8 (1)	7 (1)	7 (3)
B	9 (1)	8 (2)	7 (2)	5 (1)
C	8 (3)	7 (3)	5 (4)	3 (4)
D	6 (2)	5 (4)	3 (3)	3 (2)

Table 11.12: Sums and estimates for Example 11.7

		Blocks				$\hat{\tau}_i = \bar{Y}_{i\bullet} - \bar{Y}_{\bullet\bullet}$
		Car1	Car2	Car3	Car4	
Tire	A	10	8	7	7	8.00
	B	9	8	7	5	7.25
	C	8	7	5	3	5.75
	D	6	5	3	3	4.25
$\bar{Y}_{\bullet j}$		8.25	7.00	5.50	4.50	$\bar{Y}_{\bullet\bullet} = 6.3125$
$\hat{\beta}_j = \bar{Y}_{\bullet j} - \bar{Y}_{\bullet\bullet}$		1.9375	0.6875	-0.8125	-1.8125	

Solution: The answers are as follows:

- The RCBD is completely additive, and an interaction plot is used to verify the reasonableness of the additivity assumption before computing any sums of squares. Interaction

plots show the relative size of main effects and interaction. The pairs (i, \bar{Y}_{ij}) for all j are plotted, and points in the same block are connected. The roles of blocks and treatments can be reversed, and it is often informative to do so with interaction plots. Parallel lines are indicative of additive designs. Lines that cross should be investigated further. Figure 11.17 does not suggest any problems with the RCBD's assumption of additivity; however, since graphs are often misleading and their interpretation is subjective, other means of analyzing and evaluating interaction should also be explored. One may create an interaction plot using the function `interaction.plot()`; however, R Code 11.30 provides code that can be used to create an interaction plot with `ggplot2` similar to the one shown in Figure 11.17.

R Code 11.30

```
> ggplot(data = TIREWEAR, aes(x = treat, y = wear, shape = block,
+                               group = block, linetype = block)) +
+   stat_summary(fun.y = mean, geom = "point") +
+   stat_summary(fun.y = mean, geom = "line") +
+   labs(y = "Mean Tire Wear (thousandths of an inch)", x = "Tire Type")
> ggplot(data = TIREWEAR, aes(x = block, y = wear, shape = treat,
+                               group = treat, linetype = treat)) +
+   stat_summary(fun.y = mean, geom = "point") +
+   stat_summary(fun.y = mean, geom = "line") +
+   labs(y = "Mean Tire Wear (thousandths of an inch)")
```

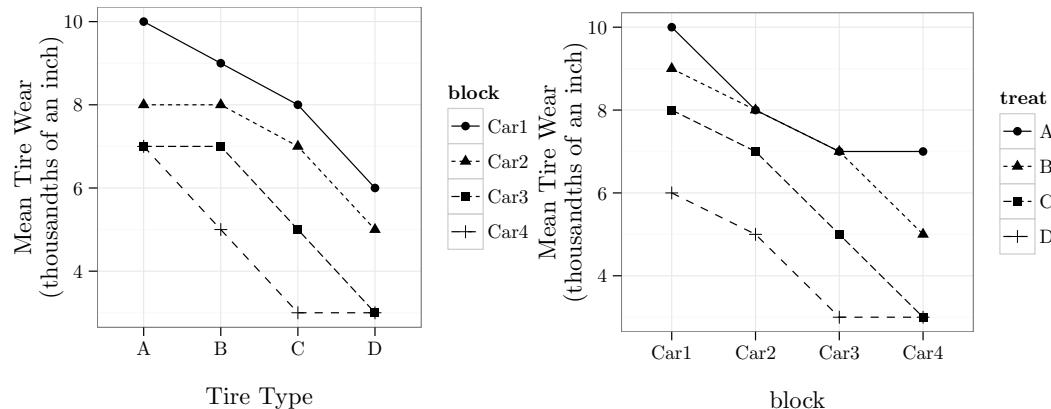


FIGURE 11.17: Left graph shows an interaction plot of blocks and treatments for the response `wear` with treatments shown along the x -axis. The right graph shows an interaction plot of treatments and blocks for the response `wear` with blocks shown along the x -axis.

The interaction plots in Figure 11.17 suggest both a treatment and a block effect. Another graph that is helpful when there is only one observation per treatment/block combination is the strip plot. Results from R Code 11.31 on the following page, which creates a strip plot with `ggplot2`, are shown in Figure 11.18 on the next page. One can see that tire wear increases with tire models in the order D, C, B, and then A. In a similar fashion, one notes that tire wear in cars increases in the order Car4, Car3, Car2, and then Car1.

R Code 11.31

```
> p1 <- ggplot(data = TIREWEAR, aes(x = wear, y = treat, shape = treat)) +
+   geom_point(size = 3) +
+   facet_grid(. ~ block) +
+   theme_bw()
> p2 <- ggplot(data = TIREWEAR, aes(x = wear, y = block, shape = block)) +
+   geom_point(size = 3) +
+   facet_grid(. ~ treat) +
+   theme_bw()
> library(gridExtra)
> grid.arrange(p1 + guides(color = FALSE), p2 + guides(color = FALSE),
+               ncol = 1)
```

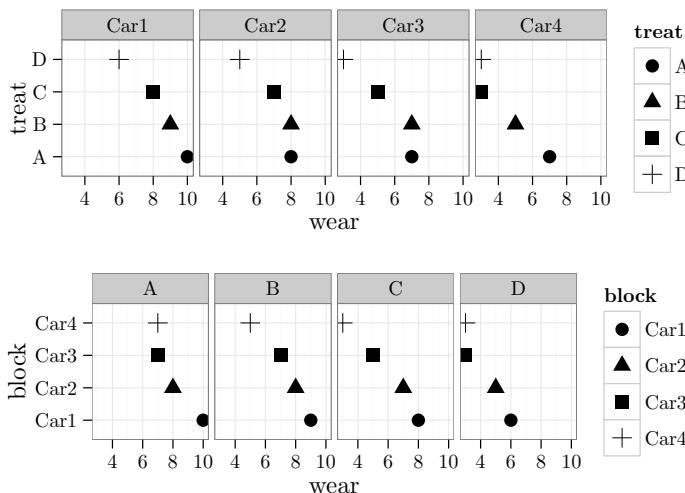


FIGURE 11.18: The `ggplot2` strip plots for Example 11.7, which show tread wear of each car by tire (top) and tread wear of each tire by car (bottom)

The graph showing tire wear means due to treatments and blocks using the function `plot.design()` is shown in Figure 11.19 on the facing page.

```
> with(data = TIREWEAR,
+       plot.design(wear ~ treat + block))
```

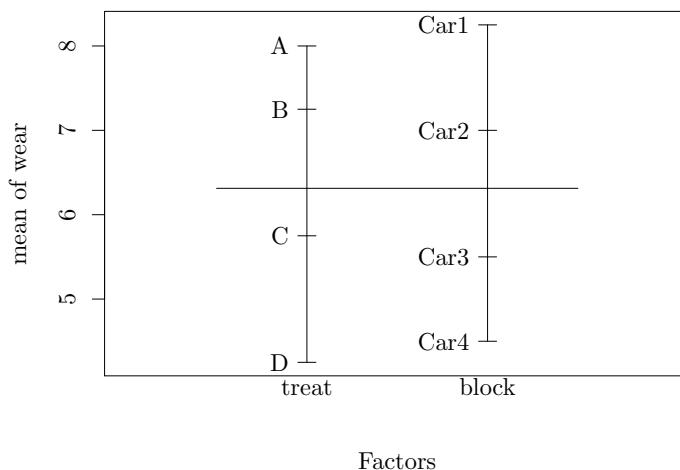


FIGURE 11.19: Tire wear means due to treatments and blocks using the function `plot.design()` for Example 11.7

(b) Using the values from Table 11.12 on page 752, the values for the ANOVA table are

$$\begin{aligned} SS_{\text{Treatment}} &= b \cdot \sum_{i=1}^a \hat{\tau}_i^2 = 4(1.6875^2 + 0.9375^2 + (-0.5625)^2 + (-2.0625)^2) \\ &= 33.1875 \\ SS_{\text{Block}} &= a \sum_{j=1}^b \hat{\beta}_j^2 = 4(1.9375^2 + 0.6875^2 + (-0.8125)^2 + (-1.8125)^2) \\ &= 32.6875 \end{aligned}$$

$$\begin{aligned} SS_{\text{Total}} &= \sum_{i=1}^a \sum_{j=1}^b (Y_{ij} - \bar{Y}_{\bullet\bullet})^2 \\ &= (10 - 6.3125)^2 + (9 - 6.3125)^2 + (8 - 6.3125)^2 + \dots + (3 - 6.3125)^2 \\ &= 69.4375 \\ SS_{\text{Error}} &= SS_{\text{Total}} - SS_{\text{Treatment}} - SS_{\text{Block}} \\ &= 69.4375 - 33.1875 - 32.6875 \\ &= 3.5625. \end{aligned}$$

To produce the ANOVA results shown in Table 11.13 on the following page with R, enter

```
> mod.aov <- aov(wear ~ treat + block, data = TIREWEAR)
> summary(mod.aov) # ANOVA

      Df Sum Sq Mean Sq F value    Pr(>F)
treat      3 33.19  11.062   27.95 6.82e-05 ***
block      3 32.69  10.896   27.53 7.25e-05 ***
Residuals  9   3.56   0.396
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 11.13: Tire wear ANOVA table

Source	<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	φ -value
Treatments	3	33.1875	11.062	27.947	$1e - 04$
Blocks	3	32.6875	10.896		
Error	9	3.5625	0.396		
Total	15	69.4375			

Note that the computer treats the blocking factor as if it were assigned at random and computes a φ -value for the blocking factor. The small φ -value suggests that blocking is appropriate. Since the φ -value = $1e - 04$, the null hypothesis ($H_0 : \tau_i = 0$) of no treatment effect is rejected.

(c) The Y_{ij} values can be decomposed as follows:

$$Y_{ij} = \hat{\mu} + \hat{\tau}_i + \hat{\beta}_j + \hat{\varepsilon}_{ij}$$

$$\begin{bmatrix} 10 & 8 & 7 & 7 \\ 9 & 8 & 7 & 5 \\ 8 & 7 & 5 & 3 \\ 6 & 5 & 3 & 3 \end{bmatrix} = \begin{bmatrix} 6.3125 & 6.3125 & 6.3125 & 6.3125 \\ 6.3125 & 6.3125 & 6.3125 & 6.3125 \\ 6.3125 & 6.3125 & 6.3125 & 6.3125 \\ 6.3125 & 6.3125 & 6.3125 & 6.3125 \end{bmatrix}$$

$$+ \begin{bmatrix} 1.6875 & 1.6875 & 1.6875 & 1.6875 \\ 0.9375 & 0.9375 & 0.9375 & 0.9375 \\ -0.5625 & -0.5625 & -0.5625 & -0.5625 \\ -2.0625 & -2.0625 & -2.0625 & -2.0625 \end{bmatrix}$$

$$+ \begin{bmatrix} 1.9375 & 0.6875 & -0.8125 & -1.8125 \\ 1.9375 & 0.6875 & -0.8125 & -1.8125 \\ 1.9375 & 0.6875 & -0.8125 & -1.8125 \\ 1.9375 & 0.6875 & -0.8125 & -1.8125 \end{bmatrix}$$

$$+ \begin{bmatrix} 0.0625 & -0.6875 & -0.1875 & 0.8125 \\ -0.1875 & 0.0625 & 0.5625 & -0.4375 \\ 0.3125 & 0.5625 & 0.0625 & -0.9375 \\ -0.1875 & 0.0625 & -0.4375 & 0.5625 \end{bmatrix}.$$

R Code 11.32 creates the four parts of each Y_{ij} .

R Code 11.32

```
> yidotbar <- tapply(TIREWEAR$wear, TIREWEAR$treat, mean)
> ydotjbar <- tapply(TIREWEAR$wear, TIREWEAR$block, mean)
> ydotdotbar <- mean(TIREWEAR$wear)
> tauihat <- yidotbar - ydotdotbar
> betajhat <- ydotjbar - ydotdotbar
> eijhat <- resid(mod.aov)
> muhatmat <- matrix(rep(ydotdotbar, 16), nrow = 4)
> muhatmat

 [,1]   [,2]   [,3]   [,4]
[1,] 6.3125 6.3125 6.3125 6.3125
[2,] 6.3125 6.3125 6.3125 6.3125
```

```
[3,] 6.3125 6.3125 6.3125 6.3125
[4,] 6.3125 6.3125 6.3125 6.3125

> treatmat <- matrix(rep(tauihat, 4), nrow = 4, byrow = FALSE)
> treatmat

[,1]      [,2]      [,3]      [,4]
[1,] 1.6875  1.6875  1.6875  1.6875
[2,] 0.9375  0.9375  0.9375  0.9375
[3,] -0.5625 -0.5625 -0.5625 -0.5625
[4,] -2.0625 -2.0625 -2.0625 -2.0625

> blockmat <- matrix(rep(betajhat, 4), nrow = 4, byrow = TRUE)
> blockmat

[,1]      [,2]      [,3]      [,4]
[1,] 1.9375  0.6875 -0.8125 -1.8125
[2,] 1.9375  0.6875 -0.8125 -1.8125
[3,] 1.9375  0.6875 -0.8125 -1.8125
[4,] 1.9375  0.6875 -0.8125 -1.8125

> residmat <- matrix(eijhat, nrow = 4, byrow = FALSE)
> residmat

[,1]      [,2]      [,3]      [,4]
[1,] 0.0625 -0.6875 -0.1875  0.8125
[2,] -0.1875  0.0625  0.5625 -0.4375
[3,] 0.3125  0.5625  0.0625 -0.9375
[4,] -0.1875  0.0625 -0.4375  0.5625

> yijmat <- muhatmat + treatmat + blockmat + residmat
> yijmat

[,1] [,2] [,3] [,4]
[1,] 10   8   7   7
[2,] 9   8   7   5
[3,] 8   7   5   3
[4,] 6   5   3   3
```

The values used in the matrices can also be obtained from using the function `proj()` (`proj(mod.aov)`).

(d) R Code 11.33 reorders the values in `TIREWEAR` according to the order the tires were assigned to a car and tested based on the values in Table 11.11 on page 752.

R Code 11.33

```
> ORD <- c(4, 1, 3, 2, 5, 6, 7, 8, 9, 10, 12, 11, 15, 13, 16, 14)
> TIREWEAR <- TIREWEAR[order(ORD),]
> head(TIREWEAR)

  wear treat block
2    9     B  Car1
4    6     D  Car1
```

```

3   8     C  Car1
1   10    A  Car1
5   8     A  Car2
6   8     B  Car2

```

```
> mod.aov0 <- aov(wear ~ treat + block, data = TIREWEARO)
```

The residuals from the model `mod.aov0` are graphed in Figure 11.20 with the function `checking.plots()` from the PASWR2 package. The top left graph in Figure 11.20 suggests that there is no problem with the independence of errors assumption. The top right and bottom right graphs in Figure 11.20 suggest the errors follow a normal distribution, while the bottom left graph suggests homogeneity of variance is reasonable.

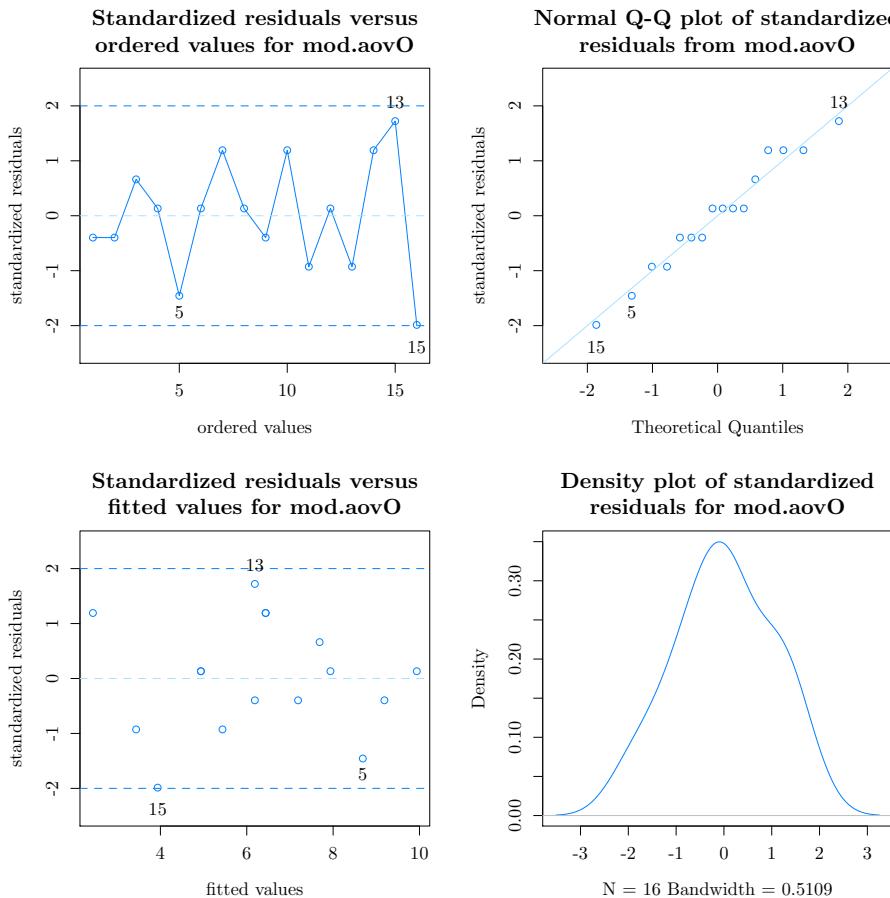


FIGURE 11.20: `checking.plots()` applied to `mod.aov0` from Example 11.7

- (e) R Code 11.34 on the facing page creates simultaneous 95% mean pairwise confidence intervals using Tukey's HSD. The confidence intervals are depicted in Figure 11.21 on the next page.

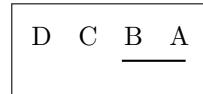
R Code 11.34

```
> CI <- TukeyHSD(mod.aov, which = "treat")
> CI

Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = wear ~ treat + block, data = TIREWEAR)

$treat
    diff      lwr       upr     p adj
B-A -0.75 -2.13882  0.6388204 0.3838264
C-A -2.25 -3.63882 -0.8611796 0.0031175
D-A -3.75 -5.13882 -2.3611796 0.0000699
C-B -1.50 -2.88882 -0.1111796 0.0343452
D-B -3.00 -4.38882 -1.6111796 0.0003981
D-C -1.50 -2.88882 -0.1111796 0.0343452
```



Tire D is significantly better (less wear) than tires C, B, and A. Tire C is significantly better than tires B and A, and tires B and A are not significantly different from one another. Figure 11.22 on the following page shows a barplot of the mean wear by tire with superimposed individual 95% confidence intervals.

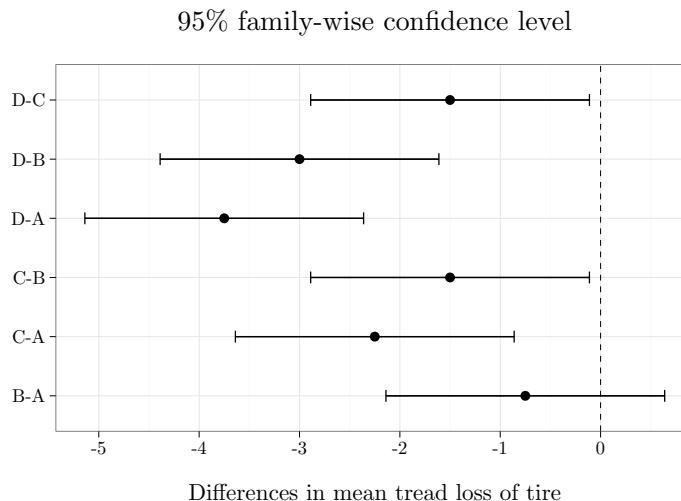


FIGURE 11.21: Simultaneous 95% mean pairwise confidence intervals using Tukey's HSD from Example 11.7

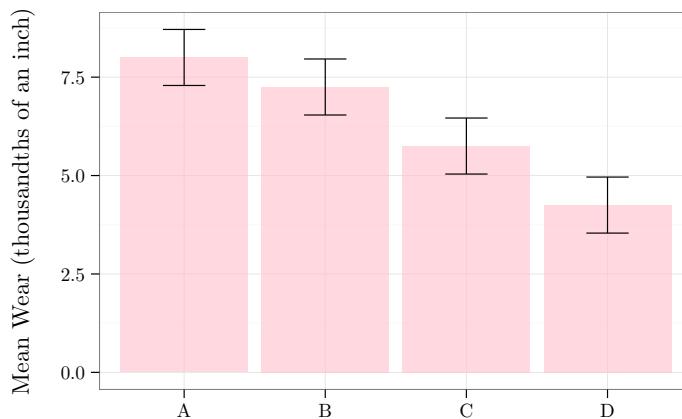


FIGURE 11.22: Barplot of the mean wear by tire with superimposed individual 95% confidence intervals from Example 11.7

11.12 Two-Factor Factorial Design

The one-way analysis of variance evaluated a single factor that had a levels. When a study involves more than one factor, say two fixed factors A and B , with a and b levels, respectively, there are a total of $a \times b = ab$ treatment combinations that need to be analyzed. An efficient method to analyze the ab treatments is with a factorial design. Such a design supplies information about all of the factors in a more efficient fashion than one-factor-at-a-time experiments and avoids potentially misleading conclusions that are possible from single-factor designs when interactions are present.

A factorial design with two fixed factors A and B , each with a and b levels, respectively, will typically have n experimental units for each of the ab treatment combinations. The ab treatments are randomly assigned to the $N = abn$ experimental units resulting in a completely randomized design. A general layout for observations from a two-factor factorial design is presented in Table 11.14.

Table 11.14: Layout for observations in a two-factor factorial design

		Factor B				
		1	2	...	b	
Factor A	1	$Y_{111}, Y_{112}, \dots, Y_{11n}$	$Y_{121}, Y_{122}, \dots, Y_{12n}$...	$Y_{1b1}, Y_{1b2}, \dots, Y_{1bn}$	$\bar{Y}_{1\bullet\bullet}$
	2	$Y_{211}, Y_{212}, \dots, Y_{21n}$	$Y_{221}, Y_{222}, \dots, Y_{22n}$...	$Y_{2b1}, Y_{2b2}, \dots, Y_{2bn}$	$\bar{Y}_{2\bullet\bullet}$
	:	:	:		:	:
	a	$Y_{a11}, Y_{a12}, \dots, Y_{a1n}$	$Y_{a21}, Y_{a22}, \dots, Y_{a2n}$...	$Y_{ab1}, Y_{ab2}, \dots, Y_{abn}$	$\bar{Y}_{a\bullet\bullet}$
		$\bar{Y}_{\bullet 1\bullet}$	$\bar{Y}_{\bullet 2\bullet}$...	$\bar{Y}_{\bullet b\bullet}$	$\bar{Y}_{\bullet\bullet\bullet}$

The observations from a two-factor factorial design are described by the linear model

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \varepsilon_{ijk} \quad \text{for } i = 1, \dots, a; j = 1, \dots, b; k = 1, \dots, n \quad (11.42)$$

where μ is the overall mean effect, α_i is the effect of the i^{th} row factor A , β_j is the effect of the j^{th} column factor B , $\alpha\beta_{ij}$ is the effect of the interaction between α_i and β_j , and ε_{ijk} is a random error. Note that $\alpha\beta$ is not $\alpha \cdot \beta$ but rather a single term. Both α_i and β_j are assumed to be fixed with the constraints

$$\sum_{i=1}^a \alpha_i = 0; \quad \sum_{j=1}^b \beta_j = 0; \quad \sum_{i=1}^a \alpha\beta_{ij} = \sum_{j=1}^b \alpha\beta_{ij} = 0. \quad (11.43)$$

That is, the treatment effects are defined as deviations from the overall mean. Given these assumptions, the least squares estimators for the parameters in the two-factor factorial design are

$$\begin{aligned}\hat{\alpha}_i &= \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet} & \hat{\beta}_j &= \bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet} \\ \hat{\alpha}\beta_{ij} &= \bar{Y}_{ij\bullet} - \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet j\bullet} + \bar{Y}_{\bullet\bullet\bullet} & \hat{\varepsilon}_{ijk} &= Y_{ijk} - \bar{Y}_{ij\bullet}\end{aligned}$$

Sums of Squares Each Y_{ijk} from (11.42) can be decomposed into five parts by substituting the least squares estimates of μ , α_i , β_j , $\alpha\beta_{ij}$, and ε_{ijk} for the parameters' values:

$$Y_{ijk} = \hat{\mu} + \hat{\alpha}_i + \hat{\beta}_j + \hat{\alpha}\beta_{ij} + \hat{\varepsilon}_{ijk}$$

$$Y_{ijk} = \bar{Y}_{\bullet\bullet\bullet} + (\bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet}) + (\bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet}) + (\bar{Y}_{ij\bullet} - \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet j\bullet} + \bar{Y}_{\bullet\bullet\bullet}) + (Y_{ijk} - \bar{Y}_{ij\bullet})$$

which implies that

$$(Y_{ijk} - \bar{Y}_{\bullet\bullet\bullet}) = (\bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet}) + (\bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet}) + (\bar{Y}_{ij\bullet} - \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet j\bullet} + \bar{Y}_{\bullet\bullet\bullet}) + (Y_{ijk} - \bar{Y}_{ij\bullet}). \quad (11.44)$$

Squaring (11.44) and summing over $i = 1, \dots, a$; $j = 1, \dots, b$; and $k = 1, \dots, n$ gives

$$\begin{aligned}\sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n (Y_{ijk} - \bar{Y}_{\bullet\bullet\bullet})^2 &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n [(\bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet}) \\ &\quad + (\bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet}) + (\bar{Y}_{ij\bullet} - \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet j\bullet} + \bar{Y}_{\bullet\bullet\bullet}) + (Y_{ijk} - \bar{Y}_{ij\bullet})]^2.\end{aligned} \quad (11.45)$$

When the right side of (11.45) is expanded, all four cross products sum to zero (which is left to the reader to verify), giving

$$\begin{aligned}\underbrace{\sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n (Y_{ijk} - \bar{Y}_{\bullet\bullet\bullet})^2}_{SS_{\text{Total}}} &= \underbrace{bn \sum_{i=1}^a (\bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet})^2}_{SS_A} + \underbrace{an \sum_{j=1}^b (\bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet})^2}_{SS_B} \\ &\quad + \underbrace{n \sum_{i=1}^a \sum_{j=1}^b (\bar{Y}_{ij\bullet} - \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet j\bullet} + \bar{Y}_{\bullet\bullet\bullet})^2}_{SS_{AB}} + \underbrace{\sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n (Y_{ijk} - \bar{Y}_{ij\bullet})^2}_{SS_{\text{Error}}}.\end{aligned} \quad (11.46)$$

That is,

$$SS_{\text{Total}} = SS_A + SS_B + SS_{AB} + SS_{\text{Error}}. \quad (11.47)$$

The corresponding degrees of freedom are

$$\underbrace{abn - 1}_{\text{total } df} = \underbrace{a - 1}_{A \text{ df}} + \underbrace{b - 1}_{B \text{ df}} + \underbrace{(a - 1)(b - 1)}_{AB \text{ interaction df}} + \underbrace{ab(n - 1)}_{\text{Error df}}. \quad (11.48)$$

The mean squares are computed by dividing each sum of squares by its degrees of freedom. The expected value of the mean squares, with fixed factors A and B , can be shown to be

$$\begin{aligned} E(MS_A) &= \sigma^2 + \frac{bn \sum_{i=1}^a \alpha_i^2}{a-1} & E(MS_B) &= \sigma^2 + \frac{an \sum_{j=1}^b \beta_j^2}{b-1} \\ E(MS_{AB}) &= \sigma^2 + \frac{n \sum_{i=1}^a \sum_{j=1}^b \alpha_i \beta_j^2}{(a-1)(b-1)} & E(MS_{\text{Error}}) &= \sigma^2 \end{aligned}$$

Consequently, to test for A and B main effects as well as the interaction between A and B , the corresponding mean square is divided by the MS_{Error} . The ANOVA table for a two-factor design is given in Table 11.15. The formal hypotheses for testing for factor A

Table 11.15: ANOVA table for two-factor factorial design

Source	df	SS	MS	F
A	$a-1$	$SS_A = bn \sum_{i=1}^a (\bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet})^2$	$MS_A = \frac{SS_A}{a-1}$	$\frac{MS_A}{MS_{\text{Error}}}$
B	$b-1$	$SS_B = an \sum_{j=1}^b (\bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet})^2$	$MS_B = \frac{SS_B}{b-1}$	$\frac{MS_B}{MS_{\text{Error}}}$
AB	$(a-1)(b-1)$	$SS_{AB} = n \sum_{i=1}^a \sum_{j=1}^b (\bar{Y}_{ij\bullet} - \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet j\bullet} + \bar{Y}_{\bullet\bullet\bullet})^2$	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$\frac{MS_{AB}}{MS_{\text{Error}}}$
Error	$ab(n-1)$	$SS_{\text{Error}} = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n (Y_{ijk} - \bar{Y}_{ij\bullet})^2$	$MS_{\text{Error}} = \frac{SS_{\text{Error}}}{ab(n-1)}$	
Total	$abn-1$	$SS_{\text{Total}} = \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n (Y_{ijk} - \bar{Y}_{\bullet\bullet\bullet})^2$		

treatment effects, factor B treatment effects, and the interaction between factor A and factor B are written, respectively, as

$$\begin{array}{lll} \text{Factor } A & \text{Factor } B & \text{Interaction} \\ H_0 : \alpha_i = 0 & \text{for all } i & H_0 : \beta_j = 0 \quad \text{for all } j \quad H_0 : \alpha_i \beta_j = 0 \quad \text{for all } (i, j) \\ H_1 : \alpha_i \neq 0 & \text{for some } i & H_1 : \beta_j \neq 0 \quad \text{for some } j \quad H_1 : \alpha_i \beta_j \neq 0 \quad \text{for some } (i, j) \end{array}$$

Example 11.8 ▷ **Television Tube Screen Brightness** ◷ The data in Table 11.16 on the next page are taken from Hicks (1956), where an experiment was designed to study the effect of glass type and phosphor type on the brightness of a television tube screen. The measured variable was the current in microamperes (μA) necessary to produce a certain level of brightness. The higher the μA required to produce a given brightness, the poorer are the tube screen characteristics. That is, optimal characteristics are obtained when the response (μA) is small. Analyze the data using a two-factor factorial design.

Table 11.16: Data from Hicks (1956) used in Example 11.8

		Phosphor		
		A	B	C
Glass	I	280, 290, 285	300, 310, 295	270, 285, 290
	II	230, 235, 240	260, 240, 235	220, 225, 230

- (a) Store the data in Table 11.16 in a data frame named **DF**.
- (b) Graphically examine the data in Table 11.16. Create interaction plots as well as a barplot of the individual means for the six treatment combinations.
- (c) Fill in the missing values to complete Table 11.17.
- (d) Create a two-way ANOVA table using the information from Table 11.17, and verify your answers using the function **anova()**.
- (e) Analyze the residuals, and discuss whether the model from (11.42) fits the data.
- (f) Is there significant interaction between glass type and phosphor type?
- (g) Using $\alpha_e = 0.05$, compute Tukey's HSD 95% confidence intervals to determine which glass type and which phosphor type require the fewest μA .

Table 11.17: Two-factor factorial design table to complete for (c) of Example 11.8

		Phosphor A	Phosphor B	Phosphor C	$\bar{Y}_{i\bullet\bullet}$	$\hat{\alpha}_i =$ $\bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet}$
Glass I	$\bar{Y}_{11\bullet} = \boxed{}$	$\bar{Y}_{12\bullet} = \boxed{}$	$\bar{Y}_{13\bullet} = \boxed{}$	$\bar{Y}_{1\bullet\bullet} =$ $\boxed{}$	$\hat{\alpha}_1 = \boxed{}$	
	$\hat{\beta}_{11} = \boxed{}$	$\hat{\beta}_{12} = \boxed{}$	$\hat{\beta}_{13} = \boxed{}$			
Glass II	$\bar{Y}_{21\bullet} = \boxed{}$	$\bar{Y}_{22\bullet} = \boxed{}$	$\bar{Y}_{23\bullet} = \boxed{}$	$\bar{Y}_{2\bullet\bullet} =$ $\boxed{}$	$\hat{\alpha}_2 = \boxed{}$	
	$\hat{\beta}_{21} = \boxed{}$	$\hat{\beta}_{22} = \boxed{}$	$\hat{\beta}_{23} = \boxed{}$			
$\bar{Y}_{\bullet j\bullet}$		$\bar{Y}_{\bullet 1\bullet} = \boxed{}$	$\bar{Y}_{\bullet 2\bullet} = \boxed{}$	$\bar{Y}_{\bullet 3\bullet} = \boxed{}$		
$\hat{\beta}_j = \bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet}$		$\hat{\beta}_1 = \boxed{}$	$\hat{\beta}_2 = \boxed{}$	$\hat{\beta}_3 = \boxed{}$		$\bar{Y}_{\bullet\bullet\bullet} = \boxed{}$

Solution: The answers are as follows:

- (a) R Code 11.35 on the following page creates the requested data frame by first storing the values from Table 11.16 in the variable **Microamps** and creating the factors **Glass** and **Phosphor**. The data frame **DF** is subsequently created with the **data.frame()** function.

R Code 11.35

```
> Microamps <- c( 280, 290, 285, 300, 310, 295, 270, 285, 290, 230,
+                  235, 240, 260, 240, 235, 220, 225, 230)
> Glass <- factor(c(rep("Glass I", 9), rep("Glass II", 9)))
> Phosphor <- factor(rep(c(rep("Phosphor A", 3),
+                           rep("Phosphor B", 3),
+                           rep("Phosphor C", 3)), 2))
> DF <- data.frame(Microamps, Glass, Phosphor)
> rm(Microamps ,Glass, Phosphor) # Clean up workspace
```

(b) The function `twoway.plots()` is used to examine the data, and the results are shown in Figure 11.23. From Figure 11.23, glass type appears important, and the lines in the interaction plot are nearly parallel, suggesting interaction between the two factors is not significant.

```
> with(data = DF,
+       twoway.plots(Microamps, Glass, Phosphor)
+     )
```

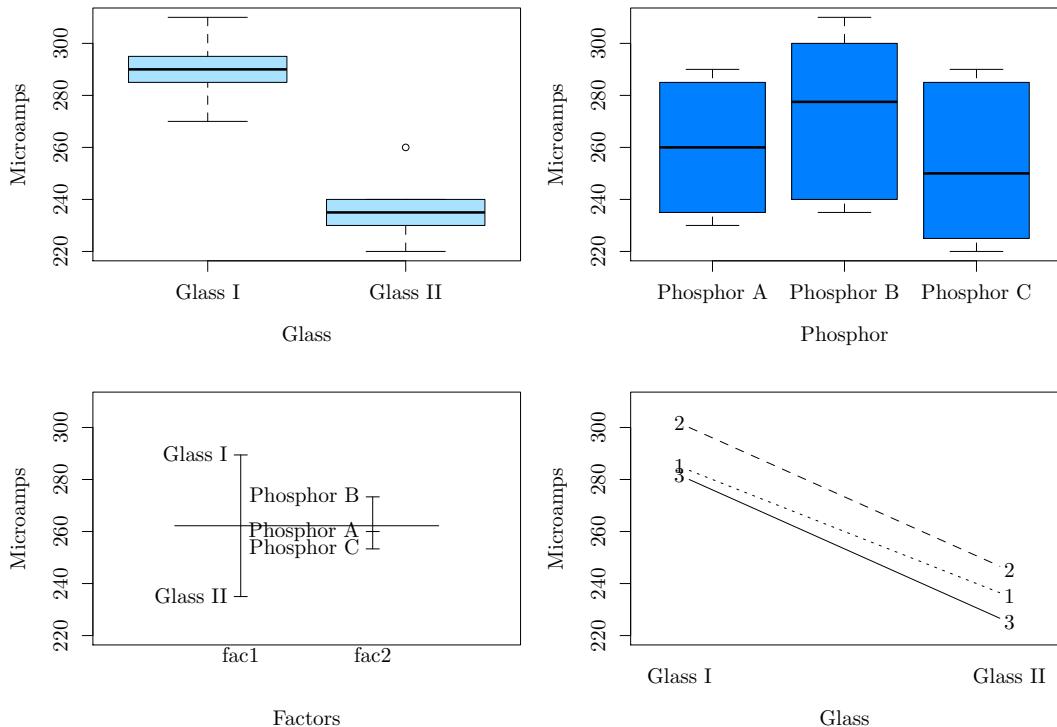


FIGURE 11.23: Graphs from using `twoway.plots()` for Example 11.8

To assess possible interaction between the factors `Glass` and `Phosphor`, two interaction plots of the same data are created with the R Code 11.36 on the next page and shown

in Figure 11.24. Since the lines in Figure 11.24 are roughly parallel in both plots, it is reasonable to assume the two factors, **Glass** and **Phosphor**, do not interact.

R Code 11.36

```
> p1 <- ggplot(data = DF, aes(x = Glass, y = Microamps,
+                               colour = Phosphor, group = Phosphor) +
+   stat_summary(fun.y = mean, geom = "point") +
+   stat_summary(fun.y = mean, geom = "line") +
+   theme_bw())
> p2 <- ggplot(data = DF, aes(x = Phosphor, y = Microamps, colour = Glass,
+                               group = Glass, linetype = Glass)) +
+   stat_summary(fun.y = mean, geom = "point") +
+   stat_summary(fun.y = mean, geom = "line") +
+   theme_bw()
> library(gridExtra)
> grid.arrange(p1, p2, nrow = 2)
```

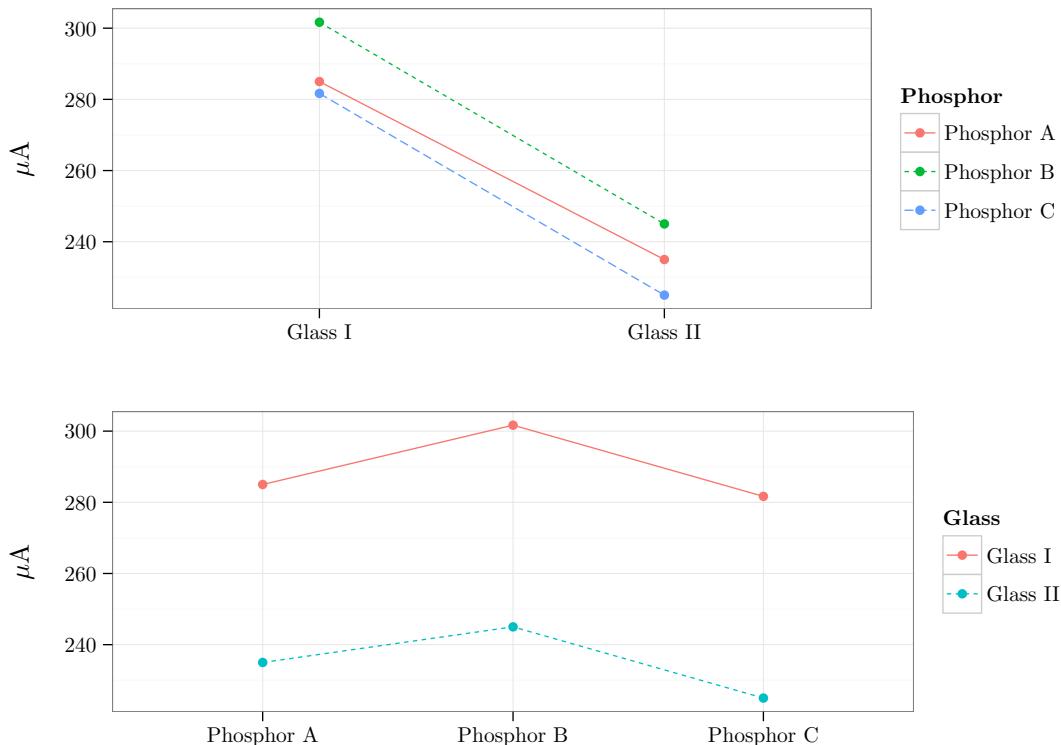


FIGURE 11.24: Top graph shows an interaction plot of **Glass** and **Phosphor** for the response μA . The bottom graph shows an interaction plot of **Phosphor** and **Glass** for the response μA .

R Code 11.37 on the next page can be modified to create a barplot showing the means for each combination of **Glass** and **Phosphor** similar to Figure 11.25 on the following page.

R Code 11.37

```
> library(plyr) # loaded for function ddply
> mdf <- ddply(DF, c("Glass", "Phosphor"), summarize,
+                 mmicro = mean(Microamps))
> mdf # Mean micoramps for 6 combinations as data frame.
> p <- ggplot(data = mdf, aes(x = Phosphor, y = mmicro, fill = Glass)) +
+   geom_bar(position = "dodge", stat = "identity") +
+   scale_fill_grey(start = 0.4, end = 0.8) +
+   labs(x = "") +
+   labs(x = "", y = "Microamps") +
+   theme_bw()
```

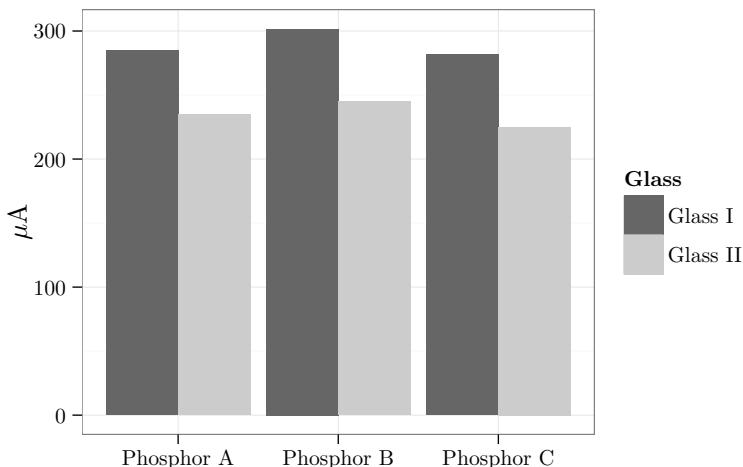


FIGURE 11.25: Barplot of means for each level of Glass and Phosphor

- (c) The values to complete Table 11.17 on page 763 are computed using the function `model.tables()` in R Code 11.38, and the completed table is shown in Table 11.18 on the next page.

R Code 11.38

```
> mod1.TVB <- aov(Microamps ~ Glass + Phosphor + Glass:Phosphor,
+                   data = DF)
> model.tables(mod1.TVB, type = "means")
```

Tables of means
Grand mean

262.2222

Glass
Glass
Glass I Glass II

```

289.44    235.00

Phosphor
Phosphor
Phosphor A Phosphor B Phosphor C
  260.00    273.33    253.33

Glass:Phosphor
    Phosphor
Glass   Phosphor A Phosphor B Phosphor C
Glass I 285.00    301.67    281.67
Glass II 235.00    245.00    225.00

> model.tables(mod1.TVB, type = "effects")

Tables of effects

Glass
Glass
Glass I Glass II
  27.222 -27.222

Phosphor
Phosphor
Phosphor A Phosphor B Phosphor C
  -2.222    11.111   -8.889

Glass:Phosphor
    Phosphor
Glass   Phosphor A Phosphor B Phosphor C
Glass I -2.2222    1.1111    1.1111
Glass II  2.2222   -1.1111   -1.1111

```

Table 11.18: Two-factor factorial design table COMPLETED for (c) of Example 11.8

	Phosphor A	Phosphor B	Phosphor C	$\bar{Y}_{i\bullet\bullet}$	$\hat{\alpha}_i = \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet}$
Glass I	$\bar{Y}_{11\bullet} = 285$ $\hat{\beta}_{11} = -2.2222$	$\bar{Y}_{12\bullet} = 301.67$ $\hat{\beta}_{12} = 1.1111$	$\bar{Y}_{13\bullet} = 281.67$ $\hat{\beta}_{13} = 1.1111$	$\bar{Y}_{1\bullet\bullet} = 289.44$	$\hat{\alpha}_1 = 27.222$
Glass II	$\bar{Y}_{21\bullet} = 235$ $\hat{\beta}_{21} = 2.2222$	$\bar{Y}_{22\bullet} = 245$ $\hat{\beta}_{22} = -1.1111$	$\bar{Y}_{23\bullet} = 225$ $\hat{\beta}_{23} = -1.1111$	$\bar{Y}_{2\bullet\bullet} = 235$	$\hat{\alpha}_2 = -27.222$
$\bar{Y}_{\bullet j\bullet}$	$\bar{Y}_{\bullet 1\bullet} = 260$	$\bar{Y}_{\bullet 2\bullet} = 273.33$	$\bar{Y}_{\bullet 3\bullet} = 253.33$		
$\hat{\beta}_j = \bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet}$	$\hat{\beta}_1 = -2.2222$	$\hat{\beta}_2 = 11.1111$	$\hat{\beta}_3 = -8.889$		$\bar{Y}_{\bullet\bullet\bullet} = 262.22$

(d) Using the results from (c), the sums of squares for the ANOVA table are computed and displayed in Table 11.19.

$$\begin{aligned} SS_A &= bn \sum_{i=1}^a (\bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet\bullet\bullet})^2 = bn \sum_{i=1}^a \hat{\alpha}_i^2 \\ &= 3 \cdot 3 \cdot [27.222^2 + (-27.222)^2] = 13338.9. \end{aligned}$$

$$\begin{aligned} SS_B &= an \sum_{j=1}^b (\bar{Y}_{\bullet j\bullet} - \bar{Y}_{\bullet\bullet\bullet})^2 = an \sum_{j=1}^b \hat{\beta}_j^2 \\ &= 2 \cdot 3 \cdot [(-2.2222)^2 + (11.1111)^2 + (-8.889)^2] = 1244.4. \end{aligned}$$

$$\begin{aligned} SS_{AB} &= n \sum_{i=1}^a \sum_{j=1}^b (\bar{Y}_{ij\bullet} - \bar{Y}_{i\bullet\bullet} - \bar{Y}_{\bullet j\bullet} + \bar{Y}_{\bullet\bullet\bullet})^2 = n \sum_{i=1}^a \sum_{j=1}^b \alpha \beta_{ij}^2 \\ &= 3 \cdot [(-2.2222)^2 + (1.1111)^2 + \dots + (-1.1111)^2] = 44.4. \end{aligned}$$

$$\begin{aligned} SS_{\text{Error}} &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n (Y_{ijk} - \bar{Y}_{ij\bullet})^2 \\ &= [(280 - 285)^2 + (290 - 285)^2 + \dots + (230 - 225)^2] = 833.3. \end{aligned}$$

$$\begin{aligned} SS_{\text{Total}} &= \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n (Y_{ijk} - \bar{Y}_{\bullet\bullet\bullet})^2 \\ &= [(280 - 262.2222)^2 + (290 - 262.2222)^2 + \dots + (230 - 262.2222)^2] = 15461.11. \end{aligned}$$

Table 11.19: ANOVA table for two-factor factorial design for Example 11.8

Source	df	SS	MS	F
Glass	1	13338.9	13338.9	192.08
Phosphor	2	1244.4	622.2	8.96
Glass:Phosphor	2	44.4	22.2	0.32
Residuals	12	833.3	69.4	
Total	17	15461.0		

The values for Table 11.19 are verified with the function `anova()` in R Code 11.39.

R Code 11.39

```
> anova(mod1.TVB)
```

```
Analysis of Variance Table
```

Response: Microamps

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Glass	1	13338.9	13338.9	192.08	9.568e-09 ***
Phosphor	2	1244.4	622.2	8.96	0.004162 **
Glass:Phosphor	2	44.4	22.2	0.32	0.732158
Residuals	12	833.3	69.4		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(e) The residuals from fitting the data to model (11.42) are analyzed using the function `checking.plots()` and shown in Figure 11.26. The first graph in Figure 11.26 is not relevant because no time component is present in the data; the second graph suggests normality is reasonable; and the third graph indicates homogeneity of variance is plausible. R Code 11.40 on the following page uses the `leveneTest()` function from the package `car` to test the assumption of homogeneity of variance. Based on a p -value of 0.8601, there is little evidence to suggest a lack of homogeneity with respect to the variances. Consequently, a two-factor factorial model seems to be a reasonable model for the data on hand.

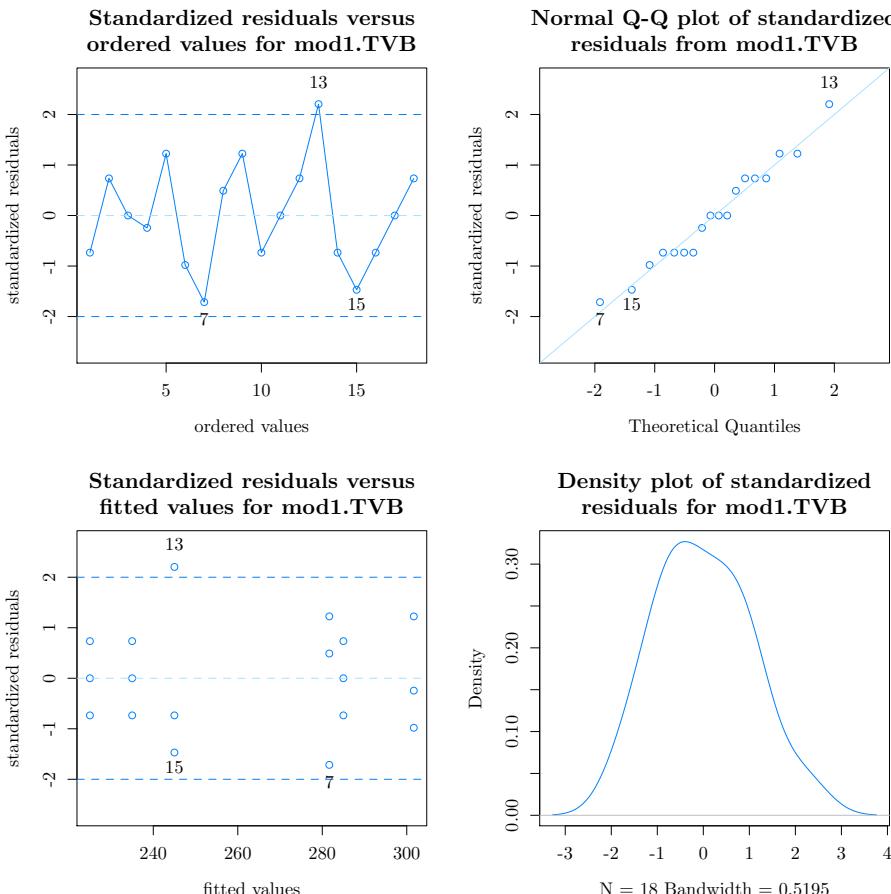


FIGURE 11.26: Graphs resulting from using `checking.plots()` on the model `mod1.TVB` from Example 11.8

R Code 11.40

```
> library(car)
> leveneTest(Microamps ~ Glass*Phosphor, data = DF)

Levene's Test for Homogeneity of Variance (center = median)
  Df F value Pr(>F)
group  5  0.3692 0.8601
      12
```

(f) The graphical interpretation of no interaction is also supported by the ϕ -value = 0.7322 for the interaction term in R Code 11.39 on page 768.

(g) Tukey 95% confidence intervals for pairwise differences are computed for the factors **Phosphor** and **Glass** using the R function **TukeyHSD()**. R Code 11.41 computes the intervals. Based on the Tukey HSD confidence intervals, **Glass II** is better (fewer μA) than **Glass I**. **Phosphor C** is better (fewer μA) than **Phosphor B** but not statistically different from **Phosphor A**.

R Code 11.41

```
> mod1.TVB <- aov(Microamps ~ Glass * Phosphor, data = DF)
> CIs <- TukeyHSD(mod1.TVB, which = c("Phosphor", "Glass"))
> CIs

Tukey multiple comparisons of means
  95% family-wise confidence level

Fit: aov(formula = Microamps ~ Glass * Phosphor, data = DF)

$Phosphor
            diff      lwr      upr    p adj
Phosphor B-Phosphor A 13.333333  0.4975684 26.169098 0.0416601
Phosphor C-Phosphor A -6.666667 -19.5024316  6.169098 0.3785691
Phosphor C-Phosphor B -20.000000 -32.8357650 -7.164235 0.0035163

$Glass
            diff      lwr      upr p adj
Glass II-Glass I -54.44444 -63.00363 -45.88526     0
```

R Code 11.42, not run, can be used to graph the confidence intervals from R Code 11.41.

R Code 11.42

```
> opar <- par(no.readonly = TRUE)
> par(mar=c(4.1, 10.1, 5.1, 2.1), cex.axis = 0.8)
> plot(TukeyHSD(mod1.TVB, which = c("Glass")), las = 1)
> plot(TukeyHSD(mod1.TVB, which = c("Phosphor")), las = 1)
> par(opar)
```

11.13 Problems

1. Develop a randomization scheme to assign three treatments A, B, and C to 15 experimental units, numbered from 1 to 15. Use the command `sample` to assign them.
2. Develop a randomization scheme for a complete block design that has 4 blocks, 3 treatments, and 12 experimental units.
3. Provide a randomized assignment for a two-factor factorial design with 36 experimental units, 4 levels for the first factor, 3 levels for the second factor, and 3 experimental units for every combination of factor levels.
4. An economic study in a particular city desires to discover the monthly expenses of consumers, based on their level of education. The survey has drawn data in three different boroughs: I, II, and III. The educational levels corresponds to low, medium low, medium high, and high. The expenses have been recorded in thousands of dollars, and the analysis of variance provides the following information:

	Df	Sum of Sq	Mean Sq	F Value	Pr(F)
Education_level	-	0.32	---	---	---
Boroughs	-	26.69	---	---	---
Residuals	-	3.64	---		

- (a) Fill in the table, and write the model corresponding to the ANOVA output.
- (b) Calculate the percentage of the total variability explained by the educational level.
- (c) What is the percentage of the total variability explained by the boroughs?
- (d) What is the value for the residual variance of this model?
- (e) Is the factor `Education_level` significant?

5. Given the following partial ANOVA information from a randomized complete block design:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
block	4	0.00073	--	--	--
factor	3	--	0.35431	--	--
Residuals	--	0.01703	0.00142		

- (a) Fill in the missing values, and give the corresponding model.
- (b) How many levels does the factor have?
- (c) How many blocks are there in the design?
- (d) Explain the meaning of the model's parameters.
- (e) Describe a scenario that could be described by this particular table.

6. An agricultural engineer wants to know what type of barley produces the greatest yield: A (ASPEN), B (ERIKA), or C (SULTANE). The results obtained from 12 experimental plots, given in tons per hectare, are displayed in the following table:

A (ASPEN)	3	2	4	3
B (ERIKA)	2	3	4	4
C (SULTANE)	7	6	5	6

Assume the treatments (types of barley) were assigned at random to the 12 plots.

- (a) Write the statistical model.
- (b) Conduct an analysis of variance and explain the results based on the model for part (a).
- (c) Write the fitted model in matrix form and provide an estimate of the error variance.
- (d) Are the assumptions satisfied for the model in part (a)?
- (e) Construct 95% confidence intervals for the pairwise mean differences using a technique that controls experiment-wise error.
- (f) Use orthogonal contrasts to assess whether differences exist between SULTANE and the other two varieties ASPEN and ERIKA.

7. As described in *Basic Statistics and Data Analysis*, *Car and Driver* (July 1995) conducted tests of five cars from five different countries: Japan's Acura NSXT, Italy's Ferrari F355, Great Britain's Lotus Esprit S4S, Germany's Porsche 911 Turbo, and the United States' Dodge Viper RT/10. The maximum speeds the cars obtained in miles per hour using as much distance as necessary without exceeding the engine's redline are given:

Acura	Ferrari	Lotus	Porsche	Viper
159.7	179.6	167.4	173.5	172.3
161.5	173.9	163.0	182.4	168.9
163.7	180.2	160.3	171.3	169.5
166.0	183.9	164.9	175.7	174.6
157.7	176.7	160.5	179.1	161.1
161.7	178.4	158.3	175.0	164.2

Data from Kitchens (2003, page 512).

- (a) What statistical model should be used to analyze this experiment?
- (b) Conduct an analysis of variance to investigate if differences exist among the maximum speeds of the cars.
- (c) Use appropriate diagnostic measures to check the adequacy of the model from part (a).
- (d) What is the mean squared error value for the model from part (a)?
- (e) Use Tukey's multiple comparison test to determine which of the cars are different according to speed. Plot the confidence intervals for the mean differences.

8. The data frame **barley** from the **lattice** package lists barley yield in bushels per acre for the years 1931 and 1932 for ten varieties of barley grown at six sites. Is there evidence to suggest the average barley yield in 1931 for the Waseca site is different from the average barley yield in 1931 for the Duluth site?

- (a) Use a t procedure to test the appropriate hypotheses using an $\alpha = 0.05$ significance level.
- (b) Solve the same problem using a RCBD.
- (c) Generalize your findings about the relationship between (a) and (b).

9. An insurance company wants to know how its resources are being used with respect to time spent issuing travel insurance policies. The company randomly selects three moments during a day and records the time required to issue a travel insurance policy to three randomly selected clients who take out a travel policy over the phone, over the Internet, and in person. The data obtained (in minutes) are

telephone	3.49	2.38	2.09
Internet	4.38	6.68	5.37
in person	7.91	8.70	8.54

- (a) What type of design structure did the company use?
- (b) Propose a statistical model to analyze the data.
- (c) Comment on any assumptions that need to be made with the model selected in part (a). Check these assumptions.
- (d) Test to see if differences exist among the methods used to issue insurance policies.
- (e) Estimate the model's parameters.
- (f) How is the standard deviation of the errors estimated?
- (g) Write the estimated model in matrix form.
- (h) Do the residuals sum to zero?
- (i) Use Tukey's HSD to determine if significant differences exist among methods.
- (j) Create a barplot of the mean times, and display the standard errors over their respective means.

10. A health-conscious pizza parlor is attempting to specify the added calories for each ingredient of its medium size pizza. Specifically, the pizza parlor wants to know if there is more variability in an olive topping due to olive suppliers or due to the olives themselves. From numerous suppliers, four are selected randomly and the calories for a pizza topping of olives are recorded for five randomly selected pizzas. The data obtained are given in the following table:

Supplier 1	133	136	142	135	134
Supplier 2	124	137	125	132	131
Supplier 3	127	126	130	120	123
Supplier 4	150	141	155	150	157

- (a) Specify a statistical model to analyze these data.
- (b) Conduct an ANOVA.
- (c) Estimate the variance components and the total variability of the data.
- (d) Interpret the results.

11. The following data were obtained from an experiment that investigated the effects of four bleaching chemicals (randomly selected from a large population of potential bleaching agents) on pulp brightness. The brightness of pulp is measured as the ability of a pulp sheet to reflect light directed at it. Brightness is affected by both the light absorption and light scattering of the pulp. It is usually a measure of reflectivity, and its value is expressed as a percent of some scale (a standard measurement is the General Electric Brightness (GEB), which is a measurement of directional reflectance and is expressed as a percentage of a maximum GEB value and can be obtained by following TAPPI Standard Method T-452).

Chemical 1	77.20	74.47	82.75	76.21	82.88	76.22	78.06	76.39	76.16	78.04
Chemical 2	80.52	79.31	81.91	80.35	78.38	81.84	82.78	80.90	79.18	80.62
Chemical 3	79.42	78.02	81.60	80.80	80.63	79.01	80.55	78.48	81.80	80.92
Chemical 4	78.00	78.36	77.54	77.36	77.55	75.91	78.04	78.94	77.15	77.39

Data from Dudewicz and Bishop (1981).

- (a) Create side-by-side boxplots of the four chemicals. Interpret the resulting graph.
- (b) Specify an appropriate model to test if the chemicals have an influence on pulp brightness. Conduct an analysis on the specified model using $\alpha = 0.05$.
- (c) Estimate the component of variance for the chemicals.
- (d) Estimate the total variability in the data.
- (e) Construct a confidence interval for the ratio of the variability due to chemicals with respect to the total variability given that $\frac{MS_{Treatment}/(n\sigma^2_\tau + \sigma^2)}{MS_{Error}/\sigma^2} \sim F_{a-1, N-a}$. Interpret your interval.

12. The household appliances section of a well-known store does research to satisfy the clients' demands for information about its products. In particular, clients are increasingly asking if the average washing times of the different brands of washing machines are the same. To discover this, the household appliances section has done the following experiment: They measured the washing time of five machines of different brands in four types of cycles (prewash, short, medium, long). The results, in minutes, are displayed in the following table:

Machines	Washing Cycle			
	Prewash	Short	Medium	Long
Machine 1	15.45	19.95	23.10	25.35
Machine 2	3.15	6.30	13.80	17.70
Machine 3	20.10	22.05	32.10	33.30
Machine 4	25.20	27.15	33.15	38.55
Machine 5	13.65	16.35	19.80	21.75

- (a) What is the design structure used in this experiment?
 - (b) Propose a statistical model for analyzing these data.
 - (c) Use a graph to check for interaction among machines and washing cycles.
 - (d) Use diagnostic measures to check the adequacy of the model from part (b).
 - (e) If the model from part (b) is appropriate, use it to test if the average washing times are the same for the five washing machines.
 - (f) What is an estimate of the model's error variance?
 - (g) What are the estimates of the model's parameters?
 - (h) Write the model in matrix form and check that the model's constraints are satisfied.
 - (i) Use Tukey's HSD to determine which washing machines have significantly different washing times.
 - (j) Is the mean washing time of machines 2, 3, and 4 significantly different from the mean washing time of machine 5?
 - (k) Use a barplot to show the mean washing times by machine. Superimpose 95% confidence intervals over the appropriate bars.
13. The Environmental Protection Agency (EPA) is interested in the fuel consumption of older vehicles. An experiment is designed where the gallons of gasoline consumed by vehicles over six years old are measured when the same driver travels 162.78 miles from Boone, North Carolina, to Durham, North Carolina, in 35 different vehicles. Seven vehicles are randomly selected from each category to be tested. The categories are compact, station wagon, minivan, van, and full-size pickup truck. The data obtained (gallons consumed) are given in the following table:

Compact	4.35	4.96	4.82	4.62	4.32	4.70	4.82
Station Wagon	5.47	6.35	5.33	6.25	5.44	5.73	5.64
Minivan	9.37	7.43	8.40	6.76	8.62	7.53	7.54
Van	8.61	8.66	10.12	8.06	9.31	6.75	8.14
Pickup Truck	20.09	14.93	13.38	16.53	13.79	12.44	14.73

- (a) Based on the described randomization, what type of design structure did the EPA use?
- (b) Propose a statistical model to analyze these data.
- (c) Are the assumptions for the model specified in part (b) satisfied? If the assumptions for the model specified in part (b) are not satisfied, suggest fixes before advancing to the next question.
- (d) Are there significant differences between the fuel consumption for the five types of vehicles?
- (e) Estimate the model's error variance.
- (f) What conclusions can be drawn from the data?

14. A turpentine manufacturer is interested in the most effective combination of acid treatment and tap hole shape for its upcoming pine resin collection. The company asks a local statistician to design an experiment to compare four tap hole shapes and to determine whether acid should be used to treat the holes. Twenty-four pine trees are selected at random from the forest where the sap will be harvested and assigned at random to the eight combinations of acid treatment (yes or no) and hole shape (circle, diagonal slash, check, rectangle). The response is total grams of resin collected from the hole.

	Circle	Diagonal Slash	Check	Rectangle
No Acid	9 13 12	43 48 57	60 65 70	77 70 91
Yes Acid	15 13 20	66 58 73	75 78 90	97 108 99

Data in this table comes from problem 8.5, page 201 of Oehlert (2000)

- (a) Analyze these data using a two-factor factorial design.
- (b) Looking at the results of the two-way ANOVA table, is there significant interaction between acid treatment and hole shape? Use $\alpha = 0.05$.
- (c) Create a graphical display of the interactions. Does this display corroborate the numerical results?
- (d) Analyze the residuals and comment on whether the chosen model fits the data.
- (e) Provide estimates of the parameters $\alpha_i, i = 1, 2$ and $\beta_j, j = 1, \dots, 4$.
- (f) Are the effects of acid treatment and hole shape statistically significant?
- (g) Using an experiment-wise error rate of $\alpha_e = 0.05$, what shape has the highest quantity of resin collected?

15. The data stored in **COWS** were extracted from a Canadian record book of purebred dairy cattle. Random samples of 10 mature (five-year-old and older) and 10 two-year-old cows were taken from each of five breeds. The average butterfat percentage of these 100 cows is stored in the variable **butterfat**, with the type of cow stored in the variable **breed** and the age of the cow stored in the variable **age**.

- (a) Create a two-way ANOVA table.
- (b) Analyze the residuals and comment on whether the two-factorial model with interaction fits the data.
- (c) If there are problems that might be remedied with a transformation, suggest an appropriate transformation and reanalyze the new model.
- (d) Create a graphical display of the interactions for the model selected in (c). Is there significant interaction between breed and age?

(e) Based on the model selected in (c), compute group means and parameter estimates to fill in a table similar to Table 11.18.

(f) Using $\alpha_e = 0.05$, which breed has the highest average butterfat percentage?

16. Photosynthesis in aquatic plants is often inhibited due to the salinity of the water. Some plants such as *Cymodocea nodosa* seagrass appear to thrive in waters with high salinity. To determine the stress of *Cymodocea nodosa* seagrass seedlings in four levels of salinity (05PSU, 11PSU, 18PSU, and 36PSU), with two levels of spermidine (NO, YES), plant stress was determined by taking the ratio of F_v/F_m of four vessels each with two deceased *nodosa* seagrass seedlings that were randomly assigned to the eight treatments. F_v is the variable fluorescence, and F_m is the maximal fluorescence. The ratio F_v/F_m is stored under the variable name `fluorescence` in the `SEAGRASS.csv` file. The treatment structure is a 2×4 factorial experiment with 32 experimental units, where an experimental unit is a vessel containing two *Cymodocea nodosa* seagrass seedlings. The data stored at <https://raw.github.com/alanarnholt/Data/master/SEAGRASS.csv> is part of a larger study by Elso et al. (2012). Plants not under stress typically have F_v/F_m values between 0.7 and 0.8. Salinity for this study was recorded in practical salinity units (PSU), where 36PSU corresponds to typical ocean salinity.

- (a) Download the `SEAGRASS.csv` file using the `source_data()` function from the `repmis` package and store the results in an object named `SEAGRASS`.
- (b) Find and report the mean and standard deviation of `fluorescence` for the 8 treatment combinations. Does it appear that plants are more stressed without `spermidine` and at lower levels of `salinity`?
- (c) Are the assumptions for a factorial model satisfied with this data?
- (d) Create interaction plots for the factors `spermidine` and `salinity`. Based on your graphs, is there interaction between `spermidine` and `salinity`?
- (e) Write the hypotheses to test the main effects and the interaction for a 2 factor factorial design.
- (f) Test the hypotheses from (e).
- (g) Create and plot 99% family-wise confidence intervals for the pair-wise differences of the factors `spermidine` and `salinity` using the function `TukeyHSD()`. Interpret your confidence intervals.
- (h) Compute the means and the effects for the variables `spermidine` and `salinity` in the factorial model using the function `model.tables()`.
- (i) Assume the true means for the eight treatments are:

```
> MEANS <- c(0.6, 0.68, 0.75, 0.78, 0.70, 0.75, 0.79, 0.8)
> M <- matrix(MEANS, byrow = TRUE, nrow = 2)
> dimnames(M) <- list(spermidine=c("NO", "YES"),
+                       salinity=c("05PSU", "11PSU", "18PSU", "36PSU"))
> M

      salinity
spermidine 05PSU 11PSU 18PSU 36PSU
      NO     0.6   0.68   0.75   0.78
      YES    0.7   0.75   0.79   0.80
```

Compute the powers for testing differences in **salinity** if there are $n = 4$ experimental units per treatment assuming $\sigma = 0.05$, $\sigma = 0.1$, and $\sigma = 0.15$ for the seedling's ratio F_v/F_m and $\alpha = 0.05$.

- (j) Assuming the standard deviation for the fluorescence ratio, F_v/F_m , of all seedlings is $\sigma = 0.1$, find the minimum number of experimental units, n , needed per treatment group to have a power of at least 0.80 to detect differences in **spermidine** using $\alpha = 0.05$.

Case Study: Sunflower Defoliation

Ideas and data for this case study come from Muro et al. (2001).

17. Quantifying the effect of the loss of leaf area (defoliation) on sunflower (*Helianthus annuus L.*) yield caused by hail, pests, and diseases is important in the management of this crop both from a technical and economic point of view. The effect of defoliation depends, however, on the foliar surface eliminated and on the growth stage at which this takes place. The aim of this case study is to determine the response of sunflower cultivation to several levels of defoliation (**defoli**) that took place at different growth stages. An overall of 72 field trials were conducted by applying four defoliation treatments (non-defoliated control, 33%, 66% and 100%) at different growth stages (**stage**) ranging from pre-flowering (1) to physiological maturity (5) in four different locations (**location**) of Navarra, Spain: Carcastillo (1), Mélida (2), Murillo (3), and Unciti (4). There are two response variables: **yield** in kg/ha of the sunflower and **numseed**, the number of seeds per sunflower head. Data are stored in the data frame **SUNFLOWER**.

- (a) To explore the contents of the data frame **SUNFLOWER**,
 - (i) Construct a table with the total of the variable **yield** for every level of **stage** and **defoli**.
 - (ii) Construct a single table to display **numseed** for every level of **defoli**, **location**, and **stage**. (Hint: Use the functions **xtabs()** and **ftable()**.)
- (b) How many observations are there for every combination of **stage** and **defoli**?
- (c) Is the design complete or incomplete?
- (d) Is the design balanced or unbalanced?
- (e) Use side-by-side boxplots to display the variable **yield** for every level of **stage**.
- (f) Use side-by-side boxplots to display the variable **yield** for every level of **defoli**.
- (g) Construct an interaction plot for **stage** and **defoli** on **yield**. Comment on the results.

Model (A) Conduct an analysis of variance for **yield~stage + defoli + stage:defoli**.

- (i) Is the interaction between **stage** and **defoli** statistically significant?
- (ii) Use diagnostic graphics and appropriate tests to see if the assumptions for this model are satisfied.

Model (B) A rogue pest infestation was found in several plots. Observations from these plots were not under experimental control. Remove any observation whose standardized residual absolute value is greater than 2 and refit a new model.

- (i) Is the interaction term statistically significant?
- (ii) Use diagnostic graphics and appropriate tests to see if the assumptions for this model are satisfied.

Model (C) Define a model that pools the interaction term of Model (B) with the full model's error.

- (i) Check this model's assumptions.
 - (ii) Estimate the model's effects. Decompose the $Y_{i,j}$ s.
 - (iii) Construct Tukey's HSD pairwise confidence intervals for `yield` differences by levels of `defoli` from this model. Plot the intervals and interpret the results.
 - (iv) Construct Tukey's HSD pairwise confidence intervals for `yield` differences by levels of `stage` from this model. Plot the intervals and interpret the results.
- (h) If an insurance company compensates the `yield` loss only when there is a 100% defoliation, can statistical differences between this level and the rest of the defoliation levels be found? (Hint: Use orthogonal contrasts.)
- (i) To illustrate the final results, provide two graphs: a boxplot and a barplot of `yield` by levels of `stage`. Calculate the numerical values of the `yield` means and the standard errors.
 - (j) To illustrate the final results, redefine two levels for `defoli`: The first level combines the original levels 100 and 66 and the second new level groups the original levels 0 and 33 into a single level. Construct a boxplot and a barplot of `yield` for these two new levels. Provide a table for the corresponding means and standard errors.

Chapter 12

Regression

12.1 Introduction

The central theme of this chapter is modeling associations among variables. Understanding these associations can be important for many reasons, including

Reason 1. Prediction of future observations

Reason 2. Variable screening

Reason 3. System explanation

Reason 4. Parameter estimation

The primary tool used to model associations among variables in this chapter is regression. Regression analysis is used for modeling the relationship between a single variable Y , called the **response** or **dependent** variable, and one or more explanatory variables, also called **predictor(s)** or **independent variable(s)**, x_1, x_2, \dots, x_{p-1} . The response variable must be a continuous variable, but the predictor variables can be either continuous, discrete, or categorical. The word “regression” is due to Sir Francis Galton, who demonstrated that offspring do not tend toward the size of the parents; rather, offspring size tends toward the mean of the population. That is, there is a “*regression* toward mediocrity.” The following examples illustrate scenarios where it is important to understand the associations among response and predictor variables.

Example 12.1 ▷ **Prediction of Future Observations** ◁ A department chair is preparing a budget for the next fiscal year and must include enough money to replace personal computers in two laboratories. The chair wants to predict the price of personal computers for next year. He decides that good predictor variables for next year’s personal computer price Y are the price x_1 of a similar personal computer this year and x_2 , the rate of inflation. ■

Example 12.2 ▷ **Variable Screening** ◁ A chemist conducts a taste-testing experiment with randomly selected individuals from a particular geographical region. The dependent variable Y is the individual’s ratings of several formulations of a soft drink. The predictor variables are the various ingredients put into the soft drink. The sole purpose of the study is to decide which ingredients influence taste. ■

Example 12.3 ▷ **System Explanation** ◁ A sociologist has historical information on an isolated people group including voting records, media infiltration, numbers of roads accessing the area, and religious preferences. The sociologist is interested in understanding the rationale for why the people in the isolated group vote as they do. ■

Example 12.4 ▷ **Parameter Estimation** ◁ An economist has data on the GDP (gross domestic product) per capita (Y) and two independent variables: the median household income and the median household expenses for food in all European countries. The 27 points are fit to a linear model where prediction of gross domestic product is unimportant; however, the estimated signs and magnitudes of the model's parameters are important in supporting or refuting a particular economic theory. ■

Models of the form

$$Y = \beta_0 + \beta_1 x + \varepsilon \quad (12.1)$$

where β_0 and β_1 are the intercept and slope, respectively, and ε is the model error, can be used to model linear relationships between two variables. The population model in (12.1) is typically seen in a data setting where observations $(x_1, Y_1), (x_2, Y_2), \dots, (x_n, Y_n)$ are taken on experimental units, and estimates of the parameters β_0 and β_1 are sought. In a data setting, the model is expressed as

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \text{ for } i = 1, \dots, n. \quad (12.2)$$

Model (12.2) is said to be simple, linear in the parameters (β_0 and β_1), and linear in the predictor variables (x_i). It is simple because there is only one predictor; it is linear in the parameters because no parameter appears as an exponent nor is multiplied or divided by another parameter; and it is linear in the predictor variable since the predictor variable is raised only to the first power. When the predictor variable is raised to a power, this power is called the **order** of the model.

The models

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 \ln(x_i) + \varepsilon_i \\ Y_i &= \beta_0 + \beta_1 x_{1i} x_{2i} + \varepsilon_i \end{aligned}$$

are statistical linear models; however,

$$\begin{aligned} Y_i &= \beta_0 \exp(\beta_1 x_i) + \varepsilon_i \\ Y_i &= \frac{\beta_0}{1 + e^{\beta_1 x_i}} + \varepsilon_i \end{aligned}$$

are not statistical linear models since the Y_i s are not linearly related to the parameters β_0 and β_1 . Thus, a “linear model” is characterized by a linear relationship between the dependent variable and the parameters, not necessarily by a linear relationship with the independent variables. The random error term represents the absence of an exact relationship between Y and x . When the variance for all error terms is constant, the errors are said to be homoscedastic. Typically, $\text{Var}(\varepsilon_i) = \sigma^2$. Furthermore, the random variability is independent of x . The expected value of Y given x is written

$$E[Y|x] = \beta_0 + \beta_1 x. \quad (12.3)$$

The distribution of Y given x when ε_i follows a normal distribution with a mean of zero and a standard deviation of σ is depicted in Figure 12.1 on the next page. Since the random variable Y is a linear combination of the xs , it follows that σ^2 is not truly the variance of Y but rather the variance of Y given x . As seen in Figure 12.1, $\sigma^2 = \text{Var}(\varepsilon) = \text{Var}(Y|x)$. Up to this point, normally distributed random variables have been denoted as $N(\mu, \sigma)$, where σ is the standard deviation.

To simplify matrix expressions, the variance will take the place of the standard deviation in normal distributions from this point forward. For example, the distribution of the error

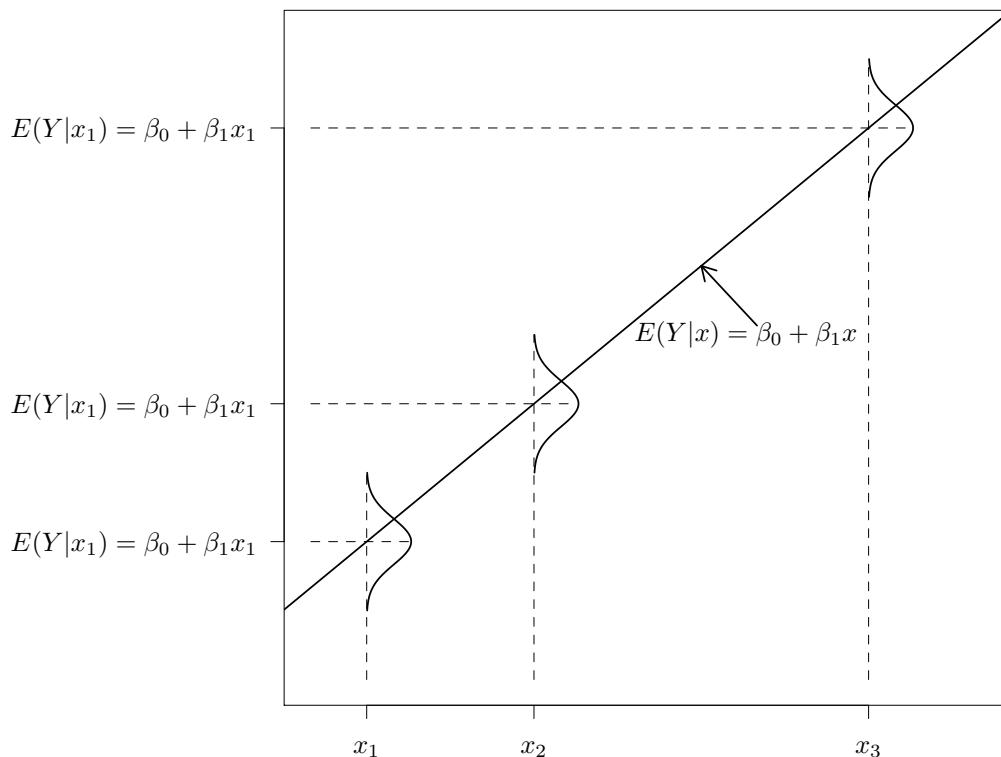


FIGURE 12.1: Graphical representation of simple linear regression model depicting the distribution of Y given x

terms in a simple linear regression model will be expressed $N(\mathbf{0}, \sigma^2 \mathbf{I})$ rather than saying each of the n errors has a $N(0, \sigma)$ distribution.

The slope, β_1 , represents the expected change in Y when a one-unit change is present in x . If $\beta_1 = 0$, Y does not depend linearly on x . When $\beta_1 < 0$, x and Y have a negative linear relationship, which means that as x increases, Y decreases. Likewise, when $\beta_1 > 0$, x and Y have a positive linear relationship, where, as x increases, so does Y .

12.2 Simple Linear Regression

The simple linear regression model when the error terms are distributed normally is

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad (12.4)$$

where

Y_i is the value of the response variable for the i^{th} trial,

β_0 and β_1 are parameters,

x_i is a known constant for the i^{th} trial, and

ε_i is a random error term that is assumed to have a $N(0, \sigma^2)$ distribution, where σ^2 (the variance) is typically unknown.

The idea that drives regression is the estimation of parameters based on n measurements. The simple linear model for n bivariate measurements is

$$\begin{aligned} Y_1 &= \beta_0 + \beta_1 x_1 + \varepsilon_1 \\ Y_2 &= \beta_0 + \beta_1 x_2 + \varepsilon_2 \\ &\vdots = \vdots + \vdots + \vdots \\ Y_n &= \beta_0 + \beta_1 x_n + \varepsilon_n, \end{aligned}$$

which can also be expressed with matrix notation as

$$\mathbf{Y}_{n \times 1} = \mathbf{X}_{n \times 2} \boldsymbol{\beta}_{2 \times 1} + \boldsymbol{\varepsilon}_{n \times 1} \quad (12.5)$$

$$\text{where } \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \text{ and } \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

Note that $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$, where $\sigma^2 \mathbf{I}$ is the variance-covariance matrix of the vector of errors.

12.3 Multiple Linear Regression

Multiple linear regression is similar to simple linear regression in several ways. The dependent variable is still Y . The intercept is still β_0 . The primary change is that instead of having only β_1 as a coefficient of a single x_i variable, there now exists an entire vector of β_j values to multiply by a matrix of x_{ij} values. The multiple linear regression model is written

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{p-1} x_{i(p-1)} + \varepsilon_i \text{ for } i = 1, 2, \dots, n. \quad (12.6)$$

The multiple regression model (12.6) describes a line when there is a single predictor (x_{i1}), a plane when there are two predictors (x_{i1}, x_{i2}), and a hyperplane for more than two predictors. The model will typically be expressed more compactly in matrix form as

$$\mathbf{Y}_{n \times 1} = \mathbf{X}_{n \times p} \boldsymbol{\beta}_{p \times 1} + \boldsymbol{\varepsilon}_{n \times 1} \quad (12.7)$$

$$\text{where } \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1(p-1)} \\ 1 & x_{21} & \dots & x_{2(p-1)} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \dots & x_{n(p-1)} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}, \text{ and } \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

Each column of \mathbf{X} contains the values for a particular independent variable. The values of \mathbf{X} are assumed to be known constants. The vectors \mathbf{Y} and $\boldsymbol{\varepsilon}$ are random vectors whose elements are random variables. The vector $\boldsymbol{\beta}$ is a vector of unknown constants that are estimated from the data. Each β_j for $j = 0, 1, \dots, p - 1$ indicates the change $E[Y|x_{ij}]$ for a fixed i when x_{ij} is increased by one unit and all the other predictors are held constant.

When ε is assumed $N(\mathbf{0}, \sigma^2 \mathbf{I})$, model (12.7) is referred to as the **normal error model**. In the normal error model, \mathbf{X} and $\boldsymbol{\beta}$ are assumed to be constants. Consequently, \mathbf{Y} is a random vector that is the sum of a constant vector $\mathbf{X}\boldsymbol{\beta}$ and the random vector ε . Since ε is assumed $N(\mathbf{0}, \sigma^2 \mathbf{I})$, it follows that $\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I})$. The tests and confidence intervals developed in later sections are based on the assumption that $\varepsilon \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$.

Assuming that there is no error in the measurement of the x_{ij} values, one can proceed with either of the two most widely used techniques used to estimate parameters (β_j s) in a regression model: **ordinary least squares** or the **method of maximum likelihood**.

12.4 Ordinary Least Squares

The ordinary least squares method of estimating parameters minimizes the sum of the squared deviations of the Y_i s from their expected values such that

$$\varepsilon_i = Y_i - E(Y_i)$$

is the i^{th} deviation (error). For the simple linear regression model, $\varepsilon_i = Y_i - (\beta_0 + \beta_1 x_i)$. The estimates $\hat{\beta}_0$ of β_0 and $\hat{\beta}_1$ of β_1 are calculated by minimizing the quantity \mathcal{Q} (the sum of the squared residuals) found in (12.8):

$$\mathcal{Q} = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (Y_i - (\beta_0 + \beta_1 x_i))^2, \quad (12.8)$$

which is equivalent to the matrix form

$$\mathcal{Q} = \varepsilon' \varepsilon = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}). \quad (12.9)$$

The values of β_0 and β_1 that minimize \mathcal{Q} are found by differentiating \mathcal{Q} with respect to β_0 and β_1 and setting the partial derivatives equal to zero. The resulting equations are known as the **normal equations**:

$$\begin{aligned} \frac{\delta \mathcal{Q}}{\delta \beta_0} &= 2 \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i)(-1) \\ &= -2 \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i). \end{aligned} \quad (12.10)$$

$$\begin{aligned} \frac{\delta \mathcal{Q}}{\delta \beta_1} &= 2 \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i)(-x_i) \\ &= -2 \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i)(x_i). \end{aligned} \quad (12.11)$$

After setting each of these partial derivatives equal to zero, the normal equations for the simple linear regression model simplify to

$$\sum_{i=1}^n Y_i = n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^n x_i. \quad (12.12)$$

$$\sum_{i=1}^n Y_i x_i = \hat{\beta}_0 \sum_{i=1}^n x_i + \hat{\beta}_1 \sum_{i=1}^n x_i^2. \quad (12.13)$$

Note that the β_j s are replaced with $\hat{\beta}_j$ s as their values are estimates once the partial derivatives are set equal to zero. These equations are now solved for $\hat{\beta}_0$ and $\hat{\beta}_1$.

Solving for $\hat{\beta}_0$ is relatively simple using (12.12):

$$\begin{aligned} \sum_{i=1}^n Y_i &= n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^n x_i \\ \sum_{i=1}^n Y_i - \hat{\beta}_1 \sum_{i=1}^n x_i &= n\hat{\beta}_0 \\ \frac{\sum_{i=1}^n Y_i}{n} - \frac{\hat{\beta}_1 \sum_{i=1}^n x_i}{n} &= \hat{\beta}_0 \\ \hat{\beta}_0 &= \bar{Y} - \hat{\beta}_1 \bar{x}. \end{aligned} \quad (12.14)$$

In solving for $\hat{\beta}_1$, two quantities appear that require simplification. The first quantity is

$$\sum_{i=1}^n Y_i x_i - \frac{\sum_{i=1}^n Y_i \sum_{i=1}^n x_i}{n}. \quad (12.15)$$

$$\begin{aligned} \sum_{i=1}^n Y_i x_i - \frac{\sum_{i=1}^n Y_i \sum_{i=1}^n x_i}{n} &= \sum_{i=1}^n Y_i x_i - \bar{Y} \sum_{i=1}^n x_i \\ &= \sum_{i=1}^n Y_i x_i - \bar{Y} \sum_{i=1}^n x_i - \bar{Y} \sum_{i=1}^n x_i + \frac{n}{n} \bar{Y} \sum_{i=1}^n x_i \\ &= \sum_{i=1}^n Y_i x_i - \bar{Y} \sum_{i=1}^n x_i - \frac{\sum_{i=1}^n Y_i}{n} \sum_{i=1}^n x_i + n \bar{Y} \bar{x} \\ &= \sum_{i=1}^n Y_i x_i - \bar{Y} \sum_{i=1}^n x_i - \bar{x} \sum_{i=1}^n Y_i + n \bar{Y} \bar{x} \\ &= \sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y}). \end{aligned}$$

The second quantity that will need to be simplified is

$$\sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n}. \quad (12.16)$$

$$\begin{aligned} \sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n} &= \sum_{i=1}^n x_i^2 - n \left(\frac{\sum_{i=1}^n x_i}{n} \right)^2 \\ &= \sum_{i=1}^n x_i^2 - n \bar{x}^2 \\ &= \sum_{i=1}^n x_i^2 - n \bar{x}^2 - n \bar{x}^2 + n \bar{x}^2 \\ &= \sum_{i=1}^n (x_i - \bar{x})^2. \end{aligned} \quad (12.17)$$

Knowing these two simplifications, $\hat{\beta}_1$ can be solved using (12.13):

$$\begin{aligned}
 \sum_{i=1}^n Y_i x_i &= \hat{\beta}_0 \sum_{i=1}^n x_i + \hat{\beta}_1 \sum_{i=1}^n x_i^2 \\
 \sum_{i=1}^n Y_i x_i &= (\bar{Y} - \hat{\beta}_1 \bar{x}) \sum_{i=1}^n x_i + \hat{\beta}_1 \sum_{i=1}^n x_i^2 \\
 \sum_{i=1}^n Y_i x_i &= \left(\frac{\sum_{i=1}^n Y_i}{n} - \hat{\beta}_1 \frac{\sum_{i=1}^n x_i}{n} \right) \sum_{i=1}^n x_i + \hat{\beta}_1 \sum_{i=1}^n x_i^2 \\
 \sum_{i=1}^n Y_i x_i &= \frac{\sum_{i=1}^n Y_i \sum_{i=1}^n x_i}{n} - \hat{\beta}_1 \frac{(\sum_{i=1}^n x_i)^2}{n} + \hat{\beta}_1 \sum_{i=1}^n x_i^2 \\
 \sum_{i=1}^n Y_i x_i - \frac{\sum_{i=1}^n Y_i \sum_{i=1}^n x_i}{n} &= \hat{\beta}_1 \sum_{i=1}^n x_i^2 - \hat{\beta}_1 \frac{(\sum_{i=1}^n x_i)^2}{n} \\
 \sum_{i=1}^n Y_i x_i - \frac{\sum_{i=1}^n Y_i \sum_{i=1}^n x_i}{n} &= \hat{\beta}_1 \left(\sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n} \right) \\
 \hat{\beta}_1 &= \frac{\sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{x})^2}. \tag{12.18}
 \end{aligned}$$

After $\hat{\beta}_0$ and $\hat{\beta}_1$ have been found, it must be shown that these values will give a minimum value for the sum of squared errors.

Proof ($\sum_{i=1}^n \hat{\varepsilon}_i^2$ is a Minimum): If the matrix of partial derivatives of \mathcal{Q} as found in (12.8) is positive definite, then our $\hat{\beta}$ values do give the minimum value for \mathcal{Q} . Recall from (12.10) that $\frac{\delta \mathcal{Q}}{\delta \beta_0} = -2 \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i)$ and from (12.11) that $\frac{\delta \mathcal{Q}}{\delta \beta_1} = -2 \sum_{i=1}^n (Y_i - \beta_0 - \beta_1 x_i)(x_i)$. This implies that the second-order partials are

$$\begin{aligned}
 \frac{\delta^2 \mathcal{Q}}{\delta \beta_0^2} &= -2 \sum_{i=1}^n (-1) = 2n \\
 \frac{\delta^2 \mathcal{Q}}{\delta \beta_1^2} &= -2 \sum_{i=1}^n (-x_i)(x_i) = 2 \sum_{i=1}^n x_i^2 \\
 \frac{\delta^2 \mathcal{Q}}{\delta \beta_0 \delta \beta_1} &= -2 \sum_{i=1}^n (-x_i) = 2 \sum_{i=1}^n x_i.
 \end{aligned}$$

The matrix of partials is then

$$\frac{\delta^2 \mathcal{Q}}{\delta \beta^2} = \begin{bmatrix} 2n & 2 \sum_{i=1}^n x_i \\ 2 \sum_{i=1}^n x_i & 2 \sum_{i=1}^n x_i^2 \end{bmatrix}. \tag{12.19}$$

The determinant of this matrix is $4n \sum_{i=1}^n x_i^2 - 4(\sum_{i=1}^n x_i)^2$. It must be shown that this quantity is always positive to prove that $\hat{\beta}_0$ and $\hat{\beta}_1$ as given provide a minimum value for \mathcal{Q} . Note that n is assumed to be greater than zero:

$$\begin{aligned}
 4n \sum_{i=1}^n x_i^2 - 4 \left(\sum_{i=1}^n x_i \right)^2 &\stackrel{?}{>} 0 \\
 \sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i \right)^2}{n} &\stackrel{?}{>} 0 \\
 \sum_{i=1}^n (x_i - \bar{x})^2 &> 0 \quad \text{from (12.17).}
 \end{aligned}$$

Therefore, the $\hat{\beta}_0$ and $\hat{\beta}_1$ calculated do give the minimum value for Q . ■

Now that the β values that will minimize Q are computed, the fitted regression line is written

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \quad (12.20)$$

where estimated (predicted) errors, also called **residuals**, are defined to be

$$\hat{\varepsilon}_i = Y_i - \hat{Y}_i. \quad (12.21)$$

12.5 Properties of the Fitted Regression Line

Several properties of the fitted regression line will be helpful in understanding the relationships between \mathbf{X} , $\boldsymbol{\beta}$, $\boldsymbol{\varepsilon}$, and \mathbf{Y} :

1. $\sum_{i=1}^n \hat{\varepsilon}_i = 0$.
2. $\sum_{i=1}^n Y_i = \sum_{i=1}^n \hat{Y}_i$.
3. $\sum_{i=1}^n x_i \hat{\varepsilon}_i = 0$.
4. $\sum_{i=1}^n \hat{Y}_i \hat{\varepsilon}_i = 0$.
5. The regression line always goes through the point (\bar{x}, \bar{Y}) .

Note that all five of these properties follow from the least squares normal equations (12.12) and (12.13).

Proof (Property 1):

$$\begin{aligned}
 \hat{\varepsilon}_i &= Y_i - \hat{Y}_i \\
 \hat{\varepsilon}_i &= Y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) \\
 \sum_{i=1}^n \hat{\varepsilon}_i &= \sum_{i=1}^n (Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) \\
 \sum_{i=1}^n \hat{\varepsilon}_i &= \sum_{i=1}^n Y_i - n\hat{\beta}_0 - \hat{\beta}_1 \sum_{i=1}^n x_i = 0 \text{ by (12.12)} \quad \blacksquare
 \end{aligned}$$

Proof (Property 2):

$$\begin{aligned}\hat{\varepsilon}_i &= Y_i - \hat{Y}_i \\ \sum_{i=1}^n \hat{\varepsilon}_i &= \sum_{i=1}^n Y_i - \sum_{i=1}^n \hat{Y}_i \\ 0 &= \sum_{i=1}^n Y_i - \sum_{i=1}^n \hat{Y}_i \\ \sum_{i=1}^n Y_i &= \sum_{i=1}^n \hat{Y}_i \quad \blacksquare\end{aligned}$$

Proof (Property 3):

$$\begin{aligned}\sum_{i=1}^n Y_i x_i &= \hat{\beta}_0 \sum_{i=1}^n x_i + \hat{\beta}_1 \sum_{i=1}^n x_i^2 \text{ by (12.13)} \\ \sum_{i=1}^n Y_i x_i - \hat{\beta}_0 \sum_{i=1}^n x_i - \hat{\beta}_1 \sum_{i=1}^n x_i^2 &= 0 \\ \sum_{i=1}^n x_i (Y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) &= 0 \\ \sum_{i=1}^n x_i (Y_i - \hat{Y}_i) &= 0 \\ \sum_{i=1}^n x_i \hat{\varepsilon}_i &= 0 \quad \blacksquare\end{aligned}$$

Proof (Property 4):

$$\begin{aligned}\sum_{i=1}^n \hat{Y}_i \hat{\varepsilon}_i &= \sum_{i=1}^n (\hat{\beta}_0 + \hat{\beta}_1 x_i) \hat{\varepsilon}_i \\ &= \hat{\beta}_0 \sum_{i=1}^n \hat{\varepsilon}_i + \hat{\beta}_1 \sum_{i=1}^n x_i \hat{\varepsilon}_i \\ &= 0 \text{ by Properties 1 and 3} \quad \blacksquare\end{aligned}$$

Proof (Property 5): Given the regression line $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$, if $x_i = \bar{x}$, then

$$\begin{aligned}\hat{Y}_i &= \hat{\beta}_0 + \hat{\beta}_1 \bar{x} \\ \hat{Y}_i &= \bar{Y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 \bar{x} \quad \text{using (12.14)} \\ \Rightarrow \hat{Y}_i &= \bar{Y} \quad \blacksquare\end{aligned}$$

12.6 Using Matrix Notation with Ordinary Least Squares

The solutions, $\boldsymbol{\beta}$, to (12.5) are generally easier to express in matrix notation than in summation notation. The normal equations are now presented in matrix form. Recall that

$$\mathcal{Q} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

This is simplified first and then differentiated with respect to β . Then, the result is set equal to $\mathbf{0}$ to solve for $\hat{\beta}$:

$$\mathcal{Q} = \mathbf{Y}'\mathbf{Y} - \beta'\mathbf{X}'\mathbf{Y} - \mathbf{Y}'\mathbf{X}\beta + \beta'\mathbf{X}'\mathbf{X}\beta.$$

Since $\beta'\mathbf{X}'\mathbf{Y}$ is a scalar (1×1), $(\beta'\mathbf{X}'\mathbf{Y})' = \mathbf{Y}'\mathbf{X}\beta$, so \mathcal{Q} simplifies to

$$\mathcal{Q} = \mathbf{Y}'\mathbf{Y} - 2\mathbf{Y}'\mathbf{X}\beta + \beta'\mathbf{X}'\mathbf{X}\beta.$$

The expression for $\frac{\delta \mathcal{Q}}{\delta \beta}$ can now be calculated:

$$\begin{aligned} \frac{\delta \mathcal{Q}}{\delta \beta} &= \frac{\delta}{\delta \beta}(\mathbf{Y}'\mathbf{Y}) - \frac{\delta}{\delta \beta}(2(\mathbf{X}'\mathbf{Y})'\beta) + \frac{\delta}{\delta \beta}(\beta'\mathbf{X}'\mathbf{X}\beta) \\ &= 0 - 2\mathbf{X}'\mathbf{Y} + [\mathbf{X}'\mathbf{X}\beta + (\mathbf{X}'\mathbf{X})'\beta] \\ &\quad \text{by Rules for Differentiation 1 and 3 on page 917} \\ &= -2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\beta. \end{aligned} \tag{12.22}$$

Setting (12.22) equal to zero and solving for β yields

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}, \tag{12.23}$$

the normal equations expressed in matrix notation. The worked-out solutions for the matrix form of the simple linear regression model are presented next.

Recall that, for the simple linear regression model, $\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$, so

$$\mathbf{X}'\mathbf{X} = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix}. \tag{12.24}$$

Also recall that the inverse of a matrix $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is $\mathbf{A}^{-1} = \frac{1}{\det \mathbf{A}} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$, where $\det \mathbf{A} = ad - bc$. Then

$$\det(\mathbf{X}'\mathbf{X}) = n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 = n \sum_{i=1}^n (x_i - \bar{x})^2. \tag{12.25}$$

So,

$$(\mathbf{X}'\mathbf{X})^{-1} = \frac{1}{n \sum_{i=1}^n (x_i - \bar{x})^2} \begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix}. \tag{12.26}$$

Likewise,

$$\mathbf{X}'\mathbf{Y} = \begin{bmatrix} \sum_{i=1}^n Y_i \\ \sum_{i=1}^n x_i Y_i \end{bmatrix}. \tag{12.27}$$

This means that

$$\begin{aligned}
 \hat{\beta} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\
 &= \frac{1}{n \sum_{i=1}^n (x_i - \bar{x})^2} \begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix} \begin{bmatrix} \sum_{i=1}^n Y_i \\ \sum_{i=1}^n x_i Y_i \end{bmatrix} \\
 &= \begin{bmatrix} \frac{\sum_{i=1}^n x_i^2 \sum_{i=1}^n Y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i Y_i}{n \sum_{i=1}^n (x_i - \bar{x})^2} \\ \frac{-\sum_{i=1}^n x_i \sum_{i=1}^n Y_i + n \sum_{i=1}^n x_i Y_i}{n \sum_{i=1}^n (x_i - \bar{x})^2} \end{bmatrix} \\
 &= \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix}.
 \end{aligned} \tag{12.28}$$

Next, it should be shown that the matrix solutions for $\hat{\beta}_0$ and $\hat{\beta}_1$ are identical to the summation solutions shown in (12.14) and (12.18). Converting the second entry in $\hat{\beta}$ from (12.28) to (12.18) is more obvious, so it will be done first:

$$\begin{aligned}
 \frac{-\sum_{i=1}^n x_i \sum_{i=1}^n Y_i + n \sum_{i=1}^n x_i Y_i}{n \sum_{i=1}^n (x_i - \bar{x})^2} &\stackrel{?}{=} \hat{\beta}_1 \\
 \frac{\frac{1}{n} \cdot \frac{n \sum_{i=1}^n x_i Y_i - \sum_{i=1}^n x_i \sum_{i=1}^n Y_i}{n \sum_{i=1}^n (x_i - \bar{x})^2}}{\frac{1}{n} \cdot \frac{\sum_{i=1}^n x_i Y_i - \frac{\sum_{i=1}^n x_i \sum_{i=1}^n Y_i}{n}}{\sum_{i=1}^n (x_i - \bar{x})^2}} &\stackrel{?}{=} \hat{\beta}_1 \\
 \frac{\sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^n (x_i - \bar{x})^2} &= \hat{\beta}_1 \text{ by simplification (12.15)}.
 \end{aligned}$$

Next, show $\hat{\beta}_0$ from (12.14) is equal to the first entry of $\hat{\beta}$:

$$\begin{aligned}
 \frac{\sum_{i=1}^n x_i^2 \sum_{i=1}^n Y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i Y_i}{n \sum_{i=1}^n (x_i - \bar{x})^2} &\stackrel{?}{=} \hat{\beta}_0 \\
 \frac{\sum_{i=1}^n x_i^2 \sum_{i=1}^n Y_i - \frac{\left(\sum_{i=1}^n x_i\right)^2 \sum_{i=1}^n Y_i}{n} + \frac{\left(\sum_{i=1}^n x_i\right)^2 \sum_{i=1}^n Y_i}{n} - \sum_{i=1}^n x_i \sum_{i=1}^n x_i Y_i}{n \sum_{i=1}^n (x_i - \bar{x})^2} &\stackrel{?}{=} \hat{\beta}_0 \\
 \frac{\sum_{i=1}^n Y_i \left[\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i\right)^2}{n} \right] - \left[\left(\sum_{i=1}^n x_i Y_i - \bar{Y} \sum_{i=1}^n x_i \right) \sum_{i=1}^n x_i \right]}{n \sum_{i=1}^n (x_i - \bar{x})^2} &\stackrel{?}{=} \hat{\beta}_0 \\
 \frac{\sum_{i=1}^n Y_i [\sum_{i=1}^n (x_i - \bar{x})^2] - [\sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y}) \sum_{i=1}^n x_i]}{n \sum_{i=1}^n (x_i - \bar{x})^2} &\stackrel{?}{=} \hat{\beta}_0 \\
 &\quad \text{by Simplifications (12.16) and (12.15)} \\
 \frac{\sum_{i=1}^n Y_i [\sum_{i=1}^n (x_i - \bar{x})^2] - \sum_{i=1}^n x_i \sum_{i=1}^n (x_i - \bar{x})(Y_i - \bar{Y})}{n \sum_{i=1}^n (x_i - \bar{x})^2} &\stackrel{?}{=} \hat{\beta}_0 \\
 &\quad \bar{Y} - \hat{\beta}_1 \bar{x} = \hat{\beta}_0.
 \end{aligned}$$

Therefore, the matrix solution is identical to the summation solution, so $\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix}$.

Example 12.5 Find the variance-covariance matrix of $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ when \mathbf{X} is an $n \times p$ matrix.

Solution: Let $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = \mathbf{A}$. Then $\hat{\beta} = \mathbf{AY}$ and $\sigma_{\hat{\beta}}^2 = \mathbf{A}\sigma_{\mathbf{Y}}^2\mathbf{A}'$ by property 3 on page 919. Note that $\sigma_{\mathbf{Y}}^2 = \sigma^2\mathbf{I}_{n \times n}$. So,

$$\begin{aligned}\sigma_{\hat{\beta}}^2 &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\sigma^2\mathbf{I}_{n \times n}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\ &= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \\ &= \sigma^2(\mathbf{X}'\mathbf{X})^{-1}.\end{aligned}$$



Example 12.6 \triangleright *Linear Relationship between GPA and SAT Scores* \triangleleft The admissions committee of a comprehensive state university selected at random the records of 200 second-semester freshmen. The results, first-semester college GPA and SAT scores, are stored in the data frame **GRADES**. The admissions committee wants to study the linear relationship between first-semester college grade point average (**gpa**) and scholastic aptitude test (**sat**) scores. Assume that the requirements for model (12.4) are satisfied.

- (a) Create a scatterplot of the data to investigate the relationship between **gpa** and **sat** scores.
- (b) Obtain the least squares estimates for β_0 and β_1 , and state the estimated regression function using
 - (i) Summation notation with (12.14) and (12.18).
 - (ii) Matrix notation with (12.23).
 - (iii) Use the R function `lm()` to verify the answers in (i) and (ii).
- (c) What is the point estimate of the change in the mean **gpa** when the **sat** score increases by 50 points?

Solution: The data frame **GRADES** is in the **PASWR2** package.

- (a) The scatterplot in Figure 12.2 on the next page created from R Code 12.1 suggests a linear relationship exists between **gpa** and **sat**.

R Code 12.1

```
> p <- ggplot(data = GRADES, aes(x = sat, y = gpa))
> p + geom_point() + geom_smooth(method = "lm", se = FALSE) + theme_bw()
```

- (b) The answers to (b) follow.

- (i) Assign **gpa** to Y and **sat** to x :

```
> Y <- GRADES$gpa
> x <- GRADES$sat
```

Solving using summation notation as in (12.18) (**b1**) and (12.14) (**b0**):

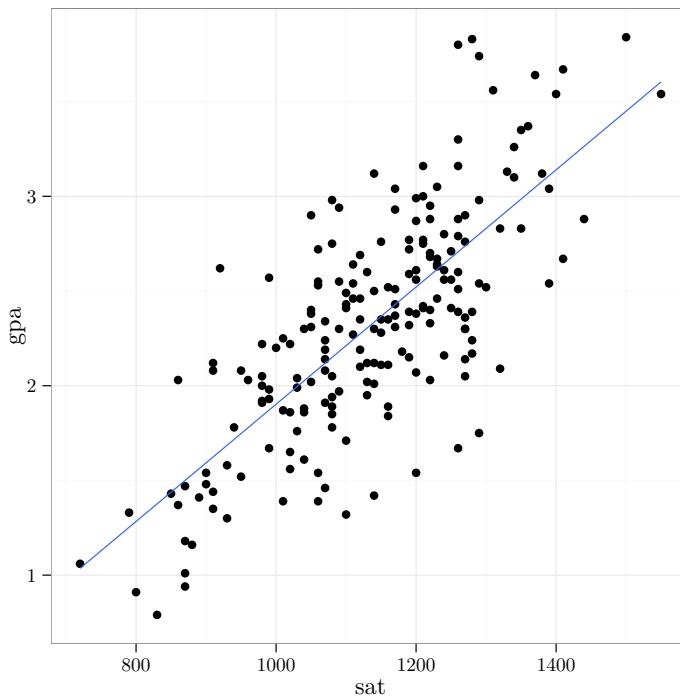


FIGURE 12.2: Scatterplot of gpa versus sat using GRADES

```
> b1 <- sum((x - mean(x)) * (Y - mean(Y))) / sum((x - mean(x))^2)
> b0 <- mean(Y) - b1 * mean(x)
> c(b0, b1)

[1] -1.19206381  0.00309427
```

The estimated regression function is $\hat{Y}_i = -1.1921 + 0.0031x_i$.

(ii) Solving using matrix notation as in (12.23):

```
> X <- cbind(rep(1, dim(GRADES)[1]), x)
> Y <- as.matrix(Y, nrow = dim(GRADES)[1])
> betahat <- solve(t(X) %*% X) %*% t(X) %*% Y
> beta0hat <- betahat[1, 1]
> beta1hat <- betahat[2, 1]
> names(beta1hat) <- NULL
> c(beta0hat, beta1hat)

[1] -1.19206381  0.00309427
```

(iii) Solving using lm():

```
> model.lm <- lm(gpa ~ sat, data = GRADES)
> coef(model.lm)

(Intercept)          sat
-1.19206381  0.00309427
```

The estimated regression function is $\hat{Y}_i = -1.1921 + 0.0031x_i$.

- (c) The point estimate of the change in the mean `gpa` when the `sat` score increases by 50 points is $\hat{\beta}_1 \cdot 50 = 0.1547$:

```
> b1 * 50

[1] 0.1547135
```



In Example 12.6 on page 792, the function `lm()` was used to find estimates for β_0 and β_1 for a simple linear regression model. To use the function `lm()` with multiple linear regression models, one specifies the predictors for a multiple linear regression model on the right side of the tilde (\sim) operator inside the `lm()` function. The data frame `HSWRESTLER` contains the body fat measurements of 78 high school wrestlers. R Code 12.2 stores the multiple linear regression model for regressing `hwfat` (hydrostatic fat) onto `abs` (abdominal fat) and `triceps` (tricep fat). The estimated coefficients for β_0 , β_1 , and β_2 determine the plane of best fit for the given values.

R Code 12.2

```
> hsw.lm <- lm(hwfat ~ abs + triceps, data = HSWRESTLER)
> coef(summary(hsw.lm)) # lm coefficients

            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.0590398 0.65219267 3.157104 2.295484e-03
abs         0.3370839 0.06414665 5.254895 1.343931e-06
triceps    0.5043030 0.09919943 5.083729 2.638708e-06
```

Figure 2.27 on page 148 shows the residuals for a simple linear regression while Figure 12.3 on the facing page shows the residuals for the multiple linear regression model $\widehat{\text{hwfat}} = 2.059 + 0.3371 \times \text{abs} + 0.5043 \times \text{triceps}$.

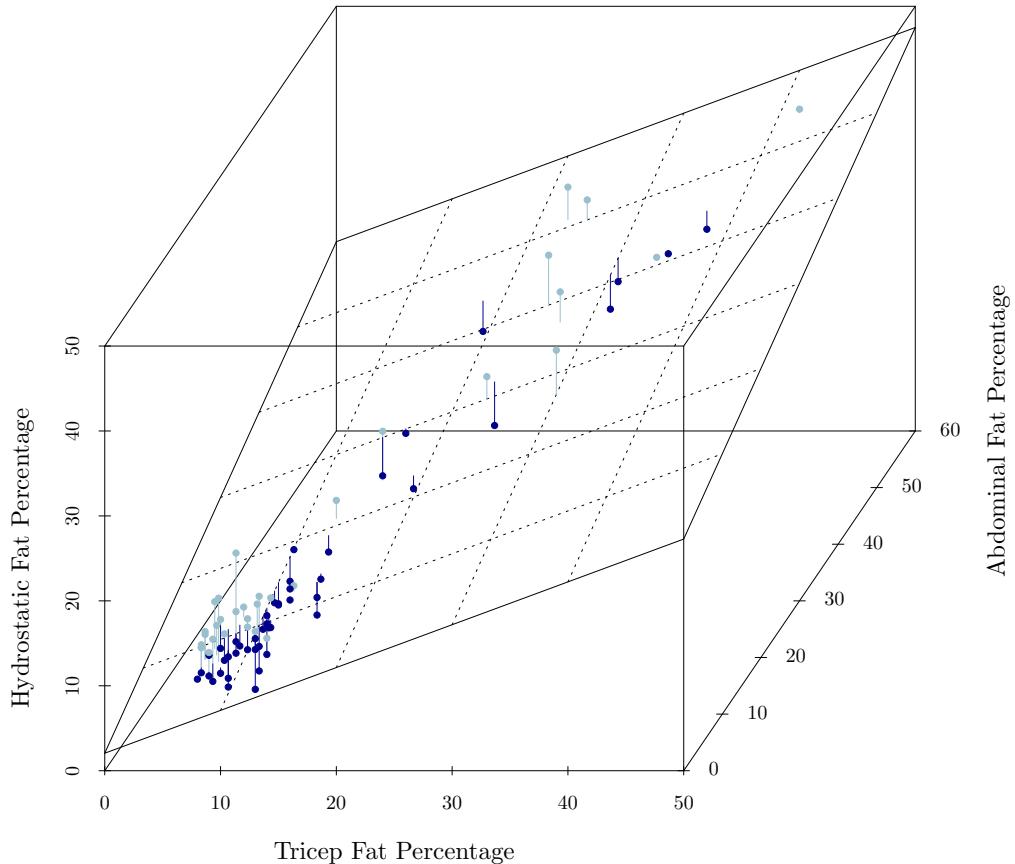


FIGURE 12.3: Plane of best fit from regressing `hwfat` onto `triceps` and `abs`. The light-colored vertical lines are positive residuals, while the darker-colored vertical lines are negative residuals.

12.7 The Method of Maximum Likelihood

The method of least squares is not the only one that can be used to construct an estimate of β . Another common method for constructing estimators is that of maximum likelihood. To construct the maximum likelihood estimator (MLE) of β when $\varepsilon \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$, first construct the likelihood function. Next, take the natural log. Finally, take appropriate partial derivatives and set them equal to zero to solve for the MLE of β , $\hat{\beta}$.

The likelihood function for β and σ^2 when \mathbf{X} is given is

$$\mathcal{L}(\beta, \sigma^2 | \mathbf{X}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[\frac{-(Y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i,p-1}))^2}{2\sigma^2} \right]. \quad (12.29)$$

In matrix form, this is

$$\mathcal{L}(\beta, \sigma^2 | \mathbf{X}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[\frac{-(\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta)}{2\sigma^2} \right]. \quad (12.30)$$

The natural log of the matrix form of the likelihood function (log-likelihood function) is

$$\ln \mathcal{L}(\beta, \sigma^2 | \mathbf{X}) = -\frac{n}{2} \ln(2\pi) - \frac{n}{2} \ln(\sigma^2) - \frac{(\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta)}{2\sigma^2}.$$

Simplifying the partial derivative of the log-likelihood function with respect to β gives

$$\begin{aligned} \frac{\delta \ln \mathcal{L}(\beta, \sigma^2 | \mathbf{X})}{\delta \beta} &= \frac{\delta}{\delta \beta} \left[-\frac{(\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta)}{2\sigma^2} \right] \\ &= \frac{\delta}{\delta \beta} \left[\frac{-\mathbf{Y}'\mathbf{Y} + \beta'\mathbf{X}'\mathbf{Y} + \mathbf{Y}'\mathbf{X}\beta - \beta'\mathbf{X}'\mathbf{X}\beta}{2\sigma^2} \right] \end{aligned}$$

Recall that $\beta'\mathbf{X}'\mathbf{Y}$ is 1×1

$$= \frac{\delta}{\delta \beta} \left[\frac{-\mathbf{Y}'\mathbf{Y} + 2(\mathbf{X}'\mathbf{Y})'\beta - \beta'\mathbf{X}'\mathbf{X}\beta}{2\sigma^2} \right]$$

By Rules of Differentiation 1 and 3 on page 917

$$\begin{aligned} &= \frac{2\mathbf{X}'\mathbf{Y}}{2\sigma^2} - \left[\frac{\mathbf{X}'\mathbf{X}\beta + (\mathbf{X}'\mathbf{X})'\beta}{2\sigma^2} \right] \\ &= \frac{\mathbf{X}'\mathbf{Y} - \mathbf{X}'\mathbf{X}\beta}{\sigma^2}. \end{aligned}$$

Setting this equal to zero and solving for β yields

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

Note that the MLE is equivalent to the ordinary least squares estimator for β given in (12.23). It is also of interest to find the MLE for σ^2 . Taking the partial derivative of the log-likelihood function in terms of σ^2 gives

$$\frac{\delta \ln \mathcal{L}(\beta, \sigma^2 | \mathbf{X})}{\delta \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \cdot (\mathbf{Y} - \mathbf{X}\beta)'(\mathbf{Y} - \mathbf{X}\beta).$$

When this quantity is set equal to zero and solved for σ^2 , the MLE is

$$\tilde{\sigma}^2 = \frac{(\mathbf{Y} - \mathbf{X}\hat{\beta})'(\mathbf{Y} - \mathbf{X}\hat{\beta})}{n} = \frac{\hat{\varepsilon}'\hat{\varepsilon}}{n}.$$

Unfortunately, $\tilde{\sigma}^2$ is a biased estimator of σ^2 . The bias is easily fixed and the unbiased estimator $\frac{\hat{\varepsilon}'\hat{\varepsilon}}{n-p}$ is typically used to estimate σ^2 .

12.8 The Sampling Distribution of $\hat{\beta}$

The matrix form of $\hat{\beta}$ was described in (12.23) to be $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. If model (12.7) assumes that $\varepsilon \sim N(\mathbf{0}, \sigma^2\mathbf{I})$, then $\mathbf{Y} \sim N(\mathbf{X}\beta, \sigma^2\mathbf{I})$ because \mathbf{X} and β are assumed to be constants. Since $\hat{\beta}$ can be expressed as constants multiplied by \mathbf{Y} , it follows that $\hat{\beta}$ also has a normal distribution. It will be shown that

$$\hat{\beta} \sim N(\beta, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}).$$

In Example 12.5 on page 792, the variance of $\hat{\beta}$ was shown to equal $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. Next, $\hat{\beta}$ is shown to be an unbiased estimator of β . Specifically,

$$\text{If } \hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$$

$$\begin{aligned} \text{Then } E[\hat{\beta}] &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}] \\ &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\beta + \varepsilon)] \\ &= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon] \\ &= E[\mathbf{I}\beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\varepsilon] \\ &= \beta \text{ since } \mathbf{I} \text{ and } (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \text{ are constants and } E(\varepsilon) = \mathbf{0}, \end{aligned}$$

under the normal error regression model. Unfortunately, unbiasedness does not guarantee uniqueness. Fortunately, the **Gauss-Markov** theorem guarantees that among the class of linear unbiased estimators for β , $\hat{\beta}$ is the best in the sense that the variances of $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ are minimized. Consequently, $\hat{\beta}$ is called a best linear unbiased estimator, or a **BLUE**. Note that the error variance σ^2 is unknown, but its unbiased estimate is given by

$$\hat{\sigma}^2 = s^2 = MSE = \frac{SSE}{n-p} = \frac{\sum_{i=1}^n \hat{\varepsilon}_i^2}{n-p}. \quad (12.31)$$

If the matrix \mathbf{V} is defined to be $(\mathbf{X}'\mathbf{X})^{-1}$, then $\sigma_{\hat{\beta}_k}^2 = \sigma^2 \cdot v_{k+1,k+1}$, where $v_{k+1,k+1}$ is the $(k+1)^{\text{st}}$ diagonal entry ($k = 0, 1, \dots, p-1$) of \mathbf{V} . It is preferable to calculate \mathbf{V} with the command `summary(lm.object)$cov.unscaled`, where `lm.object` is a linear model object, rather than with the matrix computations `t(X)%*%X`, where \mathbf{X} is the design matrix. Since $\hat{\beta} \sim N(\beta, \sigma_{\hat{\beta}}^2)$, where $\sigma_{\hat{\beta}}^2 = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$, an estimate of $\sigma_{\hat{\beta}}^2$ is

$$\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1} = MSE(\mathbf{X}'\mathbf{X})^{-1} = \begin{bmatrix} s_{\hat{\beta}_0}^2 & s_{\hat{\beta}_0, \hat{\beta}_1} & \cdots & s_{\hat{\beta}_0, \hat{\beta}_{p-1}} \\ s_{\hat{\beta}_1, \hat{\beta}_0} & s_{\hat{\beta}_1}^2 & \cdots & s_{\hat{\beta}_1, \hat{\beta}_{p-1}} \\ \vdots & \vdots & \ddots & \vdots \\ s_{\hat{\beta}_{p-1}, \hat{\beta}_0} & s_{\hat{\beta}_{p-1}, \hat{\beta}_1} & \cdots & s_{\hat{\beta}_{p-1}}^2 \end{bmatrix} = s_{\hat{\beta}}^2. \quad (12.32)$$

The function `vcov()` will compute $s_{\hat{\beta}}^2$ when applied to a linear model object. A test statistic for testing $H_0 : \beta_k = \beta_{k_0}$ versus $H_1 : \beta_k \neq \beta_{k_0}$ can be justified using the standard form of a *t*-statistic:

$$\frac{\text{unbiased estimator} - \text{hypothesized value}}{\text{standard error of estimator}}.$$

Specifically, the test statistic is

$$\frac{\hat{\beta}_k - \beta_{k_0}}{s_{\hat{\beta}_k}} \sim t_{n-p} \text{ for } k = 0, 1, \dots, p-1. \quad (12.33)$$

In the event that the hypothesis of interest is $H_0 : \beta_k = 0$ versus $H_1 : \beta_k \neq 0$, the function `summary()` applied to a linear model object will provide the $t_{\text{obs}} = \hat{\beta}_k / s_{\hat{\beta}_k}$ value and the corresponding p -value. That is, the p -value = $2 \times \mathbb{P}(t_{n-p} \geq |t_{\text{obs}}|)$.

Using (12.33) as a pivotal quantity, in a similar fashion to the derivation of a confidence interval for μ in Section 8.2.2, a $100 \cdot (1 - \alpha)\%$ confidence interval for β_k , where $k = 0, 1, \dots, p - 1$, is

$$CI_{1-\alpha}(\beta_k) = \left[\hat{\beta}_k - t_{1-\alpha/2; n-p} \cdot s_{\hat{\beta}_k}, \hat{\beta}_k + t_{1-\alpha/2; n-p} \cdot s_{\hat{\beta}_k} \right]. \quad (12.34)$$

Note that the degrees of freedom for the t -distribution are $n - p$ because σ^2 is estimated with $MSE = \frac{SSE}{n-p}$.

Example 12.7 Consider Example 12.6 on page 792, where the admissions committee of a comprehensive state university wants to study the linear relationship between first-semester college grade point averages (`gpa`) and scholastic aptitude test (`sat`) scores. These are stored in the data frame `GRADES`. Assume that the requirements for model (12.4) are satisfied.

- (a) Find the variance-covariance matrix for $\hat{\beta}$ using (12.32).
- (b) Test whether there is a linear relationship at the $\alpha = 0.10$ significance level.
- (c) Construct 90% confidence intervals for β_0 and β_1 .

Solution: (a) Recall that $\hat{\varepsilon}_i = Y_i - \hat{Y}_i$, $\hat{\sigma}^2 = \sum_{i=1}^n \frac{\hat{\varepsilon}_i^2}{n-p}$, and the variance-covariance matrix is $s_{\hat{\beta}}^2 = \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$:

```
> model.lm <- lm(gpa ~ sat, data = GRADES)
> XTXI <- summary(model.lm)$cov.unscaled
> MSE <- summary(model.lm)$sigma^2
> var.cov.b <- MSE * XTXI
> var.cov.b
```

	(Intercept)	sat
(Intercept)	4.948408e-02	-4.290866e-05
sat	-4.290866e-05	3.781665e-08

$$s_{\hat{\beta}}^2 = \begin{bmatrix} 0.0495 & 0 \\ 0 & 0 \end{bmatrix}.$$

To compute $s_{\hat{\beta}}^2$ directly, type

```
> vcov(model.lm)

          (Intercept)           sat
(Intercept) 4.948408e-02 -4.290866e-05
sat         -4.290866e-05  3.781665e-08
```

- (b) The five-step procedure is used to test for a linear relationship between `sat` and `gpa`.

Step 1: **Hypotheses** — $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$.

Step 2: **Test Statistic** — $\hat{\beta}_1 = 0.0031$ is the test statistic. Assuming the assumptions of Model (12.4) are satisfied,

$$\hat{\beta}_1 \sim N(\beta_1, \sigma_{\hat{\beta}_1}^2).$$

The standardized test statistic under the assumption that H_0 is true and its distribution are

$$\frac{\hat{\beta}_1 - \beta_1}{s_{\hat{\beta}_1}} \sim t_{200-2}.$$

Step 3: **Rejection Region Calculations** — Because the standardized test statistic is distributed t_{198} and H_1 is a two-sided hypothesis, the rejection region is $|t_{\text{obs}}| > t_{0.95; 198} = 1.6526$. The value of the standardized test statistic is $t_{\text{obs}} = \frac{0.0031 - 0}{\sqrt{2e-04}} = 15.9117$.

Step 4: **Statistical Conclusion** — The p -value is $2 \times \mathbb{P}(t_{198} \geq 15.9117) = 2 \times 0 = 0$.

- I. From the rejection region, reject H_0 because $|15.9117|$ is greater than 1.6526.
- II. From the p -value, reject H_0 because the p -value = 0 is less than 0.10.

Step 5: **English Conclusion** — There is evidence to suggest a linear relationship between sat and gpa.

To see the test statistics and their p -values for the `model.lm`, enter

```
> summary(model.lm)$coef # lm coefficients

Estimate Std. Error t value   Pr(>|t|)
(Intercept) -1.19206381 0.222450180 -5.35879 2.316666e-07
sat          0.00309427 0.000194465 15.91171 2.922995e-37
```

(c) 90% Confidence intervals for β_0 and β_1 are

$$\begin{aligned} CI_{0.90}(\beta_0) &= \left[\hat{\beta}_0 - t_{.95; n-p} \cdot s_{\hat{\beta}_0}, \hat{\beta}_0 + t_{.95; n-p} \cdot s_{\hat{\beta}_0} \right] \\ CI_{0.90}(\beta_0) &= [-1.1921 - 1.6526(0.2225), -1.1921 + 1.6526(0.2225)] \\ CI_{0.90}(\beta_0) &= [-1.5597, -0.8244] \end{aligned}$$

and

$$\begin{aligned} CI_{0.90}(\beta_1) &= \left[\hat{\beta}_1 - t_{.95; n-p} \cdot s_{\hat{\beta}_1}, \hat{\beta}_1 + t_{.95; n-p} \cdot s_{\hat{\beta}_1} \right] \\ CI_{0.90}(\beta_1) &= [0.0031 - 1.6526(2e-04), 0.0031 + 1.6526(2e-04)] \\ CI_{0.90}(\beta_1) &= [0.0028, 0.0034]. \end{aligned}$$

R Code 12.3 on the next page computes the requested confidence intervals extracting the pieces needed for (12.33) as well as computing the answers directly with the function `confint()`.

R Code 12.3

```

> b0 <- coef(summary(model.lm))[1, 1]
> s.b0 <- coef(summary(model.lm))[1, 2]
> b1 <- coef(summary(model.lm))[2, 1]
> s.b1 <- coef(summary(model.lm))[2, 2]
> ct <- qt(1 - 0.1/2, 198) # alpha = 0.10
> CI.B0 <- b0 + c(-1, 1) * ct * s.b0
> CI.B0
[1] -1.5596818 -0.8244458

> CI.B1 <- b1 + c(-1, 1) * ct * s.b1
> CI.B1
[1] 0.00277290 0.00341564

> # Or
> confint(model.lm, level = 0.9)

      5 %         95 %
(Intercept) -1.5596818 -0.82444581
sat          0.0027729  0.00341564

```



12.9 ANOVA Approach to Regression

The basic normal error term regression model (12.4) has now been developed extensively. This same model can also be considered in an analysis of variance framework. This new paradigm will prove useful when working with multiple regression models. The analysis of variance approach is based on partitioning the sums of squares and the degrees of freedom associated with the response variable Y . The total deviation, $Y_i - \bar{Y}$, can be decomposed into two components:

1. The deviation of the fitted value \hat{Y}_i around the mean \bar{Y} and
2. The deviation of the observation Y_i around the regression line.

$$\underbrace{Y_i - \bar{Y}}_{\text{Total Deviation}} = \underbrace{\hat{Y}_i - \bar{Y}}_{\substack{\text{Deviation of Fitted} \\ \text{Regression Value} \\ \text{around the Mean}}} + \underbrace{Y_i - \hat{Y}_i}_{\substack{\text{Deviation around the} \\ \text{Fitted Regression Line}}} \quad (12.35)$$

Note that the total deviation is used to measure the variation of the Y_i s without taking the predictor variable(s) into account. Recall that since $\hat{\varepsilon}_i = Y_i - \hat{Y}_i$ and referring to regression properties on page 788,

$$\begin{aligned} 2 \sum_{i=1}^n (\hat{Y}_i - \bar{Y})(Y_i - \hat{Y}_i) &= 2 \sum_{i=1}^n (\hat{Y}_i - \bar{Y})\hat{\varepsilon}_i \\ &= 2 \sum_{i=1}^n \hat{Y}_i \hat{\varepsilon}_i - 2\bar{Y} \sum_{i=1}^n \hat{\varepsilon}_i \\ &= \underbrace{2 \times 0}_{\text{by Property (4)}} - \underbrace{2 \times 0}_{\text{by Property (1)}} = 0, \end{aligned} \quad (12.36)$$

so that squaring both sides and summing from $i = 1$ to n of (12.35) yields

$$\begin{aligned} \sum_{i=1}^n (Y_i - \bar{Y})^2 &= \sum_{i=1}^n [(\hat{Y}_i - \bar{Y}) + (Y_i - \hat{Y}_i)]^2 \\ &= \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2 + \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 + \underbrace{2 \sum_{i=1}^n (\hat{Y}_i - \bar{Y})(Y_i - \hat{Y}_i)}_{=0 \text{ by (12.36)}} \\ \sum_{i=1}^n (Y_i - \bar{Y})^2 &= \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2 + \sum_{i=1}^n (Y_i - \hat{Y}_i)^2. \end{aligned} \quad (12.37)$$

The expression in (12.37) is commonly expressed as $SST = SSR + SSE$, where SST denotes total sum of squares, SSR stands for regression sum of squares, and SSE represents error (residual) sum of squares. Figure 12.4 on the following page provides a graphical representation of the decomposition of the deviations of the observations, Y_i s, around the regression line (as in (12.35)) for a simple linear regression.

12.9.1 ANOVA with Simple Linear Regression

The degrees of freedom for SST are partitioned into degrees of freedom for SSR and degrees of freedom for SSE , just as the total sum of squares (SST) itself was partitioned into SSR and SSE . There are $n - 1$ degrees of freedom associated with SST . One degree of freedom is lost since the deviations $Y_i - \bar{Y}$ are subject to one constraint, specifically, $\sum_{i=1}^n (Y_i - \bar{Y})$ must equal zero, as it always does. Another explanation is that one degree of freedom is lost since \bar{Y} is used to estimate the population mean, μ .

SSE has $n - 2$ degrees of freedom. Two degrees of freedom are lost since two parameters, β_0 and β_1 , are estimated while obtaining the fitted values of \hat{Y}_i . There are two degrees of freedom associated with the estimated regression line, that is, one for the slope and one for the intercept; however, one of the degrees of freedom is lost since $\sum_{i=1}^n (\hat{Y}_i - \bar{Y})$ must equal zero by property 2. Consequently, SSR in a simple linear regression model has one degree of freedom.

When a sum of squares is divided by its associated degrees of freedom, the result is called a **mean square** and is denoted with MS . Specifically,

$$\frac{SSR}{1} = MSR \text{ and } \frac{SSE}{n-2} = MSE.$$

Mean squares, unlike sums of squares, are not additive. That is,

$$MST \neq MSR + MSE.$$

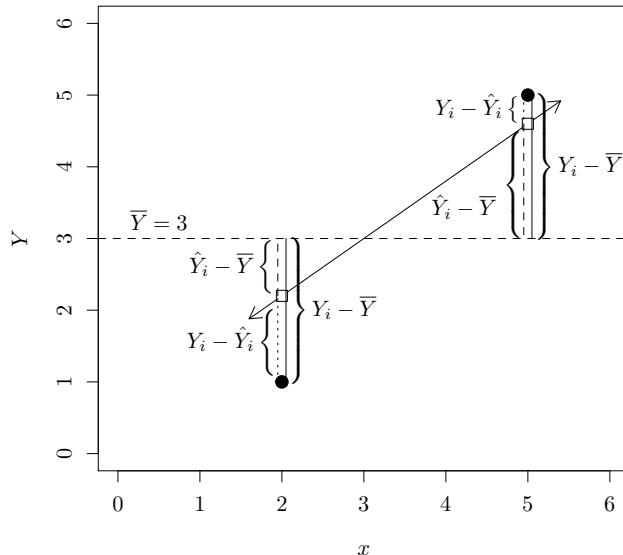


FIGURE 12.4: Decomposition of the deviations of the observations, Y_i s, around the regression line for a simple linear regression.

For the normal error regression model in (12.4), $\frac{SSE}{\sigma^2} \sim \chi_{n-2}^2$. Consequently,

$$E\left[\frac{SSE}{\sigma^2}\right] = n - 2 \implies E\left[\frac{SSE}{n-2}\right] = \sigma^2 \implies E[MSE] = \sigma^2.$$

In other words, the MSE is an unbiased estimator of σ^2 .

To find the expected value of MSR , recall from property 5 that $\bar{Y} = \hat{\beta}_0 + \hat{\beta}_1 \bar{x}$ and that the SSR for the simple linear model has one degree of freedom. This implies that $SSR = SSR/1 = MSR$. Also, note that the definition of the variance of $\hat{\beta}_1$ is $\sigma_{\hat{\beta}_1}^2 = E[\hat{\beta}_1^2] - (E[\hat{\beta}_1])^2 \Rightarrow E[\hat{\beta}_1^2] = \sigma_{\hat{\beta}_1}^2 + (E[\hat{\beta}_1])^2$.

$$\begin{aligned} SSR &= \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2 \\ SSR &= \sum_{i=1}^n ((\hat{\beta}_0 + \hat{\beta}_1 x_i) - (\hat{\beta}_0 + \hat{\beta}_1 \bar{x}))^2 \\ SSR &= \hat{\beta}_1^2 \sum_{i=1}^n (x_i - \bar{x})^2 \end{aligned}$$

Then, $E[SSR] = E[\hat{\beta}_1^2] \sum_{i=1}^n (x_i - \bar{x})^2$, since the x values are not random:

$$\begin{aligned} E[SSR] &= \left\{ \sigma_{\hat{\beta}_1}^2 + (E[\hat{\beta}_1])^2 \right\} \cdot \sum_{i=1}^n (x_i - \bar{x})^2 \\ &= \left\{ \underbrace{\frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}_{\text{by Example 12.5 and (12.26)}} + \beta_1^2 \right\} \cdot \sum_{i=1}^n (x_i - \bar{x})^2 \\ E[SSR] &= \sigma^2 + \beta_1^2 \cdot \sum_{i=1}^n (x_i - \bar{x})^2 = E[SSR/1] = \\ E[MSR] &= \sigma^2 + \beta_1^2 \cdot \sum_{i=1}^n (x_i - \bar{x})^2. \end{aligned}$$

Note that the mean of the sampling distribution of MSE is σ^2 whether a linear relationship exists between Y and x or not. The mean of the sampling distribution of MSR is also σ^2 when $\beta_1 = 0$. Consequently, MSR and MSE will be similar in magnitude when $\beta_1 = 0$. Likewise, when $\beta_1 \neq 0$, the center of the sampling distribution of MSR will be larger than the center of the sampling distribution of MSE by approximately $\beta_1^2 \cdot \sum_{i=1}^n (x_i - \bar{x})^2$.

In particular, the test statistic for testing $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$ for model (12.4) is

$$F_{\text{obs}} = \frac{MSR}{MSE}. \quad (12.38)$$

When the null hypothesis is true, $H_0 : \beta_1 = 0$, then

$$\frac{MSR}{MSE} \sim F_{1,n-2}.$$

Although it is beyond the scope of this text, it is noted that the quantities $\frac{SSR}{\sigma^2}$ and $\frac{SSE}{\sigma^2}$ are independent χ^2 random variables with 1 and $n - 2$ degrees of freedom, respectively. It then follows, using Definition 6.2, that

$$\frac{MSR}{MSE} = \frac{\frac{SSR/\sigma^2}{1}}{\frac{SSE/\sigma^2}{n-2}} = \frac{\chi_1^2/1}{\chi_{n-2}^2/(n-2)} \sim F_{1,n-2}.$$

Finally, values of F_{obs} close to 1 tend to support the null hypothesis, while large values of F_{obs} tend to support the alternative hypothesis. Specifically, the null hypothesis is rejected if $F_{\text{obs}} > f_{1-\alpha;1,n-2}$. R generates an ANOVA table on linear model objects with the function `anova(lm.object)`, which can be constructed using the equations from Table 12.1 on the next page.

Table 12.1: ANOVA table for simple linear regression

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Squares	F_{obs}
Regression	1	$SSR = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2$	$MSR = \frac{SSR}{1}$	$\frac{MSR}{MSE}$
Error	$n - 2$	$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$	$MSE = \frac{SSE}{n-2}$	
Total	$n - 1$	$SST = \sum_{i=1}^n (Y_i - \bar{Y})^2$		

Example 12.8 Construct an ANOVA table using the data in `GRADES`. Then, test if a linear relationship exists between first-semester college grade point average (`gpa`) and scholastic aptitude score (`sat`) using the information in the ANOVA table at the $\alpha = 0.05$ level.

Solution: R Code 12.4 is used to create Table 12.2.

R Code 12.4

```
> model.lm <- lm(gpa ~ sat, data = GRADES)
> anova(model.lm) # ANOVA

Analysis of Variance Table

Response: gpa
          Df Sum Sq Mean Sq F value    Pr(>F)
sat         1 40.397  40.397 253.18 < 2.2e-16 ***
Residuals 198 31.592   0.160
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 12.2: ANOVA table for `model.lm <- lm(gpa ~ sat)`

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
sat	1	40.3965	40.3965	253.1824	0
Residuals	198	31.5919	0.1596		

Step 1: **Hypotheses** — $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$.

Step 2: **Test Statistic** — F_{obs} since MSR/MSE under the assumption that $\beta_1 = 0$ has an $F_{1,198}$ distribution.

Step 3: **Rejection Region Calculations** — Because $F_{\text{obs}} \sim F_{1,198}$ and this is a one-tailed test, the rejection region is $F_{\text{obs}} > f_{0.95;1,198} = 3.8889$. The value of the standardized test statistic is $F_{\text{obs}} = \frac{40.3965}{0.1596} = 253.1824$.

Step 4: **Statistical Conclusion** — The p -value is $\mathbb{P}(F_{1,198} \geq 253.1824) = 0$.

- I. From the rejection region, reject H_0 because 253.1824 is greater than 3.8889.
- II. From the p -value, reject H_0 because the p -value = 0 is less than 0.05.

Step 5: **English Conclusion** — There is strong evidence to suggest a linear relationship exists between first-semester gpa and sat scores. 

12.9.2 ANOVA with Multiple Linear Regression

The ANOVA approach to linear regression analysis can be generalized to test hypotheses of the form

$$\begin{aligned} H_0 : \beta_1 = \beta_2 = \cdots = \beta_{p-1} = 0 \text{ versus} \\ H_1 : \text{at least one } \beta_i \neq 0 \text{ for } i = 1, 2, \dots, p-1. \end{aligned}$$

The ANOVA table for a multiple linear regression model with p parameters expressed in matrix notation is given in Table 12.3. Note that \mathbf{J} is an $n \times n$ matrix of 1s.

Table 12.3: ANOVA table for multiple linear regression

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Squares	F_{obs}
Regression	$p - 1$	$SSR = \hat{\beta}'\mathbf{X}'\mathbf{Y} - \frac{1}{n}\mathbf{Y}'\mathbf{J}\mathbf{Y}$	$MSR = \frac{SSR}{p-1}$	$\frac{MSR}{MSE}$
Error	$n - p$	$SSE = \mathbf{Y}'\mathbf{Y} - \hat{\beta}'\mathbf{X}'\mathbf{Y}$	$MSE = \frac{SSE}{n-p}$	
Total	$n - 1$	$SST = \mathbf{Y}'\mathbf{Y} - \frac{1}{n}\mathbf{Y}'\mathbf{J}\mathbf{Y}$		

An important matrix in the theory of linear models is the \mathbf{H} or “hat” matrix, defined as

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'. \quad (12.39)$$

The \mathbf{H} matrix is a symmetric, idempotent ($\mathbf{H}^2 = \mathbf{H}$), $n \times n$ matrix that transforms the Y_i s

into \hat{Y}_i s. Specifically,

$$\begin{aligned}\hat{\mathbf{Y}} &= \mathbf{X}\hat{\beta} \\ \hat{\mathbf{Y}} &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ \hat{\mathbf{Y}} &= \mathbf{HY}.\end{aligned}$$

The values for the sums of squares found in Table 12.3 on the preceding page can also be expressed in terms of the hat matrix as well as identity and \mathbf{J} matrices. Recall that $\hat{\beta}'\mathbf{X}'\mathbf{Y}$ is a 1×1 vector and is thus equal to its transpose:

$$\begin{aligned}SSE &= \mathbf{Y}'\mathbf{Y} - \hat{\beta}'\mathbf{X}'\mathbf{Y} \\ &= \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{X}\hat{\beta} \\ &= \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{HY} \\ &= \mathbf{Y}'(\mathbf{I} - \mathbf{H})\mathbf{Y}\end{aligned}\tag{12.40}$$

$$\begin{aligned}SSR &= \hat{\beta}'\mathbf{X}'\mathbf{Y} - \frac{1}{n}\mathbf{Y}'\mathbf{JY} \\ &= \mathbf{Y}'\mathbf{X}\hat{\beta} - \frac{1}{n}\mathbf{Y}'\mathbf{JY} \\ &= \mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} - \frac{1}{n}\mathbf{Y}'\mathbf{JY} \\ &= \mathbf{Y}'\mathbf{HY} - \frac{1}{n}\mathbf{Y}'\mathbf{JY} \\ &= \mathbf{Y}'(\mathbf{H} - \frac{1}{n}\mathbf{J})\mathbf{Y}\end{aligned}\tag{12.42}$$

$$\begin{aligned}SST &= \mathbf{Y}'\mathbf{Y} - \frac{1}{n}\mathbf{Y}'\mathbf{JY} \\ &= \mathbf{Y}'(\mathbf{I} - \frac{1}{n}\mathbf{J})\mathbf{Y}\end{aligned}\tag{12.41}$$

Thus it can be seen that each of the three sums of squares can be expressed as a quadratic form ($\mathbf{Y}'\mathbf{AY}$), where the \mathbf{A} matrices are $(\mathbf{I} - \mathbf{H})$, $(\mathbf{H} - \frac{1}{n}\mathbf{J})$, and $(\mathbf{I} - \frac{1}{n}\mathbf{J})$.

Knowing that the sums of squares are quadratic forms allows the statistician to prove various important results.

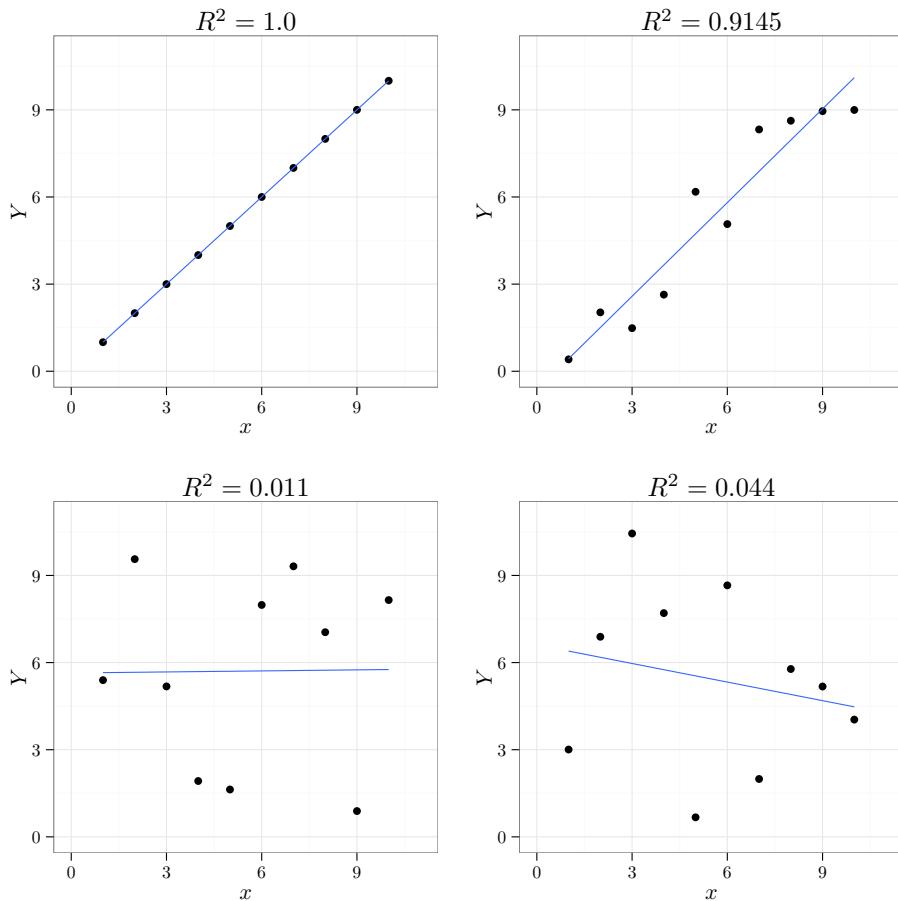
12.9.3 Coefficient of Determination

Figure 12.5 on the next page shows four different scatterplots of bivariate data. The points in the first scatterplot fall exactly on a straight line. Consequently, 100% of the variability in the y values can be attributed to the linear relationship between y and x . The points in the second scatterplot do not fall exactly along a line; however, the deviations from the least squares line (\hat{Y}_i) compared to the total deviations ($Y_i - \bar{Y}$) are relatively small. This makes it reasonable to conclude that a large proportion of the variability in y can be attributed to the linear relationship between y and x . The third and fourth scatterplots show both large deviations from the least squares lines as well as large total deviations. In these cases, a linear relationship between y and x is not overly helpful in explaining the variability of the y_i s exhibited in the scatterplots.

The sum of squares due to error (SSE) can be interpreted as the amount of variability in Y that is unexplained by a linear model. Since $SSE (\sum_{i=1}^n (Y_i - \hat{Y}_i)^2)$ is smaller than the sum of squared deviations of any other line, $SSE \leq SST$. Note that only in the case of a horizontal line would $SSE = SST$. Consequently, the ratio $\frac{SSE}{SST}$ represents the proportion of variability that cannot be explained by the linear regression model. In an analogous fashion, R^2 , the **coefficient of determination**, represents the proportion of variability in the Y_i s that can be explained by the simple linear regression model where

$$R^2 = 1 - \frac{SSE}{SST} = \frac{SSR}{SST}.\tag{12.43}$$

When working with linear regression models with $p - 1$ explanatory variables, R^2 is interpreted as the proportion of variability in the Y_i s that can be explained with a linear

FIGURE 12.5: Scatterplots to illustrate values of R^2

model containing the variables x_1, x_2, \dots, x_{p-1} . Since adding more x -variables to the regression model can only increase R^2 (as SSE never increases as more variables are added to a model and SST is constant for any set of Y_i values), a measure is needed that takes into account how many variables are in a model to determine the most appropriate variables to include. Such a measure is the **adjusted coefficient of determination**, R_a^2 . R_a^2 is computed by dividing each sum of squares by its associated degrees of freedom. That is,

$$R_a^2 = 1 - \frac{\frac{SSE}{n-p}}{\frac{SST}{n-1}} = 1 - \frac{MSE}{MST}. \quad (12.44)$$

Although R^2 and R_a^2 provide a certain measure of “goodness-of-fit” for the fitted model, they should be used with caution and never as the sole criterion for determining which among several models is best.

12.9.4 Extra Sum of Squares

An extra sum of squares measures the marginal increase in the regression sum of squares when one or more variables are added to a regression model. The marginal increase when

adding x_2 to a model that already contains x_1 will be denoted as

$$SSR(x_2|x_1) = SSR(x_2, x_1) - SSR(x_1), \quad (12.45)$$

which is equivalent to

$$SSR(x_2|x_1) = SSE(x_1) - SSE(x_1, x_2).$$

When the regression model contains r x -variables, there are $r!$ possible decompositions of the x -variables.

Example 12.9 The data frame `HSWRESTLER` contains information on nine variables for a group of 78 high school wrestlers that was collected by the human performance lab at Appalachian State University. The variables are `age` (in years), `ht` (height in inches), `wt` (weight in pounds), `abs` (abdominal skinfold measure), `triceps` (tricep skinfold measure), `subscap` (subscapular skinfold measure), `hwfat` (hydrostatic determination of fat), `tanfat` (Tanita determination of fat), and `skfat` (skinfold determination of fat). Use `hwfat` (Y), `abs` (x_1), and `triceps` (x_2) to verify empirically that

$$SSR(x_2|x_1) = SSR(x_2, x_1) - SSR(x_1)$$

and

$$SSR(x_2|x_1) = SSE(x_1) - SSE(x_1, x_2).$$

Solution: R Code 12.5 regresses Y on x_1 and x_2 and stores the result in `mod1`. Y is also regressed on x_1 alone and stored in `mod2`. The sum of squares for `mod1` and `mod2` are shown using the function `anova()`.

R Code 12.5

```
> Y <- HSWRESTLER$hwfat
> x1 <- HSWRESTLER$abs
> x2 <- HSWRESTLER$triceps
> mod1 <- lm(Y ~ x1 + x2)
> mod2 <- lm(Y ~ x1)
> anova(mod1)
```

Analysis of Variance Table

Response: Y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x1	1	5072.8	5072.8	541.365	< 2.2e-16 ***
x2	1	242.2	242.2	25.844	2.639e-06 ***
Residuals	75	702.8	9.4		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
> anova(mod2)
```

Analysis of Variance Table

Response: Y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x1	1	5072.8	5072.8	407.99	< 2.2e-16 ***
Residuals	76	945.0	12.4		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R Code 12.6 computes the quantities $SSR(x_2, x_1)$, $SSR(x_1)$, $SSR(x_2|x_1)$, $SSE(x_1)$ and the quantity $SSE(x_1, x_2)$. Note that both

$$SSR(x_2|x_1) = SSR(x_2, x_1) - SSR(x_1)$$

and

$$SSR(x_2|x_1) = SSE(x_1) - SSE(x_1, x_2)$$

return the same value (242.1727).

R Code 12.6

```
> SSRx1x2 <- anova(mod1)[1, 2] + anova(mod1)[2, 2] # SSR(x1, x2)
> SSRx1x2

[1] 5315.008

> SSRx1 <- anova(mod1)[1, 2] # SSR(x1)
> SSRx1

[1] 5072.835

> SSRx2Gx1 <- SSRx1x2 - SSRx1 # SSR(x2 | x1)
> SSRx2Gx1

[1] 242.1727

> SSEx1 <- anova(mod2)[2, 2] # SSE(x1)
> SSEx1

[1] 944.9564

> SSEx1x2 <- anova(mod1)[3, 2] # SSE(x1, x2)
> SSEx1x2

[1] 702.7837

> SSRx2Gx1 <- SSEx1 - SSEx1x2 # SSR(x2 | x1)
> SSRx2Gx1

[1] 242.1727
```

To help visualize the extra sum of squares, a schematic representation of the extra sum of squares for Example 12.9 on the facing page is shown in Figure 12.6 on the next page.

$$SSTO = 6017.792$$

$$SSR(x_1) = 5072.835$$

$$SSE(x_1) = 994.9564$$

$$SSR(x_2|x_1) = 242.1727 \rightarrow$$

$$SSR(x_1, x_2) = 5315.008$$

$$SSE(x_1, x_2) = 702.7837$$

$$SSR(x_1, x_2) = 5315.008$$

$$SSE(x_1, x_2) = 702.7837$$

$$SSR(x_1|x_2) = 258.7548 \rightarrow$$

$$SSR(x_2) = 5056.253$$

$$SSE(x_2) = 961.5385$$

$$SSTO = 6017.792$$

FIGURE 12.6: Schematic representation of extra sum of squares for Example 12.9 on page 808

■

Example 12.10 Consider the case where $r = 3$. What are the six decompositions of $SSR(x_1, x_2, x_3)$?

Solution:

$$SSR(x_1, x_2, x_3) = SSR(x_1) + SSR(x_2|x_1) + SSR(x_3|x_1, x_2) \quad (12.46)$$

$$SSR(x_1, x_3, x_2) = SSR(x_1) + SSR(x_3|x_1) + SSR(x_2|x_1, x_3)$$

$$SSR(x_2, x_1, x_3) = SSR(x_2) + SSR(x_1|x_2) + SSR(x_3|x_1, x_2)$$

$$SSR(x_2, x_3, x_1) = SSR(x_2) + SSR(x_3|x_2) + SSR(x_1|x_2, x_3)$$

$$SSR(x_3, x_1, x_2) = SSR(x_3) + SSR(x_1|x_3) + SSR(x_2|x_1, x_3)$$

$$SSR(x_3, x_2, x_1) = SSR(x_3) + SSR(x_2|x_3) + SSR(x_1|x_2, x_3) \quad (12.47)$$

■

Example 12.11 Use the data frame `HSWRESTLER` to obtain the ANOVA results when hydrostatic fat (Y) is regressed on ABS (x_1), TRICEPS (x_2), and SUBSCAP (x_3) to verify empirically the results from (12.46) and (12.47).

Solution: The order in which variables are specified in R impacts the ANOVA table since the sums of squares reported are conditional sums of squares. R Code 12.7 indirectly computes $SSR(x_1, x_2, x_3)$ and $SSR(x_3, x_2, x_1)$.

R Code 12.7

```
> mod1.HSW <- lm(hwfat ~ abs + triceps + subscap, data = HSWRESTLER)
> anova(mod1.HSW) # ANOVA

Analysis of Variance Table

Response: hwfat
            Df Sum Sq Mean Sq F value    Pr(>F)
abs          1 5072.8 5072.8 535.858 < 2.2e-16 ***
triceps      1  242.2   242.2 25.581 2.984e-06 ***
subscap       1     2.2     2.2  0.237   0.6278
Residuals 74  700.5     9.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> mod2.HSW <- lm(hwfat ~ subscap + triceps + abs, data = HSWRESTLER)
> anova(mod2.HSW) # ANOVA

Analysis of Variance Table

Response: hwfat
            Df Sum Sq Mean Sq F value    Pr(>F)
subscap      1 4939.0 4939.0 521.720 < 2.2e-16 ***
triceps      1  204.6   204.6 21.616 1.422e-05 ***
abs          1  173.6   173.6 18.341 5.473e-05 ***
Residuals 74  700.5     9.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the order of the x_i s does not impact the computation of SSR . The values used to compute $SSR(x_1, x_2, x_3)$ are taken from Table 12.4 on the next page, while the values used to compute $SSR(x_3, x_2, x_1)$ are taken from Table 12.5 on the following page.

$$\begin{aligned} SSR(x_1, x_2, x_3) &= SSR(x_1) + SSR(x_2|x_1) + SSR(x_3|x_1, x_2) \\ 5317.2517 &\stackrel{?}{=} 5072.8354 + 242.1727 + 2.2436 \\ 5317.2517 &= 5317.2517 \end{aligned}$$

$$\begin{aligned} SSR(x_3, x_2, x_1) &= SSR(x_3) + SSR(x_2|x_3) + SSR(x_1|x_2, x_3) \\ 5317.2517 &\stackrel{?}{=} 4938.9927 + 204.6305 + 173.6286 \\ 5317.2517 &= 5317.2517 \end{aligned}$$



Table 12.4: ANOVA table for `mod1.HSW`

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
abs	1	5072.8354	5072.8354	535.8578	0
triceps	1	242.1727	242.1727	25.5814	0
subscap	1	2.2436	2.2436	0.237	0.6278
Residuals	74	700.54	9.4668		

Table 12.5: ANOVA table for `mod2.HSW`

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
subscap	1	4938.9927	4938.9927	521.7196	0
triceps	1	204.6305	204.6305	21.6157	0
abs	1	173.6286	173.6286	18.3409	1e - 04
Residuals	74	700.54	9.4668		

12.9.4.1 Tests on a Single Parameter

To test whether the term $\beta_k x_k$ can be dropped from a multiple regression model, the hypotheses of interest are

$$H_0 : \beta_k = 0 \quad \text{versus} \quad H_1 : \beta_k \neq 0.$$

It was shown earlier in (12.33) that $t_{\text{obs}} = \hat{\beta}_k / s_{\hat{\beta}_k}$ could be used as an appropriate test statistic. It is also possible to test $\beta_k = 0$ using a general linear test statistic that involves an extra sum of squares. Consider a regression model with three predictor variables (which represent the full model). To test the hypothesis

$$H_0 : \beta_2 = 0 \quad \text{versus} \quad H_1 : \beta_2 \neq 0,$$

a reduced model where $\beta_2 x_2$ has been eliminated from the full model is computed. The general linear test statistic is

$$F_{\text{obs}} = \frac{\frac{SSR_F - SSR_R}{df_{rF} - df_{rR}}}{\frac{SSE_F}{df_{eF}}} = \frac{\frac{SSE_R - SSE_F}{df_{eR} - df_{eF}}}{\frac{SSE_F}{df_{eF}}} \quad (12.48)$$

where SSR and SSE represent the sum of squares due to regression and error, respectively. The notations dfe and dfr denote the degrees of freedom for error and regression, respectively, while the subscripts F and R represent the full and reduced models, respectively. This F_{obs} follows an F distribution with $(dfr_F - dfr_R = dfe_R - dfe_F, dfe_F)$ degrees of freedom, under the assumptions that H_0 is true and the normal error linear model assumptions are satisfied.

Example 12.12 Use the data frame `HSWRESTER` to show the equivalence between the t_{obs} and F_{obs} values when testing the hypothesis $H_0 : \beta_2 = 0$ versus $H_1 : \beta_2 \neq 0$ when regressing `hwfat` (Y) on `abs` (x_1), `triceps` (x_2), and `subscap` (x_3).

Solution: The important concept to remember is that both the t and F tests assume x_1 (`abs`) and x_3 (`subscap`) are in the model. Consequently, x_2 (`triceps`) must be entered into the model last.

R Code 12.8

```
> mod3.HSW <- lm(hwfat ~ abs + subscap + triceps, data = HSWRESTLER)
> anova(mod3.HSW) # ANOVA

Analysis of Variance Table

Response: hwfat
            Df Sum Sq Mean Sq F value    Pr(>F)
abs           1 5072.8 5072.8 535.858 < 2.2e-16 ***
subscap       1   132.6   132.6 14.005 0.0003577 ***
triceps       1   111.8   111.8 11.814 0.0009682 ***
Residuals 74   700.5      9.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(mod3.HSW)[3, 4] # Fobs value
[1] 11.81363

> coef(summary(mod3.HSW))

            Estimate Std. Error    t value    Pr(>|t|)
(Intercept) 2.06996703 0.65591984 3.1558232 2.314876e-03
abs          0.31893939 0.07447288 4.2826247 5.472651e-05
subscap      0.06631758 0.13622388 0.4868279 6.278192e-01
triceps      0.46069458 0.13403601 3.4370957 9.681801e-04

> coef(summary(mod3.HSW))[4, 3]^2 # tobs value squared
[1] 11.81363
```

The value of F_{obs} may also be found with function `drop1()`.

```
> drop1(mod3.HSW, test = "F") # Single term deletions

Single term deletions

Model:
hwfat ~ abs + subscap + triceps
            Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>              700.54 179.22
abs      1    173.629 874.17 194.49  18.341 5.473e-05 ***
subscap  1     2.244 702.78 177.47   0.237 0.6278192
triceps  1    111.837 812.38 188.77  11.814 0.0009682 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

R Code 12.9

```
> mod4.HSW <- lm(hwfat ~ abs + subscap, data = HSWRESTLER)
> anova(mod4.HSW) # ANOVA
```

Analysis of Variance Table

Response: hwfat

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
abs	1	5072.8	5072.8	468.33 < 2.2e-16 ***	
subscap	1	132.6	132.6	12.24 0.0007904 ***	
Residuals	75	812.4	10.8		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

From the ANOVA output for `mod3.HSW` and `mod4.HSW` in R Code 12.8 on the previous page and R Code 12.9, respectively, calculate

$$SSR_F = 5072.8354 + 132.5796 + 111.8367 = 5317.2517$$

$$SSR_R = 5072.8354 + 132.5796 = 5205.415,$$

which gives an F_{obs} value of

$$F_{\text{obs}} = \frac{\frac{SSR_F - SSR_R}{df_{r_F} - df_{r_R}}}{\frac{SSE_F}{df_{e_F}}} = \frac{\frac{5317.2517 - 5205.415}{3-2}}{\frac{700.54}{74}} = 11.8136.$$

The t_{obs} value is

$$t_{\text{obs}} = \frac{\hat{\beta}_2}{s_{\hat{\beta}_2}} = \frac{0.4607}{0.134} = 3.4371.$$

Recall from (6.30) that $t_{1-\alpha/2;\nu}^2 = f_{1-\alpha;1,\nu}$. So, the equivalence between the t_{obs} and F_{obs} values is equivalent to showing that $(3.4371)^2 = 11.8136$, which it does to four decimal places. One may also test whether $\beta_2 = 0$ using the `anova()` function. The output from the `anova()` function makes use of the second formulation of the F_{obs} given in (12.48) that uses only the SSE . Note that the SSE is denoted as `RSS` (residual sum of squares) with `anova()` output. A more general hypothesis for testing $H_0 : \beta_2 = 0$ versus $H_1 : \beta_2 \neq 0$ is that the two models specified by `mod4.HSW` and `mod3.HSW` are equivalent versus the alternative hypothesis that the full model (`mod3.HSW`) is superior. Since the p -value = 0.001, one can conclude that $\beta_2 = \beta_{\text{triceps}} \neq 0$, which is equivalent to stating that the full model (`mod3.HSW`) is superior to the reduced model (`mod4.HSW`).

```
> anova(mod4.HSW, mod3.HSW) # test models
```

Analysis of Variance Table

```
Model 1: hwfat ~ abs + subscap
Model 2: hwfat ~ abs + subscap + triceps
  Res.Df   RSS Df Sum of Sq    F    Pr(>F)
1      75 812.38
2      74 700.54  1     111.84 11.814 0.0009682 ***
---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



12.9.4.2 Tests on Subsets of the Regression Parameters

Traditional tests of hypotheses on individual regression coefficients generated with the R command `anova(lm.object)` are partitions of the regression sum of squares. Frequently, the user may want to test hypotheses containing a subset of the regression parameters. To test whether a subset of the regression parameters are equal to zero requires the general linear statistic in (12.48) on page 812. Consider a full model with three predictors, x_1 , x_2 , and x_3 . To see if $\beta_1 x_{i1}$ and $\beta_2 x_{i2}$ can be dropped from the model, the test of hypothesis is

$$H_0 : \beta_1 = \beta_2 = 0 \text{ versus } H_1 : \text{at least one } \beta_i \neq 0 \text{ for } i = 1, 2.$$

The full model is $Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$ while the reduced model is $Y_i = \beta_0 + \beta_3 x_{i3} + \varepsilon_i$. Consequently, the general linear test statistic will be

$$F_{\text{obs}} = \frac{\frac{SSR_F - SSR_R}{df_{rF} - df_{rR}}}{\frac{SSE_F}{df_{eF}}} = \frac{\frac{SSE_R - SSE_F}{df_{eR} - df_{eF}}}{\frac{SSE_F}{df_{eF}}}.$$

Example 12.13 Use the data frame `HSWRESTLER` and test whether $\beta_1 x_{i1}$ and $\beta_2 x_{i2}$ can be dropped from the full model $Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$, where $x_1 = \text{abs}$, $x_2 = \text{triceps}$, and $x_3 = \text{subscap}$ using the general linear test approach with a 0.01 significance level.

Solution: To test $H_0 : \beta_1 = \beta_2 = 0$ versus $H_1 : \text{at least one } \beta_i \neq 0 \text{ for } i = 1, 2$, one must ensure x_3 (`subscap`) is the first variable specified in the model formulation.

R Code 12.10

```
> mod4.HSW <- lm(hwfat ~ subscap, data = HSWRESTLER)
> mod5.HSW <- lm(hwfat ~ subscap + abs + triceps, data = HSWRESTLER)
> anova(mod4.HSW, mod5.HSW) # test models
```

Analysis of Variance Table

	Model 1: hwfat ~ subscap	Model 2: hwfat ~ subscap + abs + triceps				
	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	76	1078.80				
2	74	700.54	2	378.26	19.978	1.154e-07 ***

						Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

$$F_{\text{obs}} = \frac{\frac{SSE_R - SSE_F}{df_{eR} - df_{eF}}}{\frac{SSE_F}{df_{eF}}} = \frac{\frac{1078.7991 - 700.54}{76 - 74}}{\frac{700.54}{74}} = 19.9783.$$

Since $p\text{-value} = \mathbb{P}(F_{2,74} \geq 19.9783) = 0$, reject H_0 and declare the results statistically significant. The evidence suggests that `abs` and `triceps` should not be dropped from a model that already contains `subscap`. In other words, `mod5.HSW` is a superior model compared to `mod4.HSW`.

12.10 General Linear Hypothesis

Tests on individual parameters and on subsets of parameters can be expressed in a much more general fashion that provides fantastic flexibility in testing. The **general linear hypotheses** are

$$H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{m} \quad \text{versus} \quad H_1 : \mathbf{K}\boldsymbol{\beta} \neq \mathbf{m} \quad (12.49)$$

where \mathbf{K} is a $q \times p$ matrix of rank $q \leq p$ with each row corresponding to one partial hypothesis and \mathbf{m} is a numerical vector. When working with the normal error model, where $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$ and $\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2 (\mathbf{X}'\mathbf{X})^{-1})$, the parameter vector $\mathbf{K}\boldsymbol{\beta}$ is estimated by $\mathbf{K}\hat{\boldsymbol{\beta}}$, which is a linear combination of normally distributed random variables. This implies $\mathbf{K}\hat{\boldsymbol{\beta}} \sim N(\mathbf{K}\boldsymbol{\beta}, \sigma^2 \mathbf{K}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{K}')$. When \mathbf{K} is a vector ($q = 1$), the null hypothesis $\mathbf{K}\boldsymbol{\beta} = \mathbf{m}_0$ is tested with a t -statistic since

$$t = \frac{\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m}_0}{\sqrt{\hat{\sigma}^2 \mathbf{K}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{K}'}} \sim t_{n-p,\gamma} \quad (12.50)$$

where

$$\gamma = \frac{\mathbf{K}\boldsymbol{\beta} - \mathbf{m}_0}{\sqrt{\sigma^2 \mathbf{K}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{K}'}}. \quad (12.51)$$

Under the null hypothesis, $\gamma = 0$ and (12.50) is a central t -distribution with $n - p$ degrees of freedom. In the general linear hypothesis, only a two-sided alternative is given; however, when $q = 1$, the one-sided alternative, $H_1 : \mathbf{K}\boldsymbol{\beta} > \mathbf{m}$ or $H_1 : \mathbf{K}\boldsymbol{\beta} < \mathbf{m}$, may be specified and tested using (12.50).

When the rank of \mathbf{K} is greater than one ($q > 1$), the quantity

$$\frac{(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})'(\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{K}')^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})}{q\hat{\sigma}^2} \sim F_{q, n-p, \lambda} \quad (12.52)$$

where

$$\lambda = \frac{1}{\sigma^2} (\mathbf{K}\boldsymbol{\beta} - \mathbf{m})'(\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{K}')^{-1}(\mathbf{K}\boldsymbol{\beta} - \mathbf{m}) \quad (12.53)$$

is used to test the null hypothesis $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{m}$. For details, see Graybill (1976). Note that Graybill does offer a slightly different definition of λ (his λ is that of (12.53) divided by 2), but this does not complicate the exposition. This text will use the definition used by R and Rao (1973). By using (12.53), the relationship between γ and λ is $\lambda = \gamma^2$. The square of the non-central t -statistic with non-centrality parameter γ is distributed as a non-central F -statistic with non-centrality parameter $\lambda = \gamma^2$. In other words, $t_{\nu, \gamma}^2 = f_{1, \nu, \gamma^2}$. The power of the test $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{m}$ as a function of λ for a given α is $\text{Power}(\lambda) = \mathbb{P}(F_{\nu_1, \nu_2, \lambda} > f_{1-\alpha; \nu_1, \nu_2})$.

When $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{m}$ is true,

$$\frac{(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})'(\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{K}')^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})}{\sigma^2} \sim \chi_q^2 \quad (12.54)$$

and

$$F_{\text{obs}} = \frac{(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})'(\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1} \mathbf{K}')^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})}{q\hat{\sigma}^2} \sim F_{q, n-p}. \quad (12.55)$$

To test $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{m}$ at the α significance level, reject H_0 and conclude H_1 when $\mathbb{P}(F_{q, n-p} \geq F_{\text{obs}}) < \alpha$. The function `glht()` in the R package `multcomp` as well as function `linearHypothesis()` in the R package `car` can greatly ease the computation of general linear hypotheses, especially when testing for linear relationships among the β s.

Example 12.14 ▷ **General Linear Model** ◷ Use a general linear hypothesis with $\alpha = 0.05$ to

- (a) Test whether $\beta_2 x_{i2}$ and $\beta_3 x_{i3}$ can be dropped from the model $Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$, where $x_1 = \text{subscap}$, $x_2 = \text{abs}$, and $x_3 = \text{triceps}$, using information from the data frame **HSWRESTLER**.

- (b) Test the two linear relationships

$$\begin{aligned} 2\beta_1 + \beta_2 &= \beta_3 \\ -5\beta_1 + \beta_3 &= 0.20. \end{aligned}$$

- (c) Test whether $\beta_2 = \beta_3$.

Solution: Since there are 78 wrestlers in the data frame **HSWRESTLER** and the full model requires estimating four coefficients, $n = 78$ and $p = 4$.

- (a)

Step 1: **Hypotheses** —

$$H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{m} \quad \text{versus} \quad H_1 : \mathbf{K}\boldsymbol{\beta} \neq \mathbf{m}$$

where $\mathbf{K} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $\mathbf{m} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Step 2: **Test Statistic** — Under the assumption H_0 is true,

$$F_{\text{obs}} = \frac{(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})'(\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{K}')^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})}{q\hat{\sigma}^2} \sim F_{q, n-p}.$$

Step 3: **Rejection Region Calculations** — Because $F_{\text{obs}} \sim F_{2, 74}$ and this is a one-tailed test, the rejection region is $F_{\text{obs}} > f_{0.95; 2, 74} = 3.1203$. The value of the standardized test statistic is $F_{\text{obs}} = 19.9783$:

$$F_{\text{obs}} = \frac{[0.319 \quad 0.461] \begin{bmatrix} 1819.5992 & 251.6221 \\ 251.6221 & 561.7319 \end{bmatrix} \begin{bmatrix} 0.319 \\ 0.461 \end{bmatrix}}{2(9.467)} = 19.9783.$$

Step 4: **Statistical Conclusion** — The ϕ -value is $\mathbb{P}(F_{2, 74} \geq 19.9783) = 0$.

- I. From the rejection region, reject H_0 because 19.9783 is greater than 3.1203.
- II. From the ϕ -value, reject H_0 because the ϕ -value = 0 is less than 0.05.

Step 5: **English Conclusion** — There is strong evidence to suggest a linear relationship exists between **abs**, **triceps**, and **hwfat**, suggesting neither variable should be dropped from a model that currently contains **subscap**.

The F_{obs} value as well as the ϕ -value are computed in R Code 12.11 on the next page.

R Code 12.11

```
> K <- matrix(c(0, 0, 1, 0, 0, 0, 0, 1), nrow = 2, byrow = TRUE)
> q <- qr(K)$rank # computes the rank of K
> mod5.HSW <- lm(hwfat ~ subscap + abs + triceps, data = HSWRESTLER)
> b <- matrix(coef(mod5.HSW), ncol = 1) # vector of beta hats
> m <- matrix(0, byrow = TRUE, nrow = 2) # right hand side
> XTXI <- summary(mod5.HSW)$cov.unscaled #  $X'X^{-1}$  matrix
> NUM <- t(K %*% b - m) %*% solve(K %*% XTXI %*% t(K)) %*%
+      (K %*% b - m)
> MSE <- anova(mod5.HSW)[4, 3]
> Fobs <- NUM/(q * MSE)
> pvalue <- pf(Fobs, q, 74, lower = FALSE)
> ANS <- c(Fobs = Fobs, pvalue = pvalue)
> ANS

Fobs      pvalue
1.997828e+01 1.154032e-07
```

Using the function `linearHypothesis()` from the `car` package facilitates the computations as shown in R Code 12.12. Note the three arguments and their values: `model` a fitted linear model object set to `mod5.HSW`, `hypothesis.matrix` a matrix specifying the linear combinations to be tested set equal to `K`, and `rhs` the right-hand-side vector `m` for testing $\mathbf{K}\beta = \mathbf{m}$ set to `m`.

R Code 12.12

```
> library(car) # load package car
> linearHypothesis(model = mod5.HSW, hypothesis.matrix = K,
+   rhs = m)

Linear hypothesis test

Hypothesis:
abs = 0
triceps = 0

Model 1: restricted model
Model 2: hwfat ~ subscap + abs + triceps

  Res.Df    RSS Df Sum of Sq    F    Pr(>F)
1     76 1078.80
2     74  700.54  2    378.26 19.978 1.154e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(b)

Step 1: **Hypotheses** —

$$H_0 : \mathbf{K}\beta = \mathbf{m} \quad \text{versus} \quad H_1 : \mathbf{K}\beta \neq \mathbf{m}$$

where $\mathbf{K} = \begin{bmatrix} 0 & 2 & 1 & -1 \\ 0 & -5 & 0 & 1 \end{bmatrix}$ and $\mathbf{m} = \begin{bmatrix} 0 \\ .2 \end{bmatrix}$.

Step 2: **Test Statistic** — Under the assumption H_0 is true,

$$F_{\text{obs}} = \frac{(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})'(\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{K}')^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})}{q\hat{\sigma}^2} \sim F_{q, n-p}.$$

Step 3: **Rejection Region Calculations** — Because $F_{\text{obs}} \sim F_{2, 74}$ and this is a one-tailed test, the rejection region is $F_{\text{obs}} > f_{0.95; 2, 74} = 3.1203$. The value of the standardized test statistic is $F_{\text{obs}} = 0.0623$:

$$F_{\text{obs}} = \frac{[-0.009 \quad -0.071] \begin{bmatrix} 676.0834 & 297.8446 \\ 297.8446 & 146.8894 \end{bmatrix} \begin{bmatrix} -0.009 \\ -0.071 \end{bmatrix}}{2(9.467)} = 0.0623.$$

Step 4: **Statistical Conclusion** — The φ -value is $\mathbb{P}(F_{2, 74} \geq 0.0623) = 0.9396$.

- I. From the rejection region, fail to reject H_0 because 0.0623 is less than 3.1203.
- II. From the φ -value, fail to reject H_0 because the φ -value = 0.9396 is greater than 0.05.

Step 5: **English Conclusion** — There is no evidence to suggest the postulated relationships $\mathbf{K}\boldsymbol{\beta} \neq \mathbf{m}$.

The F_{obs} value as well as the φ -value are computed in R Code 12.13.

R Code 12.13

```
> K <- matrix(c(0, 2, 1, -1, 0, -5, 0, 1), nrow = 2, byrow = TRUE)
> q <- qr(K)$rank                                     # computes the rank of K
> mod5.HSW <- lm(hwfat ~ subscap + abs + triceps, data = HSWRESTLER)
> b <- matrix(coef(mod5.HSW), ncol = 1)              # vector of beta hats
> m <- matrix(c(0, 0.2), byrow = TRUE, nrow = 2)      # right hand side
> XTXI <- summary(mod5.HSW)$cov.unscaled            # X'X^-1 matrix
> NUM <- t(K%*%b - m)%*%solve(K%*%XTXI%*%t(K))%*%(K%*%b - m)
> MSE <- anova(mod5.HSW)[4,3]
> Fobs <- NUM/(q*MSE)
> pvalue <- pf(Fobs, q, 74, lower = FALSE)
> ANS <- c(Fobs = Fobs, pvalue = pvalue)
> ANS

      Fobs      pvalue
0.06230336 0.93964704
```

R Code 12.14 uses the function `linearHypothesis()` from the package `car` as well as the function `glht()` (generalized linear hypothesis test) from the `multcomp` package to verify the values from R Code 12.13.

R Code 12.14

```
> library(multcomp) # load package multcomp
> linearHypothesis(model = mod5.HSW, hypothesis.matrix = K, rhs = m)
```

```
Linear hypothesis test
```

Hypothesis:

```
2 subscap + abs - triceps = 0
- 5 subscap + triceps = 0.2
```

Model 1: restricted model

Model 2: hwfat ~ subscap + abs + triceps

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	76	701.72				
2	74	700.54	2	1.1796	0.0623	0.9396

```
> summary(glht(model = mod5.HSW, linfct = K, rhs = m), test = Ftest())
```

General Linear Hypotheses

Linear Hypotheses:

	Estimate
1 == 0	-0.00912
2 == 0.2	0.12911

Global Test:

	F	DF1	DF2	Pr(>F)
1	0.0623	2	74	0.9396

(c)

Step 1: **Hypotheses** —

$$H_0 : \mathbf{K}\beta = \mathbf{m} \quad \text{versus} \quad H_1 : \mathbf{K}\beta \neq \mathbf{m}$$

where $\mathbf{K} = [0 \ 0 \ 1 \ -1]$ and $\mathbf{m} = [0]$.

Step 2: **Test Statistic** — Under the assumption H_0 is true,

$$F_{\text{obs}} = \frac{(\mathbf{K}\hat{\beta} - \mathbf{m})'(\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{K}')^{-1}(\mathbf{K}\hat{\beta} - \mathbf{m})}{q\hat{\sigma}^2} \sim F_{q, n-p}.$$

Step 3: **Rejection Region Calculations** — Because $F_{\text{obs}} \sim F_{1, 74}$ and this is a one-tailed test, the rejection region is $F_{\text{obs}} > f_{0.95; 1, 74} = 3.9702$. The value of the standardized test statistic is $F_{\text{obs}} = 0.7056$:

$$F_{\text{obs}} = \frac{[-0.1418] [332.3932] [-0.1418]}{1(9.4668)} = 0.7056.$$

Step 4: **Statistical Conclusion** — The φ -value is $\mathbb{P}(F_{1, 74} \geq 0.7056) = 0.4036$.

- I. From the rejection region, fail to reject H_0 because 0.7056 is less than 3.9702.
- II. From the φ -value, fail to reject H_0 because the φ -value = 0.4036 is greater than 0.05.

Step 5: **English Conclusion** — There is no evidence to suggest $\beta_2 \neq \beta_3$.

The F_{obs} value as well as the p -value are computed in R Code 12.15.

R Code 12.15

```
> K <- matrix(c(0, 0, 1, -1), nrow = 1, byrow = TRUE)
> q <- qr(K)$rank                                     # computes the rank of K
> mod5.HSW <- lm(hwfat ~ subscap + abs + triceps, data = HSWRESTLER)
> b <- matrix(coef(mod5.HSW), ncol = 1)              # vector of beta hats
> m <- matrix(c(0), byrow = TRUE, nrow = 1)           # right hand side
> XTXI <- summary(mod5.HSW)$cov.unscaled            #  $X'X^{-1}$  matrix
> NUM <- t(K %*% b - m) %*% solve(K %*% XTXI %*% t(K)) %*% (K %*% b - m)
> MSE <- anova(mod5.HSW)[4,3]
> Fobs <- NUM/(q*MSE)
> pvalue <- pf(Fobs, q, 74, lower = FALSE)
> ANS <- c(Fobs = Fobs, pvalue = pvalue)
> ANS

      Fobs      pvalue
0.7055517 0.4036302
```

R Code 12.16 uses the function `linearHypothesis()` from the package `car` as well as the function `glht()` (generalized linear hypothesis test) from the `multcomp` package to verify the values from R Code 12.15.

R Code 12.16

```
> library(multcomp) # load package multcomp
> linearHypothesis(model = mod5.HSW, hypothesis.matrix = K, rhs = m)

Linear hypothesis test

Hypothesis:
abs - triceps = 0

Model 1: restricted model
Model 2: hwfat ~ subscap + abs + triceps

  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     75 707.22
2     74 700.54  1   6.6793 0.7056 0.4036

> summary(glht(model = mod5.HSW, linfct = K, rhs = m), test = Ftest())

General Linear Hypotheses

Linear Hypotheses:
  Estimate
1 == 0 -0.1418

Global Test:
  F DF1 DF2 Pr(>F)
1 0.7056   1  74 0.4036
```

12.11 Model Building

The process of building a model either for predictive or explanatory purposes involves several procedures, where the order of the procedures is not always the same. One needs always to bear in mind that regression analysis is simply a tool to understand the structure of data. In what follows, general methods to select models, to verify assumptions, and to perform transformations on both the response and predictor variables are discussed. Although Figure 12.7 on the next page provides a flow chart for how one might build a model, the analyst should always be alert for an unexpected structure in the data and be flexible in his assessment of the model.

12.11.1 Testing-Based Procedures

When building a model, it is desirable to select the “best” subset of predictors that explains the data in the simplest fashion. Adding too many variables wastes degrees of freedom and adds unwanted noise to the problem, increases the risk of adding variables that measure the same quantity, as well as increasing the effort needed to measure the redundant predictors. There are two basic approaches one can take to select variables: 1) a stepwise testing strategy that compares successive models, and 2) a criterion approach that attempts to maximize some measure of goodness-of-fit.

12.11.1.1 Backward Elimination

Backward elimination begins with a model containing all potential x -variables and identifies the one with the largest p -value. This can be done by looking at the p -values for the t -values of the $\hat{\beta}_i, i = 1, \dots, p - 1$ using the function `summary()` or by using the p -values from the function `drop1()`. If the variable with the largest p -value is above a predetermined value, α_{crit} , that x -variable is dropped. A model with the remaining x -variables is then fit and the procedure continues until all the p -values for the remaining variables in the model are below the predetermined α_{crit} . The α_{crit} is sometimes referred to as the “ p -to-remove” and is typically set to 15 or 20%.

12.11.1.2 Forward Selection

Forward selection starts with no variables in the model and then adds the x -variable that produces the smallest p -value below α_{crit} when included in the model. This procedure is continued until no new predictors can be added. The user can determine the variable that produces the smallest p -value by regressing the response variable on the x_i s one at a time using `lm()` and `summary()` or by using the function `add1()`.

12.11.1.3 Stepwise Regression

This is a combination of backward elimination and forward selection. This technique allows variables that were either removed or added early in the procedure to reenter or exit the model later in the process. At each stage, a variable may be added or removed.

Testing-based procedures are relatively straightforward to implement; however, they do have some drawbacks. One of the chief weaknesses of testing-based procedures is ending up with a model that is overly parsimonious. When the analyst has a firm grasp of the subject matter, the analyst may want to include predictors that appear to have no statistical significance. Although predictors can be added to a model developed from a testing-based

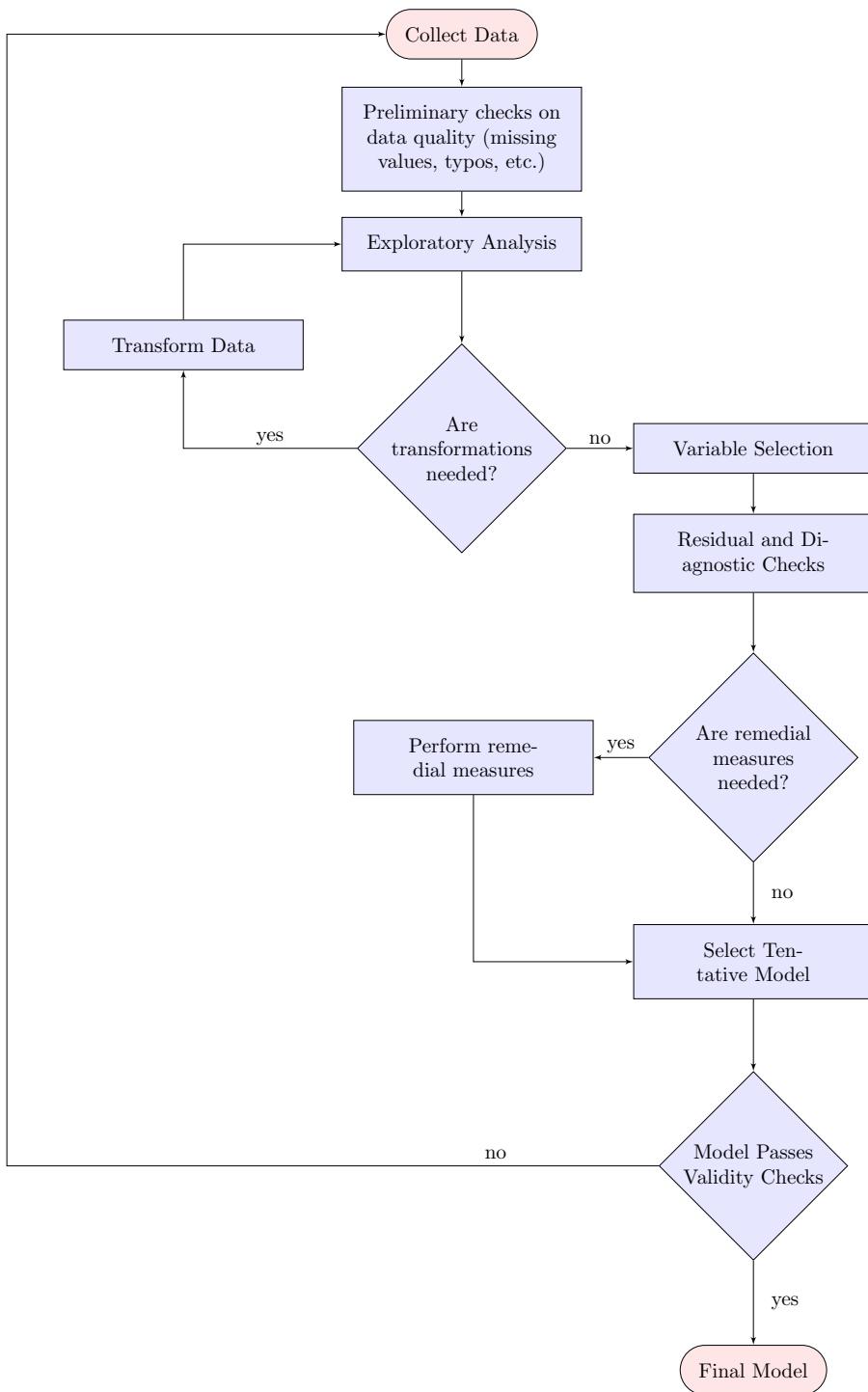


FIGURE 12.7: Regression model building flow chart modified from Neter et al. (1996, Figure 8.1)

perspective, the idea of adding predictors that are not necessarily significant conforms more to a criterion-based procedure.

Example 12.15  **Backward Elimination and Forward Selection**  Create regression models for predicting wrestlers' hydrostatic fat (`hwfat`) using the data frame `HSWRESTLER`.

- Perform quality checks on the data in `HSWRESTLER` as well as some basic exploratory analysis to see how the predictors (`age`, `ht`, `wt`, `abs`, `triceps`, and `subscap`) are related to `hwfat`.
- Use backward elimination with the predictors `age`, `ht`, `wt`, `abs`, `triceps`, and `subscap` and an α_{crit} of 0.20 to create a regression model where `hwfat` is the response variable.
- Use forward selection with the predictors `age`, `ht`, `wt`, `abs`, `triceps`, and `subscap` and an α_{crit} of 0.20 to create a regression model where `hwfat` is the response variable.

Solution: The package `car` has several extremely useful functions that will be used throughout the chapter including the function `scatterplotMatrix()` for exploring relationships among the predictors and `hwfat`.

- The function `summary()` is used to perform an initial check of the data frame `HSWRESTLER`

R Code 12.17

```
> summary(HSWRESTLER)
```

age	ht	wt	abs
Min. :13.00	Min. :56.00	Min. : 93.4	Min. : 6.00
1st Qu.:15.00	1st Qu.:64.56	1st Qu.:125.1	1st Qu.: 9.00
Median :16.00	Median :67.50	Median :142.2	Median :11.00
Mean :15.72	Mean :66.79	Mean :154.1	Mean :16.73
3rd Qu.:17.00	3rd Qu.:69.00	3rd Qu.:171.3	3rd Qu.:19.00
Max. :18.00	Max. :73.00	Max. :299.2	Max. :54.00
triceps	subscap	hwfat	tanfat
Min. : 6.000	Min. : 6.00	Min. : 3.580	Min. : 6.40
1st Qu.: 7.625	1st Qu.: 8.00	1st Qu.: 8.727	1st Qu.:11.95
Median :10.000	Median : 9.00	Median :11.135	Median :14.00
Mean :12.962	Mean :12.94	Mean :14.235	Mean :16.73
3rd Qu.:14.000	3rd Qu.:12.00	3rd Qu.:15.418	3rd Qu.:18.20
Max. :42.000	Max. :43.00	Max. :41.890	Max. :44.90
skfat			
Min. : 8.120			
1st Qu.: 9.988			
Median :11.160			
Mean :15.152			
3rd Qu.:15.165			
Max. :41.090			

Based on output from R Code 12.17, the data contains no missing values and the range of values for all of the variables seems reasonable. Since no obvious abnormalities are noted in the data, the function `scatterplotMatrix()` is used in R Code 12.18 on the next page to create Figure 12.8 on the facing page. There are a number of arguments/options for the function `scatterplotMatrix()` and the reader should refer to its help page for further details.

R Code 12.18

```
> library(car)
> scatterplotMatrix(x = HSWRESTLER[, -c(8:9)]) # remove tanfat and skfat
> # Or
> scatterplotMatrix(formula = ~ hwt + age + ht + wt + abs + triceps +
+                     subscap, data = HSWRESTLER)
```

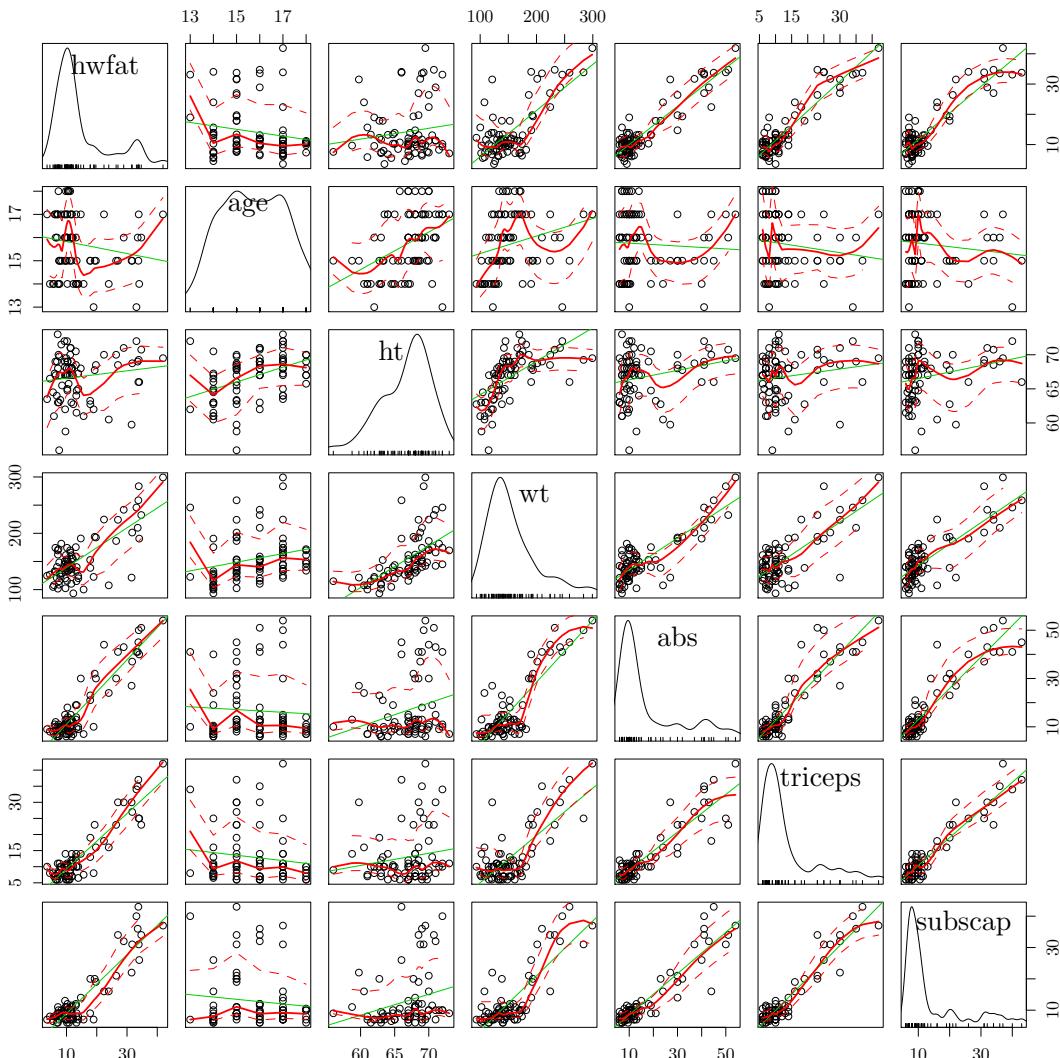


FIGURE 12.8: Enhanced scatterplot matrices with univariate density estimates on the diagonal. Superimposed straight lines are computed using the function `r1m()` from the MASS package.

Based on Figure 12.8, the variables `abs`, `triceps`, and `subscap` appear to have the strongest linear relationship with `hwt`. The strength of the linear relationships between the response

`hwfat` and the potential predictors is computed in R Code 12.19.

R Code 12.19

```
> cor(HSWRESTLER[, -c(8:9)]) # remove tanfat and skfat
```

	age	ht	wt	abs	triceps	subscap
age	1.00000000	0.4177471	0.2406721	-0.05970301	-0.1349368	-0.1077762
ht	0.41774714	1.0000000	0.6044705	0.25882739	0.1578298	0.2525007
wt	0.24067211	0.6044705	1.0000000	0.84808755	0.7526595	0.8149226
abs	-0.05970301	0.2588274	0.8480876	1.00000000	0.9057768	0.9254833
triceps	-0.13493683	0.1578298	0.7526595	0.90577680	1.0000000	0.9455530
subscap	-0.10777616	0.2525007	0.8149226	0.92548332	0.9455530	1.0000000
hwfat	-0.17053777	0.1399135	0.7334894	0.91813556	0.9166337	0.9059425
	hwfat					
age	-0.1705378					
ht	0.1399135					
wt	0.7334894					
abs	0.9181356					
triceps	0.9166337					
subscap	0.9059425					
hwfat	1.0000000					

From the output of R Code 12.19, one can verify that the correlations between `hwfat` and `abs`, `hwfat` and `triceps`, and `hwfat` and `subscap` are 0.9181, 0.9166, and 0.9059, respectively.

(b) Backward elimination starts with all the variables in the model (`model.all`) and eliminates variables with the largest (least significant) p -values. The period in the short-hand notation `hwfat ~ .` tells R to include all of the variables specified in the `data` argument. In this case, the variables `tanfat` and `skfat` are removed using negative indices.

```
> model.all <- lm(hwfat ~ ., data = HSWRESTLER[, -c(8, 9)])
> drop1(model.all, test = "F") # single term deletions
```

Single term deletions

Model:

```
hwfat ~ age + ht + wt + abs + triceps + subscap
      Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>          651.05 179.51
age     1     9.594 660.64 178.65  1.0463  0.309839
ht      1     1.613 652.66 177.70  0.1759  0.676225
wt      1     2.546 653.60 177.81  0.2777  0.599879
abs     1    162.000 813.05 194.84 17.6669 7.549e-05 ***
triceps 1    72.683 723.73 185.76  7.9264  0.006301 **
subscap 1     5.921 656.97 178.21  0.6458  0.424315
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the largest p -value from `drop1()` is 0.6762, which corresponds to the variable `ht`, one needs to update the original model by dropping the variable `ht`. The function `update()` allows the user to update a previous model (`model.all`) using the short-hand notation

. ~ . -ht. The periods on the left and right side of the tilde tell R to use what is specified in the left- and right-hand sides of the tilde in `model.all` minus the variable `ht`.

```
> mod.hsw <- update(model.all, . ~ . - ht)
> drop1(mod.hsw, test = "F") # single term deletions

Single term deletions

Model:
hwt ~ age + wt + abs + triceps + subscap
      Df Sum of Sq   RSS   AIC F value    Pr(>F)
<none>          652.66 177.70
age      1     9.875 662.54 176.87  1.0894  0.300098
wt       1    10.554 663.22 176.95  1.1643  0.284169
abs      1   189.072 841.73 195.54 20.8580 1.996e-05 ***
triceps 1    78.809 731.47 184.59  8.6941  0.004302 **
subscap 1     5.693 658.36 176.38  0.6281  0.430660
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the largest p -value from `drop1()` is 0.4307, which corresponds to the variable `subscap`, `mod.hsw` is updated by dropping the variable `subscap`.

```
> mod.hsw <- update(mod.hsw, . ~ . - subscap)
> drop1(mod.hsw, test = "F") # single term deletions

Single term deletions

Model:
hwt ~ age + wt + abs + triceps
      Df Sum of Sq   RSS   AIC F value    Pr(>F)
<none>          658.36 176.38
age      1    13.615 671.97 175.97  1.5097  0.2231
wt       1     6.833 665.19 175.18  0.7577  0.3869
abs      1   220.994 879.35 196.95 24.5043 4.621e-06 ***
triceps 1   201.768 860.12 195.23 22.3725 1.068e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the largest p -value from `drop1()` is 0.3869, which corresponds to the variable `wt`, `mod.hsw` is updated by dropping the variable `wt`.

```
> mod.hsw <- update(mod.hsw, . ~ . - wt)
> drop1(mod.hsw, test = "F") # single term deletions

Single term deletions

Model:
hwt ~ age + abs + triceps
      Df Sum of Sq   RSS   AIC F value    Pr(>F)
<none>          665.19 175.18
age      1    37.595 702.78 177.47  4.1823  0.04441 *
abs      1   282.896 948.08 200.82 31.4712 3.323e-07 ***
```

```
triceps   1    198.891 864.08 193.59 22.1259 1.159e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

At this point, the p -values are all below the α_{crit} of 0.20, so the final model based on backward elimination should contain the predictors `age`, `abs`, and `triceps`.

```
> mod.be <- lm(hwfat ~ age + abs + triceps, data = HSWRESTLER)
> summary(mod.be) # lm summary
```

Call:

```
lm(formula = hwfat ~ age + abs + triceps, data = HSWRESTLER)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.8374	-2.0468	-0.4215	2.3076	7.9850

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.61606	4.23272	2.508	0.0143 *
age	-0.53309	0.26067	-2.045	0.0444 *
abs	0.35643	0.06354	5.610	3.32e-07 ***
triceps	0.46561	0.09898	4.704	1.16e-05 ***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 2.998 on 74 degrees of freedom

Multiple R-squared: 0.8895, Adjusted R-squared: 0.885

F-statistic: 198.5 on 3 and 74 DF, p-value: < 2.2e-16

(c) The functions `add1()` and `update()` are used to create a model using forward selection.

```
> SCOPE <- (~ . + age + ht + wt + abs + triceps + subscap)
> mod.fs <- lm(hwfat ~ 1, data = HSWRESTLER)
> add1(mod.fs, scope = SCOPE, test = "F")
```

Single term additions

Model:

hwfat ~ 1	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
<none>		6017.8	340.97			
age	1	175.0	5842.8	340.67	2.2765	0.1355
ht	1	117.8	5900.0	341.43	1.5175	0.2218
wt	1	3237.6	2780.2	282.74	88.5045	2.219e-14 ***
abs	1	5072.8	945.0	198.57	407.9929	< 2.2e-16 ***
triceps	1	5056.3	961.5	199.92	399.6462	< 2.2e-16 ***
subscap	1	4939.0	1078.8	208.90	347.9456	< 2.2e-16 ***

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

> mod.fs <- update(mod.fs, . ~ . + abs)
> add1(mod.fs, scope = SCOPE, test = "F")

Single term additions

Model:
hwfat ~ abs
      Df Sum of Sq   RSS   AIC F value    Pr(>F)
<none>           944.96 198.57
age      1     80.876 864.08 193.59  7.0199 0.0098255 ***
ht       1     61.598 883.36 195.31  5.2298 0.0250250 *
wt       1     43.734 901.22 196.87  3.6396 0.0602498 .
triceps 1    242.173 702.78 177.47 25.8443 2.639e-06 ***
subscap  1    132.580 812.38 188.77 12.2400 0.0007904 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> mod.fs <- update(mod.fs, . ~ . + triceps)
> add1(mod.fs, scope = SCOPE, test = "F")

Single term additions

Model:
hwfat ~ abs + triceps
      Df Sum of Sq   RSS   AIC F value    Pr(>F)
<none>           702.78 177.47
age      1     37.595 665.19 175.18  4.1823 0.04441 *
ht       1     25.246 677.54 176.62  2.7574 0.10104
wt       1     30.812 671.97 175.97  3.3932 0.06947 .
subscap  1     2.244 700.54 179.22  0.2370 0.62782
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> formula(mod.fs)

hwfat ~ abs + triceps

```

Forward selection picks the variables **abs** and **triceps**.



12.11.1.4 Criterion-Based Procedures

There are several well-defined optimality criteria used in model building, including R_a^2 (R^2 adjusted), Mallows's C_p , Bayes Information Criterion (BIC), and Akaike Information Criterion (AIC). R_a^2 is used instead of R^2 since R^2 will always increase with the addition of more variables to the model. Recall that $R_a^2 = 1 - ((n-1)/(n-p)) \cdot (SSE/SST)$.

The C_p statistic is defined as $C_p = SSE/\hat{\sigma}^2 + 2p - n$, where $\hat{\sigma}^2$ is from the model with all predictors and SSE is for the model with p parameters. When all p parameters are used in the model, $C_p = p$. A model with a bad fit will produce a C_p much bigger than p . Desirable models have small p and C_p less than or equal to p . It is common practice to plot C_p against p .

Recall that $\ln \mathcal{L}(\boldsymbol{\beta}, \sigma^2 | \mathbf{X})$ is called the log-likelihood function. The BIC for linear re-

gression models is defined as

$$-2 \max(\ln \mathcal{L}(\boldsymbol{\beta}, \sigma^2 | \mathbf{X})) + p \cdot \ln(n) = n \ln(SSE/n) + p \cdot \ln(n) + \text{constant},$$

while the AIC for linear regression models is defined as

$$-2 \max(\ln \mathcal{L}(\boldsymbol{\beta}, \sigma^2 | \mathbf{X})) + 2p = n \ln(SSE/n) + 2p + \text{constant}.$$

Since the constant is the same for a given data set and error distribution, it can be ignored when comparing models based on the same data. This is what the function `stepAIC()` does. The astute reader will note that Mallow's C_p is proportional to the Akaike Information Criterion (AIC), which means they will select identical models.

There are various R functions including `step()`, `add1()`, `drop1()`, and `stepAIC()` (from the package `MASS`) to perform criterion-based searches. The goal when using BIC or AIC is to create a model that minimizes either BIC or AIC. Both AIC and BIC search for models that have small SSE , but BIC penalizes larger models more so than does AIC (assuming $n > e^2 = 7.3891$). Consequently, BIC will favor smaller models than will AIC. When building a model to be used for predictive purposes, AIC will generally be favored over BIC. The package `leaps` contains the function `regsubsets()`, which is very useful for computing R_a^2 , Mallows's C_p , as well as BIC.

Example 12.16 ▷ Criterion-Based Variable Selection ◁ Create regression models for predicting wrestlers' hydrostatic fat (`hwfat`) using the data frame `HSWRESTLER`.

- (a) Use the function `regsubsets()` from the package `leaps` to build a regression model where `hwfat` is the response using the predictors `age`, `ht`, `wt`, `abs`, `triceps`, and `subscap` when R_a is the criterion used for variable selection.
- (b) Use the function `regsubsets()` from the package `leaps` to build a regression model where `hwfat` is the response using the predictors `age`, `ht`, `wt`, `abs`, `triceps`, and `subscap` when Mallow's C_p is the criterion used for variable selection.
- (c) Use the function `regsubsets()` from the package `leaps` to build a regression model where `hwfat` is the response using the predictors `age`, `ht`, `wt`, `abs`, `triceps`, and `subscap` when BIC is the criterion used for variable selection.
- (d) Verify that the model selected using AIC as a criterion from the function `stepAIC()` from the `MASS` package returns the same model as the model created in (b) when Mallow's C_p was used as the criterion.

Solution: The function `regsubsets()` by default performs an exhaustive search using all possible combinations of the user-specified predictors for model selection with one required argument `x`. The value passed to `x` may be a design matrix, a model formula for the full model, or a `biglm` object. If the number of predictors available is greater than eight and the user wants to examine models with more than eight predictors, the default argument `nvmax = 8` will need to be changed. The function returns separate best models of all sizes up to `nvmax`.

- (a) The short-hand notation `hwfat ~ .` is used to indicate `regsubsets()` should consider all of the variables in the data frame passed to the `data` argument.

R Code 12.20

```
> library(leaps)
> models <- regsubsets(hwfat ~ ., data = HSWRESTLER[, -c(8, 9)])
> summary(models)
```

```

Subset selection object
Call: regsubsets.formula(hwfat ~ ., data = HSWRESTLER[, -c(8, 9)])
6 Variables (and intercept)
      Forced in Forced out
age        FALSE      FALSE
ht         FALSE      FALSE
wt         FALSE      FALSE
abs        FALSE      FALSE
triceps   FALSE      FALSE
subscap   FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: exhaustive
      age ht  wt  abs triceps subscap
1 ( 1 ) " " " " " * " " " "
2 ( 1 ) " " " " " * " * " "
3 ( 1 ) * " " " " * " * " "
4 ( 1 ) * " * " " " * " * " "
5 ( 1 ) * " " " * " * " * " "
6 ( 1 ) * " * " * " * " * " "

> R2adj <- summary(models)$adjr2
> R2adj

[1] 0.8409068 0.8801014 0.8849817 0.8846381 0.8840129 0.8826699

> which.max(R2adj)

[1] 3

```

The output from R Code 12.20 on the facing page indicates the best single predictor model is that with the variable **abs**, and that particular model has $R_a^2 = 0.8409$, the best model with two predictors is the model with the variables **abs** and **triceps** with $R_a^2 = 0.8801$, and so forth. Either by visual inspection or by using the code `which.max(R2adj)`, one notes that the largest R_a^2 is 0.885, which corresponds to the model with predictors **age**, **abs**, and **triceps**.

(b) The best models of each size are stored in the object **models**, and Mallow's C_p values are extracted in R Code 12.21.

R Code 12.21

```

> MCP <- summary(models)$cp
> MCP

[1] 29.051861 4.641808 2.541953 3.775400 5.175856 7.000000

> which.min(MCP)

[1] 3

```

The smallest Mallow's C_p value is 2.542, indicating the best model according to Mallow's C_p is the one with the predictors **age**, **abs**, and **triceps**.

(c) The best models of each size are stored in the object **models** and BIC values are extracted

in R Code 12.22.

R Code 12.22

```
> BIC <- summary(models)$bic
> BIC

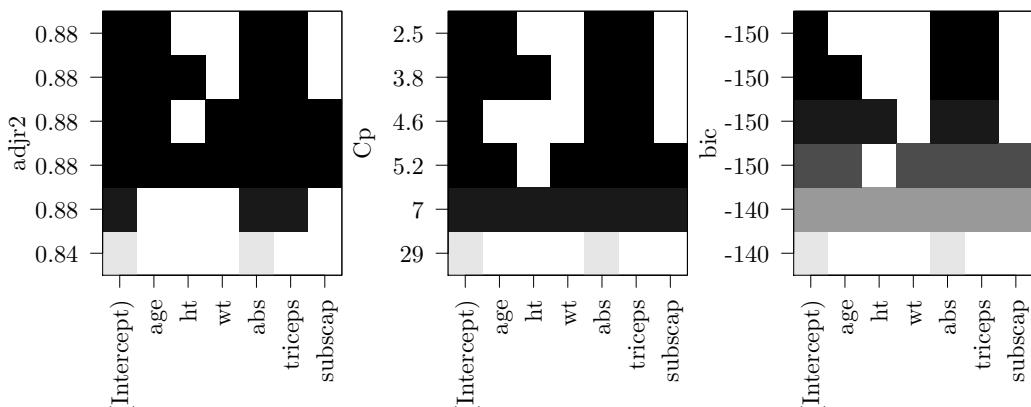
[1] -135.6909 -154.4291 -154.3607 -150.8326 -147.1302 -142.9664

> which.min(BIC)

[1] 2
```

The smallest BIC value is -154.4291 indicating the best model according to BIC is the one with the predictors `age` and `abs`. To graphically represent models one might select using the criteria R_a^2 , Mallow's C_p , and BIC; one can use the function `plot()` and pass the string `adjr2`, `Cp`, or `bic` to the argument `scale`. Using the R_a^2 criterion, one would select the variables that yield the highest R_a^2 value. The best model according to the R_a^2 criterion is found by reading the shaded variables at the top of the graph from left to right. Reading the graph from top row from left to right, one is led to choose a model that includes an intercept, `age`, `abs`, and `triceps`. When using the C_p criterion, one should select a model that was as small a C_p as possible. In the C_p versus variables plot, the C_p values decrease as one reads up; consequently, the best model according to the C_p criterion is found by reading the shaded variables at the top of the graph from left to right. The C_p criterion selects the same model as the R_a^2 . When using the BIC criterion, one should select a model that was as small a BIC as possible. In the BIC versus variables plot, the BIC values decrease as one reads up; consequently, the best model according to the BIC criterion is found by reading the shaded variables at the top of the graph from left to right. The BIC criterion leads one to select a model with an intercept, `abs`, and `triceps`.

```
> plot(models, scale = "adjr2")
> plot(models, scale = "Cp")
> plot(models, scale = "bic")
```



- (d) The function `stepAIC()` from the `MASS` package has two required arguments: `object` and `scope`. The value passed to `object` is an initial model. The value passed to `scope` is either a single formula or a list containing the components `upper` and `lower`, both as formulas. By default, the value for the argument `k` is 2, which returns the Akaike Information Criterion (AIC). By setting `k` equal to `log(n)`, the criterion will switch to BIC.

R Code 12.23

```

> mod.lm <- lm(hwfat ~ ., data = HSWRESTLER[, -c(8, 9)])
> SCOPE <- (~.)
> stepAIC(mod.lm, scope = SCOPE, k = 2)

Start: AIC=179.51
hwfat ~ age + ht + wt + abs + triceps + subscap

          Df Sum of Sq    RSS    AIC
- ht      1   1.613 652.66 177.70
- wt      1   2.546 653.60 177.81
- subscap 1   5.921 656.97 178.21
- age     1   9.594 660.64 178.65
<none>            651.05 179.51
- triceps 1   72.683 723.73 185.76
- abs     1 162.000 813.05 194.84

Step: AIC=177.7
hwfat ~ age + wt + abs + triceps + subscap

          Df Sum of Sq    RSS    AIC
- subscap 1   5.693 658.36 176.38
- age     1   9.875 662.54 176.87
- wt      1 10.554 663.22 176.95
<none>            652.66 177.70
+ ht      1   1.613 651.05 179.51
- triceps 1   78.809 731.47 184.59
- abs     1 189.072 841.73 195.54

Step: AIC=176.38
hwfat ~ age + wt + abs + triceps

          Df Sum of Sq    RSS    AIC
- wt      1   6.833 665.19 175.18
- age     1 13.615 671.97 175.97
<none>            658.36 176.38
+ subscap 1   5.693 652.66 177.70
+ ht      1   1.385 656.97 178.21
- triceps 1 201.768 860.12 195.23
- abs     1 220.994 879.35 196.95

Step: AIC=175.18
hwfat ~ age + abs + triceps

          Df Sum of Sq    RSS    AIC
<none>            665.19 175.18
+ ht      1   7.029 658.16 176.35
+ wt      1   6.833 658.36 176.38
+ subscap 1   1.972 663.22 176.95
- age     1 37.595 702.78 177.47
- triceps 1 198.891 864.08 193.59

```

```

- abs      1  282.896 948.08 200.82

Call:
lm(formula = hwtfat ~ age + abs + triceps, data = HSWRESTLER[, 
  -c(8, 9)])

Coefficients:
(Intercept)        age         abs       triceps
  10.6161     -0.5331      0.3564     0.4656

```

Based on the output from R Code 12.23 on the previous page, the smallest AIC value is obtained by regressing `hwtfat` onto `age`, `abs`, and `triceps`. This is the same model Mallow's C_p returned.



12.11.1.5 Summary

Variable selection is simply a means to select variables for inclusion or exclusion in a model that can be used for explanatory or predictive purposes. That is, the goal is not variable selection per se, rather, the goal is to create a model that adequately explains or predicts from the data. Stepwise selection procedures do not always guarantee a model will be selected that meets the user's need to explain or predict from the data. Criterion-based methods typically involve a wider search than do stepwise procedures, and many argue that they return models that are better than those from stepwise procedures. Regardless of the methods one uses to select a model, additional factors such as the cost to measure the variables and model diagnostics should be considered in developing a model.

12.11.2 Diagnostics

While fitting a model using the principle of least squares regression requires no distributional assumptions, using the model for inferential purposes does depend on specific assumptions. If (12.7) on page 784 assumes $\varepsilon \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$, that is, the errors in the model are assumed to be independent and to follow a normal distribution with a mean of zero and a constant variance, then (12.7) is called the normal error model. Regression diagnostics play a critical role in the verification of these assumptions. Regression diagnostics are also used to learn about unusual observations. The diagnostics will often dictate changes in the model selected initially. These changes emphasize the fact that model building is an iterative process.

12.11.2.1 Checking Error Assumptions

The assumption in the normal error model deals with an unobservable quantity ε ; however, the residuals $\hat{\varepsilon}_i$ can be computed and analyzed. While the residuals do not have the same properties as the errors (ε), the differences between residuals and errors are slight, and examining the residuals is a reasonable approach to use in checking the assumptions about models' errors.

Simple techniques such as a histogram or a density plot of the residuals can be used to study the distribution of the residuals. Care needs to be exercised when interpreting such graphs since histograms and density plots of data that come from a normal distribution when the sample size is small will not always look normal. Furthermore, the residuals do

not have a constant variance. In fact, the variance-covariance matrix for $\hat{\varepsilon}$ is

$$\text{Var}(\hat{\varepsilon}) = \sigma^2[\mathbf{I} - \mathbf{H}], \text{ where } \mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'. \quad (12.56)$$

Proof:

$$\begin{aligned}\hat{\varepsilon} &= \mathbf{Y} - \hat{\mathbf{Y}} \\ &= \mathbf{Y} - \mathbf{X}\hat{\beta} \\ &= \mathbf{Y} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= \mathbf{Y} - \mathbf{HY} \\ &= (\mathbf{I} - \mathbf{H})\mathbf{Y},\end{aligned}$$

which implies that

$$\text{Var}(\hat{\varepsilon}) = (\mathbf{I} - \mathbf{H})\text{Var}(\mathbf{Y})(\mathbf{I} - \mathbf{H})' \text{ by property 3 on page 919}$$

$$= (\mathbf{I} - \mathbf{H})\sigma^2\mathbf{I}(\mathbf{I} - \mathbf{H})'$$

$$= \sigma^2(\mathbf{I} - \mathbf{H})(\mathbf{I} - \mathbf{H})'$$

$$\text{Var}(\hat{\varepsilon}) = \sigma^2(\mathbf{I} - \mathbf{H}) \text{ because } (\mathbf{I} - \mathbf{H}) \text{ is symmetric and idempotent. } \blacksquare$$

The diagonal entry of \mathbf{H} is denoted as h_{ii} , which is referred to as the leverage. Note that the trace of \mathbf{H} is p , the number of parameters $(\beta_0, \beta_1, \dots, \beta_{p-1})$ in the linear model. (See Problem 12.18 on page 893.)

Example 12.17 Find the \mathbf{H} matrix, and display the first five h_{ii} values for the model selected with the AIC statistic from part (d) of Example 12.16 on page 830. Verify that the sum of the h_{ii} values equals p . Recall that the variables selected were `age`, `abs`, and `triceps`.

Solution: The \mathbf{H} matrix is first constructed using the definition from (12.56) and stored in the object `H`. There are two R functions, `hat()` and `hatvalues()`, which take different arguments but return the h_{ii} values of the \mathbf{H} matrix. To obtain the h_{ii} values with `hat()`, one must pass the design matrix to the argument `x`. To obtain the h_{ii} values with `hatvalues()`, one should pass a linear model object to `model`.

R Code 12.24

```
> mod3.lm <- lm(hwfat ~ age + abs + triceps, data = HSWRESTLER)
> X <- model.matrix(mod3.lm)                                # n*p design matrix
> H <- X %*% solve(t(X) %*% X) %*% t(X)                  # H matrix
> dim(H)                                                 # n by n

[1] 78 78

> hii <- diag(H)                                         # extract diagonal values
> hii[1:5]                                              # show first 5 values

      1          2          3          4          5
0.055555037 0.02273942 0.03266124 0.02786396 0.20830418

> hat(x = model.matrix(mod3.lm))[1:5]                      # show first 5 values
```

```
[1] 0.05555037 0.02273942 0.03266124 0.02786396 0.20830418

> hatvalues(model = mod3.lm)[1:5]           # show first 5 values

 1         2         3         4         5
0.05555037 0.02273942 0.03266124 0.02786396 0.20830418

> sum(hii)                                # sum all hii values

[1] 4
```



12.11.2.1.1 Assessing Normality and Constant Variance Although formal hypothesis tests for normality, such as the Shapiro-Wilk test, can be applied to the residuals, they lack power to detect non-normal distributions. Recall that the null hypothesis in the Shapiro-Wilk test is that the distribution is normal and the alternative is that the distribution is not normal. Consider Figure 12.9 on the next page, where `mod2`, `mod3`, and `mod4` show residuals that suggest problems with either the constant variance or the normality of the errors assumption. Note that `mod1-mod4` are linear model objects. The second residual plot on the top row (`mod2`) shows a pattern of increasing variability. The Shapiro-Wilk test is run on the residuals of `mod2` to test for normality, the p -value is 0.0015, which leads one to conclude the residuals do not follow a normal distribution. The decreasing variance model (`mod3`) also rejects the idea that the residuals follow a normal distribution with a p -value of 3e-04. When a Shapiro-Wilk test is applied to the residuals in the bottom right plot (`mod4`), the p -value is 0.0794. Consequently, it is wiser to use a combination of graphical tests as well as hypothesis tests when studying the properties of the residuals from a particular model. Using `qqnorm()` on the residuals is a good starting point for assessing normality graphically. Other graphs one might use include, but are not limited to, histograms, box-plots, and density plots. As noted earlier, care needs to be taken when interpreting such graphs.

The assumption of constant variance is typically checked by plotting the $\hat{\varepsilon}_i$ s versus the \hat{Y}_i s. Constant variance is a reasonable assumption when the residuals are scattered in a band of constant width. When the band falls around the line $y = 0$, the regression model is appropriate. An example of constant variance is provided in the top left (`mod1`) residual plot of Figure 12.9 on the facing page. One formal large sample test for constant variance is the Breusch-Pagan test, which can be performed with the function `bptest()` from the package `lmtest`; however, the test has no power asymptotically (Zaman, 2000).

12.11.2.1.2 Testing Autocorrelation Whenever data are obtained in a time sequence, it is possible to have correlation (called autocorrelation) among the errors. A frequently used test for detecting autocorrelation is the Durbin-Watson test. The hypotheses of the test are specified in terms of the autocorrelation coefficient ρ , $H_0 : \rho = 0$ versus $H_1 : \rho \neq 0$, and are tested with the statistic

$$DW = \frac{\sum_{i=2}^n (\hat{\varepsilon}_i - \hat{\varepsilon}_{i-1})^2}{\sum_{i=1}^n \hat{\varepsilon}_i^2}. \quad (12.57)$$

Small values of DW lead to the conclusion that $\rho \neq 0$ because adjacent error terms $\hat{\varepsilon}_i - \hat{\varepsilon}_{i-1}$ tend to be similar when the data are correlated. The package `car` has a function `durbinWatsonTest()` that can be used to test for autocorrelation. The p -value of the Durbin-Watson test for the bottom right plot of Figure 12.9 on the next page (`mod4`) is 0. R

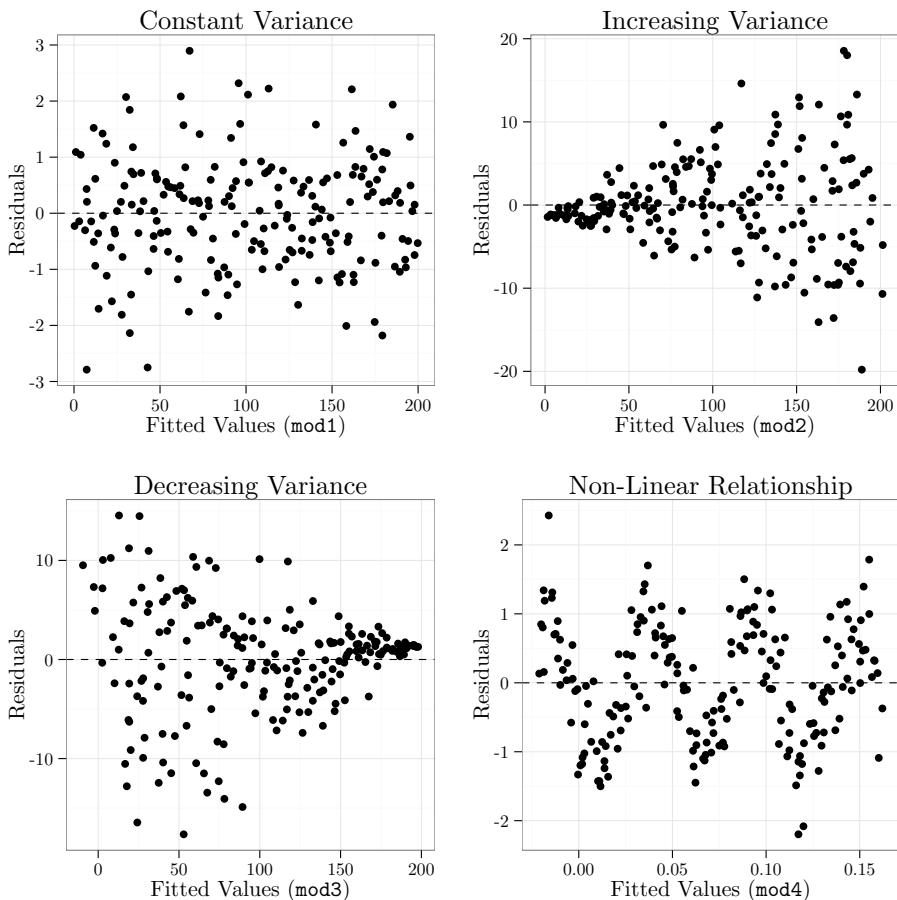


FIGURE 12.9: Residual plots for four different models with different residual patterns

Code 12.25 applies the Durbin-Watson test and the Shapiro-Wilk test to the residuals from mod4 of Figure 12.9. If correlation is present among the errors, it will not be possible to model the data using ordinary regression, which assumes the errors are independent. While it is possible to use generalized least squares to model data where errors are dependent, doing so is beyond the scope of this text.

R Code 12.25

```
> library(car)
> durbinWatsonTest(mod4)

lag Autocorrelation D-W Statistic p-value
 1      0.6510465    0.6967797      0
Alternative hypothesis: rho != 0

> shapiro.test(resid(mod4))

Shapiro-Wilk normality test

data:  resid(mod4)
W = 0.98763, p-value = 0.0794
```

Scatterplots of residuals versus a time, sequence, or order variable can often detect non-independence of error terms. When a linear model is created and stored in an object, the function `plot()` can be applied to the linear model object and several diagnostic plots will appear on the graphics device. Figure 12.10 on the facing page shows the four default graphs produced with R using the function `plot()` for `mod1`. The plot in the upper left panel shows residuals plotted against fitted values. This plot can be used to detect lack of fit. If the residuals show some curvilinear trend, the current model is not appropriate; however, transforming one or more of the variables can often remedy this problem. In this graph, such a problem does not exist. The same plot can be used to assess the constant variance assumption on the errors. In this case, the variance appears constant as the fitted values vary. The second default graph is a normal quantile-quantile plot of the residuals (upper right corner of Figure 12.10). In this case, there is not a clear deviation from normality. The lower left graph plots the square root of the residuals versus the fitted values. Assuming symmetry of the errors, this graph helps assess the constant variance of the errors, which in this case seems to be a reasonable assumption. The lower right panel shows standardized residuals (as defined in (12.58)) versus leverage points. Contours for Cook's distance (as defined in (12.63)) of 0.5 and 1 facilitate an understanding of the relationship among the residuals, leverage values, and Cooks's distance. R will actually produce six diagnostic graphs, but they must be specified using the argument `which = 1:6`, where the `1:6` is a vector with any or all of the values 1 through 6.

12.11.2.2 Identifying Unusual Observations

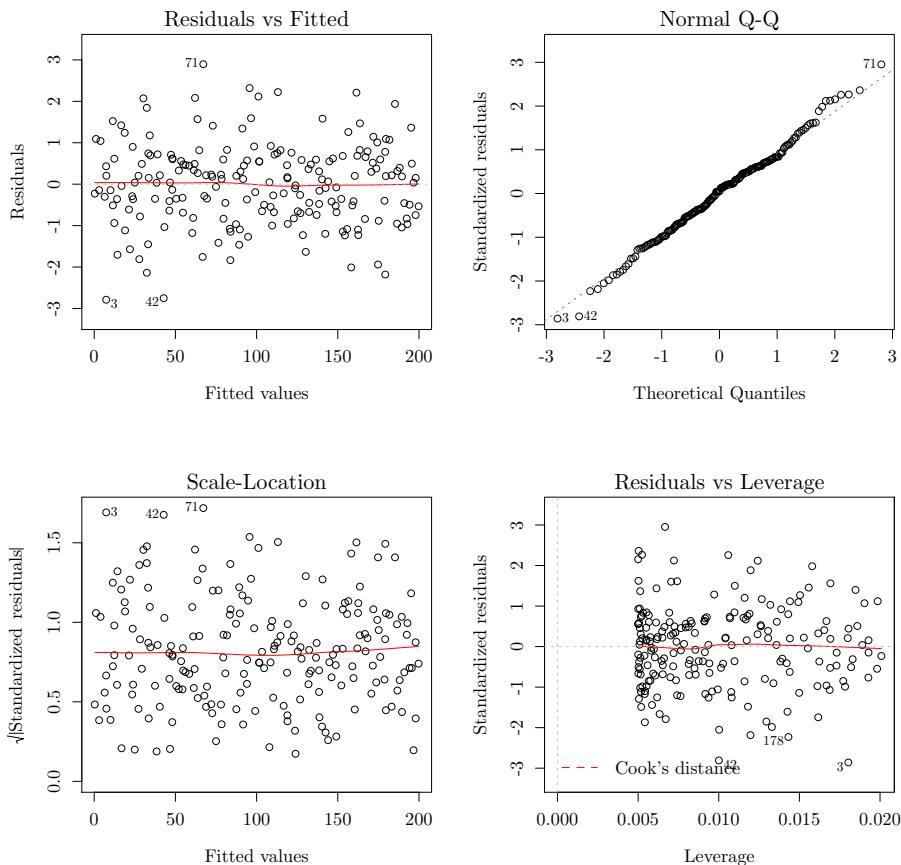
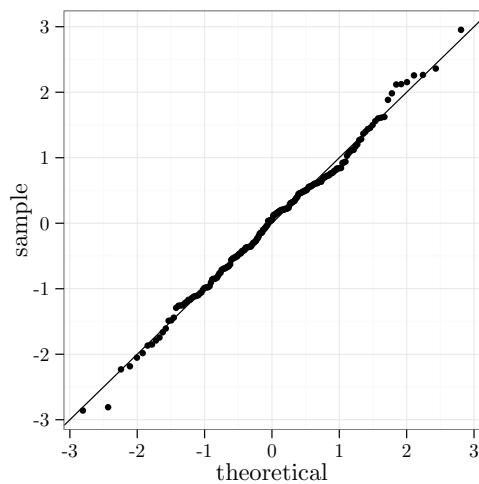
Quite often in regression models, certain observations do not seem to fit the overall pattern of the data. These cases may have a large residual and have the potential to alter dramatically the fitted regression model. An observation may be an outlier with respect to its Y values, its x values, or both, yet not all outlying observations will have a dramatic impact on the fitted regression model. One of the ways used to measure outlying Y values is to evaluate standardized residuals. This is done because residuals may have substantially different variances. Consequently, it makes sense to consider $\hat{\varepsilon}_i$ relative to its estimated standard deviation. When the residuals are rescaled to have unit variance, the resulting residuals (r_i) are known as internally studentized residuals or **standardized residuals**, where

$$r_i = \frac{\hat{\varepsilon}_i}{\sqrt{\widehat{Var}(\hat{\varepsilon}_i)}} = \frac{\hat{\varepsilon}_i}{\hat{\sigma} \cdot \sqrt{1 - h_{ii}}}. \quad (12.58)$$

The R function `rstandard()` computes standardized residuals according to (12.58). Standardized residuals are sometimes preferred in residual plots since they have been standardized to have unit variance; however, in many cases, no appreciable difference will be seen between the raw residuals and the standardized residuals. Only when there is an unusually large leverage (large is generally taken to be 2 or 3 times p/n) will differences be noticeable. When standardized residuals are displayed in a quantile-quantile plot, because the residuals are standardized, the points should fall along the line $y = x$ if the normality assumption is reasonable. Figure 12.11 on the next page shows a quantile-quantile plot of the standardized residuals from the model shown in the upper left (`mod1`) of Figure 12.9 on the preceding page with a superimposed $y = x$ line.

Another refinement to make the residuals more effective in detecting outlying observations is to use deleted residuals. Specifically, when a regression model is computed where the i^{th} case is excluded, the i^{th} prediction is denoted $\hat{Y}_{i(i)}$, and the deleted residual ($\hat{\varepsilon}_{(i)}$) is then defined as

$$\hat{\varepsilon}_{(i)} = Y_i - \hat{Y}_{i(i)}. \quad (12.59)$$

FIGURE 12.10: Diagnostic plots for `mod1` in Figure 12.9FIGURE 12.11: Normal quantile-quantile plot for the standardized residuals of `mod1` in Figure 12.9 on page 837

Fortunately, an algebraic equivalent expression for $\hat{\varepsilon}_{(i)}$ exists that does not require the

computation of $\widehat{Y}_{i(i)}$ for each omitted case. Specifically, it can be shown that $\widehat{\varepsilon}_{(i)} = Y_i - \widehat{Y}_{i(i)} = \frac{\widehat{\varepsilon}_i}{1-h_{ii}}$. The estimated variance of the $\widehat{\varepsilon}_{(i)}$ is

$$\widehat{Var}[\widehat{\varepsilon}_{(i)}] = \frac{\widehat{\sigma}_{(i)}^2}{1-h_{ii}} = \frac{MSE_{(i)}}{1-h_{ii}}. \quad (12.60)$$

Ordinarily, one prefers to study the **studentized deleted residuals** (r_i^*) rather than the ordinary deleted residuals. The i^{th} studentized deleted residual is defined as

$$r_i^* = \frac{\widehat{\varepsilon}_{(i)}}{\sqrt{\widehat{Var}(\widehat{\varepsilon}_{(i)})}} = \frac{\widehat{\varepsilon}_i}{\sqrt{\frac{\widehat{\sigma}_{(i)}^2}{1-h_{ii}}}} = \frac{\widehat{\varepsilon}_i}{\widehat{\sigma}_{(i)} \cdot \sqrt{1-h_{ii}}}. \quad (12.61)$$

Again, there is an algebraic equivalent to (12.61) that avoids doing n regressions. The algebraic equivalent definition of r_i^* is

$$r_i^* = r_i \cdot \left(\frac{n-p-1}{n-p-r_i^2} \right)^{\frac{1}{2}} \sim t_{n-p-1}. \quad (12.62)$$

When the model is correct, each studentized deleted residual follows a t -distribution with $n-p-1$ degrees of freedom. Even though it is very likely only a few “large” r_i^* ’s will be of interest, by identifying them as large, all cases have implicitly been tested. To control the overall significance level, a Bonferroni approach is often used where r_i^* values are declared significant if their absolute value exceeds $t_{1-\alpha/2n;n-p-1}$. However, this approach does tend to be conservative, especially for large n . The R function `rstudent()` computes the studentized deleted residuals according to (12.62). The functions `outlierTest()`, `qqPlot()`, and `influenceIndexPlot()` from the `car` package report the Bonferroni adjusted p -value for the largest studentized residual in absolute value of a linear model, plot the studentized residuals versus a t_{n-p-1} distribution, and provide index plots of Cook’s distance, leverages, studentized residuals, and outlier significance levels for linear model objects, respectively.

Example 12.18 Compute and plot using `ggplot2` the studentized residuals for the regression model where `hwfat` is regressed onto `age`, `abs`, and `triceps` versus the fitted values using the data frame `HSWRESTLER`. Store the linear model object in the variable `mod3.hsw`.

- (a) What are the three largest standardized residuals in absolute value?
- (b) Can any of the studentized residuals be considered outliers according to the Bonferroni approach if the significance level is 0.20? Do the results from the functions `outlierTest()` and `influenceIndexPlot()` agree with your findings?

Solution: The package `ggplot2` has the function `fortify()`, which converts a linear model object into a data frame useful for plotting that has columns named `.hat`, `.sigma`, `.cooksdi`, `.fitted`, `.resid`, and `.stdresid`. These contain the diagonal elements of the hat matrix, an estimate of the residual standard deviation when the corresponding observation is dropped from the model, Cook’s distance, fitted values for the linear model, residuals for the linear model, and standardized residuals for the linear model, respectively.

```
> mod3.hsw <- lm(hwfat ~ age + abs + triceps, data = HSWRESTLER)
> fmod <- fortify(mod3.hsw)
> head(fmod, n = 3)
```

hwfat	age	abs	triceps	.hat	.sigma	.cooksdi	.fitted
-------	-----	-----	---------	------	--------	----------	---------

```

1 10.71 18 8      6 0.05555037 2.979079 2.833328e-02 6.665447
2 8.53 15 10     8 0.02273942 3.014222 1.258905e-03 9.908808
3 6.78 17 6      6 0.03266124 3.018436 8.408945e-05 6.485680
  .resid   .stdresid
1 4.044553 1.38811158
2 -1.378808 -0.46520239
3 0.294320 0.09980993

```

R Code 12.26 is used to create Figure 12.12.

R Code 12.26

```

> p <- ggplot(data = fmod, aes(x = .fitted, y = .stdresid))
> p + geom_point() + geom_hline(y = 0, lty = "dashed") +
+   labs(x = "Fitted Values", y = "Standardized Residuals") +
+   theme_bw()

```

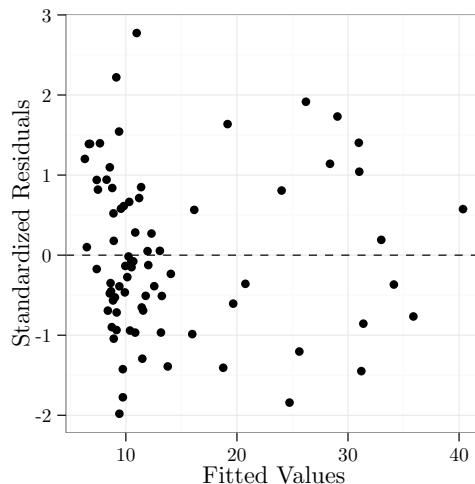


FIGURE 12.12: Standardized residuals versus fitted values for `mod3.hsw`

(a) R Code 12.27 shows two approaches for finding the three largest standardized residuals in absolute value for `mod3.hsw`.

R Code 12.27

```

> sort(abs(rstandard(mod3.hsw)), decreasing = TRUE)[1:3]
35      22      42
2.773943 2.220556 1.980789

> sort(abs(fmod$.stdresid), decreasing = TRUE)[1:3]
[1] 2.773943 2.220556 1.980789

```

(b) The Bonferroni critical value is $t_{1-\alpha/2n;n-p-1} = t_{1-0.20/(2 \times 78);78-4-1} = 3.1233$.

R Code 12.28

```
> BCV <- qt(1 - 0.2/(2*78), 73)           # Bonferroni Critical Value
> BCV

[1] 3.123317

> sum(abs(rstudent(mod3.hsw)) > BCV)          # how many r values > BCV

[1] 0

> max(abs(rstudent(mod3.hsw)))                  # value of largest r

[1] 2.910616

> which.max(abs(rstudent(mod3.hsw)))            # find row for largest r

35
35

> outlierTest(mod3.hsw)

No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
  rstudent unadjusted p-value Bonferonni p
35 2.910616      0.0047798     0.37283
```

R Code 12.28 indicates there are no studentized residuals with a value greater than the critical Bonferroni value of 3.1233. The function `outlierTest()` verifies this result and returns the Bonferroni ρ -value = 0.3728 for the largest studentized residual that is greater than $\alpha = 0.20$. R Code 12.29 is used to create Figure 12.13 on the facing page which depicts a quantile-quantile plot of the studentized residuals versus the t quantiles, and Figure 12.14 on the next page which shows several diagnostic plots. Note that the three largest (in absolute value) studentized residuals shown in Figure 12.13 on the facing page are also shown in the second plot of Figure 12.14 on the next page, while the third plot of Figure 12.14 on the facing page shows the only studentized residual to have a Bonferroni ρ -value less than 0.40 is observation 35. The first and last plot of Figure 12.14 on the next page are discussed in more detail in Section 12.11.2.3 on page 844.

R Code 12.29

```
> qqPlot(mod3.hsw, id.n = 3)                 # label largest 3
> influenceIndexPlot(mod3.hsw, id.n = 3)      # label largest 3
```



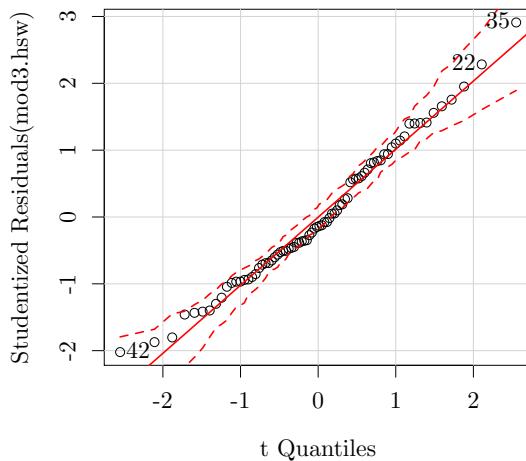


FIGURE 12.13: Quantile-quantile plot of studentized residuals from `mod3.hsw` with the three largest (in absolute value) studentized residuals labeled

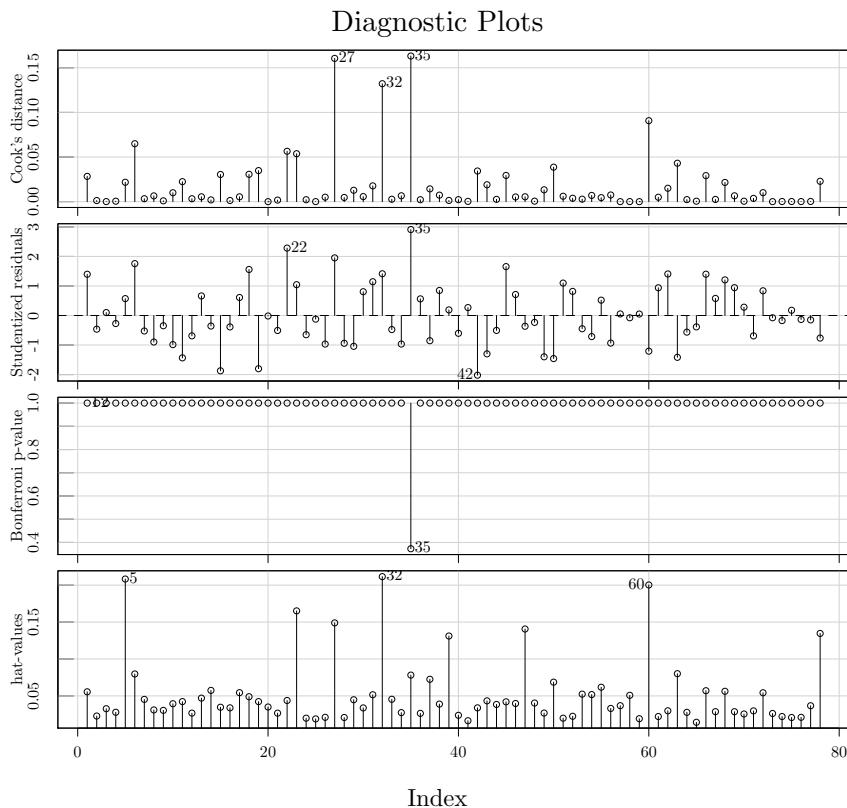


FIGURE 12.14: Diagnostic plots of `mod3.hsw` with the three most prominent observations for each diagnostic plot labeled

12.11.2.3 High Leverage Observations

While residuals were used to identify outlying Y values, the hat matrix provides an analog for the x values. The diagonal entry h_{ii} of the hat matrix \mathbf{H} provides a measure of the distance of the i^{th} case from the centroid of the x observations. That is, h_{ii} can be used to assess whether an observation is outlying from the other xs by examining its h_{ii} value. The limits on h_{ii} are $1/n \leq h_{ii} \leq 1/c$, where c is the number of rows of \mathbf{X} that have the same values as the i^{th} row. Note that the upper limit is never greater than 1. In general, a leverage value, h_{ii} , is considered large if it is more than twice as large as the mean leverage value ($2p/n$). Observations with large h_{ii} are called high leverage points, and each case should be investigated to see if the point estimates in the model under consideration change when the i^{th} case is included versus excluded from the analysis. It is important to note that not all points with high leverage will dramatically alter the estimation of parameters in the model. When the estimated parameters are substantially different with and without the i^{th} case, the i^{th} case is said to be **influential**. That is, not all high leverage observations are influential. Clearly, which cases are influential (if any) may change when the model is changed.

Influential Observations Some influence measures examined next, all of which measure the effect of deleting the i^{th} observation, include Cook's distance, D_i , which measures the effect on the $\hat{\beta}$ s or, equivalently, on the predicted values (see (12.63)); DFFITS $_i$, which measure the effect on the predicted \hat{Y}_i s; and DFBETAS $_{k(i)}$, which measure the effect on the $\hat{\beta}_j$ s. Fortunately, all of the influence measures considered can be computed from the results of a single regression using all of the data.

Cook's Distance Cook's distance evaluates the influence of the i^{th} case on all of the n fitted values. It is a combined measure of the standardized residual (r_i) and the leverage value (h_{ii}) that produces a number used to assess the impact of removing the i^{th} observation on the all regression coefficients (β). Cook's D_i is defined as

$$\frac{(\hat{\mathbf{Y}}_{(i)} - \hat{\mathbf{Y}})'(\hat{\mathbf{Y}}_{(i)} - \hat{\mathbf{Y}})}{p\hat{\sigma}^2} = \frac{(\hat{\beta}_{(i)} - \hat{\beta})'(\mathbf{X}'\mathbf{X})(\hat{\beta}_{(i)} - \hat{\beta})}{p\hat{\sigma}^2}. \quad (12.63)$$

An algebraically equivalent expression for D_i is

$$D_i = \frac{\hat{\varepsilon}_i^2}{p\hat{\sigma}^2} \left[\frac{h_{ii}}{(1-h_{ii})^2} \right] = \frac{r_i^2}{p} \left(\frac{h_{ii}}{1-h_{ii}} \right). \quad (12.64)$$

D_i values are generally flagged for further scrutiny when they exceed $f_{0.50;p,n-p}$; however, the exact distribution of D_i is unknown, and the use of $f_{0.50;p,n-p}$ is only a suggestion. Oftentimes, a simple graph of the D_i s will indicate values that require further scrutiny. One can always program a function according to (12.64) to compute the D_i s; however, a better approach is to use built-in functions on linear model objects. The function `cooks.distance()` will compute the D_i s. The function `influence.measures()` computes basic quantities used in many diagnostics, including h_{ii} values and the DFBETAS.

DFFITS A measure related to D_i is DFFITS, which is an abbreviation for “difference in fits.” DFFITS is a standardized measure of the amount by which the predicted value \hat{Y}_i changes when the i^{th} case is deleted from the data. The definition of DFFITS is

$$\text{DFFITS}_i = \frac{\hat{Y}_i - \hat{Y}_{i(i)}}{\hat{\sigma}_{(i)}\sqrt{h_{ii}}}, \quad (12.65)$$

while a computationally equivalent definition of DFFITS is

$$\text{DFFITS}_i = r_i^* \left(\frac{h_{ii}}{1 - h_{ii}} \right)^{\frac{1}{2}}, \quad (12.66)$$

where r_i^* is the studentized deleted residual. DFFITS values whose absolute value exceeds $2\sqrt{p/n}$ generally require further scrutiny. To compute DFFITS, use `dffits(linear model object)`.

It bears pointing out that there are n D_i values and n DFFITS values. The next influence measure considered is DFBETAS, which measures the influence of the i^{th} case on each regression coefficient. That is, there will be np DFBETAS values.

DFBETAS A standardized measure of the amount by which the k^{th} regression coefficient changes when the i^{th} observation is omitted from the data set is DFBETAS. A case is considered to have a large DFBETAS value if its absolute value exceeds $2/\sqrt{n}$. The DFBETAS measure is defined as

$$\text{DFBETAS}_{k(i)} = \frac{\hat{\beta}_k - \hat{\beta}_{k(i)}}{\sqrt{\hat{\sigma}_{(i)}^2 \cdot v_{k+1,k+1}}}. \quad (12.67)$$

If the matrix \mathbf{V} is defined to be $(\mathbf{X}'\mathbf{X})^{-1}$, then $\sigma_{\hat{\beta}_k}^2 = \sigma^2 \cdot v_{k+1,k+1}$, where $v_{k+1,k+1}$ is the $(k+1)^{\text{st}}$ diagonal entry ($k = 0, 1, \dots, p-1$) of \mathbf{V} . To compute DFBETAS, use `dfbetas(linear model object)`. Table 12.6 provides a list of influence measures and summarizes when the measures may be influential.

Table 12.6: Summary of measures of influential observations

Influence Measure	Formula	Case i May Be Influential if:
Cook's D_i	$\frac{r_i^2}{p} \left(\frac{h_{ii}}{1 - h_{ii}} \right)$	$D_i > f_{0.5;p,n-p}$
DFFITS	$r_i^* \left(\frac{h_{ii}}{1 - h_{ii}} \right)^{\frac{1}{2}}$	$ DFFITS > 2\sqrt{\frac{p}{n}}$
DFBETAS $_{k(i)}$	$\frac{\hat{\beta}_k - \hat{\beta}_{k(i)}}{\sqrt{\hat{\sigma}_{(i)}^2 \cdot v_{k+1,k+1}}}$	$ \text{DFBETAS}_{k(i)} > \frac{2}{\sqrt{n}}$

Example 12.19 \triangleright *Influential Observations* \triangleleft In part (b) of Example 12.16 Mallow's C_p as well as AIC suggested the regression model `hwtfat ~ age + abs + triceps`. Regress `hwtfat` onto `age`, `abs`, and `triceps` and store the result in `mod3.hsw`.

- How many of the leverage values (h_{ii}) from `mod3.hsw` are greater than $2p/n$? Which observations have leverage values greater than $2p/n$?
- How many of the $|DFFITS|$ values from `mod3.hsw` are greater than $2\sqrt{p/n}$? Which observations have $|DFFITS|$ values greater than $2\sqrt{p/n}$?
- How many of the $|\text{DFBETAS}|$ values from `mod3.hsw` are greater than $2/\sqrt{n}$? Which observations have $|\text{DFBETAS}|$ values greater than $2/\sqrt{n}$?

- (d) Use the function `influencePlot()` from the package `car` to create a bubble-plot where the studentized residuals are plotted against the leverage values and each point in the plot has an area that is proportional to Cook's distance. Based on the resulting graph and parts (a), (b), and (c), are the observations 22, 27, 32, 35, and 60 influential?

Solution: R Code 12.30 regresses `hfwat` onto `age`, `abs`, and `triceps`.

R Code 12.30

```
> mod3.hsw <- lm(hfwat ~ age + abs + triceps, data = HSWRESTLER)
> summary(mod3.hsw) # lm summary
```

Call:

```
lm(formula = hfwat ~ age + abs + triceps, data = HSWRESTLER)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.8374	-2.0468	-0.4215	2.3076	7.9850

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.61606	4.23272	2.508	0.0143 *
age	-0.53309	0.26067	-2.045	0.0444 *
abs	0.35643	0.06354	5.610	3.32e-07 ***
triceps	0.46561	0.09898	4.704	1.16e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.998 on 74 degrees of freedom

Multiple R-squared: 0.8895, Adjusted R-squared: 0.885

F-statistic: 198.5 on 3 and 74 DF, p-value: < 2.2e-16

- (a) Observations 5, 23, 27, 32, 39, 47, 60, and 78 all have leverage values greater than $2p/n = 2 \times 4/78 = 0.1026$. R Code 12.31 extracts the h_{ii} values and determines which ones are greater than 0.1026.

R Code 12.31

```
> hii.hsw <- hatvalues(mod3.hsw)
> hcv <- 2*4/78           # 2*p/n
> hcv
[1] 0.1025641

> sum(hii.hsw > hcv)      # how many hii.hsw > hcv
[1] 8

> which(hii.hsw > hcv)
5 23 27 32 39 47 60 78
5 23 27 32 39 47 60 78
```

(b) Observations 6, 22, 23, 27, 32, and 35 all have DFFITS values greater than $2\sqrt{p/n} = 2\sqrt{4/78} = 0.4529$. R Code 12.32 computes the DFFITS values and determines which ones are greater than 0.4529.

R Code 12.32

```
> dffitsCV <- 2*sqrt(4/78)           # 2*sqrt(p/n)
> dffitsCV
[1] 0.4529108

> dffits.hsw <- dffits(mod3.hsw)      # DFFITS values
> sum(dffits.hsw > dffitsCV)         # how many dffits.hsw > dffitsCV
[1] 6

> which(dffits.hsw > dffitsCV)

6 22 23 27 32 35
6 22 23 27 32 35
```

(c) Observations 1, 6, 11, 18, 19, 22, 35, 45, and 66 all have DFBETAS₀ values greater than $2/\sqrt{n} = 2/\sqrt{78} = 0.2265$, observations 1, 6, 18, 19, 22, 35, 42, 45, 66, and 68 all have DFBETAS₁ values greater than $2/\sqrt{n} = 2/\sqrt{78} = 0.2265$, and observations 6, 23, 27, 32, 60, and 63 all have DFBETAS₂ values greater than $2/\sqrt{n} = 2/\sqrt{78} = 0.2265$. R Code 12.33 computes the DFBETAS_k for $k = 0, 1, 2$ values and determines which ones are greater than 0.2265.

R Code 12.33

```
> dfbetasCV <- 2/sqrt(78)  # 2/sqrt(n)
> dfbetasCV
[1] 0.2264554

> dfbetas.hsw <- dfbetas(mod3.hsw)
> which(abs(dfbetas.hsw[, 1]) > dfbetasCV)

1 6 11 18 19 22 35 45 66
1 6 11 18 19 22 35 45 66

> which(abs(dfbetas.hsw[, 2]) > dfbetasCV)

1 6 18 19 22 35 42 45 66 68
1 6 18 19 22 35 42 45 66 68

> which(abs(dfbetas.hsw[, 3]) > dfbetasCV)

6 23 27 32 60 63
6 23 27 32 60 63
```

(d) R Code 12.34 on the following page shows the changes in the coefficients when observations 22, 27, 32, 35, and 60 are omitted, the DFFITS for the same observations, and the

original measurements for the observations in question. Not only are all five of the observations influential observations, there is also a lack of agreement between `hwt`, `tanfat`, and `hwt` for these values, possibly suggesting poor or inaccurate measurements of the variables `abs`, `triceps`, and `subscap`. Consider observation 22: `skfat` (8.88) is determined by the `abs` (7), `triceps` (7.5), and `subscap` (7) measurements; yet the value for `hwt` (15.65) and `tanfat` (16.4) are significantly larger than `skfat` (8.88). This disagreement in fat measures suggest the initial measurements of `abs`, `triceps`, and `subscap` may be incorrect. Similar arguments can be made for the other four influential values. If information had been collected on the individuals taking the wrestlers' measurements, one might be able to make a stronger case that the measurements in question are poor because the person measuring the wrestlers' body fat did not know how to use skin calipers to measure body fat.

R Code 12.34

```
> study <- c(22, 27, 32, 35, 60)
> dfbetas.hsw[study, ]

(Intercept)      age       abs     triceps
22  0.39451981 -0.35749145 -0.04742097 -0.05066954
27 -0.08251435  0.02687861 -0.60005768  0.74693181
32  0.03291160 -0.03223409  0.67113231 -0.51401921
35  0.79223839 -0.74692760  0.12497641 -0.24106405
60  0.06379316 -0.07725617 -0.55396994  0.45479108

> dffits.hsw[study]

          22        27        32        35        60
0.4885608  0.8163586  0.7319061  0.8476917 -0.6042154

> HSWRESTLER[study, ]

    age      ht      wt abs triceps subscap hwt tanfat skfat
22 14 61.00 100.2   7     7.5      7 15.65  16.4 8.88
27 15 59.75 121.4  27    30.0     22 31.51  25.7 26.00
32 16 70.75 232.4  51    23.0     34 34.71  44.9 34.24
35 13 62.00 122.6  10     8.0      7 18.96  15.3 9.95
60 17 70.00 224.4  44    18.0     16 22.39  28.4 25.71
```

The bubble-plot shown in Figure 12.15 on the next page created from R Code 12.35 shows that observations 22, 27, 32, 35, and 60 all have relatively large Cook's distance values in combination with either a large studentized residual or a large leverage value.

R Code 12.35

```
> influencePlot(mod3.hsw, id.n = 3)

    StudRes      Hat      CookD
5  0.5723248 0.20830418 0.1474568
22 2.2828699 0.04379513 0.2376124
27 1.9516411 0.14891399 0.4006462
32 1.4129540 0.21155606 0.3635138
35 2.9106163 0.07818935 0.4039433
42 -2.0216847 0.03383905 0.1853500
60 -1.2074144 0.20026941 0.3011774
```

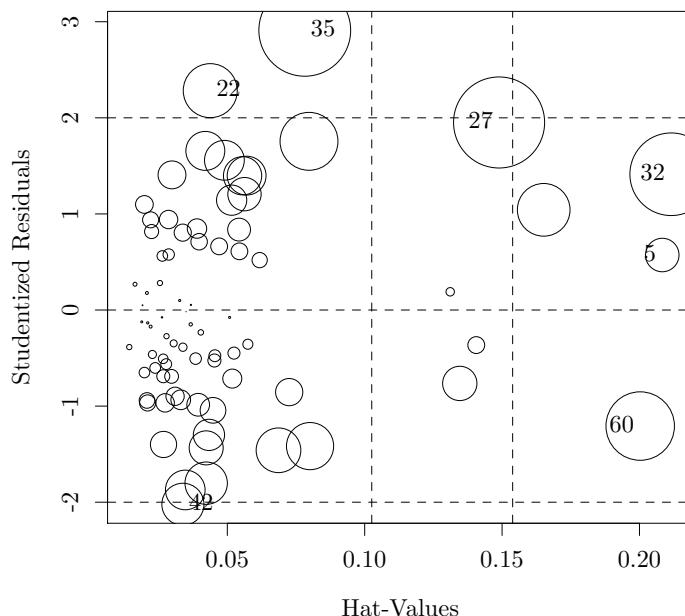


FIGURE 12.15: Bubble-plot of studentized residuals versus leverage values with plotted points proportional to Cook's distance for `mod3.hsw`

R Code 12.36 uses the function `influence.measures()` to compute regression diagnostics and then returns a table listing potentially influential observations with the `summary()` function.

R Code 12.36

```
> inflm.SR <- influence.measures(mod3.hsw)
> summary(inflm.SR) # which observations 'are' influential

Potentially influential observations of
lm(formula = hwfat ~ age + abs + triceps, data = HSWRESTLER) :

  dfb.1_ dfb.age dfb.abs dfb.trcpdffit cov.r cook.d hat
5 -0.14  0.11 -0.06  0.17  0.29  1.31_*  0.02  0.21_*
23 -0.10  0.09  0.37 -0.25  0.46  1.19_*  0.05  0.17_*
27 -0.08  0.03 -0.60  0.75  0.82_*  1.01  0.16  0.15
32  0.03 -0.03  0.67 -0.51  0.73_*  1.20_*  0.13  0.21_*
35  0.79 -0.75  0.12 -0.24  0.85_*  0.74_*  0.16  0.08
39 -0.02  0.02 -0.03  0.05  0.07  1.21_*  0.00  0.13
47 -0.06  0.07  0.03 -0.07 -0.15  1.22_*  0.01  0.14
60  0.06 -0.08 -0.55  0.45 -0.60  1.22_*  0.09  0.20_*
78  0.03  0.00  0.09 -0.20 -0.30  1.18_*  0.02  0.13
```

12.11.3 Transformations

When residual analysis reveals serious problems or when the relationships between the response and predictors are clearly non-linear, regression may still yield a reasonable model with either a transformation of the response variable, the predictors, or both response and predictors. When a scatterplot between the response and a predictor shows a non-linear relationship where the residuals are reasonably normal in distribution, appropriate transformations on the predictor may linearize the relationship between the variables without drastically altering the distribution of the residuals. After the transformation of the predictor(s), the residuals produced with the transformed variable(s) in the new model will need to be reanalyzed to assure normality assumptions are still satisfied.

Example 12.20 ▷ *Transformation of Predictors* ◷ The data frame **SIMDATAXT** contains simulated data for the response, y , and predictors, x_1 , x_2 , and x_3 . Apply appropriate transformations to x_1 , x_2 , and x_3 to linearize the relationships between the response and predictors one at a time.

Solution: Figures 12.16 on the facing page, 12.17 on page 852, and 12.18 on page 853 were created with code similar to R Code 12.37, R Code 12.38 on the facing page, and R Code 12.39 on page 852, respectively.

Transform x_1 : The top left graph in Figure 12.16 on the facing page shows a non-linear relationship between y and x_1 . The second graph shows the residuals from regressing y on x_1 ; both the first and second graphs suggest a simple transformation on x_1 . The pattern suggests a square root transformation. The resulting scatterplot and residual analysis for regressing y on $x_1^{0.5} = \sqrt{x_1}$ are illustrated in the bottom row of graphs. The curvilinear relationship evident in both the scatterplot and the residual plot using the untransformed x_1 disappear once a square root transformation is applied to x_1 .

R Code 12.37

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(2, 3))
> plot(y ~ x1, data = SIMDATAXT)
> with(data = SIMDATAXT,
+       lines(x1, x1^.5))           # function y = x1^.5
> plot(lm(y ~ x1, data = SIMDATAXT), which = c(1, 2))
> plot(y ~ I(x1^.5), data = SIMDATAXT)
> mod1 <- lm(y ~ I(x1^.5), data = SIMDATAXT)
> abline(mod1)
> plot(mod1, which = c(1, 2))
> par(opar)
```

The identity function, `I()`, is used to inhibit the interpretation of `^` as a formula operator. Operators such as `+`, `-`, `*`, and `^` have different meanings in formulas. In cases where the user wants to use arithmetical operators in a formula, they should be protected with the identity function. There are negative values in x_1 , and taking their square root produces NA values. R, by default, removes missing observations in its `lm()` function with `na.action = na.omit`. The first graph in the bottom row of Figure 12.16 on the next page shows a linear relationship between y and $\sqrt{x_1}$ suggesting the transformation was appropriate. The second graph in the bottom row of Figure 12.16 reveals no problems with the residuals when y is regressed on the transformed variable, $\sqrt{x_1}$, as a band of points without a pattern around $y = 0$ is observed. The third graph in the bottom row of Figure 12.16 suggests the errors

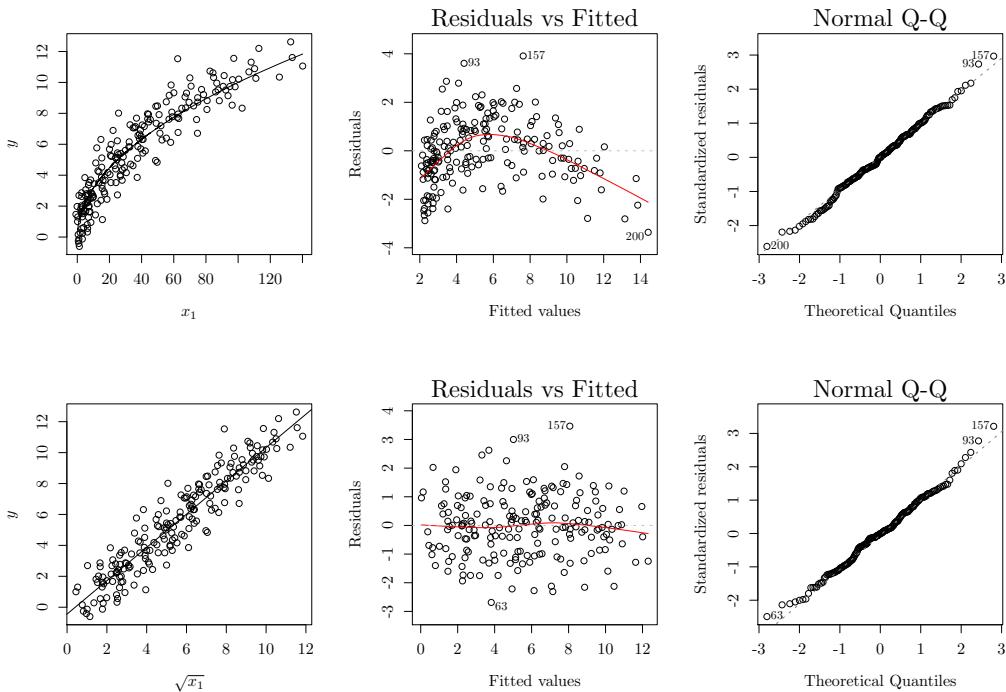


FIGURE 12.16: Scatterplot, residuals versus fitted values, and quantile-quantile plot of standardized residuals for y versus x_1 and y versus $\sqrt{x_1}$ models

from the model where y is regressed on the transformed variable, $\sqrt{x_1}$, do not deviate in any serious fashion from normality.

Transform x_2 : The concave up relationship depicted in the first two graphs of Figure 12.17 on the following page suggests a quadratic transformation on x_2 . The resulting scatterplot and residual graphs for the transformed predictor are depicted in the bottom row of graphs.

R Code 12.38

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(2, 3))
> plot(y ~ x2, data = SIMDATAXT)
> with(data = SIMDATAXT, lines(x2, x2^2)) # function y = x2^2
> plot(lm(y ~ x2, data = SIMDATAXT), which = c(1, 2))
> plot(y ~ I(x2^2), data = SIMDATAXT)
> mod2 <- lm(y ~ I(x2^2), data = SIMDATAXT)
> abline(mod2)
> plot(mod2, which = c(1, 2))
> par(opar)
```

The first graph in the bottom row of Figure 12.17 on the next page shows a linear relationship between y and x_2^2 suggesting the transformation was appropriate. The second graph in the bottom row of Figure 12.17 on the following page reveals no problems with the residuals when y is regressed on the transformed variable, x_2^2 , as a band of points without a pattern around $y = 0$ is observed. The third graph in the bottom row of Figure 12.17 on the next page suggests the errors from the model where y is regressed on the transformed variable,

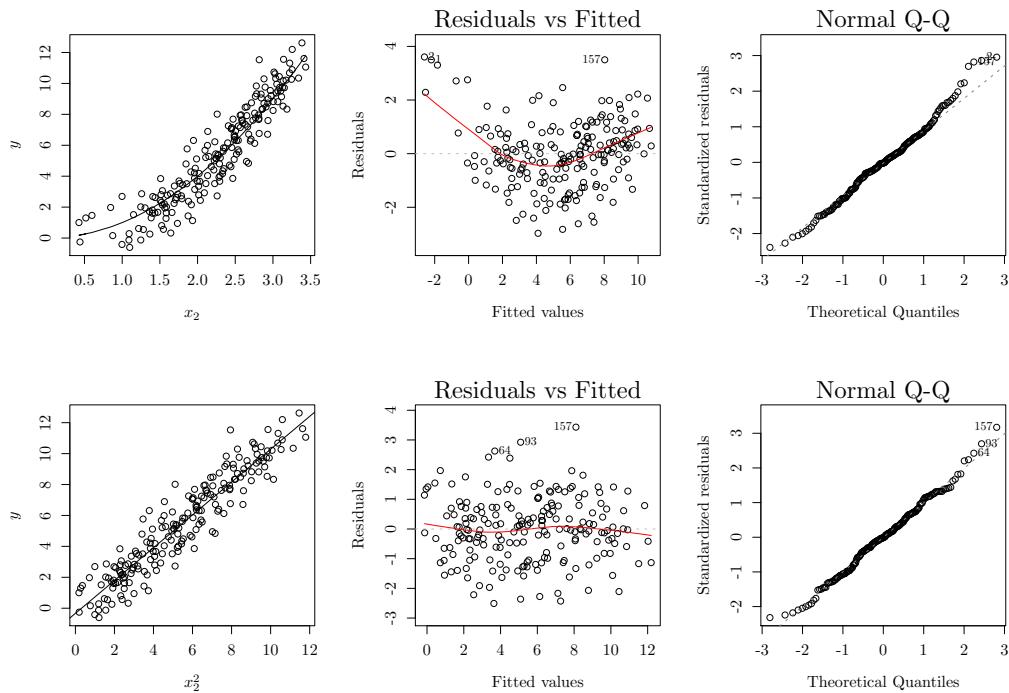


FIGURE 12.17: Scatterplot, residuals versus fitted values, and quantile-quantile plot of standardized residuals for y versus x_2 and y versus x_2^2 models

x_2^2 , do not deviate in any serious fashion from normality.

Transform x_3 : The first two graphs of Figure 12.18 on the facing page suggest a reciprocal transformation on x_3 . As before, the graphs in the second row of Figure 12.18 are for the transformed predictor (x_3).

R Code 12.39

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(2, 3))
> plot(y ~ x3, data = SIMDATAXT)
> with(data = SIMDATAXT,
+       lines(x3, x3^(-1)))          # function Y = 1/x3
> plot(lm(y ~ x3, data = SIMDATAXT), which = c(1, 2))
> plot(y ~ I(x3^(-1)), data = SIMDATAXT)
> mod3 <- lm(y ~ I(x3^(-1)), data = SIMDATAXT)
> abline(mod3)
> plot(mod3, which = c(1, 2))
> par(opar)
```

The first graph in the bottom row of Figure 12.18 on the next page shows a linear relationship between y and x_3^{-1} suggesting the transformation was appropriate. The second graph in the bottom row of Figure 12.18 on the facing page reveals no problems with the residuals when y is regressed on the transformed variable, x_3^{-1} , as a band of points without a pattern around $y = 0$ is observed. The third graph in the bottom row of Figure 12.18 on the next page suggests the errors from the model where y is regressed on the transformed variable,

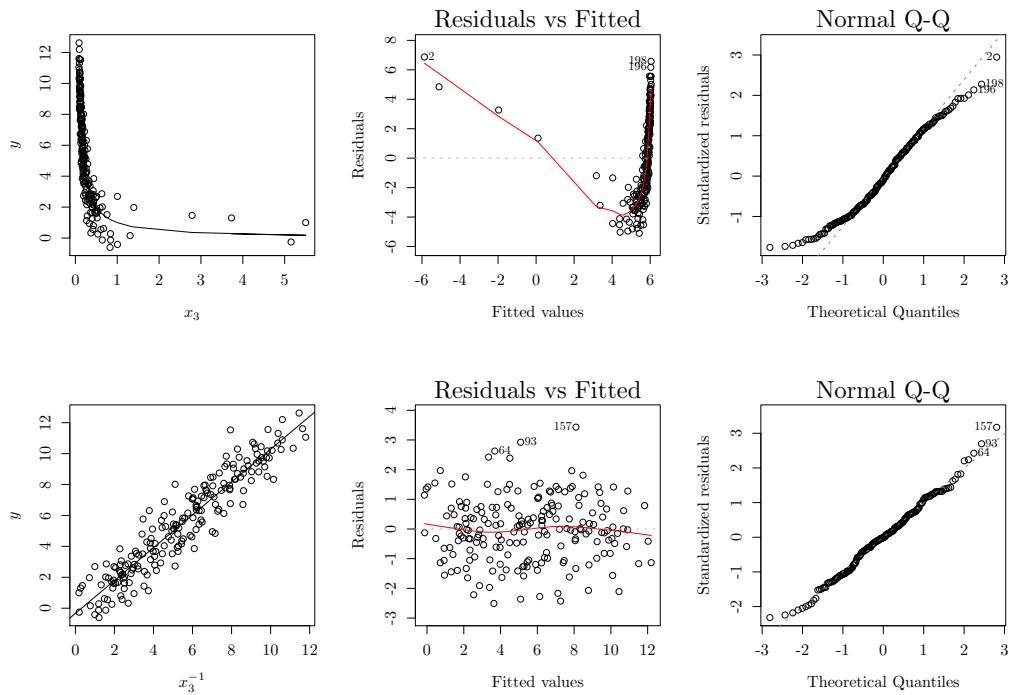


FIGURE 12.18: Scatterplot, residuals versus fitted values, and quantile-quantile plot of standardized residuals for y versus x_3 and Y versus x_3^{-1} models

x_3^{-1} , do not deviate in any serious fashion from normality.

12.11.3.1 Collinearity

Collinearity in regression occurs when some of the predictors are a linear combination of other predictors. When $\mathbf{X}'\mathbf{X}$ is singular, there is said to be exact collinearity, and there is no unique estimate of β . When $\mathbf{X}'\mathbf{X}$ is nearly singular, the problem is often called **multicollinearity**. Multicollinearity causes problems with the estimation of β and its subsequent interpretation. Severe multicollinearity can cause the sign of the coefficients to be opposite what is expected and typically inflates the standard errors of the estimates to the point where variables appear no longer to be significant. Two techniques to detect collinearity include computation of the condition number and computation of the variance inflation factor.

The **condition number** κ is defined as the square root of the largest eigenvalue of $\mathbf{X}'\mathbf{X}$ divided by the smallest eigenvalue of $\mathbf{X}'\mathbf{X}$. κ values between 30 and 100 indicate that there are moderate to strong dependencies among the predictors. κ values greater than 100 indicate serious multicollinearity problems. The function `kappa()` can be used to estimate the condition number of a matrix.

A related method of detecting multicollinearity is to regress x_j on all of the other predictors. When the coefficient of determination (R_j^2) from regressing x_j on all of the other predictors is near one, there is multicollinearity among the predictors. The **variance**

inflation factor is defined as

$$\text{VIF}_j = \frac{1}{1 - R_j^2}. \quad (12.68)$$

When there are dependencies among the predictors, R_j^2 will be near one and VIF_j will be large. VIF_j values greater than 10 suggest serious collinearity. The VIF_j for a predictor x_j can be interpreted as the factor ($\sqrt{\text{VIF}_j}$) by which the standard error of $\hat{\beta}_j$ is increased due to the presence of multicollinearity.

Example 12.21 \triangleright **Multicollinearity** \triangleleft In Example 12.20 on page 850, using the data frame **SIMDATAXT**, y was regressed on the transformed variables x_1 , x_2 , and x_3 one at a time.

- (a) Regress y on $x_1^{0.5}$, x_2^2 , and x_3^{-1} and store the results in the object **modC**. Are there any linear dependencies among the predictors?
- (b) Regress y on $x_1^{0.5}$ and x_2^2 and store the results in the object **modB**. Compute the condition number for **modB** and the VIF for $x_1^{0.5}$ and x_2^2 . Verify that the standard error for $\hat{\beta}_1$ from a model where y is regressed solely on $x_1^{0.5}$ (**mod1**) and the standard error for $\hat{\beta}_1$ from **modB** increases by approximately $\sqrt{\text{VIF}_1}$.

Solution: The answers are as follows:

- (a) From the output of R Code 12.40 it is seen that $\mathbf{X}'\mathbf{X}$ is singular.

R Code 12.40

```
> modC <- lm(y ~ I(x1^0.5) + I(x2^2) + I(x3^(-1)), data = SIMDATAXT)
> summary(modC) # lm summary
```

Call:

```
lm(formula = y ~ I(x1^0.5) + I(x2^2) + I(x3^(-1)), data = SIMDATAXT)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.5639	-0.7755	-0.0117	0.7532	3.4386

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.4219	0.1724	-2.447	0.0153 *
I(x1^0.5)	0.4500	0.5412	0.831	0.4068
I(x2^2)	0.6244	0.5384	1.160	0.2475
I(x3^(-1))	NA	NA	NA	NA

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.082 on 195 degrees of freedom

(2 observations deleted due to missingness)

Multiple R-squared: 0.8861, Adjusted R-squared: 0.8849

F-statistic: 758.2 on 2 and 195 DF, p-value: < 2.2e-16

In particular, x_2 is a function of x_3 ($x_2 \equiv \frac{1}{\sqrt{x_3}}$). Note the agreement in the first five observations for x_2 and $\frac{1}{\sqrt{x_3}}$ in the following code.

```
> cbind(SIMDATAXT$x2, SIMDATAXT$x3^(-0.5))[1:5, ]
```

[,1]	[,2]
[1,] 0.5177716	0.5177716
[2,] 0.4262969	0.4262969
[3,] 0.4405244	0.4405244
[4,] 0.5988884	0.5988884
[5,] 0.8465002	0.8465002

(b) R Code 12.41 computes the condition number for `modB`.

R Code 12.41

```
> modB <- lm(y ~ I(x1^.5) + I(x2^2), data = SIMDATAXT)
> X <- model.matrix(modB)
> eigen(t(X)%*%X, only.values = TRUE)$values # extract eigenvalues
```

[1]	15394.239140	39.566738	2.008989
-----	--------------	-----------	----------

```
> lambda.max <- max(eigen(t(X)%*%X, only.values = TRUE)$values)
> lambda.min <- min(eigen(t(X)%*%X, only.values = TRUE)$values)
> condition.number <- sqrt(lambda.max / lambda.min)
> condition.number
```

[1]	87.53672
-----	----------

The condition number for `modB` is 87.5367, suggesting strong dependencies exist between $x_1^{0.5}$ and x_2^2 . The function `kappa()` may also be used to compute the condition number, provided the argument `exact = TRUE` is used.

```
> kappa(X, exact = TRUE)
```

[1]	87.53672
-----	----------

R Code 12.42 computes the VIF according to (12.68) and verifies the answer using the function `vif()` from the package `car`.

R Code 12.42

```
> VIF1 <- 1/(1 - summary(lm(I(x1^.5) ~ I(x2^2), data = SIMDATAXT))$r.sq)
> VIF1
```

[1]	382.7241
-----	----------

```
> library(car)
> VIF2 <- vif(modB)
> VIF2
```

I(x1^.5)	I(x2^2)
382.7241	382.7241

The variance inflation factor for $x_1^{0.5}$ and x_2^2 is 382.7241. R Code 12.43 on the following page verifies that the standard error for $\hat{\beta}_1$ from a model where y is regressed solely on $x_1^{0.5}$ to the standard error for $\hat{\beta}_1$ in `modB` (19.5461) increases by approximately $\sqrt{VIF_1} = 19.5633$.

R Code 12.43

```

> mod1 <- lm(y ~ I(x1^0.5), data = SIMDATAXT)
> coef(summary(mod1)) # coefficients mod1

            Estimate Std. Error    t value   Pr(>|t|) 
(Intercept) -0.4410092 0.17178289 -2.567247 1.099639e-02
I(x1^0.5)     1.0768822 0.02769023 38.890330 4.239490e-94

> se.x1.mod1 <- coef(summary(mod1))[2, 2]
> se.x1.modB <- coef(summary(modB))[2, 2]
> ratio <- se.x1.modB/se.x1.mod1
> ratio

[1] 19.54613

> sqrt(vif(modB))

I(x1^0.5)    I(x2^2)
19.56334 19.56334

```

The standard error for $\hat{\beta}_1$ based on `mod1` is 0.0277 while the standard error for $\hat{\beta}_1$ based on `modB` is 0.5412. The ratio of 0.5412 to 0.0277 is 19.5461, which is approximately equal to the square root of the VIF for `modB` (19.5633). In this problem, the introduction of x_2^2 to a model that already contained $x_1^{0.5}$ increased the standard error for $\hat{\beta}_1$ by 19.5461. From the summary of `modB`, neither of the estimated coefficients for β_1 or β_2 are significant, yet from Example 12.20 on page 850, the coefficients for both $x_1^{0.5}$ and x_2^2 , when taken alone, are significant. ■

12.11.3.2 Transformations for Non-Normality and Unequal Error Variances

With the normal distribution, the mean and variance are independent of one another. This is not the case with many other distributions. One such example is the Poisson distribution, where the mean is equal to the variance. Quite often, non-normality and unequal error variances appear together. This “double” problem can often be remedied by transforming the response variable \mathbf{Y} . The “double” problem can be identified by an increasing or decreasing band in a curvilinear residual plot. Transformations on the response variable will frequently both linearize a curvilinear relationship and fix the problem of unequal error variances. Other times, transformations on both the response and predictors will be required to meet the assumptions of the normal linear model. One technique that searches computationally for an appropriate transformation of the response variable that directly addresses normality is the Box-Cox method. The Box-Cox method estimates the parameter λ for the transformation $Y' = Y^\lambda$, where

$$Y' = \begin{cases} \frac{Y^\lambda - 1}{\lambda} & \text{for } \lambda \neq 0, \\ \ln Y & \text{for } \lambda = 0, \end{cases} \quad (12.69)$$

by the method of maximum likelihood. The functions `boxcox()`, and `boxCox()` from the packages `MASS` and `car`, respectively, both produce a plot of the log-likelihood against the transformation parameter λ for a particular model. By default, the range of λ for both functions is from -2 to 2 ; however, once the value of λ that maximizes the log-likelihood is known, the range of the plot can be tightened to highlight the area where the function

is maximized with the argument `lambda =`. For more details, see the help file for either `boxcox()` or `boxCox()`. The functions `boxcox()` and `boxCox()` are generally used to obtain an idea for an appropriate transformation. The value of λ that maximizes the log-likelihood function may turn out to be 0.53; but if there is a possible explanation for taking the square root of the response, the transformation applied should be $\lambda = 0.5$ and not the value that maximizes the log-likelihood function.

Example 12.22 \triangleright **Box-Cox Transformation** \triangleleft Use the data frame `SIMDATAST` and the `boxCox()` function to find the transformation on y_1 that maximizes the log-likelihood of the model created by regressing y_1 on x_1 . Once the value of λ that maximizes the log-likelihood is known, reduce the range of the plot produced with `boxCox()` to focus on the area around the value of λ that maximizes the log-likelihood.

Solution: Using the default range $-2 < \lambda < 2$, the `boxCox()` function shows that the transformation $\lambda = 0$, that is, $\ln Y$, comes close to maximizing the log-likelihood and is included in the 95% confidence band for λ , as seen in Figure 12.19. Consequently, the range of λ is reduced and plotted over the region -0.3 to 0.3 using the argument `lambda = seq(-0.2, 0.2, 0.01)` in R Code 12.44.

R Code 12.44

```
> library(car)
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(1, 2)) # split graphics device 1 row 2 columns
> modx1 <- lm(y1 ~ x1, data = SIMDATAST)
> boxCox(modx1)
> boxCox(modx1, lambda = seq(-0.2, 0.2, 0.01))
> par(opar)
```

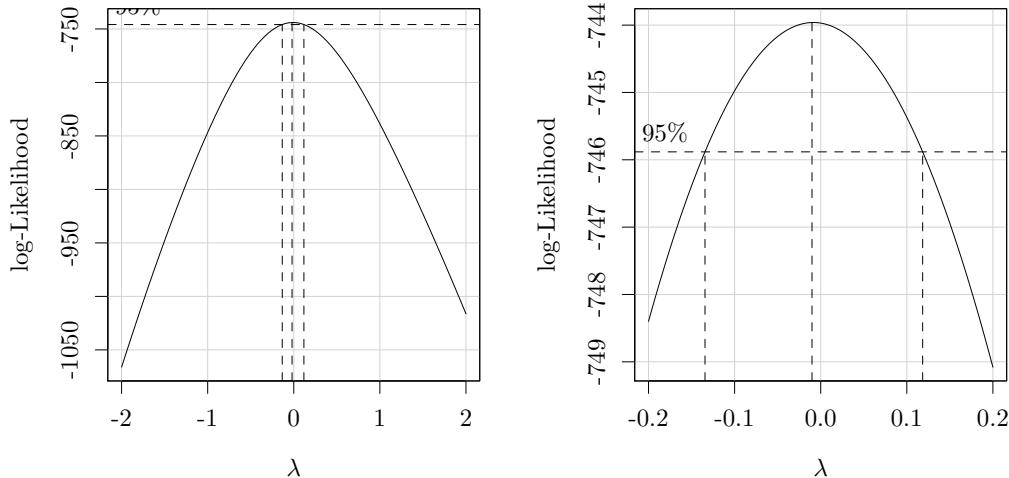


FIGURE 12.19: Box-Cox graph of λ for Example 12.22

The appropriate transformation based on the graphs in Figure 12.19 is to take the natural logarithm of the response variable. ■

Example 12.23 ▷ **Transforming Y with ln** ◷ Use the data frame **SIMDATAST** to create and display six graphs using a 2 by 3 layout.

- (a) Produce a scatterplot of y_1 versus x_1 .
- (b) Plot the residuals versus fits for the model created by regressing y_1 on x_1 (call this model **modx1**). Based on the first two graphs, does a logarithmic transformation for the response variable make sense?
- (c) Use and plot the results from the **boxCox()** function applied to **modx1**.
- (d) In the second row of graphs, create a scatterplot of $\ln y_1$ versus x_1 .
- (e) Create a plot of the residuals versus the fits for the model $\log(y_1) \sim x_1$.
- (f) Create a quantile-quantile normal plot of the residuals from the model $\log(y_1) \sim x_1$.
- (g) Based on the second row of graphs, do the assumptions for the normal error model seem to be satisfied for the model $\log(y_1) \sim x_1$? Note that the default understanding of **log** in R is $\log_e = \ln = \log$.

Solution: R Code 12.45 reads in the current settings and splits the graphics device into two rows and three columns.

R Code 12.45

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(2, 3)) # split graphics device 2 rows 3 columns
```

- (a) R Code 12.46 creates a scatterplot of y_1 versus x_1 shown in the top left of Figure 12.20 on the facing page.

R Code 12.46

```
> plot(y1 ~ x1, data = SIMDATAST)
```

- (b) R Code 12.47 stores the results from regressing y_1 on x_1 in the object **modx1**. The residuals from **modx1** are plotted and shown in the middle graph, top row, of Figure 12.20 on the facing page.

R Code 12.47

```
> modx1 <- lm(y1 ~ x1, data = SIMDATAST)
> plot(modx1, which = 1)
```

- (c) R Code 12.48 creates a Box-Cox plot of **modx1**. The resulting output is shown in the top right graph of Figure 12.20 on the facing page.

R Code 12.48

```
> boxCox(modx1, lambda = seq(-0.2, 0.2, 0.01))
```

Based on the top right graph of Figure 12.20 on the next page, transforming the response variable by taking the natural logarithm is recommended.

- (d) R Code 12.49 on the facing page creates a scatterplot of $\ln y_1$ versus x_1 . The scatterplot is the first graph in the bottom row of Figure 12.20 on the next page.

R Code 12.49

```
> plot(log(y1) ~ x1, data = SIMDATAST)
```

(e) R Code 12.50 creates a residuals versus fitted values plot for the regression model where $\ln y_1$ is regressed on x_1 . The residual versus fitted values plot is the second graph in the bottom row of Figure 12.20.

R Code 12.50

```
> plot(lm(log(y1) ~ x1, data = SIMDATAST), which = 1)
```

(f) R Code 12.51 creates a normal quantile-quantile plot for the regression model where $\ln y_1$ is regressed on x_1 . The normal quantile-quantile plot is the third graph in the bottom row of Figure 12.20.

R Code 12.51

```
> plot(lm(log(y1) ~ x1, data = SIMDATAST), which = 2)
> par(opar)
```

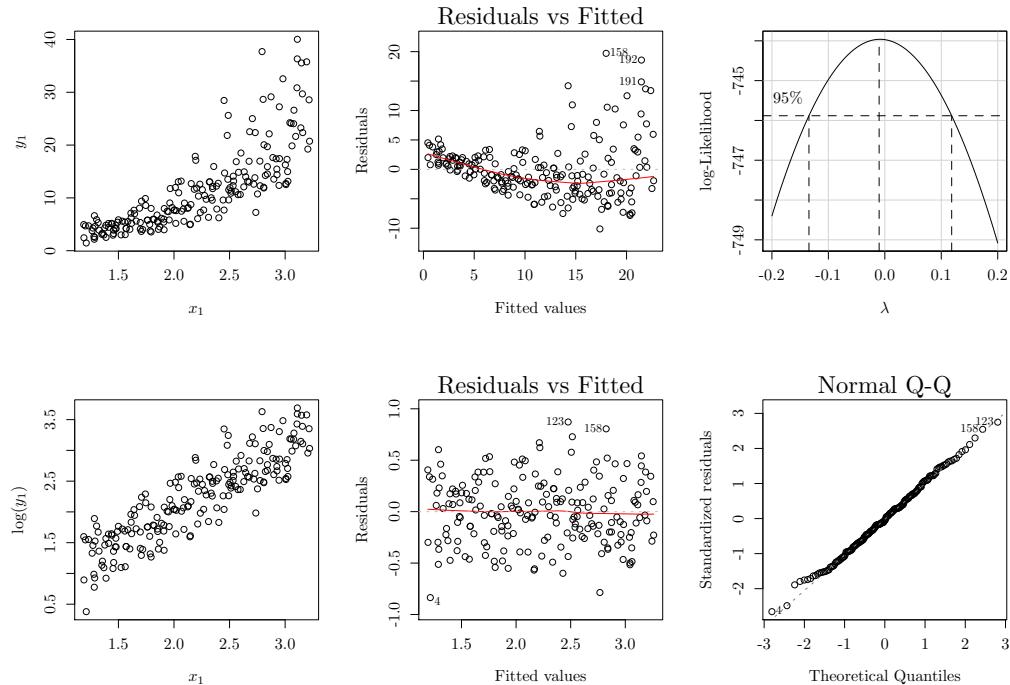


FIGURE 12.20: Scatterplot and residual versus fitted plot of y_1 versus x_1 ; Box-Cox plot of λ ; scatterplot, residual versus fitted plot, and quantile-quantile plot of $\ln(y_1)$ versus x_1

Based on the first two graphs of the first row of Figure 12.20, and the subsequent plot of λ , the transformation $\lambda = 0$, that is, $\ln(Y)$, is justified. Once the response is transformed, the problems of non-normality and unequal variance of the errors apparently disappear. ■

Example 12.24 ▷ **Transforming Y with a Reciprocal** ◁ Use the data frame **SIMDATAST** to create and display six graphs using a 2 by 3 layout.

- (a) Produce a scatterplot of y_2 versus x_2 .
- (b) Plot the residuals versus fits for the model created by regressing y_2 on x_2 (call this model **modx2**). Based on the first two graphs, does a reciprocal transformation for the response variable make sense?
- (c) Use and plot the results from the **boxCox()** function applied to **modx2**.
- (d) In the second row of graphs, create a scatterplot of y_2^{-1} versus x_2 .
- (e) Create a plot of the residuals versus the fits for the model $y2^{-1} \sim x2$.
- (f) Create a normal quantile-quantile plot of the residuals from the model $y2^{-1} \sim x2$.
- (g) Based on the second row of graphs, do the assumptions for the normal error model seem to be satisfied for the model $y2^{-1} \sim x2$?

Solution: R Code 12.52 reads in the current settings and splits the graphics device into two rows and three columns.

R Code 12.52

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(2, 3)) # split graphics device 2 rows 3 columns
```

- (a) R Code 12.53 creates a scatterplot of y_2 versus x_2 shown in the top left plot of Figure 12.21 on the next page.

R Code 12.53

```
> plot(y2 ~ x2, data = SIMDATAST)
```

- (b) R Code 12.54 stores the results from regressing y_2 on x_2 in the object **modx2**. The residuals from **modx2** are plotted and shown in the middle graph, top row, of Figure 12.21 on the next page.

R Code 12.54

```
> modx2 <- lm(y2 ~ x2, data = SIMDATAST)
> plot(modx2, which = 1)
```

- (c) R Code 12.55 creates a Box-Cox plot of **modx2**. The resulting output is shown in the top right of Figure 12.21 on the next page.

R Code 12.55

```
> boxCox(modx2, lambda = seq(-1.4, -0.5, 0.01))
```

Based on the top right graph of Figure 12.21 on the facing page, transforming the response by taking the reciprocal is recommended.

- (d) R Code 12.56 on the next page creates a scatterplot of y_2^{-1} versus x_2 . The scatterplot is the first graph in the bottom row of Figure 12.21 on the facing page.

R Code 12.56

```
> plot(y2^(-1) ~ x2, data = SIMDATAST)
```

(e) R Code 12.57 creates a residuals versus fitted values plot for the regression model where y_2^{-1} is regressed on x_2 . The residuals versus fitted values plot is the second graph on the bottom row of Figure 12.21.

R Code 12.57

```
> plot(lm(y2^(-1) ~ x2, data = SIMDATAST), which = 1)
```

(f) R Code 12.58 creates a normal quantile-quantile plot of the residuals from the regression model where y_2^{-1} is regressed on x_2 . The residuals versus fitted values plot is the third graph on the bottom row of Figure 12.21.

R Code 12.58

```
> plot(lm(y2^(-1) ~ x2, data = SIMDATAST), which = 2)
> par(opar)
```

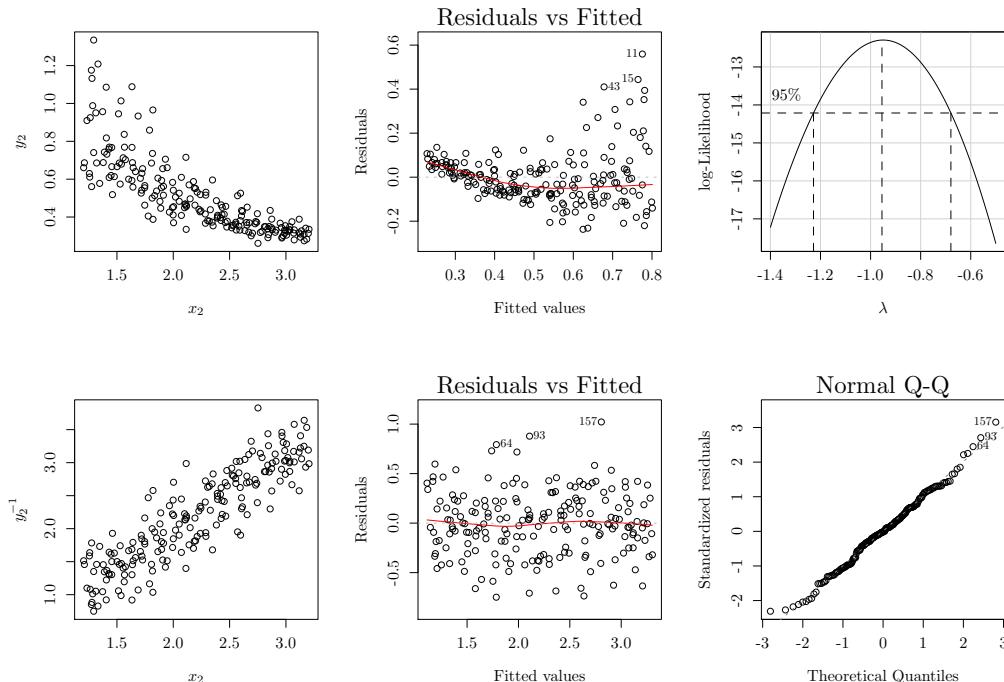


FIGURE 12.21: Scatterplot and residual versus fitted plot of Y_2 versus x_2 ; Box-Cox plot of λ ; scatterplot, residual versus fitted plot, and quantile-quantile plot of Y_2^{-1} versus x_2

Based on the first two graphs of the first row of Figure 12.21, and the subsequent plot of λ , the transformation $\lambda = -1$, that is, y_2^{-1} , is justified. Once the response is transformed,

the non-normality problem as well as the unequal variance of the errors problem appear to vanish.

As mentioned earlier, at times it will be necessary to transform the response and the predictor. Consider the top left graph (I) in Figure 12.22 on the facing page along with the top middle graph (II) produced with `boxCox()` suggesting a $\ln Y$ transformation. Graph III, which depicts a scatterplot of $\log_e Y$ versus x_3 with a superimposed line from the least squares fit of the model $\log(y_1) \sim x_3$ reveals a slight curvilinear pattern. The curvilinear pattern is further highlighted in graph IV of the residuals versus fits plot for the same model shown in graph III. The curvilinear pattern suggests some type of transformation for x_3 . The slight curvature is eliminated in both the scatterplot (graph V) and the residual versus fitted values plot (graph VI) by applying a square root transformation to x_3 . Graphs similar to Figure 12.22 can be created using R Code 12.59.

R Code 12.59

```
> opar <- par(no.readonly = TRUE) # copy of current settings
> par(mfrow = c(2, 3)) # split graphics device 2 rows 3 columns
> plot(y1 ~ x3, data = SIMDATAST, main = "(I)")
> mod1 <- lm(y1 ~ x3, data = SIMDATAST)
> boxCox(mod1, lambda = seq(-0.2, 0.37, 0.01))
> title(main = "(II)")
> mod2 <- lm(log(y1) ~ x3, data = SIMDATAST)
> plot(log(y1) ~ x3, data = SIMDATAST, main = "(III)")
> abline(mod2)
> plot(mod2, which = 1, main = "(IV)")
> plot(log(y1) ~ I(x3^0.5), data = SIMDATAST, main = "(V)")
> mod3 <- lm(log(y1) ~ I(x3^.5), data = SIMDATAST)
> abline(mod3)
> plot(mod3, which = 1, main ="(VI)")
> par(opar)
```

12.12 Model Validation

Model validation generally includes an assessment of whether the regression model provides an adequate description of the data, an analysis of the residuals, and some check on the predictive ability of the chosen model. Specifically, the predictive performance of a regression model is evaluated with an independent data set that was not used to build the regression model. This section focuses on how to assess the predictive performance of a regression model. In the absence of a designated independent testing set, the predictive performance of a regression model can be estimated with either the validation set approach, leave-one-out cross-validation, or k -fold cross-validation. Each method of assessing a regression model's predictive performance is discussed and the rationale for using k -fold cross-validation over the other two methods is provided.

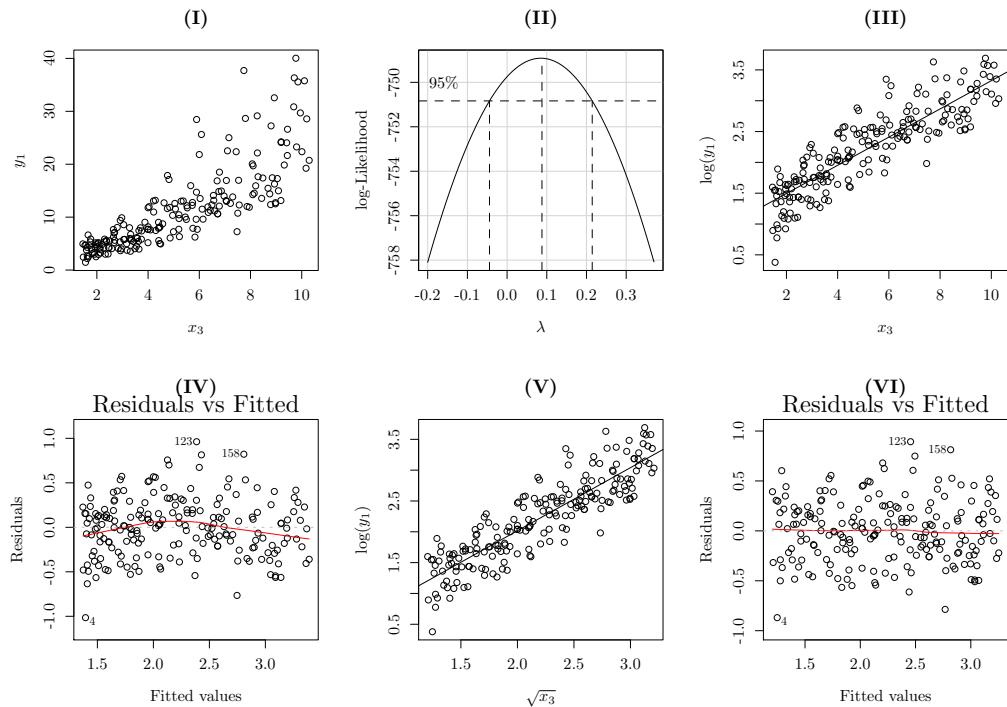


FIGURE 12.22: Process of model building with transformations: (I) original scatterplot, (II) `boxCox()` transformation suggestion, (III) scatterplot with Y transformed, (IV) residual plot shows curvature, (V) x -variable transformed, (VI) residuals appear normalized

12.12.1 The Validation Set Approach

The basic idea behind the **validation set approach** is to split the available data into a **training** set and a **testing** set. A regression model is developed using only the training set. Figure 12.23 on the following page illustrates an even split of the available data into a training set and a testing set. The percent of values that are allocated into training and testing may vary based on the size of the available data. It is not unusual to allocate 70–75% of the available data as the training set and the remaining 25–30% as the testing set. The predictive performance of a regression model is assessed using the testing set. One of the more common methods to assess the predictive performance of a regression model is the mean square prediction error (*MSPE*). The *MSPE* is defined as

$$MSPE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2. \quad (12.70)$$

To have an accurate estimate of the regression model's validation set error, (12.70) should be computed using the testing set with the model developed with the training set. Since the n available observations are randomly assigned to the training set and the testing set, different random assignments will return different values (sometimes quite different) for the *MSPE* computed from the test set of observations.

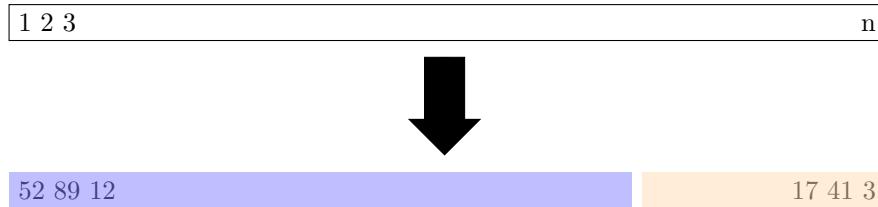


FIGURE 12.23: Schematic display of the validation set approach based on James et al. (2013, Figure 5.1). A set of n observations are randomly split into a training set (darker shade starting with data observations 52, 89, and 12) and a testing set (lighter shade ending with data observations 17, 41, and 3).

12.12.2 Leave-One-Out Cross-Validation

The leave-one-out cross-validation (LOOCV) eliminates the problem of variability in *MSPE* present in the validation set approach. The LOOCV is similar to the validation set approach as the available n observations are split into training and testing sets. The difference is that each of the available n observations are split into n training and n testing sets where each of the n training sets consist of $n - 1$ observations and each of the testing sets consists of a single different value from the original n observations. Figure 12.24 provides a schematic display of the leave-one-out cross-validation process with testing sets (light shade) and training sets (dark shade) for a data set of n observations. The *MSPE* is computed with each testing set resulting in n values of *MSPE*. The LOOCV estimate for the test *MSPE* is the average of these n *MSPE* values denoted as

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n MSPE_i. \quad (12.71)$$

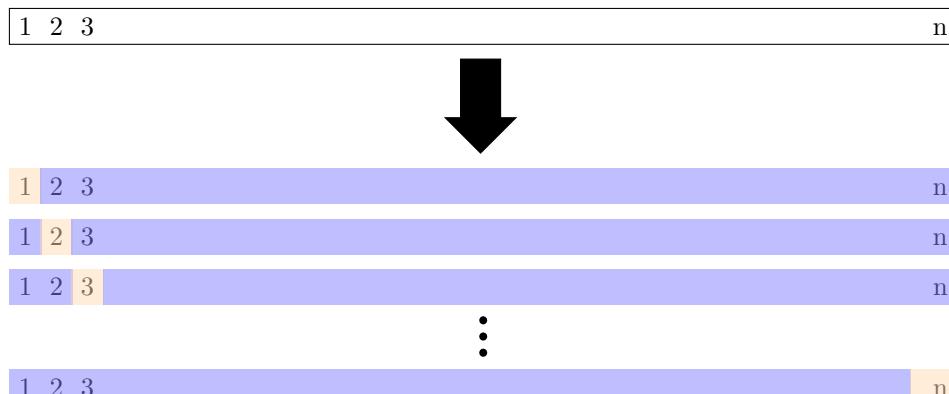


FIGURE 12.24: Schematic display of the leave-one-out cross-validation based on James et al. (2013, Figure 5.3). The first training set consists of all n observations minus the first observation (which is the first testing set), the second training set consists of all n observations minus the second observation (which is the second testing set), and so forth.

Performing LOOCV multiple times will always return the same value for $CV_{(n)}$ since there is no randomization in the training and testing sets.

12.12.3 k-Fold Cross-Validation

k -fold cross-validation is similar to LOOCV in that the available data is split into training sets and testing sets; however, instead of creating n different training and testing sets, k folds/groups of training and testing sets are created where $k < n$ and each fold consists of roughly n/k values in the testing set and $1 - n/k$ values in the training set. Figure 12.25 shows a schematic display of 5-fold cross-validation. The lightly shaded rectangles are the testing sets and the darker shaded rectangles are the training sets. The $MSPE$ is computed on each of the k folds using the testing set to evaluate the regression model built from the training set. The average of k $MSPE$ values is denoted as

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^k MSPE_i. \quad (12.72)$$

Note that LOOCV is a special case of k -fold cross-validation where k is set equal to n . An important advantage k -fold cross-validation has over LOOCV is that $CV_{(k)}$ for $k = 5$ or $k = 10$ provides a more accurate estimate of the test error rate than does $CV_{(n)}$ (James et al., 2013, page 183).

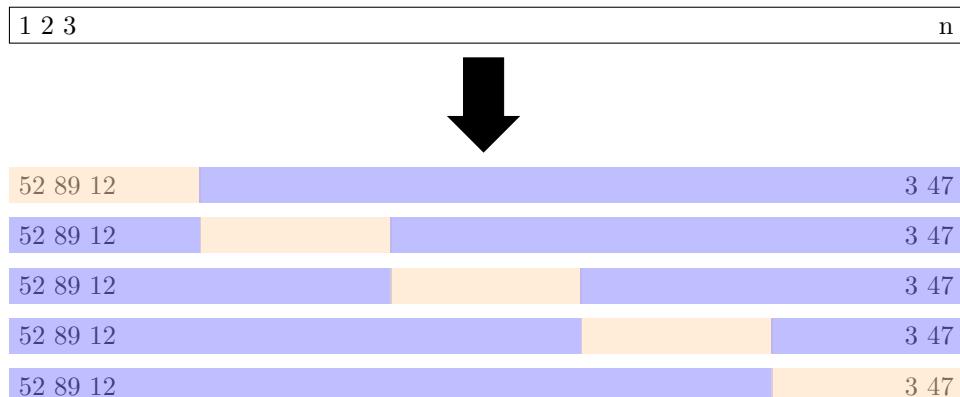


FIGURE 12.25: Schematic display of 5-fold cross-validation based on James et al. (2013, Figure 5.5). The lightly shaded rectangles represent the testing sets and the darker shaded rectangles represent the training data sets.

Example 12.25 Create and validate regression models for predicting wrestlers' hydrostatic fat (`hwfat`) using the data frame `HSWRESTLER`. Do not use the variables `tanfat` or `skfat` as possible predictors. Remove observations 22, 27, 32, 35, and 60, which may have poorly measured skin fold values (`abs`, `triceps`, and `subscap`) according to part (d) of Example 12.19 on page 845.

- (a) Use the validation set approach to choose the best regression model returned from the function `regsubsets()` when `hwfat` is the response and possible predictors include `age`, `ht`, `abs`, `triceps`, and `subscap`. Use `set.seed = 5` and split the available data into

training and testing sets where roughly 70% of the available data is used for the training set and the remainder is used for the testing set.

- (b) Use leave-one-out cross-validation to choose the best regression model returned from the function `regsubsets()` when `hwfat` is the response and possible predictors include `age`, `ht`, `abs`, `triceps`, and `subscap`.
- (c) Use the 5-fold cross-validation to choose the best regression model returned from the function `regsubsets()` when `hwfat` is the response and possible predictors include `age`, `ht`, `abs`, `triceps`, and `subscap`.

Solution: R Code 12.60 creates a vector `study` where the questionable observations are stored. The observations in `study` are excluded from possible selection and the function `sample()` is used to generate observations for a training set (`train`) and a testing set (`test`) where roughly 70% of the available observations are assigned to `train`.

R Code 12.60

```
> study <- c(22, 27, 32, 35, 60) # poorly measured values
> set.seed(5)
> train <- sample(c(TRUE, FALSE), size = nrow(HSWRESTLER[-study, ]),
+                  replace = TRUE, prob = c(0.70, 0.30))
> prop.table(table(train)) # compute percent in train

train
  FALSE      TRUE
0.369863 0.630137

> test <- (!train)
> prop.table(table(test)) # compute percent in test

test
  FALSE      TRUE
0.630137 0.369863
```

- (a) R Code 12.61 computes all possible models for the predictors `age`, `ht`, `abs`, `triceps`, and `subscap`. Note that only the training set is used.

R Code 12.61

```
> library(leaps)
> model.exh <- regsubsets(hwfat ~ ., data = HSWRESTLER[train, 1:7],
+                           method = "exhaustive")
> summary(model.exh)

Subset selection object
Call: regsubsets.formula(hwfat ~ ., data = HSWRESTLER[train, 1:7],
  method = "exhaustive")
6 Variables (and intercept)
  Forced in  Forced out
age        FALSE      FALSE
ht         FALSE      FALSE
wt         FALSE      FALSE
abs        FALSE      FALSE
```

```

triceps      FALSE      FALSE
subscap      FALSE      FALSE
1 subsets of each size up to 6
Selection Algorithm: exhaustive
      age ht  wt  abs triceps subscap
1 ( 1 ) " " " " " " " *"      " "
2 ( 1 ) " " " " " " *" *"      " "
3 ( 1 ) " " *" " " " *" *"      " "
4 ( 1 ) " " *" " " " *" " *"      "*"
5 ( 1 ) " " *" *" " *" " *"      "*"
6 ( 1 ) *" *" *" " *" " *"      "*"

```

R Code 12.62 computes validation set errors for the best model of each size (1-6 possible predictors).

R Code 12.62

```

> test.mat <- model.matrix(hwfat ~ ., data = HSWRESTLER[test, 1:7])
> val.errors <- rep(NA, 6)
> for(i in 1:6){
+   coefi = coef(model.exh, id = i)
+   pred <- test.mat[, names(coefi)]%*%coefi
+   val.errors[i] = mean((HSWRESTLER[test, ]$hwfat - pred)^2)
+ }
> val.errors

[1] 12.670744 10.016085  9.650927  9.742823 10.013797  9.864935

> coef(model.exh, 3) # coefficients for model with 3 predictors

(Intercept)          ht          abs        triceps
15.7861224 -0.2067732  0.3608162  0.4693222

```

The validation set approach indicates a model with three predictors has the smallest *MSPE* (9.6509). Next, the entire data set (minus the questionable observations stored in *study*) is used to fit all possible models and the best model with three predictors is selected using R Code 12.63. Note that when the entire data set is used, the best three variable model does not have the same predictors as the best three variable model selected from the testing set.

R Code 12.63

```

> regfit.best <- regsubsets(hwfat ~ ., data = HSWRESTLER[-study,
+           1:7])
> coef(regfit.best, 3) # coef for best 3 predictor model

(Intercept)          age          abs        triceps
 5.4651481 -0.2198196  0.3994437  0.4060953

> mod.VSA <- lm(hwfat ~ wt + abs + triceps, data = HSWRESTLER[-study,
+           1:7])
> summary(mod.VSA)

```

Call:

```

lm(formula = hwfat ~ wt + abs + triceps, data = HSWRESTLER[-study,
  1:7])

Residuals:
    Min      1Q  Median      3Q     Max 
-6.1759 -1.7494 -0.2497  2.0080  5.5631 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 3.12319   1.66018   1.881 0.064160 .  
wt          -0.01128   0.01447  -0.780 0.438106  
abs         0.42693   0.08654   4.933 5.38e-06 *** 
triceps     0.41961   0.11024   3.806 0.000302 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 2.656 on 69 degrees of freedom
Multiple R-squared:  0.9059, Adjusted R-squared:  0.9018 
F-statistic: 221.3 on 3 and 69 DF,  p-value: < 2.2e-16

```

The final model based on the validation set approach contains the predictors `wt`, `abs`, and `triceps`. R Code 12.62 on the preceding page is fairly involved because there is no `predict()` method for `regsubsets()`. R Code 12.64 taken from James et al. (2013, page 249) creates a `predict` method for `regsubsets()` that automates the steps taken in R Code 12.62 on the preceding page.

R Code 12.64

```

> predict.regsubsets <- function(object, newdata, id, ...){
+   form <- as.formula(object$call[[2]])
+   mat <- model.matrix(form, newdata)
+   coefi = coef(object, id = id)
+   xvars <- names(coefi)
+   mat[, xvars] %*% coefi
+ }

```

The validation set *MSPE* for the best model of each size found in R Code 12.62 on the previous page is repeated in R Code 12.65 so the reader can verify how the new `predict.regsubsets()` method works. Note that the new method can be called with either `predict()` or `predict.regsubsets()`. Recall that the information in `model.exh` was computed with the `train` data. The values stored in `Y` are the `hwfat` values from the `test` set, and the values stored in `Yhat` are the new predictions using the `test` data with the best models developed with the `train` set, which are stored in `model.exh`. As before, the smallest *MSPE* (9.6509) suggests the best three predictor models using the entire data set minus the influential values should include the predictors `age`, `abs`, and `triceps`.

R Code 12.65

```

> val.errors <- rep(NA, 6)
> Y <- HSWRESTLER[test, ]$hwfat # hwfat values from test
> for(i in 1:6){
+   Yhat <- predict(model.exh, newdata = HSWRESTLER[test, ], id = i)
+   val.errors[i] <- mean((Y - Yhat)^2)

```

```
+ }
> val.errors
[1] 12.670744 10.016085 9.650927 9.742823 10.013797 9.864935

> coef(model.exh, which.min(val.errors))

(Intercept) ht abs triceps
15.7861224 -0.2067732 0.3608162 0.4693222

> regfit.best <- regsubsets(hwfat ~ ., data = HSWRESTLER[-study, 1:7])
> coef(regfit.best, which.min(val.errors))

(Intercept) age abs triceps
5.4651481 -0.2198196 0.3994437 0.4060953
```

(b) R Code 12.66 uses the LOOCV method to compute the average $MSPE$, $CV_{(n)}$. Since LOOCV is a special case of k -fold cross-validation, R Code 12.66 was written so the reader can make minimal changes to answer (c), which asks the reader to compute the average $MSPE$, $CV_{(k)}$, for the best regression model of size $i = 1 \dots 6$ using 5-fold cross-validation. Note that the data indexed with `HSW[folds != j,]` consists of the training data and the data indexed with `HSW[folds == j,]` consists of the testing data for the LOOCV method.

R Code 12.66

```
> HSW <- HSWRESTLER[-study, 1:7]
> n <- nrow(HSW)
> k <- n                      # set the number of folds equal to n
> set.seed(5)                  # set for reproducible results
> folds <- sample(x = 1:k, size = nrow(HSW), replace = FALSE)
> cv.errors <- matrix(NA, k, 6, dimnames = list(NULL, paste(1:6)))
> for(j in 1:k){
+   best.fit <- regsubsets(hwfat ~ ., data = HSW[folds != j, ])
+   for(i in 1:6){
+     pred <- predict(best.fit, newdata = HSW[folds == j, ], id = i)
+     cv.errors[j, i] <- mean((HSW$hwfat[folds == j] - pred)^2)
+   }
+ }
> mean.cv.errors <- apply(cv.errors, 2, mean)
> mean.cv.errors

      1       2       3       4       5       6
8.686337 7.361655 7.856348 8.314736 8.291063 8.223795

> coef(regfit.best, which.min(mean.cv.errors))

(Intercept) abs triceps
1.9119410 0.3929936 0.4211225
```

Using the LOOCV method, the smallest average $MSPE$, $CV_{(n)}$, is 7.3617. The smallest $CV_{(n)}$ value corresponds to the regression model with predictors `abs` and `triceps`.

(c) R Code 12.67 on the next page uses 5-fold cross-validation to compute the average $MSPE$, $CV_{(5)}$. Note that the data indexed with `HSW[folds != j,]` consists of the training data

and the data indexed with `HSW[folds == j,]` consists of the testing data for the 5-fold cross-validation method. R Code 12.67 has two changes from R Code 12.66 on the previous page: the number of folds k is changed from $k = n = 73$ to $k = 5$, and the argument `replace =` in `sample()` is changed from `FALSE` to `TRUE`.

R Code 12.67

```
> HSW <- HSWRESTLER[-study, 1:7]
> n <- nrow(HSW)
> k <- 5                      # set the number of folds equal to 5
> set.seed(5)                  # set for reproducible results
> folds <- sample(x = 1:k, size = nrow(HSW), replace = TRUE)
> cv.errors <- matrix(NA, k, 6, dimnames = list(NULL, paste(1:6)))
> for(j in 1:k){
+   best.fit <- regsubsets(hwfat ~ ., data = HSW[folds != j, ])
+   for(i in 1:6){
+     pred <- predict(best.fit, newdata = HSW[folds == j, ], id = i)
+     cv.errors[j, i] <- mean((HSW$hwfat[folds == j] - pred)^2)
+   }
+ }
> mean.cv.errors <- apply(cv.errors, 2, mean)
> mean.cv.errors

      1         2         3         4         5         6
14.297668  8.881967  9.262515  9.848583 10.204780 10.150723

> coef(regfit.best, which.min(mean.cv.errors))

(Intercept)          abs       triceps
  1.9119410    0.3929936    0.4211225
```

The smallest average *MSPE* using the 5-fold cross-validation method, $CV_{(5)}$, is 8.882, which corresponds to the regression model with predictors `abs` and `triceps`. R Code 12.68 (output and graphs not shown) evaluates the final model `mod.5fcv` one last time with respect to residuals and influential observations. Based on the analysis, the assumptions required for a linear regression model appear to be satisfied. At this point, one should feel confident using model `mod.5fcv`. The function `scatter3d()` from the package `car` shows the fitted regression plane for the model `mod.5fcv`.

R Code 12.68

```
> mod.5fcv <- lm(hwfat ~ abs + triceps, data = HSW)
> summary(mod.5fcv)
> qqPlot(mod.5fcv)
> residualPlots(mod.5fcv)
> influenceIndexPlot(mod.5fcv)
> scatter3d(hwfat ~ abs + triceps, data = HSW, surface.alpha = 0.1,
+            point.col = "lightblue", grid = TRUE)
```

12.13 Interpreting a Logarithmically Transformed Model

Variables are often transformed to fix constant variance or normality assumptions; however, transformations can complicate the interpretation of the model. Unlike many other transformations, models that use logarithmic transformations have approximate explanations without back transforming the variables.

When x has been transformed with a natural log transformation, the change in the $\ln(x)$ is roughly equal to the change in x provided the changes in x are small. Consider Figure 12.26, which graphically illustrates how changing the x values 3 and 6 by 10% corresponds to an approximate increase in $\ln(x)$ of about 10%.

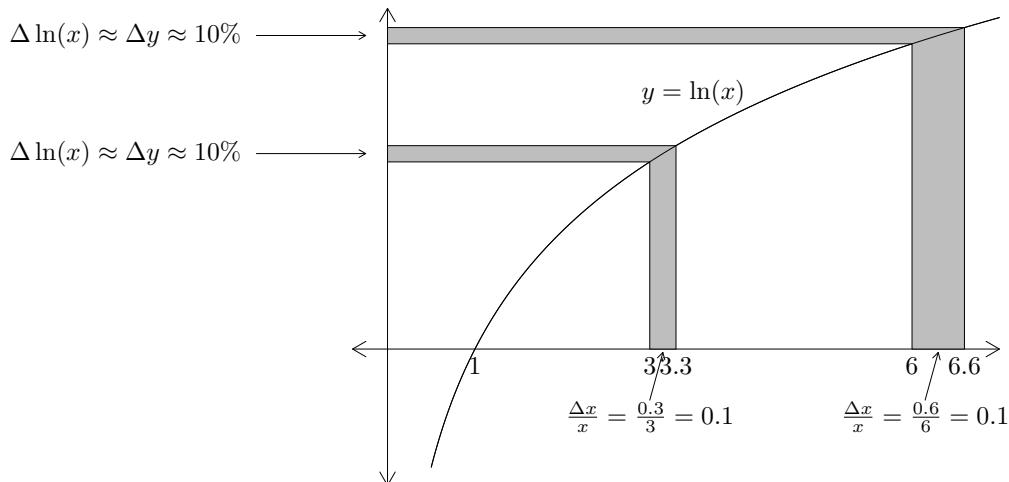


FIGURE 12.26: Small change in x gives a similar small change in $\ln(x)$

An example from economics that has multiplicative error terms is the demand function ($Q = \alpha P^\beta \varepsilon$), where Q = quantity demanded, P = price, α and β are unknown parameters, and ε is the error term. This function is often transformed by taking the natural logarithm of both sides. That is,

$$\ln(Q) = \ln(\alpha) + \beta \ln(P) + \ln(\varepsilon), \quad (12.73)$$

which is in the form of a simple linear model ($Y = \beta_0 + \beta_1 x + \varepsilon$). Note that the errors in a simple linear model are additive.

The parameter β in (12.73) can be interpreted as the percent change in Q over the percent change in P , which is the definition of **price elasticity**. In other words, $|\beta|$ = price elasticity. When dealing with a simple linear model of the form

$$\ln(Y) = \ln(\beta_0) + \beta_1 \ln(x) + \varepsilon, \quad (12.74)$$

β_1 can be interpreted as

$$\beta_1 = \frac{\% \Delta Y}{\% \Delta x}. \quad (12.75)$$

In Example 2.33 on page 147, the data frame **Animals** from the **MASS** package was used to find the least squares line for regressing `log(brain)` on `log(body)` once the three dinosaurs

were removed from the data. (Note that `log(x)` in R is the natural logarithm function, $\ln(x)$.) The resulting least squares estimates of β_0 and β_1 after the dinosaurs are removed are computed in R Code 12.69.

R Code 12.69

```
> library(MASS)
> SA <- Animals[order(Animals$body), ] # sorted by body
> NoDINO <- SA[-c(28:26), ] # remove dinosaurs
> simple.model <- lm(log(brain) ~ log(body), data = NoDINO)
> coef(simple.model)

(Intercept)    log(body)
2.1504121    0.7522607
```

If the body weight of an animal increases by 1%, the approximate increase in brain weight is $(0.01 \times 0.7523 = 0.0075 = 0.7523\%)$ since $\hat{\beta}_1 = 0.7523$. The predicted brain weight of the jaguar, whose weight is listed as 100 kg with the fitted model, is $2.1504 + 0.7523 \times \ln(100) = 5.6147$. This must be back transformed to get the units of original brain measurement (grams). The brain weight predicted by the model is $\exp(5.6147) = 274.4312$ g. If said jaguar were to increase its weight by 10%, the expected increase in brain weight would be approximately 7.5% for a new weight of $1.075 \times 274.4312 = 295.0136$ g. The actual brain weight change predicted by the model for a body weight of 110 kg is 294.83 g and the change in brain weight as predicted from the model is 7.4331% (see Table 12.7). Note that for this model, $\hat{\beta}_1 = 0.7523 \approx \% \Delta Y / \% \Delta x = 0.0743 / 0.1 = 0.7433$. In fact, when both the response and the predictors have been transformed with a natural logarithm, one can use the percentage interpretation of β_1 as in (12.75) and be very close to the actual change given by the model for small changes in the x -variables.

Table 12.7: Actual change in jaguar brain weight

	x	$\ln(x)$	$\ln(Y)$	Y
	100.0	4.6052	5.6147	274.4312
	110.0	4.7005	5.6864	294.8300
$\% \Delta$	0.1			0.0743

The parameters of growth models of the form $P(t) = ce^{\beta t}$ are often estimated with ordinary least squares regression after taking the natural logarithms of both sides since $\ln P(t) = \ln(c) + \beta t$ is the form of a simple linear model. When the slope, β , is estimated for such a model, it provides an estimate of the approximate growth rate in units of t . More generally, for models of the form $\ln Y = \beta_0 + \beta_1 x$, for each unit of increase in x , Y increases roughly by $\beta_1 \times 100\%$.

12.14 Qualitative Predictors

Up to this point, only quantitative (continuous) predictor variables have been used in regression models. Quantitative variables take on values on a well-defined scale. Examples include height, weight, income, and age, to name a few; however, many predictor variables are qualitative. For example, gender (male/female) or race (Caucasian, Hispanic, Asian, etc.) are qualitative variables that appear in many regression models. Regression using quantitative variables can be generalized to qualitative variables with the use of dummy variables. A **dummy variable** is any variable in a regression model that takes on a finite number of values so that different categories of a nominal variable can be identified. Provided the regression model has an intercept, one must define $k - 1$ dummy variables to define a qualitative variable with k categories. There are many ways to define the $k - 1$ dummy variables. R uses treatment contrasts by default to define qualitative variables (factors). To see the values R uses to define a qualitative variable with four levels, enter

```
> contr.treatment(4)

 2 3 4
1 0 0 0
2 1 0 0
3 0 1 0
4 0 0 1
```

The rows of this matrix (4×3) are the levels of the qualitative predictor and the columns are the dummy variables. R assigns levels to a qualitative variable in alphabetical order by default.

Example 12.26 ▷ *Ease Levels Dummy Variables* ◷ Consider the variable `ease` from the `EPIDURAL` data frame. Define appropriate dummy variables to specify the three levels of this variable.

Solution: The three levels of `ease` (Difficult, Easy, and Impossible) require two dummy variables to be able to identify all three levels of `ease`:

```
> contrasts(EPIDURAL$ease)

      Difficult Impossible
Easy            0          0
Difficult       1          0
Impossible      0          1
```

Note that the first level in alphabetical order is `Difficult`. To change the first level of `ease` to `Easy`, enter

```
> EPIDURAL$ease <- factor(EPIDURAL$ease,
+                           levels = c("Easy", "Difficult", "Impossible"))
> levels(EPIDURAL$ease)
[1] "Easy"      "Difficult"   "Impossible"
> contrasts(EPIDURAL$ease)
```

	Difficult	Impossible
Easy	0	0
Difficult	1	0
Impossible	0	1



The simplest situation where dummy variables might be used in a regression model is when the qualitative predictor has only two levels. The regression model for a single quantitative predictor (x_1) and a dummy variable (D_1) is written

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 D_1 + \beta_3 x_1 D_1 + \varepsilon \quad (12.76)$$

where

$$D_1 = \begin{cases} 0 & \text{for the first level} \\ 1 & \text{for the second level.} \end{cases}$$

The model in (12.76) when D_1 has two levels will yield one of four possible scenarios, as shown in Figure 12.27 on the next page. This type of model requires the user to answer **three basic questions**:

- (1) Are the lines the same?
- (2) Are the slopes the same?
- (3) Are the intercepts the same?

To address basic question (1), the null hypothesis $H_0 : \beta_2 = \beta_3 = 0$ must be tested. One way to perform the test is to use the general linear test statistic based on the full model found in (12.76) and the reduced model $Y = \beta_0 + \beta_1 x_1 + \varepsilon$. If the null hypothesis is not rejected, the interpretation is that there is one line present (the intercept and the slope are the same for both levels of the dummy variable). This is the case for graph I of Figure 12.27 on the facing page. If the null hypothesis is rejected, either the slopes, the intercepts, or possibly both the slope and the intercept are different for the different levels of the dummy variable, as seen in graphs II, III, and IV of Figure 12.27, respectively.

To answer basic question (2), the null hypothesis $H_0 : \beta_3 = 0$ must be tested. If the null hypothesis is not rejected, the two lines have the same slope, but different intercepts, as shown in graph II of Figure 12.27 on the next page. The two parallel lines that result when $\beta_3 = 0$ are

$$Y = \beta_0 + \beta_1 x_1 + \varepsilon \text{ for } (D_1 = 0) \quad \text{and} \quad Y = (\beta_0 + \beta_2) + \beta_1 x_1 + \varepsilon \text{ for } (D_1 = 1).$$

When $H_0 : \beta_3 = 0$ is rejected, one concludes that the two fitted lines are not parallel as in graphs III and IV of Figure 12.27 on the facing page.

To answer basic question (3), the null hypothesis $H_0 : \beta_2 = 0$ for model (12.76) must be tested. The reduced model for this test is $Y = \beta_0 + \beta_1 x_1 + \beta_3 x_1 D_1 + \varepsilon$. If the null hypothesis is not rejected, the two fitted lines have the same intercept but different slopes:

$$Y = \beta_0 + \beta_1 x_1 + \varepsilon \text{ for } (D_1 = 0) \quad \text{and} \quad Y = \beta_0 + (\beta_1 + \beta_3)x_1 + \varepsilon \text{ for } (D_1 = 1).$$

Graph III of Figure 12.27 on the next page represents this situation. If the null hypothesis is rejected, one concludes that the two lines have different intercepts, as in graphs II and IV of Figure 12.27.

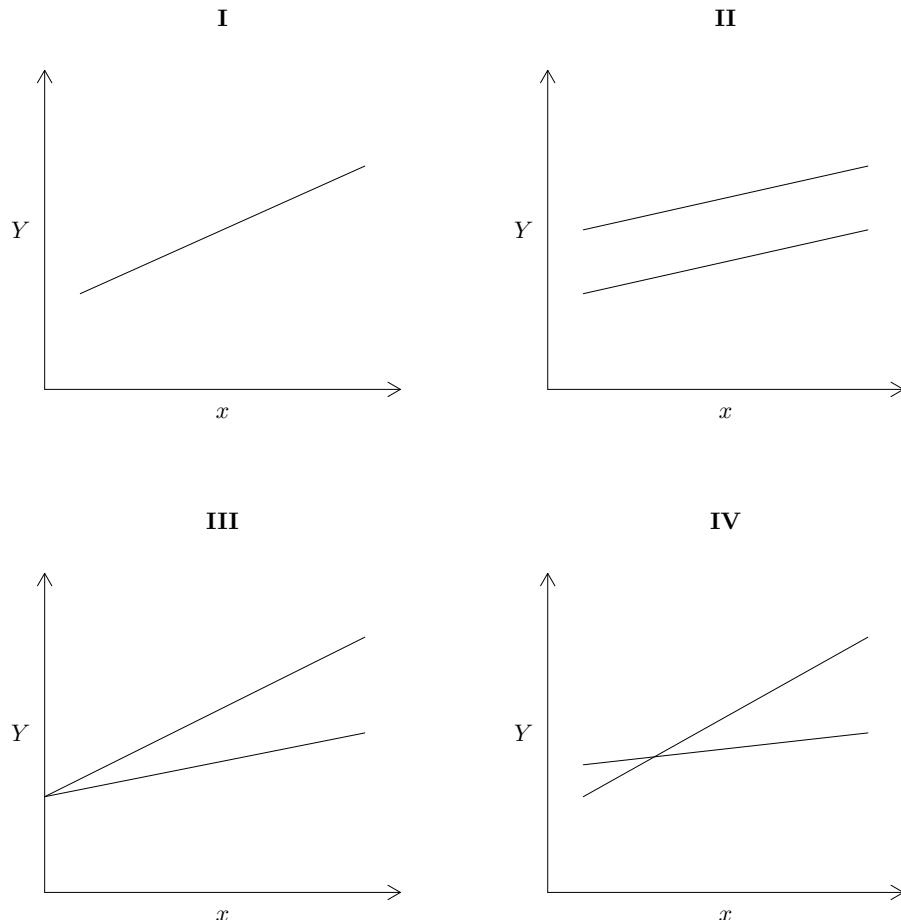


FIGURE 12.27: Four possible results for a single dummy variable with two levels. Graph I has the intercept and the slope the same for both levels of the dummy variable. Graph II has the two lines with the same slope, but different intercepts. Graph III shows the two fitted lines with the same intercept but different slopes. Graph IV shows the two lines with different intercepts and different slopes.

Example 12.27 \triangleright *Elevators* \triangleleft Suppose a realtor wants to model the appraised price of an apartment as a function of the predictors living area (in m^2) and the presence or absence of elevators. Consider the data frame **VIT2005**, which contains data about apartments in Vitoria, Spain, including **totalprice**, **area**, and **elevator**, which are the appraised apartment value in Euros, living space in square meters, and the absence or presence of at least one elevator in the building, respectively.

- The realtor first wants to know if there is any relationship between appraised price (Y) and living area (x_1).
- Next, the realtor wants to know how adding a dummy variable for whether or not an elevator is present changes the relationship:
 - Are the lines the same?
 - Are the slopes the same?

(iii) Are the intercepts the same?

Solution: (a) A linear regression model of the form

$$Y = \beta_0 + \beta_1 x_1 + \varepsilon \quad (12.77)$$

is fit yielding

$$\hat{Y}_i = 40822.4159 + 2704.751x_{i1}$$

and a scatterplot of `totalprice` versus `area` with the fitted regression line superimposed over the scatterplot is shown in Figure 12.28 on the facing page. The levels of the factor `elevator` are labeled `No` and `Yes` in R Code 12.70 and `totalprice` is regressed on `area` in R Code 12.71.

R Code 12.70

```
> VIT2005$elevator <- factor(VIT2005$elevator, labels = c("No", "Yes"))
> contrasts(VIT2005$elevator)

      Yes
No      0
Yes     1
```

R Code 12.71

```
> mod.simple <- lm(totalprice ~ area, data = VIT2005)
> summary(mod.simple)

Call:
lm(formula = totalprice ~ area, data = VIT2005)

Residuals:
    Min      1Q  Median      3Q      Max 
-156126 -21564 -2155  19493  120674 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 40822.4     12170.1   3.354  0.00094 ***
area        2704.8      133.6   20.243 < 2e-16 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 40810 on 216 degrees of freedom
Multiple R-squared:  0.6548, Adjusted R-squared:  0.6532 
F-statistic: 409.8 on 1 and 216 DF,  p-value: < 2.2e-16
```

Based on Figure 12.28 on the next page created from R Code 12.72 on the facing page, there appears to be a linear relationship between appraised price and living area. Further, this relationship is statistically significant, as the p -value for testing $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$ is less than 2.0×10^{-16} .

R Code 12.72

```
> p <- ggplot(data = VIT2005,
+               aes(x = area, y = totalprice, color = elevator))
> p + geom_point() +
+   scale_color_grey(start = 0.2, end = 0.8) +
+   theme_bw() +
+   labs(x = "\nLiving Area in Square Meters",
+        y = "Appraised Price in Euros\n") +
+   geom_abline(intercept = coef(mod.simple)[1],
+               slope = coef(mod.simple)[2], colour = "blue")
```

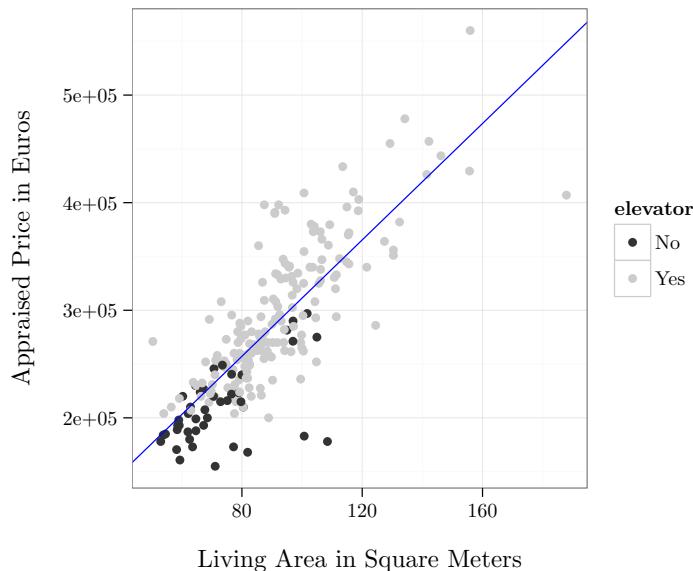


FIGURE 12.28: Scatterplot of `totalprice` versus `area` with the fitted regression line superimposed from `mod.simple`

(b) The regression model including the dummy variable for `Elevator` is written

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 D_1 + \beta_3 x_1 D_1 + \varepsilon \quad (12.78)$$

where

$$D_1 = \begin{cases} 0 & \text{when a building has no elevators} \\ 1 & \text{when a building has at least one elevator.} \end{cases}$$

(i) To determine if the lines are the same (which means that the linear relationship between appraised price and living area is the same for apartments with and without elevators), the hypotheses are

$$H_0 : \beta_2 = \beta_3 = 0 \text{ versus } H_1 : \text{at least one } \beta_i \neq 0 \text{ for } i = 2, 3.$$

The hypotheses are tested in R Code 12.73 on the next page.

R Code 12.73

```
> mod.total <- lm(totalprice ~ area + elevator + area:elevator,
+       data = VIT2005)
> mod.simple <- lm(totalprice ~ area, data = VIT2005)
> anova(mod.simple, mod.total) # compare models

Analysis of Variance Table

Model 1: totalprice ~ area
Model 2: totalprice ~ area + elevator + area:elevator
  Res.Df   RSS Df Sum of Sq    F    Pr(>F)
1     216 3.5970e+11
2     214 3.0267e+11  2 5.704e+10 20.165 9.478e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In this problem, one may conclude that at least one of β_2 and β_3 is not zero since the p -value = 0. In other words, the lines have either different intercepts, different slopes, or different intercepts and slopes.

(ii) To see if the lines have the same slopes (which means that the presence of an elevator adds constant value over all possible living areas), the hypotheses are

$$H_0 : \beta_3 = 0 \text{ versus } H_1 : \beta_3 \neq 0.$$

The hypotheses are tested in R Code 12.74.

R Code 12.74

```
> anova(mod.total) # ANOVA

Analysis of Variance Table

Response: totalprice
  Df   Sum Sq Mean Sq F value    Pr(>F)
area      1 6.8239e+11 6.8239e+11 482.4846 < 2.2e-16 ***
elevator  1 4.5308e+10 4.5308e+10 32.0352 4.83e-08 ***
area:elevator 1 1.1732e+10 1.1732e+10  8.2949  0.00438 **
Residuals 214 3.0267e+11 1.4143e+09
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the p -value = 0.0044, it may be concluded that $\beta_3 \neq 0$, which implies that the lines are not parallel.

(iii) To test for equal intercepts (which means that appraised price with and without elevators starts at the same value), the hypotheses to be evaluated are

$$H_0 : \beta_2 = 0 \text{ versus } H_1 : \beta_2 \neq 0.$$

R Code 12.75

```
> mod.total <- lm(totalprice ~ area + elevator + area:elevator,
+   data = VIT2005)
> mod.inter <- lm(totalprice ~ area + area:elevator, data = VIT2005)
> anova(mod.inter, mod.total) # compare models
```

Analysis of Variance Table

	Model 1: totalprice ~ area + area:elevator	Model 2: totalprice ~ area + elevator + area:elevator			
Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	215	3.0624e+11			
2	214	3.0267e+11	1	3576497188	2.5288 0.1133

Since the p -value for testing the null hypothesis is 0.1133, one fails to reject H_0 and should conclude that the two lines have the same intercept but different slopes.

```
> coef(summary(mod.inter))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	71352.0844	12309.17938	5.796656	2.389680e-08
area	1897.9368	180.59084	10.509596	4.082287e-21
area:elevatorYes	553.9856	90.42399	6.126534	4.227047e-09

The fitted model is $\hat{Y}_i = 71352.0844 + 1897.9368x_{i1} + 553.9856x_{i1}D_{i1}$, and the fitted regression lines for the two values of D_1 are shown in Figure 12.29. The fitted model using the same intercept with different slopes has an R^2_a of 0.7034, a modest improvement over the model without the variable `elevator`, which had an R^2_a value of 0.6532.

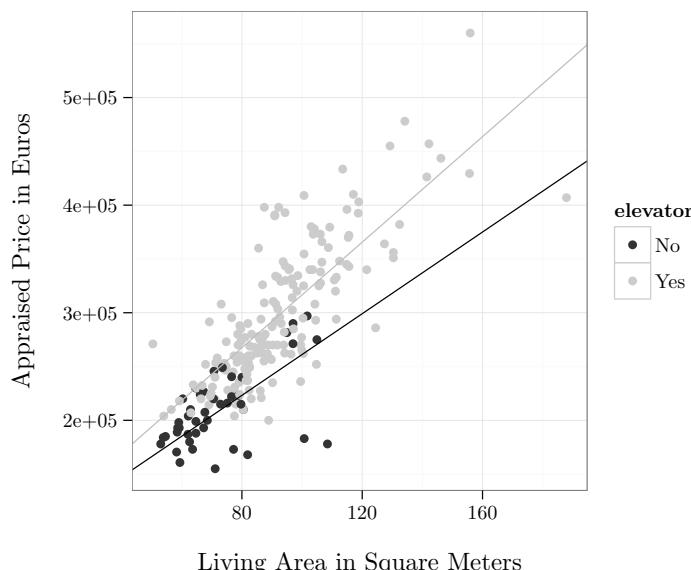


FIGURE 12.29: Fitted regression lines for `mod.inter`

As the numbers of levels in the qualitative variables increases, the number of dummy variables required to represent all of the possible combinations of variables (both dummy and numerical) increases rapidly, and the comparison of regression equations becomes virtually intractable. Further exploration of this topic could be carried out with a book dedicated to regression.

12.15 Estimation of the Mean Response for New Values \mathbf{X}_h

Not only is it desirable to create confidence intervals on the parameters of the regression models, but it is also common to estimate the mean response ($E(Y_h)$) for a particular set of \mathbf{X} values. The particular values where an estimate is desired will be denoted $\mathbf{X}_h = [1, x_{h,1}, x_{h,2}, \dots, x_{h,p-1}]$. Since $\widehat{\mathbf{Y}} = \mathbf{X}\hat{\beta}$, it follows that $\widehat{Y}_h = \mathbf{X}_h\hat{\beta}$. For the normal error model ($\varepsilon \sim N(\mathbf{0}, \sigma^2 \mathbf{I})$),

$$\widehat{Y}_h \sim N(Y_h = X_h\beta, \sigma^2 \mathbf{X}_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'_h). \quad (12.79)$$

Recall (12.32) states that $s_{\hat{\beta}}^2 = \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1} = MSE(\mathbf{X}'\mathbf{X})^{-1}$, while (12.79) gives $\sigma_{\widehat{Y}_h}^2 = \sigma^2 \mathbf{X}_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'_h$, from which it follows that

$$s_{\widehat{Y}_h}^2 = MSE \cdot \mathbf{X}_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'_h = \mathbf{X}_h s_{\hat{\beta}}^2 \mathbf{X}'_h. \quad (12.80)$$

Consequently, for a vector of given values (\mathbf{X}_h), a $(1 - \alpha) \cdot 100\%$ confidence interval for the mean response $E(Y_h)$ is

$$CI_{1-\alpha}[E(Y_h)] = [\widehat{Y}_h - t_{1-\alpha/2; n-p} \cdot s_{\widehat{Y}_h}, \widehat{Y}_h + t_{1-\alpha/2; n-p} \cdot s_{\widehat{Y}_h}]. \quad (12.81)$$

The function `predict()` applied to a linear model object will compute \widehat{Y}_h and $s_{\widehat{Y}_h}$ for a given \mathbf{X}_h . R output has \widehat{Y}_h labeled `fit` and $s_{\widehat{Y}_h}$ labeled `se.fit`. The function `predict()` can be used for a wide range of applications where the statistician would like to predict values of new data. One of `predict()`'s arguments is `newdata=`, where what follows the `=` should be a data frame whose columns have identical names to those of the variables that were used in constructing the original model.

12.16 Prediction and Sampling Distribution of New Observations $Y_{h(\text{new})}$

In Section 12.15, a confidence interval was found for the mean response, $E(Y_h)$. In contrast, it is not unusual to require a confidence interval on a single, new observation instead. For example, suppose a linear model that describes course grade as a function of time studied is calculated. As the user of this model, you might be interested in your predicted grade given the amount of time you study rather than the average grade that is received by all people who study for the same amount of time you do. Although the point estimates for the average grade given time studied ($E(Y_h)$) and your grade given time studied ($Y_{h(\text{new})}$) are identical, the confidence intervals for these two quantities are not

the same because $s_{\hat{Y}_{h(\text{new})}}^2$ accounts for an additional source of variability not present in $s_{\hat{Y}_h}^2$. Specifically, $s_{\hat{Y}_{h(\text{new})}}^2$ estimates the variance of the distribution of \hat{Y}_h at $\mathbf{X} = \mathbf{X}_h$, which has a value of σ^2 with MSE as well as the variance of the sampling distribution of \hat{Y}_h with $s_{\hat{Y}_h}^2$.

For the normal error model,

$$\hat{Y}_{h(\text{new})} \sim N\left(Y_h = \mathbf{X}_h\beta, \sigma^2(1 + \mathbf{X}_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'_h)\right). \quad (12.82)$$

It follows that $s_{\hat{Y}_{h(\text{new})}}^2 = MSE(1 + \mathbf{X}_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'_h)$ and the $(1 - \alpha) \cdot 100\%$ prediction interval for new observation $\hat{Y}_{h(\text{new})}$ is written as

$$PI_{1-\alpha}[\hat{Y}_{h(\text{new})}] = \left[\hat{Y}_h - t_{1-\alpha/2; n-p} \cdot s_{\hat{Y}_{h(\text{new})}}, \hat{Y}_h + t_{1-\alpha/2; n-p} \cdot s_{\hat{Y}_{h(\text{new})}}\right]. \quad (12.83)$$

To compute prediction intervals, the function `predict()` may be applied to a linear model using the argument `interval = "pred"`.

Example 12.28 Use the `GRADES` data set and model `gpa` as a function of `sat` assuming that the requirements for model (12.4) are satisfied.

- (a) Compute the expected GPA (`gpa`) for an SAT score (`sat`) of 1300.
- (b) Construct a 90% confidence interval for the mean GPA for students scoring 1300 on the SAT.
- (c) Find the prediction limits on GPA for a future student who scores 1300 on the SAT.

Solution: R Code 12.76 creates the linear model object `mod.lm` from regressing `gpa` on `sat`.

R Code 12.76

```
> mod.lm <- lm(gpa ~ sat, data = GRADES)
> betahat <- coef(mod.lm)
> betahat

(Intercept)          sat
-1.19206381  0.00309427
```

- (a) The expected GPA for an SAT score of 1300 is $\hat{Y}_h = \mathbf{X}_h \cdot \hat{\beta} = 2.8305$, where $\mathbf{X}_h = (1, 1300)$ and $\hat{\beta} = [-1.1921, 0.0031]'$

```
> Xh <- matrix(c(1, 1300), nrow = 1)
> Yhath <- Xh %*% betahat
> Yhath

[,1]
[1,] 2.830488
```

A method that requires less typing is

```
> predict(mod.lm, newdata=data.frame(sat = 1300))

1
2.830488
```

(b) A 90% confidence interval for the mean gpa for students scoring 1300 on the SAT using (12.81) is $CI_{0.90}(E(Y_h)) = [2.7598, 2.9012]$. R Code 12.77 computes $s_{\hat{Y}_h}^2$ using (12.80), and the confidence interval using (12.81).

R Code 12.77

```
> MSE <- anova(mod.lm)[2, 3]
> MSE

[1] 0.1595551

> XTXI <- summary(mod.lm)$cov.unscaled
> XTXI

(Intercept)          sat
(Intercept) 0.310137964 -2.689270e-04
sat         -0.000268927  2.370131e-07

> var.cov.b <- MSE*XTXI
> var.cov.b

(Intercept)          sat
(Intercept) 4.948408e-02 -4.290866e-05
sat         -4.290866e-05  3.781665e-08

> s2yhath <- Xh%*%var.cov.b%*%t(Xh)
> s2yhath

[,1]
[1,] 0.001831706

> syhath <- sqrt(s2yhath)
> syhath

[,1]
[1,] 0.04279843

> crit.t <- qt(0.95, 198)
> CI.EYh <- Yhath + c(-1, 1)*crit.t*syhath
> CI.EYh

[1] 2.759760 2.901216
```

The function `predict()` may also be used to compute the requested interval.

```
> predict(mod.lm, newdata = data.frame(sat = 1300),
+         interval = "conf", level = 0.90)

    fit      lwr      upr
1 2.830488 2.75976 2.901216
```

(c) The prediction limits on GPA for a future student who scores 1300 on the SAT are $PI_{0.90} = [2.1666, 3.4944]$ using (12.83).

```

> s2yhathnew <- MSE + s2yhath
> syhathnew <- sqrt(s2yhathnew)
> syhathnew

      [,1]
[1,] 0.4017297

> PI <- Yhath + c(-1, 1)*crit.t*syhathnew
> PI

[1] 2.166595 3.494380

```

Using the `predict()` function with the argument `interval = "pred"` also returns the requested prediction limits.

```

> PI <- predict(mod.lm, newdata = data.frame(sat = 1300),
+                 interval = "pred", level = 0.90)
> PI

    fit      lwr      upr
1 2.830488 2.166595 3.49438

```

12.17 Simultaneous Confidence Intervals

Now that a determination has been made of a correct $(1 - \alpha) \cdot 100\%$ confidence interval for a single β_k , confidence intervals for multiple β_k s are desired such that the significance level of all the intervals together will be only a specified α . For example, if $\alpha = 5\%$ and two independent confidence intervals were created for a β_0 and a β_1 , the probability that both would contain their parameters would be only $(0.95)^2 = 0.9025$, giving a family $\alpha = 0.0975$. The goal is to create intervals such that the family α is a given value. This goal is more difficult than the example because the same data are used to construct all the confidence intervals, so they are not independent and the α calculation is not straightforward. A **family confidence coefficient** is the proportion of confidence intervals that contain all the β_k parameters specified for the entire family of $g \leq p$ parameters for a given sample.

One approach to calculating these simultaneous confidence intervals is named the **Bonferroni** method. In this method, the joint interval estimates for $\beta_k, k = 0, \dots, g$ parameters are

$$CI_{1-\alpha}(\beta_k) = \left[\hat{\beta}_k - t_{1-\frac{\alpha}{2g}; n-p} \cdot s_{\hat{\beta}_k}, \hat{\beta}_k + t_{1-\frac{\alpha}{2g}; n-p} \cdot s_{\hat{\beta}_k} \right]. \quad (12.84)$$

A second approach is to construct a simultaneous confidence region for the β_k coefficients. Any set of parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_g)$ that satisfy the inequality

$$\frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})'(\mathbf{X}'\mathbf{X})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{q \cdot MSE} \leq f_{1-\alpha; q, n-p} \quad (12.85)$$

fall inside a $(1 - \alpha) \cdot 100\%$ ellipsoidal confidence region for $\boldsymbol{\beta}$ where MSE is that of the full model. Note that q is the rank of \mathbf{K} for the hypothesis $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{m}$ discussed in Section 12.10. When the simultaneous confidence limits are for $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_k)$,

excluding β_0 , q will be equal to $p - 1$, the number of predictors in the full model. This is a rather computationally intensive method. The simultaneous Scheffé confidence limits for the individual β_k s based on (12.85) are

$$CI_{1-\alpha}(\beta_k) = \left[\hat{\beta}_k - \sqrt{q \cdot f_{1-\alpha; q, n-p}} \cdot s_{\hat{\beta}_k}, \hat{\beta}_k + \sqrt{q \cdot f_{1-\alpha; q, n-p}} \cdot s_{\hat{\beta}_k} \right]. \quad (12.86)$$

For simple linear regression, the function `confidence.ellipse()` in the `car` package will construct and display a simultaneous confidence region for β_0 and β_1 ($q = 2$). For models with $p > 2$, `confidence.ellipse()` will draw a simultaneous confidence region for any two β_k s, $k = 1, \dots, p - 1$, specified by the user. Note that in the $p > 2$ case, q will equal $p - 1$. The R function `confint()` will compute individual confidence intervals for one or more parameters in a fitted model. The Bonferroni intervals from (12.84) will be wider than those from (12.86) whenever $t_{1-\frac{\alpha}{2g}; n-p} > \sqrt{q \cdot f_{1-\alpha; q, n-p}}$.

12.17.1 Simultaneous Confidence Intervals for Several Mean Responses — Confidence Band

To construct several confidence intervals for the mean response, $E(Y_h)$, corresponding to different \mathbf{X}_h vectors such that the family confidence coefficient is $1 - \alpha$, use (12.87), where $s_{\hat{Y}_h} = \sqrt{\mathbf{X}_h \mathbf{s}_{\hat{\beta}}^2 \mathbf{X}_h'}$:

$$CI_{1-\alpha}[E(Y_h)] = \left[\hat{Y}_h - \sqrt{p \cdot f_{1-\alpha; p, n-p}} \cdot s_{\hat{Y}_h}, \hat{Y}_h + \sqrt{p \cdot f_{1-\alpha; p, n-p}} \cdot s_{\hat{Y}_h} \right] \quad (12.87)$$

or

$$CI_{1-\alpha}[E(Y_h)] = \left[\hat{Y}_h - t_{1-\frac{\alpha}{2g}; n-p} \cdot s_{\hat{Y}_h}, \hat{Y}_h + t_{1-\frac{\alpha}{2g}; n-p} \cdot s_{\hat{Y}_h} \right], \quad (12.88)$$

whichever produces narrower intervals.

A **confidence band** is a region of confidence around the entire regression line constructed by plotting the upper and lower values of (12.87) over the range of \mathbf{X}_h and subsequently connecting all of the upper values with a curve and all of the lower values with a curve. See Figure 12.30 on page 886 for an example.

12.17.2 Predictions of g New Observations

To create simultaneous prediction intervals for g new observations, corresponding to g \mathbf{X}_h vectors with a family confidence coefficient of $1 - \alpha$, use

$$PI_{1-\alpha}[Y_{h(\text{new})}] = \left[\hat{Y}_h - \sqrt{g \cdot f_{1-\alpha; g, n-p}} \cdot s_{\hat{Y}_{h(\text{new})}}, \hat{Y}_h + \sqrt{g \cdot f_{1-\alpha; g, n-p}} \cdot s_{\hat{Y}_{h(\text{new})}} \right] \quad (12.89)$$

or

$$\left[\hat{Y}_h - t_{1-\alpha/2g; n-p} \cdot s_{\hat{Y}_{h(\text{new})}}, \hat{Y}_h + t_{1-\alpha/2g; n-p} \cdot s_{\hat{Y}_{h(\text{new})}} \right], \quad (12.90)$$

whichever produces narrower intervals.

12.17.3 Distinguishing Pointwise Confidence Envelopes from Confidence Bands

There is a distinction between connected intervals with $(1 - \alpha) \cdot 100\%$ confidence at each single point and an entire band with $(1 - \alpha) \cdot 100\%$ confidence of containing the regression

line. In this text, when each single interval has $(1 - \alpha) \cdot 100\%$ confidence of containing a mean response ($E(Y_h)$), and the upper and lower endpoints of the intervals are connected over the range of possible \mathbf{X}_h values, a **pointwise confidence envelope** is created. If the confidence for containing the entire regression line ($E(Y_h|\mathbf{X}_h)$) is $(1 - \alpha) \cdot 100\%$, a **confidence band** is being calculated. A confidence band is constructed by plotting the upper and lower values of (12.87) over the range of \mathbf{X}_h and subsequently connecting all of the upper values with a curve and all of the lower values with a curve from (12.87). A graphical representation of 90% pointwise confidence intervals, 90% confidence bands, and 90% pointwise prediction intervals for the regression of `gpa` on `sat` using the **GRADES** data frame from the **PASWR2** package is shown in Figure 12.30 on the next page. R Code 12.78 was used to create Figure 12.30 on the next page.

R Code 12.78

```
> pred.frame <- data.frame(sat = seq(700, 1600, 5))
> CE <- predict(mod.lm, interval = "conf", newdata = pred.frame,
+                 level = 0.90)
> PE <- predict(mod.lm, interval = "pred", newdata = pred.frame,
+                 level = 0.90)
> plot(gpa ~ sat, data = GRADES)
> matlines(pred.frame$sat, CE, lty = c(1, 3, 3), col = "black")
> matlines(pred.frame$sat, PE, lty = c(1, 2, 2), col = "black")
> syhath <- predict(mod.lm, interval = "conf", level = 0.90,
+                     newdata = pred.frame, se = TRUE)$se.fit
> Yhath <- CE[, "fit"]
> CV <- sqrt(2*qf(0.95, 2, 198))
> LL <- Yhath - CV*syhath
> UL <- Yhath + CV*syhath
> CB <- cbind(LL, UL)
> matlines(pred.frame$sat, CB, lty = c(1, 1), col = "gray")
```

Example 12.29 Use the data **HSWRESTLER** and the linear model in (12.6) with `hwfat` as the response and `age`, `abs`, and `triceps` as the predictors, assuming the errors from this model are normally distributed with mean zero and constant variance σ^2 .

- Obtain joint interval estimates for β_1 , β_2 , and β_3 using a 90% family confidence coefficient with both the Bonferroni and Scheffé approaches.
- Use the function `confidenceEllipse()` from the package `car` to construct a 90% simultaneous confidence region for β_2 and β_3 . Use the function `abline()` to verify visually that the limits of the simultaneous confidence region drawn by `confidenceEllipse()` agree with the values found in part (a).
- Find 90% joint interval estimates for the mean `hwfat` of wrestlers with values of \mathbf{X}_{hi} given in Table 12.8.
- Find 90% joint prediction intervals for three new wrestlers with values of \mathbf{X}_{hi} given in Table 12.8.

Solution: The answers are as follows:

- The estimates of the $\hat{\beta}_k$ s, $s_{\hat{\beta}}$ s, and the Bonferroni critical value $t_{1-\alpha/2,g}$ are computed in R Code 12.79 on the following page and then used with (12.84) to compute three simul-

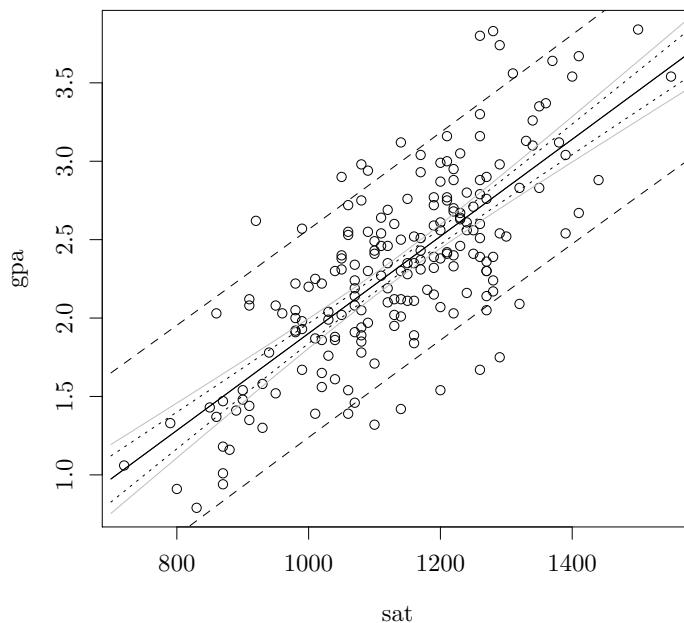


FIGURE 12.30: Representation of 90% pointwise confidence intervals (dotted lines), 90% prediction intervals (dashed lines), and a 90% confidence band (solid lines) for the regression of `gpa` on `sat` using the data in `GRADES`

Table 12.8: Values of \mathbf{X}_{hi} for `HSWRESTLER`

	age	abs	$triceps$
\mathbf{X}_{h1}	16	10	9
\mathbf{X}_{h2}	17	11	11
\mathbf{X}_{h3}	18	8	8

taneous confidence intervals. The Bonferroni simultaneous confidence intervals are

$$CI_{0.90}(\beta_1) = [-1.0984, 0.0322]$$

$$CI_{0.90}(\beta_2) = [0.2187, 0.4942]$$

$$CI_{0.90}(\beta_3) = [0.251, 0.6803]$$

R Code 12.79

```
> alpha <- 0.1
> mult.model <- lm(hwfat ~ age + abs + triceps, data = HSWRESTLER)
> coef(summary(mult.model))

            Estimate Std. Error    t value   Pr(>|t|)    
(Intercept) 10.6160623 4.23272425  2.508092 1.433001e-02
age        -0.5330948 0.26067474 -2.045057 4.440545e-02
abs         0.3564311 0.06353588  5.609918 3.323075e-07
triceps     0.4656071 0.09898493  4.703819 1.158514e-05
```

```

> b <- coef(summary(mult.model))[2:4, 1]
> s.b <- coef(summary(mult.model))[2:4, 2]
> g <- 3
> B <- qt((1 - alpha/(2 * g)), 78 - 4)
> B

[1] 2.168523

> BonSimCI.b <- matrix(c(b - B * s.b, b + B * s.b), ncol = 2)
> conf <- c("5%", "95%")
> bnam <- c("age", "abs", "triceps")
> dimnames(BonSimCI.b) <- list(bnam, conf)
> BonSimCI.b # Bonferroni simultaneous CIs

      5%      95%
age    -1.0983739 0.0321843
abs     0.2186521 0.4942101
triceps 0.2509561 0.6802582

```

The Scheffé critical value $\sqrt{pf_{1-\alpha; q, n-p}}$ is computed in R Code 12.80 and then used in (12.86) to compute three simultaneous confidence intervals. The Scheffé simultaneous confidence intervals are

$$\begin{aligned} CI_{0.90}(\beta_1) &= [-1.1966, 0.1304] \\ CI_{0.90}(\beta_2) &= [0.1947, 0.5181] \\ CI_{0.90}(\beta_3) &= [0.2137, 0.7175]. \end{aligned}$$

R Code 12.80

```

> Q <- 3
> S <- sqrt(Q*qf(0.9, Q, 78 - 4))
> S

[1] 2.545185

> SchSimCI.b <- matrix(c(b - S*s.b, b + S*s.b), ncol = 2)
> dimnames(SchSimCI.b) <- list(bnam, conf)
> SchSimCI.b

      5%      95%
age    -1.1965602 0.1303706
abs     0.1947205 0.5181416
triceps 0.2136722 0.7175421

```

(b) R Code 12.81 is used to create the left graph in Figure 12.31 on the next page, which depicts a joint confidence region for β_2 and β_3 enclosed by the Bonferroni confidence limits.

R Code 12.81

```

> confidenceEllipse(mult.model, level = 0.90, which.coef = c(3, 4),
+                   Scheffe = FALSE, main = "")
> title(main="Bonferroni Confidence Region")

```

```
> abline(v=BonSimCI.b[2, ])
> abline(h=BonSimCI.b[3, ])
```

In a similar fashion, the right graph of Figure 12.31 depicts a joint confidence region for β_2 and β_3 enclosed by the Scheffé confidence limits. R Code 12.82 was used to produce the right graph of Figure 12.31.

R Code 12.82

```
> confidenceEllipse(mult.model, level = 0.90, which.coef = c(3, 4),
+                     Scheffe = TRUE, main = "")
> title(main="Scheffe Confidence Region")
> abline(v=SchSimCI.b[2, ])
> abline(h=SchSimCI.b[3, ])
```

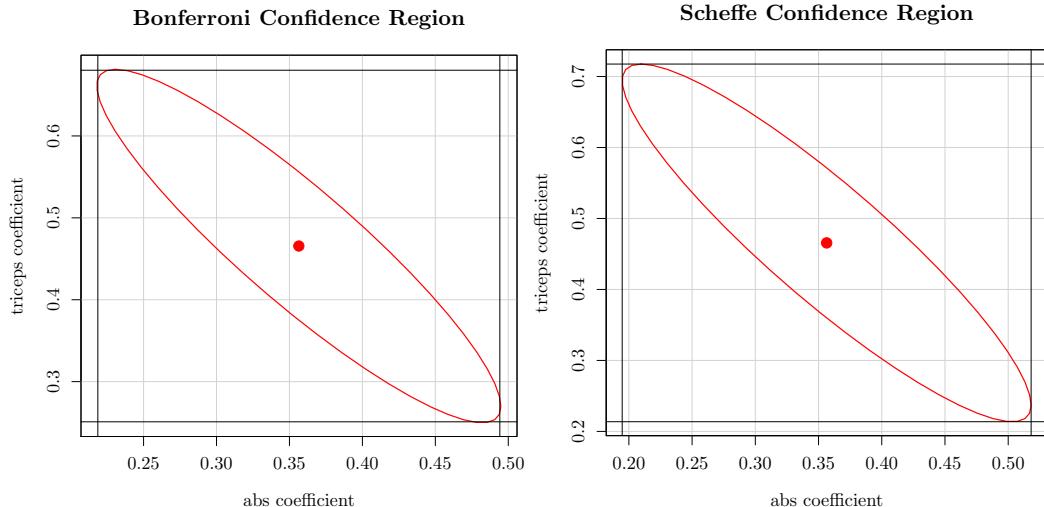


FIGURE 12.31: Joint confidence region for β_2 and β_3 enclosed by the Bonferroni (left graph) and Scheffé (right graph) confidence limits

(c) The 90% simultaneous confidence intervals for the mean `hwfat` of wrestlers with values of \mathbf{X}_{hi} given in Table 12.8 on page 886 using (12.88) since $t_{1-\frac{\alpha}{2g}; n-p} = 2.1685 < \sqrt{g \cdot f_{1-\alpha; g, n-p}} = 2.5452$ are

$$\begin{aligned} CI_{0.90}[E(Y_{h1})] &= [8.998, 10.6846] \\ CI_{0.90}[E(Y_{h2})] &= [9.4478, 11.744] \\ CI_{0.90}[E(Y_{h3})] &= [6.0479, 9.1454]. \end{aligned}$$

R Code 12.83 on the next page is used to compute the three simultaneous confidence intervals for the mean `hwfat` of wrestlers with values of \mathbf{X}_{hi} given in Table 12.8 on page 886.

R Code 12.83

```

> g <- 3
> alpha <- 0.10
> SC <- sqrt(g*qf(1 - alpha, 3, 74))
> TC <- qt(1 - alpha/(2*g), 74)
> RES <- predict(mult.model, newdata = data.frame(age = c(16, 17, 18),
+                                     abs = c(10, 11, 8), triceps =c(9, 11, 8)), se.fit = TRUE)
> Yhath <- RES$fit
> Syhath <- RES$se.fit
> ll <- Yhath - TC*Syhath
> ul <- Yhath + TC*Syhath
> BCI <- cbind(Yhath, Syhath, ll, ul)
> BCI

      Yhath     Syhath       ll       ul
1  9.841321 0.3888869 8.998010 10.684631
2 10.595871 0.5294386 9.447771 11.743971
3  7.596662 0.7141915 6.047921  9.145402

```

(d) The 90% joint prediction intervals for three new wrestlers with values of \mathbf{X}_{hi} given in Table 12.8 on page 886 using (12.83) since $t_{1-\alpha/2g; n-p} = 2.1685 < \sqrt{g f_{1-\alpha; g, n-p}} = 2.5452$ are

$$\begin{aligned} PI_{0.90}[Y_{h1(new)}] &= [3.2853, 16.3974] \\ PI_{0.90}[Y_{h2(new)}] &= [3.9937, 14.2802] \\ PI_{0.90}[Y_{h3(new)}] &= [0.9131, 14.2802]. \end{aligned}$$

R Code 12.84 is used to compute the three simultaneous prediction intervals for three new wrestlers with values of \mathbf{X}_{hi} given in Table 12.8 on page 886.

R Code 12.84

```

> g <- 3
> alpha <- 0.10
> SC <- sqrt(g*qf(1 - alpha, 3, 74))
> TC <- qt(1 - alpha/(2*g), 74)
> c(SC, TC)

[1] 2.545185 2.168523

> MSE <- anova(mult.model)[4, 3]
> MSE

[1] 8.989042

> s2yhathnew <- MSE + Syhath^2
> Syhathnew <- sqrt(s2yhathnew)
> ll <- Yhath - TC*Syhathnew
> ul <- Yhath + TC*Syhathnew
> SPI <- cbind(Yhath, Syhathnew, ll, ul)
> SPI

```

	Yhath	Syhathnew	11	ul
1	9.841321	3.023289	3.2852500	16.39739
2	10.595871	3.044560	3.9936729	17.19807
3	7.596662	3.082063	0.9131382	14.28019



12.18 Problems

1. The manager of a URL commercial address is interested in predicting the number of megabytes downloaded, `megasd`, by clients according to the number of minutes they are connected, `mconnected`. The manager randomly selects (megabyte, minute) pairs, records the data, and stores the pairs (`megasd`, `mconnected`) in the file **URLADDRESS**.
 - (a) Create a scatterplot of the data. Characterize the relationship between `megasd` and `mconnected`.
 - (b) Fit a regression line to the data. Superimpose the resulting line in the plot created in part (a).
 - (c) Compute the covariance matrix of the $\hat{\beta}$ s.
 - (d) What is the standard error of $\hat{\beta}_1$?
 - (e) What is the covariance between $\hat{\beta}_0$ and $\hat{\beta}_1$?
 - (f) Construct a 95% confidence interval for the slope of the regression line.
 - (g) Compute R^2 , R_a^2 , and the residual variance for the fitted regression.
 - (h) What assumptions need to be satisfied in order to use the model from part (b) for inferential purposes?
 - (i) Are there any outlying observations?
 - (j) Are there any influential observations? Compute and graph Cook's distances, DFFITS, and DFBETAS to answer this question. Create a bubble plot of studentized residuals versus leverage values with plotted points proportional to Cook's distance using the function `influencePlot()` from the `car` package. Does the bubble plot confirm your answer with respect to influential observations?
 - (k) Estimate the mean value of megabytes downloaded by clients spending 5, 10, and 15 minutes on line. Construct the corresponding 90% confidence intervals.
 - (l) Predict the megabytes downloaded by a client spending 30 minutes on line. Construct the corresponding 90% prediction interval.
2. A metallurgic company is investigating lost revenue due to worker illness. It is interested in creating a table of lost revenue to be used for future budgets and company forecasting plans. The data are stored in the data frame **LOSTR**.
 - (a) Create a scatterplot of lost revenue versus number of ill workers. Characterize the relationship between `lostrevenue` and `numbersick`.
 - (b) Fit a regression line to the data. Superimpose the resulting line in the plot created in part (a).
 - (c) Compute the covariance matrix of the $\hat{\beta}$ s.
 - (d) Create a 95% confidence interval for β_1 .

- (e) Compute the coefficient of determination and the adjusted coefficient of determination. Provide contextual interpretations of both values.
- (f) What assumptions need to be satisfied in order to use the model from part (b) for inferential purposes? If there is/are any outlier/s in the data, remove it/them prior to answering the remainder of the questions.
- (g) Determine the expected lost revenues when 5, 15, and 25 workers are absent due to illness.
- (h) Compute a 95% prediction interval of lost revenues when 14 workers are absent due to illness.

3. To obtain a linear relationship between the employment (number of employed people = dependent variable) and the GDP (gross domestic product = response variable), a researcher has taken data from 12 regions. Use the following information to answer the questions:

$$\sum_{i=1}^{12} x_i = 581 \quad \sum_{i=1}^{12} x_i^2 = 28507 \quad \sum_{i=1}^{12} x_i Y_i = 2630 \quad \sum_{i=1}^{12} Y_i = 53 \quad \sum_{i=1}^{12} Y_i^2 = 267$$

Source	df	SS	MS	F_{obs}	$p\text{-value}$
Regression	*	*	*	*	*
Error	*	22.08	*	*	*

- (a) Complete the ANOVA table.
- (b) Decide if the regression is statistically significant.
- (c) Compute and interpret the coefficient of determination.
- (d) Calculate the model's residual variance.
- (e) Write out the fitted regression line and construct a 90% confidence interval for the slope.

4. The speed of a tennis ball after being struck with a tennis racket depends on the length of the racket and the string tension. A multiple regression model is fit where Y is the speed of the struck tennis ball, x_1 is the length of the racket, and x_2 is the string tension, for 16 different tennis rackets. The following table displays the analysis of variance for the fitted regression model:

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x1	--	3797.0	--	--	--
x2	--	1331.3	--	--	--
Residuals	--	174.1	--	--	--

- (a) Complete the table.
- (b) Compute the regression sum of squares. Is the regression statistically significant?
- (c) Estimate the model's error variance.
- (d) Compute both R^2 and R_a^2 coefficients.

- (e) Given $\hat{\beta}_0 = -8.355$, $\hat{\beta}_1 = 3.243$, $\hat{\beta}_2 = -1.711$, $Var[\hat{\beta}_0] = 292.280$, $Var[\hat{\beta}_1] = 0.051$ and $Var[\hat{\beta}_2] = 0.029$, conduct the following tests of hypotheses and comment on the results:

$$\begin{array}{ll} H_0 : \beta_0 = 0 & H_0 : \beta_1 = 3 \\ H_1 : \beta_0 \neq 0, & H_1 : \beta_1 > 3, \end{array} \quad \begin{array}{ll} H_0 : \beta_2 = -1 & \\ H_1 : \beta_2 < -1. \end{array}$$

5. Given a simple linear regression model, show

(a) $\hat{\sigma}^2 = \frac{\sum_i \hat{\epsilon}_i^2}{n-2}$ is an unbiased estimator of σ^2 .

(b) The diagonal element of the hat matrix can be expressed as

$$h_{ii} = \mathbf{x}'_i (\mathbf{X}' \mathbf{X})^{-1} \mathbf{x}_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_i (x_i - \bar{x})^2},$$

where $\mathbf{x}'_i = (1, x_i)$.

6. Show that (12.63) and (12.64) are algebraically equivalent:

$$D_i = \frac{(\hat{\beta}_{(i)} - \hat{\beta})' (\mathbf{X}' \mathbf{X}) (\hat{\beta}_{(i)} - \hat{\beta})}{p \hat{\sigma}^2} = \frac{r_i^2}{p} \frac{h_{ii}}{1 - h_{ii}}.$$

Note: $r_i = \frac{\hat{\epsilon}_i}{\hat{\sigma} \sqrt{1 - h_{ii}}}$ and $h_{ii} = \mathbf{x}'_i (\mathbf{X}' \mathbf{X})^{-1} \mathbf{x}_i$.

HINT:

$$\left(\mathbf{X}'_{(i)} \mathbf{X}_{(i)} \right)^{-1} = (\mathbf{X}' \mathbf{X})^{-1} + \underbrace{(\mathbf{X}' \mathbf{X})^{-1} x_i x'_i (\mathbf{X}' \mathbf{X})^{-1}}_{1 - h_{ii}}. \quad (12.91)$$

7. Show that (12.65) and (12.66) are algebraically equivalent:

$$\text{DFFITS}_i = \frac{\hat{Y}_i - \hat{Y}_{i(i)}}{\hat{\sigma}_{(i)} \sqrt{h_{ii}}} = r_i^* \sqrt{\frac{h_{ii}}{1 - h_{ii}}}.$$

8. Show that the SSE in a linear model expressed in summation notation is equivalent to the SSE expressed in matrix notation:

$$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 = \mathbf{Y}' \mathbf{Y} - \hat{\beta}' \mathbf{X}' \mathbf{Y}.$$

9. Show that the SSR in a linear model expressed in summation notation is equivalent to the SSR expressed in matrix notation:

$$SSR = \sum_{i=1}^n (\hat{Y}_i - \bar{Y})^2 = \hat{\beta}' \mathbf{X}' \mathbf{Y} - \frac{1}{n} \mathbf{Y}' \mathbf{J} \mathbf{Y}.$$

10. Show that the trace of the hat matrix \mathbf{H} is equal to p , the number of parameters (β s), in a multiple linear regression model.

11. The data frame **HSWRESTLER** contains information on nine variables for a group of 78 high school wrestlers that was collected by the human performance lab at Appalachian State University. The variables are **age** (in years), **ht** (height in inches), **wt** (weight in pounds), **abs** (abdominal skinfold measure), **triceps** (tricep skinfold measure), **subscap** (subscapular skinfold measure), **hwfat** (hydrostatic determination of fat), **tanfat** (Tanita determination of fat), and **skfat** (skinfold determination of fat). Use **hwfat** (Y), **abs** (x_1), and **triceps** (x_2) to verify empirically the value obtained for $SSR(x_2, x_1)$ using quadratic forms.
12. The data frame **KINDER** contains the height in inches and weight in pounds of 20 children from a kindergarten class. Use all 20 observations and construct a regression model where the results are stored in the object **mod** by regressing weight on height.
- Create a scatterplot of weight versus height to verify a possible linear relationship between the two variables.
 - Compute and display the hat values for **mod** in a graph. Use the graph to identify the two largest hat values. Superimpose a horizontal line at $2p/n$. Remove the values that exceed $2p/n$ and regress weight on height, storing the results in an object named **modk**.
 - Remove case 19 from the original data frame **KINDER** and regress weight on height, storing the results in **modk19**. Is the child with the largest hat value an influential observation if one considers the 19 observations without case 19 from the original data frame? Compute and consider Cook's D_i , $DFFITS_i$, and $DFBETAS_{k(i)}$, in reaching a conclusion. Specifically, produce a graph showing h_{ii} , the differences in $\hat{\beta}_{1(i)} - \hat{\beta}_1$, $DFBETAS_{k(i)}$, studentized residuals, $DFFITS_i$, Cook's D_i , and a bubble-plot of studentized residuals versus leverage values with plotted points proportional to Cook's distance along with the corresponding values that flag observations for further scrutiny assuming $\alpha = 0.10$. Hint: Use the functions **fortify()** from the **ggplot2** package and **lm.influence()**.
 - Remove case 20 from the data frame **KINDER** and regress weight on height, storing the results in **modk20**. Is the child with the largest hat value an influential observation if one considers the 19 observations without case 20 from the original data frame? Compute and consider Cook's D_i , $DFFITS_i$, and $DFBETAS_{k(i)}$ in reaching a conclusion. Specifically, produce a graph showing h_{ii} , the differences in $\hat{\beta}_{1(i)} - \hat{\beta}_1$, $DFBETAS_{k(i)}$, studentized residuals, $DFFITS_i$, Cook's D_i , and a bubble-plot of studentized residuals versus leverage values with plotted points proportional to Cook's distance along with the corresponding values that flag observations for further scrutiny assuming $\alpha = 0.10$.
 - Create a scatterplot showing all 20 children. Use a solid circle to identify case 19 and a solid triangle to identify case 20. Superimpose the lines for models **mod** (**lty** = 1), **modk** (**lty** = 2), **mod19** (**lty** = 3), and **mod20** (**lty** = 4).
13. Suppose a realtor wants to model the appraised price of an apartment in Vitoria as a function of the predictors living area and the status of the apartment's conservation. Consider the data frame **VIT2005**, which contains data about apartments in Vitoria, Spain, including total price, area, and conservation. The variable conservation has four levels: **1A**, **2A**, **2B**, and **3A**.
- Define a new conservation variable called **conservation1** with three levels, **A**, **B**, and **C**, where **A** = **1A**, **B** = **2A**, and **C** = **2B** and **3A** together. Define the corresponding dummy variables considering **A** (the first category) as the reference category.

- (b) Write and fit separate linear regression models (different intercepts and different slopes) for each `conservation1` category.
- (c) Construct a single scatterplot of the data where the fitted models are superimposed over the scatterplot.

Case Study: Biomass

Data and ideas for this case study come from (Goicoa et al., 2011).

14. To estimate the amount of carbon dioxide retained in a tree, its biomass needs to be known and multiplied by an expansion factor (there are several alternatives in the literature). To calculate the biomass, specific regression equations by species are frequently used. These regression equations, called allometric equations, estimate the biomass of the tree by means of some known characteristics, typically diameter and/or height of the stem and branches. The **BIOMASS** file contains data of 42 beeches (*Fagus Sylvatica*) from a forest of Navarra (Spain) in 2006, where

- `diameter`: diameter of the stem in centimeters
 - `height`: height of the tree in meters
 - `stemweight`: weight of the stem in kilograms
 - `aboveweight`: aboveground weight in kilograms
- (a) Create a scatterplot of `aboveweight` versus `diameter`. Is the relationship linear? Superimpose a regression line over the plot just created.
 - (b) Create a scatterplot of $\log(\text{aboveweight})$ versus $\log(\text{diameter})$. Is the relationship linear? Superimpose a regression line over the plot just created.
 - (c) Fit the regression model $\log(\text{aboveweight}) = \beta_0 + \beta_1 \log(\text{diameter})$, and compute R^2 , R_a^2 , and the variance of the residuals.
 - (d) Introduce $\log(\text{height})$ as an explanatory variable and fit the model $\log(\text{aboveweight}) = \beta_0 + \beta_1 \log(\text{diameter}) + \beta_2 \log(\text{height})$. What is the effect of introducing $\log(\text{height})$ in the model?
 - (e) Complete the ANALYSIS QUESTIONS for the model in (d).

ANALYSIS QUESTIONS:

- (1) Estimate the model's parameters and their standard errors. Provide an interpretation for the model's parameters.
- (2) Compute the variance-covariance matrix of the $\hat{\beta}$ s.
- (3) Provide 95% confidence intervals for β_1 and β_2 .
- (4) Compute the R^2 , R_a^2 , and the residual variance.
- (5) Construct a graph with the default diagnostics plots of R.
- (6) Can homogeneity of variance be assumed?
- (7) Do the residuals appear to follow a normal distribution?
- (8) Are there any outliers in the data?

- (9) Are there any influential observations in the data?
- (f) Obtain predictions of the aboveground biomass of trees with diameters `diameter = seq(12.5, 42.5, 5)` and heights `height = seq(10, 40, 5)`. Note that the weight predictions are obtained from back transforming the logarithm. The bias correction is obtained by means of the lognormal distribution: If \hat{Y}_{pred} is the prediction, the corrected (back-transformed) prediction \tilde{Y}_{pred} is given by

$$\tilde{Y}_{\text{pred}} = \exp(\hat{Y}_{\text{pred}} + \hat{\sigma}^2/2)$$

where $\hat{\sigma}^2$ is the variance of the error term.

Case Study: Fruit Trees

Data and ideas for this case study come from Militino et al. (2006).

15. To estimate the total surface occupied by fruit trees in three small areas (R63, R67, and R68) of Navarra in 2001, a sample of 47 square segments has been taken. The experimental units are square segments or quadrats of 4 hectares, obtained by random sampling after overlaying a square grid on the study domain. The focus of this case study is illustrating two different techniques used to obtain estimates: direct estimation and small area estimation. The direct technique estimates the total surface area by multiplying the mean of the observed surface area in the sampled segments by the total number of segments in every small area. The small area technique consists of creating a regression model where the dependent variable is the observed surface area occupied by fruit trees in every segment and the explanatory variables are the classified cultivars by satellite in the same segment and the small areas to which they belong. The final surface area totals are obtained by multiplying the total classified surface area of every small area by the β 's parameter estimates obtained from the regression model (observed surface area \sim classified surface area + small areas).

The surface variables in the data frame `SATFRUIT` are given in m^2 :

- `quadrat` is the number of the sampled segment or quadrat
- `smallarea` are the small areas' labels
- `wheat` is the classified surface of wheat in the sampled segment
- `barley` is the classified surface of barley in the sampled segment
- `nonarable` is the classified surface of fallow or non-arable land in the sampled segment
- `corn` is the classified surface of corn in the sampled segment
- `sunflower` is the classified surface of sunflowers in the sampled segment
- `vineyard` is the classified surface of vineyards in the sampled segment
- `grass` is the classified surface of grass in the sampled segment
- `asparagus` is the classified surface of asparagus in the sampled segment
- `alfalfa` is the classified surface of lucerne (type of alfalfa) in the sampled segment

- `rape` is the classified surface of rape *Brassica napus* in the sampled segment
 - `rice` is the classified surface of rice in the sampled segment
 - `almonds` is the classified surface of almonds in the sampled segment
 - `olives` is the classified surface of olives in the sampled segment
 - `fruit` is the classified surface of fruit trees in the sampled segment
 - `observed` is the observed surface of fruit trees in the sampled segment
- (a) Characterize the shape, center, and spread for the variable `fruit`.
- (b) What is the maximum number of m^2 of classified fruits by segment?
- (c) How many observations are there by small area?
- (d) Use `scatterplotMatrix()` from `car` or `pairs()` to explore the linear relationships between `observed` and the remainder of the numerical variables. Comment on the results.
- (e) Create density plots of the observed fruits' surface area (`observed`) by small areas (`smallarea`).
- (f) Use boxplots and barplots with standard errors to compare the observed surface area (`observed`) and the classified surface area (`fruit`) by small areas (`smallarea`).
- (g) Compute the correlation between `observed` and all other numerical variables. List the three variables in order along with their correlation coefficients that have the highest correlation with `observed`.

Model (A) Use backward elimination to develop a model that predicts `observed` using the data frame **SATFRUIT** without considering `smallarea`. Start the backward elimination process by considering all of the numerical variables in **SATFRUIT** as potential predictors. Use a “ p -value-to-remove” of 10%. Store the final model in the object `modelA`.

- i. Compute CV_n , the leave-one-out cross validation error, for `modelA`. Set the seed to 5 and compute CV_5 , the five-fold cross validation error, for `modelA`. The cross validation error for a generalized linear model can be computed using the `cv.glm()` from the `boot` package. Using the function `glm()` without passing a `family` argument is the same as using the function `lm()`. R Code 12.85 provides a template of how to use the `cv.glm()` function. Note that CV_n is returned with `cv.error$delta[1]`. To compute CV_5 , pass the value 5 to the argument `K` inside the `cv.glm()` function.

R Code 12.85

```
> mod.glm <- glm(y ~ x1 + x2, data = DF)
> library(boot)
> cv.error <- cv.glm(data = DF, glmfit = mod.glm)
> cv.error$delta[1]
```

- ii. Compute R^2 , R_a^2 , the AIC, and the BIC for Model (A). What is the proportion of total variability explained by Model (A)?

- Model (B) Use the criterion-based procedure AIC, which for linear regression is equivalent to Mallow's C_p , to develop a model that predicts **observed** using all of the numerical variables in **SATFRUIT**. Store the model in the object **modelB**. Verify that the model suggested using BIC is the same model as the one suggested by AIC or Mallow's C_p , which are all the same as model (A).
- Model (C) Use mean squared prediction error (*MSPE*) to select a model using all of the numerical variables in **SATFRUIT** as potential predictors for predicting **observed**. Store the model in the object **modelC**. Specifically, select a model using both leave-one-out cross validation (LOOCV) and five-fold cross validation.
- Compute CV_n for **modelC**. Set the seed to 5 and compute CV_5 for **modelC**.
 - Compute R^2 , R_a^2 , the AIC, and the BIC for Model (C). What is the proportion of total variability explained by Model (C)?
- Model (D) Use whichever of model (A) or (C) has the smaller cross-validation error, and introduce **smallarea** into the chosen model. Store the new model that includes **smallarea** in **modelD**.
- Eliminate any variables from **modelD** that are not statistically significant ($\alpha = 0.10$). Store the resulting model in **modelD**.
 - Compute CV_n for **modelD**. Set the seed to 5 and compute CV_5 for **modelD**.
 - Compute R^2 , R_a^2 , the AIC, and the BIC for Model (D). What is the proportion of total variability explained by Model (D)?
 - Does model (D) have a smaller cross validation error than the cross validation error for either model (A) or model (C)?
 - Plot the Cook distances, the studentized residuals, the diagonal elements of the hat matrix, the DFFITS, and DFBETAS₁ of Model (D) versus the index.
 - Are there any leverage points? Justify the answer given.
 - Are there any outliers? Justify the answer given.
 - Check normality and homoscedasticity for Model (D) using graphics and hypotheses tests.
 - Calculate a 95% confidence interval for the **fruit** coefficient.
- (h) How many hectares of observed fruits are expected to be incremented if the classified hectares of fruit trees by the satellite are increased by 10,000 m² (1 ha)?
- (i) Suppose the total classified fruits by the satellite in area R63 is 97,044.28 m², in area R67 is 4,878,603.43 m², and in area R68 is 2,883,488.24 m². Predict the total area of fruit trees by small areas.
- (j) Create a plot of **observed** versus **fruit** with the points color coded according to **smallarea**. Superimpose the corresponding regression lines for each small area.
- (k) Plot the individual predictions for **modelD** versus the observed data. Add a diagonal line to the plot.
- (l) Create a barplot that displays the predicted area occupied by fruit trees based on **modelD** for each small area and the direct estimates of the area occupied by fruit trees by small area knowing that the total number of classified segments in areas R63, R67, and R68, are 119, 703, and 564, respectively.

Case Study: Real Estate

Data and ideas for this case study come from (Militino et al., 2004).

16. The goal of this case study is to walk the user through the creation of a parsimonious multiple linear regression model that can be used to predict the total price (`totalprice`) of apartments by their hedonic (structural) characteristics. The data frame `VIT2005` contains several variables, and further description of the data can be found in the help file.

- (a) Characterize the shape, center, and spread of the variable `totalprice`.
- (b) Use `scatterplotMatrix()` from `car` or `pairs()` to explore the relationships between `totalprice` and the numerical explanatory variables `area`, `age`, `floor`, `rooms`, `toilets`, `garage`, `elevator`, and `storage`.
- (c) Compute the correlation between `totalprice` and all of the other numerical variables. List the three variables in order along with their correlation coefficients that have the highest correlation with `totalprice`.

Model (A) Use backward elimination to develop a model that predicts `totalprice` using the data frame `VIT2005`. Use a “ α -value-to remove” of 5%. Store the final model in the object `modelA`.

- (i) Compute CV_n , the leave-one-out cross validation error, for `modelA`. Set the seed to 5 and compute CV_5 , the five-fold cross validation error, for `modelA`. The cross validation error for a generalized linear model can be computed using the `cv.glm()` from the `boot` package. Using the function `glm()` without passing a `family` argument is the same as using the function `lm()`. R Code 12.86 provides a template of how to use the `cv.glm()` function. Note that CV_n is returned with `cv.error$delta[1]`. To compute CV_5 , pass the value 5 to the argument `K` inside the `cv.glm()` function.

R Code 12.86

```
> mod.glm <- glm(y ~ x1 + x2, data = DF)
> library(boot)
> cv.error <- cv.glm(data = DF, glmfit = mod.glm)
> cv.error$delta[1]
```

- (ii) Compute R^2 , R_a^2 , the AIC, and the BIC for Model (A). What is the proportion of total variability explained by Model (A)?

Model (B) Use the criterion-based procedure AIC, which for linear regression is equivalent to Mallow's C_p , to develop a model that predicts `totalprice` using the variables in `VIT2005`. Store the model in the object `modelB`.

- (i) Compute CV_n for `modelB`. Set the seed to 5 and compute CV_5 for `modelB`.
- (ii) Compute R^2 , R_a^2 , the AIC, and the BIC for Model (B). What is the proportion of total variability explained by Model (B)?

Model (C) Use the criterion-based procedure BIC to develop a model that predicts `totalprice` using the variables in `VIT2005`. Store the model in the object `modelC`.

- (i) Compute CV_n for `modelC`. Set the seed to 5 and compute CV_5 for `modelC`.

- (ii) Compute R^2 , R_a^2 , the AIC, and the BIC for Model (C). What is the proportion of total variability explained by Model (C)?
- Model (D) Use forward selection to develop a model that predicts `totalprice` using the variables in `VIT2005`. Use a “ p -value-to-add” of 5%. Store the final model in the object `modelD`.
- (i) Compute CV_n for `modelD`. Set the seed to 5 and compute CV_5 for `modelD`.
 - (ii) Compute R^2 , R_a^2 , the AIC, and the BIC for Model (D). What is the proportion of total variability explained by Model (D)?
- (d) Explore the residuals of the Models (A), (B), (C), and (D) using the function `residualPlot()` or `residualPlots()` from the package `car`. Comment on the results.
- (e) Use the function `boxCox()` from `car` to find a suitable transformation for `totalprice`.
- Model (E) Use backward elimination to develop a model that predicts `log(totalprice)` using the data frame `VIT2005`. Use a “ p -value-to-remove” of 5%. Store the final model in the object `modelE`.
- (i) Compute CV_n for `modelE`. Set the seed to 5 and compute CV_5 for `modelE`.
 - (ii) Compute R^2 , R_a^2 , the AIC, and the BIC for Model (E). What is the proportion of total variability explained by Model (E)?
- Model (F) Use the criterion-based procedure AIC, which for linear regression is equivalent to Mallow’s C_p , to develop a model that predicts `log(totalprice)` using the variables in `VIT2005`. Store the model in the object `modelF`.
- (i) Compute CV_n for `modelF`. Set the seed to 5 and compute CV_5 for `modelF`.
 - (ii) Compute R^2 , R_a^2 , the AIC, and the BIC for Model (F). What is the proportion of total variability explained by Model (F)?
- Model (G) Use the criterion-based procedure BIC to develop a model that predicts `log(totalprice)` using the variables in `VIT2005`. Store the model in the object `modelG`.
- (i) Compute CV_n for `modelG`. Set the seed to 5 and compute CV_5 for `modelG`.
 - (ii) Compute R^2 , R_a^2 , the AIC, and the BIC for Model (G). What is the proportion of total variability explained by Model (G)?
- Model (H) Use forward selection to develop a model that predicts `log(totalprice)` using the variables in `VIT2005`. Use a “ p -value-to-add” of 5%. Store the final model in the object `modelH`.
- (i) Compute CV_n for `modelH`. Set the seed to 5 and compute CV_5 for `modelH`.
 - (ii) Compute R^2 , R_a^2 , the AIC, and the BIC for Model (H). What is the proportion of total variability explained by Model (H)?
- (f) Which model has the smallest CV_5 as well as the smallest CV_n error among Models (E), (F), (G), and (H)?
- (g) Use the model selected from part (f) and explore its residuals using the function `residualPlots()` from `car`. Comment on the results.

Model (I) Refer to the model selected in part (e) as `modelI`.

- (i) Plot the Cook distances, the studentized residuals, and the diagonal elements of the hat matrix of Model (I) versus the index. Based on the graphs, are there any outliers?
- (ii) Create a bubble-plot of the studentized residuals versus the hat values with the function `influencePlot()`. Are any of the points influential?
- (iii) The original researchers evaluated the apartments in rows 3 and 93 and decided they were not representative and decided to remove them from the study. Remove observations 3 and 93 from consideration in `modelI`.
- (iv) Check normality and homoscedasticity for `modelI` using graphs and hypotheses tests.
- (v) Find the variance inflation factors for Model (I). Is multicollinearity a problem?
- (vi) Find the parameter estimates, and compute 95% confidence intervals for the parameters of Model (I).
- (vii) Find the relative contribution of the explanatory variables to explaining the variability of the prices in Model (I).
- (viii) What is the variable that explains the most variability in Model (I)?
- (ix) What variables jointly explain 80% of the total variability of `log(totalprice)`?
- (x) Find the predictions of Model (I) with bias correction and without bias correction. The bias correction is obtained by means of the lognormal distribution: If \hat{Y}_{pred} is the prediction of Model (I), the corrected (backtransformed) prediction \tilde{Y}_{pred} of Model (I) is given by

$$\tilde{Y}_{\text{pred}} = \exp(\hat{Y}_{\text{pred}} + \hat{\sigma}^2/2)$$

where $\hat{\sigma}^2$ is the variance of the error term, and the confidence interval is given by

$$l_{\text{inf}} = \exp(\hat{Y}_{\text{pred}} + \hat{\sigma}^2/2 - z_{1-\alpha/2} \sqrt{\widehat{\text{Var}}(\hat{Y}_{\text{pred}}) + \widehat{\text{Var}}(\hat{\sigma}^2)/4})$$

$$l_{\text{sup}} = \exp(\hat{Y}_{\text{pred}} + \hat{\sigma}^2/2 + z_{1-\alpha/2} \sqrt{\widehat{\text{Var}}(\hat{Y}_{\text{pred}}) + \widehat{\text{Var}}(\hat{\sigma}^2)/4})$$

and $\widehat{\text{Var}}(\hat{\sigma}^2) = \frac{2\hat{\sigma}^4}{df_{\text{residual}}}$.

- (xi) For Model (I), plot the predicted values (with and without bias correction) versus observed values. Comment on the results.
- (xii) Show that in Model (I) an increment of 10 m² in the area of a flat implies an increment of roughly 4% in the predicted total price. To verify this, find the predicted price of three apartments with `areas` of 80, 90, and 100 m², respectively, and keep the rest of the explanatory variables fixed. For example, assign the following values to the explanatory variables: `zone = Z32`, `elevator = 1`, `toilets = 1`, `garage = 1`, `elevator = 1`, `toilets = 1`, `garage = 0`, `category = 3B`, `out = E50`, `storage = 1`, `heating = 3A`, and `streetcategory = S3`. Compute the corresponding 90% prediction intervals.
- (xiii) What is the percentage change in the total price of an apartment when the number of garages changes from one to two?
- (xiv) What is the percentage change in the total price of an apartment when the heating type changes from “1A” to “3B”?

Appendix A

R Commands

More information on all functions in the appendices can be obtained in their respective help files. To access a help file, type `?function_name` at the R prompt.

Table A.1: Useful Commands When Working with Numeric Vectors

Function	Description
<code>abs(x)</code>	computes $ x $.
<code>cat(x, y)</code>	concatenates <code>x</code> and <code>y</code> and then outputs a single vector.
<code>cbind(x, y)</code>	joins vectors <code>x</code> and <code>y</code> as columns of vectors.
<code>ceiling(x)</code>	computes the smallest integers not less than the corresponding elements of <code>x</code> .
<code>choose(x, y)</code>	computes $\frac{x!}{(x-y)!y!}$.
<code>combn(x, m)</code>	generates all combinations of the elements of <code>x</code> taken <code>m</code> at a time.
<code>cor(x, y)</code>	computes the correlation coefficient.
<code>cos(x)</code>	returns the cosine for all values in <code>x</code> .
<code>cov(x, y)</code>	computes the covariance between <code>x</code> and <code>y</code> .
<code>diff(x)</code>	calculates lagged and iterated differences between values of <code>x</code> .
<code>exp(x)</code>	computes e^x for all values in <code>x</code> .
<code>expand.grid(x, y)</code>	creates a data frame from all combinations of the supplied vectors or factors.
<code>factorial(x)</code>	computes $x!$ for non-negative integers <code>x</code> .
<code>fivenum(x)</code>	computes the minimum value, the lower hinge, the median, the upper hinge, and the maximum value of a vector <code>x</code> .
<code>floor(x)</code>	returns a numeric vector containing the largest integers not greater than the corresponding elements of <code>x</code> .
<code>fractions(x)</code>	finds fractional representations of <code>x</code> (MASS package).
<code>integrate(f, lower, upper)</code>	calculates the integral of a function <code>f</code> between <code>lower</code> and <code>upper</code> .
<code>IQR(x)</code>	returns the interquartile range of <code>x</code> .
<code>length(x)</code>	gets or sets the length of <code>x</code> .
<code>log(x)</code>	computes the natural logarithm for all values in <code>x</code> .
<code>log10(x)</code>	returns the base 10 logarithm for all values in <code>x</code> .

Table A.1: Useful Commands When Working with Numeric Vectors (continued)

Function	Description
<code>mad(x, constant = 1)</code>	returns the median absolute deviation of <code>x</code> .
<code>max(x)</code>	computes the largest value of <code>x</code> .
<code>mean(x)</code>	returns the sample mean of <code>x</code> .
<code>median(x)</code>	computes the sample median of <code>x</code> .
<code>min(x)</code>	returns the smallest value of <code>x</code> .
<code>pretty(x)</code>	computes a sequence of about $n+1$ equally spaced ‘round’ values that cover the range of the values in <code>x</code> .
<code>prod(x)</code>	returns the product of all the values in <code>x</code> .
<code>quantile(x, probs)</code>	produces sample quantiles corresponding to the probabilities given in <code>probs</code> for <code>x</code> .
<code>range(x)</code>	returns the smallest and largest values in <code>x</code> .
<code>rbind(x, y)</code>	joins vectors <code>x</code> and <code>y</code> as rows of vectors.
<code>rep(x, times)</code>	replicates elements in <code>x</code> number of times based on values in vector <code>times</code> .
<code>round(x, n)</code>	rounds the number of decimals to <code>n</code> for object <code>x</code> .
<code>scale(x)</code>	computes the z -score of <code>x</code> .
<code>sd(x)</code>	returns the sample standard deviation of <code>x</code> .
<code>seq(from, to, by)</code>	creates a sequence of numbers starting at <code>from</code> with increments of the value in <code>by</code> ending at <code>to</code> .
<code>sin(x)</code>	returns the sine for all values in <code>x</code> .
<code>sqrt(x)</code>	computes the square root for all values in <code>x</code> .
<code>sum(x)</code>	returns the sum of all the values in <code>x</code> .
<code>summary(x)</code>	computes the minimum, the first quartile, the median, the mean, the third quartile, and the maximum of <code>x</code> .
<code>tan(x)</code>	returns the tangent for all values in <code>x</code> .
<code>var(x)</code>	computes the sample variance of <code>x</code> .
<code>which(x == n)</code>	gives the index of number <code>n</code> in vector <code>x</code> .

Table A.2: Vector and Matrix Functions

Function	Description
<code>A %*% B</code>	multiplies matrices A and B .
<code>diag(matrix)</code>	extracts the diagonal elements of the <code>matrix</code> .
<code>diag(vector)</code>	produces a diagonal matrix with the elements from the <code>vector</code> .
<code>dim(matrix)</code>	returns the dimensions of <code>matrix</code> .

Table A.2: Vector and Matrix Functions (continued)

Function	Description
<code>dimnames(matrix)</code>	returns or sets the dimnames of <code>matrix</code> .
<code>eigen(matrix)</code>	computes eigenvalues and eigenvectors of <code>matrix</code> .
<code>matrix(vector, nrow = r, byrow = TRUE)</code>	creates a matrix by rows with r rows from values in <code>vector</code> .
<code>names(vector)</code>	gets or sets the names of <code>vector</code> .
<code>solve(A)</code>	finds the inverse of a matrix \mathbf{A} .
<code>solve(A, b)</code>	solves systems of equations $\mathbf{Ax} = \mathbf{b}$.
<code>sort(vector)</code>	produces an ordered vector.
<code>svd(matrix)</code>	computes the singular value decomposition of <code>matrix</code> .
<code>t(A)</code>	returns the transpose of a matrix \mathbf{A} .

Table A.3: Functions Used with Arrays, Factors, and Lists

Function	Description
<code>addmargins(A)</code>	adds margins to a table or array (\mathbf{A}).
<code>aggregate(x, by, FUN, ...)</code>	applies function <code>FUN</code> to each element of <code>x</code> based on the categories stored in <code>by</code> and gives the results in a data frame.
<code>apply(X, MARGIN, FUN, ...)</code>	applies function <code>FUN</code> to each <code>MARGIN</code> of <code>X</code> , where <code>X</code> is an array. <code>MARGIN</code> is a vector giving the subscripts over which the function will be applied. 1 indicates rows, 2 indicates columns, ‘ <code>c(1, 2)</code> ’ indicates rows and columns, etc.
<code>attach(object)</code>	makes the columns of an object (e.g., data frame) available by column names.
<code>data.frame(...)</code>	creates a data frame.
<code>detach(object)</code>	detaches an object when one is finished working with it.
<code>dump()</code>	saves the contents of an object.
<code>file.show(name)</code>	displays one or more files specified with <code>name</code> .
<code>ftable(CT)</code>	creates a flat three-way contingency table.
<code>head(object)</code>	shows the first parts of a vector, matrix, table, data frame, or function.
<code>lapply(X, FUN)</code>	applies function <code>FUN</code> to each element of <code>X</code> , where <code>X</code> is a list and the answer is given in the form of a list.
<code>load()</code>	reads a file created with <code>save()</code> in R.
<code>margin.table()</code>	adds margins to a contingency table.
<code>merge(DF1, DF2)</code>	merges two data frames.

Table A.3: Functions Used with Arrays, Factors, and Lists (continued)

Function	Description
<code>prop.table()</code>	calculates proportions in a contingency table.
<code>read.table()</code>	reads a file in table format.
<code>row.names(DF)</code>	gets or sets the names of rows of a data frame.
<code>sapply(X, FUN)</code>	calls the function <code>lapply</code> , which applies function <code>FUN</code> to each element of <code>X</code> , where <code>X</code> is either a list or vector. Note that even if <code>X</code> is a list, <code>sapply(X, FUN)</code> returns either a vector or matrix, not a list, as would <code>lapply(X, FUN)</code> .
<code>save()</code>	writes an external representation of R objects.
<code>scan()</code>	reads data into a vector or list from the console or file.
<code>source()</code>	reads a dumped file.
<code>split(x, f)</code>	returns a list of vectors containing the values for the resulting groups when the vector <code>x</code> is split by the factor <code>f</code> .
<code>table()</code>	creates a contingency table based on the supplied factors.
<code>tail(object)</code>	shows the last parts of a vector, matrix, table, data frame or function.
<code>tapply(x,y,FUN)</code>	applies function <code>FUN</code> to each element of <code>x</code> based on the categories stored in <code>y</code> .
<code>write.table(x)</code>	prints contents of <code>x</code> to a file or connection.

Table A.4: Graphs Frequently Used with Descriptive Statistics

Function	Description
<code>barplot(height, ...)</code>	creates a bar plot with vertical or horizontal bars where <code>height</code> is a matrix or vector giving the heights (positive or negative) of the bars.
<code>boxplot(x)</code>	produces a boxplot of the values stored in <code>x</code> .
<code>boxplot(split(x, f))</code>	produces side-by-side boxplots of the values in <code>x</code> based on the factor <code>f</code> .
<code>dotchart(table, ...)</code>	creates a multi-way dotplot.
<code>ecdf(x)</code>	computes an empirical cumulative distribution function.
<code>hist(x, ...)</code>	creates a histogram of the values in <code>x</code> .
<code>interaction.plot(x.factor, trace.factor, response, ...)</code>	plots the mean (or other summary) of the response for two-way combinations of factors where <code>x.factor</code> contains the levels for the <code>x</code> axis, <code>trace.factor</code> is another factor whose levels form the traces, and <code>response</code> is a numeric variable containing the responses at the various factor combinations.
<code>lines(density())</code>	adds a density to an existing plot (for example, a histogram).

Table A.4: Graphs Frequently Used with Descriptive Statistics (continued)

Function	Description
<code>pairs(x, ...)</code>	creates a scatterplot for each pair of variables in <code>x</code> .
<code>persp(x, y, z, ...)</code>	produces a three-dimensional perspective plot. See system help for more details.
<code>pie(x, ...)</code>	returns a pie chart where the values in <code>x</code> are displayed as the areas of the pie slices.
<code>plot(density(x))</code>	graphs kernel density estimates.
<code>plot(x, ...)</code>	is a generic function for plotting R objects.
<code>plot(x, y)</code>	produces a scatterplot.
<code>plot.design(x, y, fun =)</code>	plots univariate effects of one or more factors, typically for a designed experiment. A function such as mean or median must be typed after <code>fun=</code> , which is then applied to each subset.
<code>qqnorm(x)</code>	produces a quantile-quantile plot that is used to assess how close the values in <code>x</code> follow the normal distribution.
<code>qqline(x)</code>	plots a line through the first and third quartiles of the data, and the corresponding quantiles of the standard normal distribution.
<code>qqplot(x, y)</code>	creates a quantile-quantile plot.
<code>stem(x)</code>	creates a stem-and-leaf plot.
<code>stripchart(x ~ g)</code>	creates a dotplot or strip chart that permits one to compare the distribution of <code>x</code> over <code>g</code> groups.

Table A.5: Basic Plotting Functions

Function	Description
<code>abline(a, b)v</code>	adds a straight line with intercept <code>a</code> and slope <code>b</code> .
<code>abline(h = c,...)</code>	draws a horizontal line at $y = c$.
<code>abline(v = c,...)</code>	draws a vertical line at $x = c$.
<code>curve(f, from, to, ...)</code>	draws a curve of a function <code>f</code> over the interval [<code>from</code> , <code>to</code>].
<code>identify(x, y, labels)</code>	reads the position of the graphics pointer when the (first) mouse button is clicked. <code>x</code> and <code>y</code> are the coordinates of points in a scatterplot and are required arguments. <code>labels</code> is an optional vector, the same length as <code>x</code> and <code>y</code> , giving labels for the points.
<code>legend(x, y, legend, ...)</code>	adds a legend to the current plot, where <code>x</code> and <code>y</code> determine the legend coordinates, and <code>legend</code> is a vector of text values to appear in the legend. To determine the coordinates where we want to place our legend, use the function <code>locator()</code> . It is possible to combine the functions <code>legend()</code> and <code>locator()</code> into one step by using <code>legend(locator(), legend, ...)</code> .

Table A.5: Basic Plotting Functions (continued)

Function	Description
<code>lines(x, y, ...)</code>	adds points or lines to the current plot.
<code>locator()</code>	reads the position of the graphics cursor when the left mouse button is pressed.
<code>mtext()</code>	writes text in the margins of a plot.
<code>points(x, y, ...)</code>	adds points or lines to the current plot at the coordinates specified in the vectors <code>x</code> and <code>y</code> .
<code>segments(x1, y1, x2, y2)</code>	adds the line segment AB with coordinates $A = (x_1, y_1)$ and $B = (x_2, y_2)$ to an existing graph.
<code>text(x, y, labels)</code>	draws the strings given in the vector <code>labels</code> at the coordinates given by <code>x</code> and <code>y</code> . Note: <code>labels</code> is one or more character strings or expressions specifying the text to be written.
<code>title("Title")</code>	adds titles to the current plot. To create a multi-line title, type <code>\n</code> at each place we want the text to start another line.

Table A.6: Commonly Used Graphical Parameters

Parameter	Description
<code>adj = 0</code>	string justification: 0 means left justify, 1 means right justify, 0.5 means center the text. Other numbers are a corresponding distance between the extremes.
<code>axes = TRUE / FALSE</code>	<code>axes = TRUE</code> draws a box around the graph, which is the default value. <code>axes = FALSE</code> removes the box surrounding the graph.
<code>cex = 1</code>	sets character expansion. For example, when <code>cex = 2</code> , characters are twice as big as normal.
<code>col = 1</code>	is used to set color for drawing lines, points, etc.
<code>las = 0</code>	is the style of axis labels (0 = always parallel to the axis — the default, 1 = always horizontal, 2 = always perpendicular to the axis, 3 = always vertical).
<code>lty = 1</code>	is the line type (1 = solid, 2 = small breaks, etc.).
<code>lwd = 1</code>	is the line width (1 = default, 2 = twice as thick, etc.).
<code>main = "title"</code>	is the title for the graph.
<code>par()</code>	is used to set or query graphical parameters. See R help files for more detail.
<code>pch = 19</code>	is the plotting symbol used. For instance, 19 is a solid circle and 22 is a square.
<code>pty = "m"</code>	is the type of plotting region: The default value for <code>pty</code> is <code>m</code> , which generates a maximal size plotting region. <code>pty = "s"</code> generates a square plotting region.

Table A.6: Commonly Used Graphical Parameters (continued)

Parameter	Description
<code>sub = "subtitle"</code>	is the subtitle for a graph.
<code>type = "b"</code>	produces both points and lines and lines between points that are used to represent data values.
<code>type = "h"</code>	produces height bars (vertical) to represent data values.
<code>type = "l"</code>	produces lines that are used to connect data values.
<code>type = "p"</code>	produces points that are used to represent data values, the default argument.
<code>xlab = "label"</code>	is the label for the x -axis.
<code>xlim = c(xmin, xmax)</code>	is the range for the x -axis.
<code>ylab = "label"</code>	is the label for the y -axis.
<code>ylim = c(ymin, ymax)</code>	is the range for the y -axis.

Table A.7: Lattice Functions

Function	Description
<code>barchart(f ~ x z)</code>	produces a bar chart, categorized according to <code>f</code> , conditioning on a factor <code>z</code> .
<code>bwplot(f ~ x z)</code>	creates boxplots for the levels of <code>f</code> (a factor) conditioning on <code>z</code> (another factor).
<code>densityplot(~x z)</code>	creates a density estimate graph.
<code>dotplot(f ~ x z)</code>	returns a dotplot with data categorized by <code>f</code> .
<code>histogram(~x z)</code>	creates a histogram.
<code>qq(f ~ x z)</code>	produces a quantile-quantile plot, <code>f</code> having two levels.
<code>qqmath(~x z)</code>	produces a quantile-quantile graph of <code>x</code> versus a distribution's quantiles while conditioning on <code>z</code> .
<code>stripplot(f ~ x z)</code>	produces strip plots for the levels of <code>f</code> (a factor) conditioning on <code>z</code> (another factor).
<code>xyplot(y ~ x z)</code>	creates an x - y scatterplot.

Note: `x` and `y` represent any numeric variable and `f` and `z` any factor or character variable.

Table A.8: Important Probability Distributions That Work with `rdist`, `pdist`, `ddist`, and `qdist`

Distribution	R name	Parameters
beta	<code>beta</code>	<code>shape1, shape2</code>
binomial	<code>binom</code>	n, π
chi-square	<code>chisq</code>	$df = \nu$
exponential	<code>exp</code>	λ
F	<code>f</code>	ν_1, ν_2
Gamma	<code>gamma</code>	<code>shape, rate</code>
geometric	<code>geom</code>	π
hypergeometric	<code>hyper</code>	m, n, k , where m = number of black balls in urn, n = number of white balls in urn, k = number of balls drawn from the urn
negative binomial	<code>nbinom</code>	n, π
normal	<code>norm</code>	μ, σ
Poisson	<code>pois</code>	λ
Student's t	<code>t</code>	$df = \nu$
uniform	<code>unif</code>	a, b
Weibull	<code>weibull</code>	<code>shape, scale</code>
Wilcoxon rank sum	<code>wilcox</code>	n, m (number of observations on the first and second sample, respectively)
Wilcoxon signed rank	<code>signrank</code>	n

Table A.9: Useful Functions for Parametric Inference

Function	Description
<code>fisher.test(x, y = NULL, ...)</code>	performs Fisher's exact test for testing the null hypothesis of independence between rows and columns in a contingency table (<code>x</code>) with fixed marginals.
<code>power.anova.test()</code>	computes the power of an analysis of variance test or determines the parameters to obtain a target power.
<code>power.t.test()</code>	computes the power of a t -test or determines the parameters to obtain a target power.
<code>prop.test(x, n, p, alternative = "two.sided", conf.level = 0.95, correct = TRUE)</code>	compares proportions against hypothesized values, where <code>x</code> is a vector of successes, <code>n</code> is a vector containing the number of trials, and <code>p</code> is a vector of probabilities of success specified by the null hypothesis. A continuity correction (<code>correct = TRUE</code>) is used by default.

Table A.9: Useful Functions for Parametric Inference (continued)

Function	Description
<code>t.test(x, y = NULL, alternative = "two.sided", mu = 0, paired = FALSE, var.equal = FALSE, conf.level = 0.95)</code>	performs a one-sample, two-sample, or paired <i>t</i> -test, or a Welch modified two-sample <i>t</i> -test.
<code>var.test(x, y, ratio = 1, alternative = "two.sided", conf.level = 0.95)</code>	performs an <i>F</i> -test to compare the variances of two samples from normal populations.

Table A.10: Useful Functions for Nonparametric Inference

Function	Description
<code>binom.test(x, n, p = 0.5, alternative = "two.sided")</code>	performs an exact test of a simple null hypothesis about the probability of success in a Bernoulli experiment.
<code>chisq.test(x, y = NULL, correct = TRUE)</code>	performs a Pearson's chi-square test on a two-dimensional contingency table, where <i>x</i> is either a matrix or a contingency table.
<code>friedman.test(y, groups, blocks)</code>	performs a Friedman rank-sum test with unreplicated blocked data, where <i>y</i> is numeric vector, <i>groups</i> is a category object specifying group membership, and <i>blocks</i> is a category object specifying the block membership.
<code>kruskal.test(y, groups)</code>	performs a Kruskal-Wallis rank-sum test on data, where <i>y</i> is a numeric vector and <i>groups</i> denotes a category object of the same length as <i>y</i> , specifying the group for each corresponding element of <i>y</i> .
<code>ks.test(x, y, ...)</code>	performs a one- or two-sample Kolmogorov-Smirnov test, which tests the relationship between two distributions.
<code>wilcox.test(x, y, alternative = "two.sided", mu = 0, paired = FALSE, exact = FALSE, correct = TRUE)</code>	computes Wilcoxon signed-rank test for paired or one-sample data and Wilcoxon rank-sum test (Mann-Whitney test) for two-sample data.

Table A.11: Useful Functions in R for Linear Regression and Analysis of Variance

Function	Description
<code>aov(formula, data)</code>	fits an analysis of variance model according to the specified formula using the specified data.
<code>coefficients(lm object)</code>	returns the coefficients from a fitted linear regression model. A shorter command with identical results is <code>coef(lm object)</code> .
<code>data.matrix(proj(aov.object))</code>	returns columns containing estimates for both factors and residuals of <code>aov</code> -type objects.
<code>formula(lm object)</code>	returns the formula used to fit the linear model.
<code>lm(formula, data)</code>	fits a linear model to the data according to the user-specified formula.
<code>ls.diag(lm object)</code>	is used on an <code>ls-</code> or <code>lm</code> -type of object that returns a list containing several quantities for assessing the fit of a least squares regression model including the standard deviation of the residuals, studentized residuals, and the standard errors of the parameter estimates.
<code>lsfit(explanatory variables, response variable(s))</code>	fits a model using least squares multivariate regression. A list of the estimated coefficients and residuals as well as the QR decomposition of the matrix of explanatory variables is returned. Although the fitted model from <code>lsfit()</code> is identical to <code>lm()</code> , the manner in which the model is specified and the output for the two functions are different.
<code>model.matrix(lm object)</code>	creates a design matrix.
<code>pairwise.t.test(x, g, ...)</code>	computes pairwise comparisons between group levels with corrections for multiple testing.
<code>shapiro.test(x)</code>	computes the Shapiro-Wilk W -statistic, a well-known goodness-of-fit test for the normal distribution.
<code>TukeyHSD(x, ...)</code>	creates a set of confidence intervals on the differences between the means of the levels of a factor with the specified family-wise probability of coverage.

Table A.12: Useful Contrast Functions in R for Linear Regression and Analysis of Variance

Function	Description
<code>contr.helmert(n)</code>	returns a matrix of orthogonal contrasts for different combinations of the factor with <code>n</code> levels where contrast i is the difference between level $i + 1$ and the average of levels 1 through i .
<code>contr.poly(n)</code>	returns a matrix of orthogonal contrasts for different combinations of the factor with <code>n</code> levels.

Table A.12: Useful Contrast Functions in R for Linear Regression and Analysis of Variance
(continued)

Function	Description
<code>contr.sum(n)</code>	returns a matrix of non-orthogonal contrasts based on “sum to zero contrasts.”
<code>contr.treatment(n)</code>	returns the coding that is not technically a set of contrasts at all.

Table A.13: Useful Model-Building Functions for Linear Regression and Analysis of Variance

Function	Description
<code>add1(object, scope, ...)</code>	computes all the single terms in the scope argument that can be added to the model, fits those models, and computes a table of the changes in fit.
<code>drop1(object, scope, ...)</code>	computes all the single terms in the scope argument that can be dropped from the model, fits those models, and computes a table of the changes in fit.
<code>leaps(x, y)</code>	performs an exhaustive search for the best subsets of the variables in <code>x</code> for predicting <code>y</code> in linear regression, is part of the package <code>leaps</code> .
<code>regsubsets(x, ...)</code>	performs model selection by exhaustive search, forward or backward stepwise, or sequential replacement, is part of the package <code>leaps</code> .
<code>step(object, scope, ...)</code>	performs stepwise model selection: The starting model is specified in the first argument (<code>object</code>) and the range of models is specified in the <code>scope</code> argument.
<code>update(object, ~ . ± explanatory.variables)</code>	allows a linear model object to be updated by including, eliminating, or modifying the variables.

Table A.14: Useful Diagnostic Functions for Linear Regression and Analysis of Variance

Function	Description
<code>fitted(object)</code>	returns the fitted values from the fitted linear model object.
<code>plot(object)</code>	creates six diagnostic plots (four by default).
<code>predict(object)</code>	returns a vector or an array of predictions using the fitted model specified in <code>object</code> .
<code>residuals(object)</code>	returns the residuals for the fitted linear model object. A shorter command with identical results is <code>resid(object)</code> .

Table A.14: Useful Diagnostic Functions for Linear Regression and Analysis of Variance
(continued)

Function	Description
<code>summary(object)</code>	returns a complete statistical summary for the fitted linear model object.

Table A.15: Functions from PASWR2

Function	Description
<code>checking.plots(object)</code>	creates four graphs that can be used to help assess independence, normality, and constant variance.
<code>cisim()</code>	simulates random samples from which it constructs confidence intervals for either the population mean, the population variance, or the population proportion of successes.
<code>eda(x)</code>	produces a histogram, density plot, boxplot, and Q-Q plot.
<code>interval.plot(l1, u1)</code>	graphs intervals given a vector of lower values (<code>l1</code>) and a vector of upper values (<code>u1</code>).
<code>multiplot(...)</code>	arranges multiple objects of class <code>ggplot</code> on a graphics device.
<code>normarea(lower, upper, m, sig)</code>	computes and draws the area between two user-specified values in a user-specified normal distribution with a given mean and standard deviation.
<code>nsize(b)</code>	determines the required sample size to be within a given margin of error.
<code>ntester(data)</code>	creates Q-Q plots of randomly generated normal data of the same sample size as the tested data are generated and plotted on the perimeter of the graph, while a Q-Q plot of the actual data is depicted in the center of the graph.
<code>oneway.plots(Y, fac1)</code>	creates dotplots, boxplots, and design plot (means) for single factor designs.
<code>SIGN.test()</code>	tests a hypothesis based on the sign test and reports linearly interpolated confidence intervals for one-sample problems.
<code>srs(popvalues, n)</code>	computes all possible samples from a given population using simple random sampling.
<code>tsum.test()</code>	performs a one-sample, two-sample, or Welch modified two-sample <i>t</i> -test based on user-supplied summary information. Output is identical to that produced with <code>t.test()</code> .
<code>twoway.plots(Y, fac1, fac2)</code>	creates side-by-side boxplots for each factor, a design plot (means), and an interaction plot.
<code>wilcoxe.test()</code>	performs exact one-sample and two-sample Wilcoxon tests on vectors of data.

Table A.15: Functions from PASWR2 (continued)

Function	Description
<code>z.test()</code>	is based on the standard normal distribution, and creates confidence intervals and tests hypotheses for both one- and two-sample problems.
<code>zsum.test()</code>	is based on the standard normal distribution, and creates confidence intervals and tests hypotheses for both one- and two-sample problems based on summarized information the user passes to the function. Output is identical to that produced with <code>z.test()</code> .

Appendix B

Quadratic Forms and Random Vectors and Matrices

B.1 Quadratic Forms

DEFINITION B.1: Assume that the scalar W can be expressed as a function of the n variables Y_1, Y_2, \dots, Y_n . That is,

$$W = f(Y_1, Y_2, \dots, Y_n) = f(\mathbf{Y}) \text{ and } \frac{\delta W}{\delta \mathbf{Y}} = \begin{bmatrix} \frac{\delta W}{\delta Y_1} \\ \vdots \\ \frac{\delta W}{\delta Y_n} \end{bmatrix}.$$

DEFINITION B.2: Let \mathbf{A} be an $n \times n$ matrix and $\mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}$ be an $n \times 1$ column vector of real variables. Then $\mathbf{q} = \mathbf{Y}' \mathbf{A} \mathbf{Y}$ is called a **quadratic form** in \mathbf{Y} , and \mathbf{A} is called the matrix of the quadratic form.

Rules for Differentiation

1. Let $\mathbf{W} = \mathbf{A}' \mathbf{Y}$, where \mathbf{A} is a vector of scalars. Then $\frac{\delta \mathbf{W}}{\delta \mathbf{Y}} = \mathbf{A}$.
2. Let $\mathbf{W} = \mathbf{Y}' \mathbf{Y}$. Then, $\frac{\delta \mathbf{W}}{\delta \mathbf{Y}} = 2\mathbf{Y}$.
3. Let $\mathbf{W} = \mathbf{Y}' \mathbf{A} \mathbf{Y}$, where \mathbf{A} is an $n \times n$ matrix. Then $\frac{\delta \mathbf{W}}{\delta \mathbf{Y}} = \mathbf{A} \mathbf{Y} + \mathbf{A}' \mathbf{Y}$.

Example B.1 Let $\mathbf{A} = \begin{bmatrix} 5 & 2 & 1 \\ 2 & 3 & -6 \\ 1 & -6 & 4 \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix}$.

Then, $\mathbf{W} = \mathbf{Y}' \mathbf{A} \mathbf{Y} = 5Y_1^2 + 3Y_2^2 + 4Y_3^2 + 4Y_1 Y_2 + 2Y_1 Y_3 - 12Y_2 Y_3$, and the partial derivatives of \mathbf{W} are

$$\begin{aligned} \frac{\delta \mathbf{W}}{\delta \mathbf{Y}_1} &= 10Y_1 + 4Y_2 + 2Y_3 \\ \frac{\delta \mathbf{W}}{\delta \mathbf{Y}_2} &= 6Y_2 + 4Y_1 - 12Y_3 \\ \frac{\delta \mathbf{W}}{\delta \mathbf{Y}_3} &= 8Y_3 + 2Y_1 - 12Y_2 \end{aligned}$$

or by Rule for Differentiation 3

$$\frac{\delta \mathbf{W}}{\delta \mathbf{Y}} = \mathbf{A} \mathbf{Y} + \mathbf{A}' \mathbf{Y} = \begin{bmatrix} 10Y_1 + 4Y_2 + 2Y_3 \\ 4Y_1 + 6Y_2 - 12Y_3 \\ 2Y_1 - 12Y_2 + 8Y_3 \end{bmatrix}.$$

B.2 Random Vectors and Matrices

A random vector or a random matrix contains elements that are themselves random variables rather than real variables or scalar values.

DEFINITION B.3: Given a $p \times 1$ random vector $\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{bmatrix}$, the expected value of \mathbf{Y} , denoted by $E(\mathbf{Y})$, is defined as $E(\mathbf{Y}) = \begin{bmatrix} E(Y_1) \\ E(Y_2) \\ \vdots \\ E(Y_p) \end{bmatrix}$.

Basically, the expected value of a random vector is the vector of the expected values of the elements in the random vector. This concept extends to the expected value of a random matrix as well. That is, given a random $n \times p$ matrix \mathbf{Y} , $E(\mathbf{Y}) = [E(Y_{ij})]$ for all $i = 1, \dots, n$ and $j = 1, \dots, p$ pairs.

B.3 Variance of Random Vectors

Recall that the variance of a random variable Y defined in (3.9) on page 220 measures the variability of Y about its mean μ . Specifically,

$$\sigma_Y^2 = \text{Var}(Y) = E[(Y - E(Y))^2] = E[(Y - \mu)^2].$$

The notion of variability is slightly more challenging to extend to vectors and matrices. The difficulty arises because of the covariance between random variables. Recall that the covariance between random variables X and Y was defined as

$$\text{Cov}[X, Y] = E[(X - \mu_X)(Y - \mu_Y)]$$

in (5.14) on page 327. To compute both the variances and covariances of random variables in

the $p \times 1$ random vector $\mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_p \end{bmatrix}$, we construct a $p \times p$ matrix where the diagonal entries are the variances of Y_1 to Y_p , while the off-diagonal entries are the covariances between Y_i and Y_j , where $i \neq j$.

DEFINITION B.4: The **variance-covariance** matrix of \mathbf{Y} , denoted $\sigma_{\mathbf{Y}}^2$, is defined as

$$\sigma_{\mathbf{Y}}^2 = E[(\mathbf{Y} - \mu_{\mathbf{Y}})(\mathbf{Y} - \mu_{\mathbf{Y}})']. \quad (\text{B.1})$$

The calculations of the expanded form of $\sigma_{\mathbf{Y}}^2$ are

$$\begin{aligned}
 \sigma_{\mathbf{Y}}^2 &= E[(\mathbf{Y} - \mu_{\mathbf{Y}})(\mathbf{Y} - \mu_{\mathbf{Y}})'] \\
 &= E \left\{ \begin{bmatrix} Y_1 - \mu_1 \\ Y_2 - \mu_2 \\ \vdots \\ Y_p - \mu_p \end{bmatrix} [Y_1 - \mu_1, Y_2 - \mu_2, \dots, Y_p - \mu_p] \right\} \\
 &= \begin{bmatrix} E[(Y_1 - \mu_1)^2] & E[(Y_1 - \mu_1)(Y_2 - \mu_2)] & \cdots & E[(Y_1 - \mu_1)(Y_p - \mu_p)] \\ E[(Y_2 - \mu_2)(Y_1 - \mu_1)] & E[(Y_2 - \mu_2)^2] & \cdots & E[(Y_2 - \mu_2)(Y_p - \mu_p)] \\ \vdots & \vdots & & \vdots \\ E[(Y_p - \mu_p)(Y_1 - \mu_1)] & E[(Y_p - \mu_p)(Y_2 - \mu_2)] & \cdots & E[(Y_p - \mu_p)^2] \end{bmatrix} \\
 &= \begin{bmatrix} \sigma_{Y_1}^2 & \sigma_{Y_1 Y_2} & \cdots & \sigma_{Y_1 Y_p} \\ \sigma_{Y_2 Y_1} & \sigma_{Y_2}^2 & \cdots & \sigma_{Y_2 Y_p} \\ \vdots & \vdots & & \vdots \\ \sigma_{Y_p Y_1} & \sigma_{Y_p Y_2} & \cdots & \sigma_{Y_p}^2 \end{bmatrix}.
 \end{aligned}$$

The following rules will help simplify complex expressions so that their variances can be determined more easily. It is frequently the case that a random vector, \mathbf{Z} , is obtained by premultiplying the random vector \mathbf{Y} by a constant matrix \mathbf{A} . That is, $\mathbf{Z} = \mathbf{AY}$.

1. $E[\mathbf{A}] = \mathbf{A}$
2. $E[\mathbf{Z}] = E[\mathbf{AY}] = \mathbf{AE}[\mathbf{Y}]$
3. $\sigma_{\mathbf{Z}}^2 = \sigma_{\mathbf{AY}}^2 = \mathbf{A}\sigma_{\mathbf{Y}}^2\mathbf{A}'$

where $\sigma_{\mathbf{Y}}^2$ is the variance-covariance matrix of \mathbf{Y} .

Bibliography

- Agresti, A. 1996. *An Introduction to Categorical Data Analysis*. Wiley Series in Probability and Statistics: Applied Probability and Statistics. New York: John Wiley & Sons.
- _____. 2002. *Categorical Data Analysis*. Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons.
- Agresti, A. and B. A. Coull. 1998. Approximate is better than *exact* for interval estimation of binomial proportions. *The American Statistician* 52, no. 2:119–126.
- Anturane Reinfarction Trial Research Group. 1980. Sulfapyrazone in the prevention of sudden death after myocardial infarction. *New England Journal of Medicine* 302:250–256.
- Arnhold, A. T. 2012a. *BSDA: Basic Statistics and Data Analysis*. <http://CRAN.R-project.org/package=BSDA>. R package version 1.01.
- _____. 2012b. *PASWR: PROBABILITY and STATISTICS WITH R*. <http://CRAN.R-project.org/package=PASWR>. R package version 1.1.
- _____. 2014. *PASWR2: Probability and Statistics with R, Second Edition*. <http://CRAN.R-project.org/package=PASWR2>. R package version 1.0.
- Artuch, R., C. Colomé, C. Sierra, N. Brandi, N. Lambruschini, J. Campistol, M. D. Ugarte, and M. Villaseca. 2004. Study of antioxidant status in phenylketonuric patients. *Clinical Biochemistry* 37:198–203.
- Auguie, B. 2012. *gridExtra: functions in Grid graphics*. <http://CRAN.R-project.org/package=gridExtra>. R package version 0.9.1.
- Belsley, D. A., E. Kuh, and R. E. Welsch. 1980. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons.
- Blaker, H. 2000. Confidence curves and improved exact confidence intervals for discrete distributions. *Canadian Journal of Statistics* 28:783–798.
- Blom, G. 1958. *Statistical Estimates and Transformed Beta-variables*. New York: John Wiley & Sons.
- Boettiger, C. 2014. *knitcitations: Citations for knitr markdown files*. <http://CRAN.R-project.org/package=knitcitations>. R package version 1.0.5.
- Brown, L. D., T. T. Cai, and A. DasGupta. 05 2001. Interval estimation for a binomial proportion. *Statistical Science* 16, no. 2:101–133. <http://dx.doi.org/10.1214/ss/1009213286>.
- Brownrigg, R. 2014. *maps: Draw Geographical Maps*. <http://CRAN.R-project.org/package=maps>. R package version 2.3-9.
- Burton, D. M. 2010. *Elementary Number Theory*. Boston: McGraw-Hill Science/Engineering/Math, seventh ed.

- Canty, A. and B. Ripley. 2015. *boot: Bootstrap Functions (Originally by Angelo Canty for S)*. <http://CRAN.R-project.org/package=boot>. R package version 1.3-16.
- Cao, R., M. Francisco, S. Naya, M. Presedo, M. Vázquez, J. A. Vilar, and J. M. Vilar. 2001. *Introducción a la Estadística y sus Aplicaciones*. Madrid: Ediciones Pirámide.
- Casas, J. M., C. García, L. P. Rivera, and A. I. Zamora. 1998. *Problemas de Estadística*. Madrid: Ediciones Pirámide.
- Casella, G. and R. L. Berger. 1990. *Statistical Inference*. The Wadsworth & Brooks/Cole Statistics/Probability Series. Pacific Grove, CA: Wadsworth & Brooks/Cole Advanced Books & Software.
- . 2002. *Statistical Inference*. The Wadsworth & Brooks/Cole Statistics/Probability Series. Pacific Grove, CA: Duxbury, second ed.
- Chambers, J. 2008. *Software for Data Analysis: Programming with R*. Statistics and Computing. New York: Springer.
- Chang, W. 2013. *R Graphics Cookbook*. Sebastopol, CA: O'Reilly Media, Inc.
- . 2014. *extrafont: Tools for using fonts*. <http://CRAN.R-project.org/package=extrafont>. R package version 0.17.
- . 2015. Cookbook for R. <http://www.cookbook-r.com/>.
- Chang, W., A. Kryukov, and P. Murrell. 2014. *fontcm: Computer Modern font for use with extrafont package*. <http://CRAN.R-project.org/package=fontcm>. R package version 1.1.
- Chang, W., J. Cheng, J. Allaire, Y. Xie, and J. McPherson. 2015. *shiny: Web Application Framework for R*. <http://CRAN.R-project.org/package=shiny>. R package version 0.11.1.
- Chatterjee, S. and A. S. Hadi. 1988. *Sensitivity Analysis in Linear Regression*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. New York: John Wiley & Sons.
- Chatterjee, S. and B. Price. 1991. *Regression Diagnostics*. New York: John Wiley & Sons.
- Chihara, L. and T. Hesterberg. 2011. *Mathematical Statistics with Resampling and R*. Hoboken, NJ: John Wiley & Sons, Inc.
- Christensen, R. 1996. *Analysis of Variance, Design and Regression*. New York: Chapman and Hall.
- Chu, S. 2003. Using soccer goals to motivate the poisson process. *INFORMS Transaction on Education* 3, no. 2:62–68.
- Cleveland, W. S. 1993. *Visualizing Data*. Summit, NJ: Hobart Press.
- . 1994. *The Elements of Graphing Data*. Summit, NJ: Hobart Press.
- Cochran, W. G. 1977. *Sampling Techniques*. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons, third ed.
- Conover, W. J. 1999. *Practical Nonparametric Statistics*. Monographs on Statistics and Applied Probability. New York: John Wiley & Sons, third ed.
- Cook, R. D. and S. Weisberg. 1982. *Residuals and Influence in Regression*. Monographs on

- Statistics and Applied Probability. London: Chapman & Hall.
- Cuesta, M. J., M. D. Ugarte, T. Goicoa, S. Eraso, and V. Peralta. 2007. A taxometric analysis of schizophrenia symptoms. *Psychiatry Research* 150:245–253.
- Dahl, D. B. 2014. *xtable: Export tables to LaTeX or HTML*. <http://CRAN.R-project.org/package=xtable>. R package version 1.7-4.
- Dalgaard, P. 2002. *Introductory Statistics with R*. Statistics and Computing. New York: Springer-Verlag.
- . 2008. *Introductory Statistics with R*. Statistics and Computing. New York: Springer-Verlag, second ed.
- Dallal, G. E. and L. Wilkinson. 1986. An analytic approximation to the distribution of Lilliefors' test for normality. *The American Statistician* 40:294–296.
- Darwin, C. 1876. *The Effects of Cross Fertilisation in the Vegetable Kingdom*. J. Murray.
- Davis, J. 1986. *Statistics and Data Analysis in Geology*. New York: John Wiley & Sons.
- Davison, A. C. and D. V. Hinkley. 1997. *Bootstrap Methods and Their Application*, vol. 1 of Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press. With 1 IBM-PC floppy disk (3.5 inch; HD).
- Devore, J. 2004. *Probability and Statistics for Engineering and the Sciences*. Belmont, CA: Brooks/Cole - THomson Learning, sixth ed.
- Dorai-Raj, S. 2014. *binom: Binomial Confidence Intervals For Several Parameterizations*. <http://CRAN.R-project.org/package=binom>. R package version 1.1-1.
- Dowle, M., T. Short, S. Lianoglou, and A. Srinivasan. 2014. *data.table: Extension of data.frame*. <http://CRAN.R-project.org/package=data.table>. R package version 1.9.4.
- Dragulescu, A. A. 2014a. *xlsx: Read, write, format Excel 2007 and Excel 97/2000/XP/2003 files*. <http://CRAN.R-project.org/package=xlsx>. R package version 0.5.7.
- . 2014b. *xlsxjars: Package required POI jars for the xlsx package*. <http://CRAN.R-project.org/package=xlsxjars>. R package version 0.6.1.
- Draper, N. R. and H. Smith. 1998. *Applied Regression Analysis*. Wiley Series in Probability and Statistics: Texts and References Section. New York: John Wiley & Sons, third ed. With 1 IBM-PC floppy disk (3.5 inch; DD).
- Dudewicz, E. J. and T. A. Bishop. 1981. Analysis of variance with unequal variances. *Journal of Quality Technology* 13, no. 2:111–114.
- Efron, B. 1979. Bootstrap methods: another look at the jackknife. *Annals of Statistics* 7:1–26.
- Efron, B. and R. J. Tibshirani. 1993. *An Introduction to the Bootstrap*, vol. 57 of Monographs on Statistics and Applied Probability. New York: Chapman and Hall.
- Else, M. Z., P. García-Jiménez, and R. Robaina. 2012. Endogenous polyamine content and photosynthetic performance under hypo-osmotic conditions reveal *Cymodocea nodosa* as an obligate halophyte. *Aquat. Biol.* 17, no. 1:7–17. <http://dx.doi.org/10.3354/ab00454>.
- Faraway, J. 2014. *faraway: Functions and datasets for books by Julian Faraway*. <http://>

- //CRAN.R-project.org/package=faraway. R package version 1.0.6.
- Faraway, J. J. 2005. *Linear Models with R*. Chapman & Hall/CRC Texts in Statistical Science Series. Boca Raton, FL: Chapman & Hall/CRC.
- . 2006. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. Texts in Statistical Science Series. Chapman & Hall/CRC, Boca Raton, FL.
- Fdez. Militino, A., S. Gómez, and G. Aldaz. 1994. *Problemas Resueltos y Aplicaciones de Estadística*. Pamplona, Spain: UNED.
- Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, no. II:179–188.
- . 1971. *The Design of Experiments*. New York: Macmillan Publishing Co., ninth ed.
- Fleiss, J. L. 1981. *Statistical Methods for Rates and Proportions*. Wiley Series in Probability and Mathematical Statistics. New York: John Wiley & Sons, second ed.
- Fox, J. 1997. *Applied Regression Analysis, Linear Models, and Related Methods*. Thousand Oaks, CA: SAGE Publications, first ed.
- . 2002. *An R and S-Plus Companion to Applied Regression*. Thousand Oaks, CA: SAGE Publications, first ed.
- Fox, J. and S. Weisberg. 2011. *An R Companion to Applied Regression*. Thousand Oaks, CA: SAGE Publications, second ed.
- . 2015. *car: Companion to Applied Regression*. <http://CRAN.R-project.org/package=car>. R package version 2.0-25.
- Francois, R. 2014. *bibtex: bibtex parser*. <http://CRAN.R-project.org/package=bibtex>. R package version 0.4.0.
- Friendly, M. 2015. *vcdExtra: 'vcd' Extensions and Additions*. <http://CRAN.R-project.org/package=vcdExtra>. R package version 0.6-8.
- Gandrud, C. 2014. *Reproducible Research with R and RStudio*. The R Series. Boca Raton, FL: Chapman & Hall/CRC.
- . 2015. *repmis: Miscellaneous Tools for Reproducible Research*. <http://CRAN.R-project.org/package=repmis>. R package version 0.4.2.
- Genz, A., F. Bretz, T. Miwa, X. Mi, and T. Hothorn. 2014. *mvtnorm: Multivariate Normal and t Distributions*. <http://CRAN.R-project.org/package=mvtnorm>. R package version 1.0-2.
- Gibbons, J. D. 1997. *Nonparametric Methods for Quantitative Analysis*, vol. 2 of American Series in Mathematical and Management Sciences. Columbus, OH: American Sciences Press, third ed.
- Gibbons, J. D. and S. Chakraborti. 1992. *Nonparametric Statistical Inference*, vol. 131 of Statistics: Textbooks and Monographs. New York: Marcel Dekker, third ed.
- . 2003. *Nonparametric Statistical Inference*, vol. 168 of Statistics: Textbooks and Monographs. New York: Marcel Dekker, fourth ed.
- Gilmour, S. G. 1996. The interpretation of Mallows's c_p -statistic. *The Statistician* 45, no. 1:49–56.

- Goicoa, T., A. F. Militino, and M. Ugarte. 2011. Modelling aboveground tree biomass while achieving the additivity property. *Environmental and Ecological Statistics* 18, no. 2:367–384.
- Gómez, K. and A. Gómez. 1984. *Statistical Procedures for Agricultural Research*. New York: John Wiley & Sons.
- Graves, S., H.-P. Piepho, and L. Selzer. 2012. *multcompView: Visualizations of Paired Comparisons*. <http://CRAN.R-project.org/package=multcompView>. R package version 0.1-5.
- Gray, J. B. and H. Woodall. 1994. The maximum size of standardized and internally studentized residuals in regression analysis. *The American Statistician* 48, no. 2:111–113.
- Graybill, F. A. 1976. *Theory and Application of the Linear Model*. Belmont, CA: Wadsworth Publishing Company.
- Gross, J. and U. Ligges. 2015. *nortest: Tests for Normality*. <http://CRAN.R-project.org/package=nortest>. R package version 1.0-3.
- Hand, D. J., F. Daly, A. D. Lunn, K. J. McConway, and E. Ostrowski. 1994. *A Handbook of Small Data Sets*. London: Chapman & Hall.
- Harrell, F. E., Jr. 2015. *Hmisc: Harrell Miscellaneous*. <http://CRAN.R-project.org/package=Hmisc>. R package version 3.15-0.
- Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, second ed. <http://dx.doi.org/10.1007/978-0-387-84858-7>. Data mining, inference, and prediction.
- Hennekens, C. 1988. Preliminary report: Findings from the aspirin component of the ongoing physician's health study. *New England Journal of Medicine* 318:262–264.
- Hicks, C. R. 1956. Fundamentals of analysis of variance, part ii—the components of variance model and the mixed model. *Industrial Quality Control* 13:5–8.
- Hines, W. G. S. 1996. Pragmatics of pooling in ANOVA tables. *The American Statistician* 50, no. 2:127–139.
- Hines, W. W. and D. C. Montgomery. 1990. *Probability and Statistics in Engineering and Management Science*. New York: John Wiley & Sons, third ed.
- Hocking, R. R. 1996. *Methods and Applications of Linear Models*. New York: John Wiley & Sons.
- Hollander, M. and D. A. Wolfe. 1999. *Nonparametric Statistical Methods*. Wiley Series in Probability and Statistics: Texts and References Section. New York: John Wiley & Sons, second ed.
- Hothorn, T., K. Hornik, M. A. van de Wiel, and A. Zeileis. 2006. A Lego system for conditional inference. *The American Statistician* 60, no. 3:257–263.
- Hothorn, T., K. Hornik, M. A. van de Wiel, and A. Zeileis. 2014. *coin: Conditional Inference Procedures in a Permutation Test Framework*. <http://CRAN.R-project.org/package=coin>. R package version 1.0-24.
- Hothorn, T., F. Bretz, and P. Westfall. 2015. *multcomp: Simultaneous Inference in General Parametric Models*. <http://CRAN.R-project.org/package=multcomp>. R package version 1.4-0.

- James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An Introduction to Statistical Learning*, vol. 103 of Springer Texts in Statistics. Springer, New York. <http://dx.doi.org/10.1007/978-1-4614-7138-7>. With applications in R.
- Johnson, N. L. and S. Kotz. 2011. *Leading Personalities in Statistical Sciences: From the Seventeenth Century to the Present*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. New York: John Wiley & Sons.
- Johnson, N. L., S. Kotz, and N. Balakrishnan. 1995. *Continuous Univariate Distributions. Vol. 2*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. New York: John Wiley & Sons, second ed.
- Johnson, S. G. and R. by Balasubramanian Narasimhan. 2013. *cubature: Adaptive multivariate integration over hypercubes*. <http://CRAN.R-project.org/package=cubature>. R package version 1.1-2.
- Kalbfleisch, J. G. 1985a. *Probability and Statistical Inference. Vol. 1*. Springer Texts in Statistics. New York: Springer-Verlag, second ed.
- . 1985b. *Probability and Statistical Inference. Vol. 2*. Springer Texts in Statistics. New York: Springer-Verlag, second ed.
- Kitchens, L. J. 2003. *Basic Statistics and Data Analysis*. Pacific Grove, CA: Brooks/Cole, a division of Thomson Learning.
- Kleinbaum, D. and L. Kupper. 1998. *Applied Regression Analysis and Other Multivariable Methods*. London: Duxbury Press, third ed.
- Kopka, H. and P. W. Daly. 1995. *A Guide to L^AT_EX2e*. New York: Addison-Wesley, second ed.
- Krause, A. and M. Olson. 1997. *The Basics of S and S-PLUS*. New York: Springer-Verlag.
- Kutner, M. H., C. J. Nachtsheim, J. Neter, and W. Li. 2004. *Applied Linear Statistical Models*. Boston: McGraw-Hill/Irwin, fifth ed.
- Lapin, L. L. 1990. *Probability and Statistics for Modern Engineering*. Boston: PWS-KENT Publishing Company, second ed.
- Lawless, J. 1982. *Statistical Models and Methods for Lifetime Data*. New York: John Wiley & Sons.
- Lebanon, G. 2012. *Probability: The Analysis of Data, Volume 1*. <http://theanalysisofdata.com>.
- Levy, P. S. and S. Lemeshow. 1999. *Sampling of Populations*. Boston: John Wiley & Sons, third ed.
- Ligges, U., M. Maechler, and S. Schnackenberg. 2014. *scatterplot3d: 3D Scatter Plot*. <http://CRAN.R-project.org/package=scatterplot3d>. R package version 0.3-35.
- Lilliefors, H. W. 1967. On the Kolmokorov-Smirnov tests for normality with mean and variance unknown. *Journal of the American Statistical Association* 62:399–402.
- López, J. 1994. *Problemas de Inferencia Estadística, (Muestreo y Control de Calidad)*. Albacete, Spain: Tébar Flores, third ed.
- Lumley, T. 2009. *leaps: regression subset selection*. <http://CRAN.R-project.org/package=leaps>. R package version 2.9.

- Lunneborg, C. E. 2000. *Data Analysis by Resampling: Concepts and Applications*. Pacific Grove, CA: Duxbury, first ed.
- Maindonald, J. and J. Braun. 2003. *Data Analysis and Graphics Using R—An Example-Based Approach*, vol. 10 of Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press.
- Matloff, N. 2011. *The Art of R Programming: A Tour of Statistical Software Design*. San Francisco, CA: No Starch Press.
- Mazess, R. B., W. W. Pepperl, and M. Gibbons. 1984. Total body composition by dual-photon (^{153}gd) absorptiometry. *American Journal of Clinical Nutrition* 40, no. 4:834–839.
- McIlroy, D., R. Brownrigg, T. P. Minka, and R. Bivand. 2014. *mapproj: Map Projections*. <http://CRAN.R-project.org/package=mapproj>. R package version 1.2-2.
- Meyer, D., E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. 2014a. *e1071: Misc Functions of the Department of Statistics (e1071)*, TU Wien. <http://CRAN.R-project.org/package=e1071>. R package version 1.6-4.
- Meyer, D., A. Zeileis, and K. Hornik. 2014b. *vcd: Visualizing Categorical Data*. <http://CRAN.R-project.org/package=vcd>. R package version 1.3-2.
- Militino, A. F., M. D. Ugarte, and L. Garcia-Reinaldos. 2004. Alternative models for describing spatial dependence among dwelling selling prices. *The Journal of Real Estate Finance and Economics* 29, no. 2:193–209.
- Militino, A. F., M. D. Ugarte, T. Goicoa, and M. González-Audicana. 2006. Using small area models to estimate the total area occupied by olive trees. *Journal of Agricultural, Biological and Environmental Statistics* 11:450–461.
- Montgomery, D. C. 1991. *Design and Analysis of Experiments*. New York: John Wiley & Sons, third ed.
- Muro, J., I. Irigoyen, A. Militino, and C. Lamsfus. 2001. Defoliation effects on sunflower yield reduction. *Agronomy Journal* 93:634–637.
- Murrell, P. 2006. *R Graphics*. Computer Science and Data Analysis Series. Boca Raton, FL: Chapman & Hall/CRC.
- . 2011. *R graphics*. The R Series. Boca Raton, FL: CRC Press, second ed.
- Neter, J., M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. 1996. *Applied Linear Statistical Models*. Boston: McGraw-Hill, fourth ed.
- Oehlert, G. W. 2000. *A First Course in Design and Analysis of Experiments*. New York: W. H. Freeman and Company.
- Ott, L. and W. Mendenhall. 1985. *Understanding Stastistics*. Boston: Duxbury Press.
- Peña, D. 2001. *Fundamentos de Estadística*. Madrid: Alianza Editorial.
- . 2002. *Regresión y Diseño de Experimentos*. Madrid: Alianza Editorial.
- Pearson, K. 1900. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5* 157–175.
- Petersen, R. 1985. *Design and Analysis of Experiments*. New York: Marcel Dekker.

- Petrucelli, J. D., B. Nandram, and M. Chen. 1999. *Applied Statistics for Engineers and Scientists*. Upper Saddle River, NJ: Prentice Hall.
- Pinheiro, J., D. Bates, and R-core. 2015. *nlme: Linear and Nonlinear Mixed Effects Models*. <http://CRAN.R-project.org/package=nlme>. R package version 3.1-120.
- Pinheiro, J. C. and D. M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer-Verlag.
- Pratt, J. W. and J. D. Gibbons. 1981. *Concepts of Nonparametric Theory*. New York: Springer-Verlag.
- Qiu, Y. and Y. Xie. 2015. *highr: Syntax Highlighting for R Source Code*. <http://CRAN.R-project.org/package=highr>. R package version 0.5.
- R Core Team. 2015a. *foreign: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ...* <http://CRAN.R-project.org/package=foreign>. R package version 0.8-63.
- . 2015b. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org/>.
- Rao, C. R. 1973. *Linear Statistical Inference and Its Applications*. New York: John Wiley & Sons.
- Rinker, T. 2014. *reports: Assist the Workflow of Writing Academic Articles and Other Reports*. <http://CRAN.R-project.org/package=reports>. R package version 0.1.4.
- Ripley, B. 2015. *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. <http://CRAN.R-project.org/package=MASS>. R package version 7.3-40.
- Rizzo, M. 2008. *Statistical Computing with R*. Boca Raton, FL: Chapman & Hall/CRC.
- Ross, S. 1994. *A First Course in Probability*. Upper Saddle River, NJ: Prentice Hall, fifth ed.
- Rousseeuw, P. J. and B. C. V. Zomeren. 1990. Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association* 85, no. 411:633–639.
- Ruiz-Maya, L. 1986. *Métodos Estadísticos de Investigación*. Madrid: Editorial INE, second ed.
- Ruiz-Maya, L. and F. J. Martín Pliego. 2001. *Estadística. II Inferencia*. Madrid: Editorial AC, second ed.
- Ryan, T. P. 1997. *Modern Regression Methods*. New York: John Wiley & Sons.
- Sarkar, D. 2008. *Lattice: Multivariate Data Visualization with R*. New York: Springer. <http://lmdvr.r-forge.r-project.org>. ISBN 978-0-387-75968-5.
- . 2015. *lattice: Lattice Graphics*. <http://CRAN.R-project.org/package=lattice>. R package version 0.20-31.
- Sarkar, D. and F. Andrews. 2013. *latticeExtra: Extra Graphical Utilities Based on Lattice*. <http://CRAN.R-project.org/package=latticeExtra>. R package version 0.6-26.
- Scheaffer, R. L., W. Mendenhall, and L. Ott. 1986. *Elementos de Muestreo*. Mexico: Grupo Editorial Iberoamérica.
- Scott, D. W. 1992. *Multivariate Density Estimation: Theory, Practice, and Visualization*.

- New York: John Wiley & Sons.
- Seber, G. A. F. 1977. *Linear Regression Analysis*. New York: John Wiley & Sons.
- Selvin, S. 1998. *Modern Applied Biostatistical Methods Using S-PLUS*. New York: Oxford University Press.
- Shapiro, S. S. and M. B. Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52, no. 3 and 4:591–611.
- Sharpsteen, C. and C. Bracken. 2015. *tikzDevice: R Graphics Output in LaTeX Format*. <http://CRAN.R-project.org/package=tikzDevice>. R package version 0.8.1.
- Sheskin, D. J. 1997. *Parametric and Nonparametric Statistical Procedures*. New York: CRC Press.
- Singh, R. and S. N. 1996. *Elements of Survey Sampling*. Dordrecht, Netherlands: Kluwer Academic Publishers.
- Sokal, R. R. and F. J. Rohlf. 1994. *Biometry*. New York: W. H. Freeman, third ed.
- Spector, P. 2008. *Data Manipulation with R*. Use R! New York: Springer.
- Sternberg, D. E., D. P. Van Kammen, and W. E. Bunney. 1982. Schizophrenia: Dopamine β -hydroxylase activity and treatment response. *Science* 216:1423–1425.
- Suess, E. A. and B. E. Trumbo. 2010. *Introduction to Probability Simulation and Gibbs Sampling with R*. Use R! New York: Springer. <http://dx.doi.org/10.1007/978-0-387-68765-0>.
- Technology, T. and LLC. 2013. *shinyAce: Ace editor bindings for Shiny*. <http://CRAN.R-project.org/package=shinyAce>. R package version 0.1.0.
- Temple Lang, D. 2014. *RCurl: General network (HTTP/FTP/...) client interface for R*. <http://CRAN.R-project.org/package=RCurl>. R package version 1.95-4.5.
- Tibshirani, R. 2015. *bootstrap: Functions for the Book "An Introduction to the Bootstrap"*. <http://CRAN.R-project.org/package=bootstrap>. R package version 2015.2.
- Ugarte, M. D. and A. F. Militino. 2002. *Estadística Aplicada con S-PLUS*. Pamplona, Spain: Universidad Pública de Navarra, second ed.
- Ugarte, M. D., A. F. Militino, and A. T. Arnholt. 2008. *Probability and Statistics with R*. Boca Raton, FL: CRC Press.
- Venables, W. N. and B. D. Ripley. 1999. *Modern Applied Statistics with S-PLUS*. New York: Springer-Verlag, third ed.
- . 2000. *S Programming*. New York: Springer-Verlag.
- . 2002. *Modern Applied Statistics with S*. New York: Springer-Verlag, fourth ed. <http://www.stats.ox.ac.uk/pub/MASS4>.
- Verzani, J. 2005. *Using R for Introductory Statistics*. Boca Raton, FL: Chapman & Hall/CRC.
- Wand, M. 2015. *KernSmooth: Functions for Kernel Smoothing Supporting Wand & Jones (1995)*. <http://CRAN.R-project.org/package=KernSmooth>. R package version 2.23-14.
- Weindling, A. M., F. N. Bamford, and R. A. Whittall. 1986. Health of juvenile delinquents. *British Medical Journal* 292:447.

- Welch, B. L. 1951. On the comparison of several mean values: an alternative approach. *Biometrika* 38:330–336.
- Wickham, H. 2009. *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer. <http://had.co.nz/ggplot2/book>.
- . 2012. *gridExtra: Arrange grobs in tables*. <http://CRAN.R-project.org/package=gtable>. R package version 0.1.2.
- . 2014. *reshape2: Flexibly Reshape Data: A Reboot of the Reshape Package*. <http://CRAN.R-project.org/package=reshape2>. R package version 1.4.1.
- . 2015a. *httr: Tools for Working with URLs and HTTP*. <http://CRAN.R-project.org/package=httr>. R package version 0.6.1.
- . 2015b. *plyr: Tools for Splitting, Applying and Combining Data*. <http://CRAN.R-project.org/package=plyr>. R package version 1.8.2.
- Wickham, H. and W. Chang. 2015a. *devtools: Tools to Make Developing R Packages Easier*. <http://CRAN.R-project.org/package=devtools>. R package version 1.7.0.
- . 2015b. *ggplot2: An Implementation of the Grammar of Graphics*. <http://CRAN.R-project.org/package=ggplot2>. R package version 1.0.1.
- Wickham, H. and R. Francois. 2015. *dplyr: A Grammar of Data Manipulation*. <http://CRAN.R-project.org/package=dplyr>. R package version 0.4.1.
- Wickham, H., P. Danenberg, and M. Eugster. 2015. *roxygen2: In-Source Documentation for R*. <http://CRAN.R-project.org/package=roxygen2>. R package version 4.1.1.
- Wilcoxon, F. 1945. Individudal comparisons by ranking methods. *Biometrics Bulletin* 1, no. 6:80–83.
- Wilkinson, L. 2005. *The Grammar of Graphics (Statistics and Computing)*. Secaucus, NJ: Springer-Verlag.
- Xie, Y. 2014. *Dynamic Documents with R and knitr*. The R Series. Boca Raton, FL: Chapman & Hall/CRC.
- . 2015a. *formatR: Format R Code Automatically*. <http://CRAN.R-project.org/package=formatR>. R package version 1.2.
- . 2015b. *knitr: A General-Purpose Package for Dynamic Report Generation in R*. <http://CRAN.R-project.org/package=knitr>. R package version 1.10.
- Yandell, B. S. 1997. *Practical Data Analysis for Designed Experiments*. New York: Chapman & Hall.
- Yau, N. 2011. *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. Indianapolis, IN: John Wiley & Sons, Inc.
- . 2013. *Data Points: Visulazation That Means Something*. Indianapolis, IN: John Wiley & Sons, Inc.
- Zaman, A. 2000. The inconsistency of the Breusch-Pagan test. *Journal of Economic and Social Research* 2:1–11.
- Zeileis, A. and T. Hothorn. 2002. Diagnostic checking in regression relationships. *R News* 2, no. 3:7–10. <http://CRAN.R-project.org/doc/Rnews/>.

PROBABILITY and STATISTICS with R

Since the publication of the popular first edition, the contributed R packages on CRAN have increased from around 1,000 to over 6,000. **Probability and Statistics with R, Second Edition** explores how some of these new packages make analysis easier and more intuitive as well as create more visually pleasing graphs.

New to the Second Edition

- Improvements to existing examples, problems, concepts, data, and functions
- New examples and exercises that use the most modern functions
- Coverage probability of a confidence interval and model validation
- Highlighted R code for calculations and graph creation

Keeping pace with today's statistical landscape, this textbook expands your knowledge of the practice of statistics. It effectively links statistical concepts with R procedures, empowering you to solve a vast array of real statistical problems with R.

Features

- Presents extensive treatments of data analysis using parametric and nonparametric techniques, including bootstrap
- Explains the design of experiments and regression analysis
- Shows you how to use readily available data from numerous sites
- Contains many real-world examples, case studies, worked-out derivations, and detailed graphs that facilitate hands-on comprehension
- Includes end-of-chapter exercises that teach you how to solve problems by hand and using R
- Provides the PASWR2 package of data sets and functions on CRAN

María Dolores Ugarte is a professor of statistics in the Department of Statistics and Operations Research at the Public University of Navarre.

Ana F. Militino is a professor of statistics in the Department of Statistics and Operations Research at the Public University of Navarre.

Alan T. Arnholt is a professor in the Department of Mathematical Sciences at Appalachian State University.



CRC Press
Taylor & Francis Group
an informa business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
711 Third Avenue
New York, NY 10017
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

