Texts in Statistical Science

Analysis of Categorical Data with R

Christopher R. Bilder
Thomas M. Loughin

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Analysis of Categorical Data with R

CHAPMAN & HALL/CRC
Texts in Statistical Science Series

Statistical Theory: A Concise Introduction
F. Abramovich and Y. Ritov

Practical Multivariate Analysis, Fifth Edition
A. Afifi, S. May, and V.A. Clark

Practical Statistics for Medical Research
D.G. Altman

Interpreting Data: A First Course in Statistics
A.J.B. Anderson

Introduction to Probability with R
K. Baclawski

Linear Algebra and Matrix Analysis for Statistics
S. Banerjee and A. Roy

Analysis of Categorical Data with R
C. R. Bilder and T. M. Loughin

Statistical Methods for SPC and TQM
D. Bissell

Introduction to Probability
J. K. Blitzstein and J. Hwang

Bayesian Methods for Data Analysis, Third Edition
B.P. Carlin and T.A. Louis

Second Edition
R. Caulcutt

The Analysis of Time Series: An Introduction, Sixth Edition
C. Chatfield

Introduction to Multivariate Analysis
C. Chatfield and A.J. Collins

Problem Solving: A Statistician's Guide, Second Edition
C. Chatfield

Statistics for Technology: A Course in Applied Statistics, Third Edition
C. Chatfield

Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians
R. Christensen, W. Johnson, A. Branscum, and T.E. Hanson

Modelling Binary Data, Second Edition
D. Collett

Modelling Survival Data in Medical Research, Second Edition
D. Collett

Introduction to Statistical Methods for Clinical Trials
T.D. Cook and D.L. DeMets

Applied Statistics: Principles and Examples
D.R. Cox and E.J. Snell

Multivariate Survival Analysis and Competing Risks
M. Crowder

Statistical Analysis of Reliability Data
M.J. Crowder, A.C. Kimber, T.J. Sweeting, and R.L. Smith

An Introduction to Generalized Linear Models, Third Edition
A.J. Dobson and A.G. Barnett

Nonlinear Time Series: Theory, Methods, and Applications with R Examples
R. Douc, E. Moulines, and D.S. Stoffer

Introduction to Optimization Methods and Their Applications in Statistics
B.S. Everitt

Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models
J.J. Faraway

Linear Models with R, Second Edition
J.J. Faraway

A Course in Large Sample Theory
T.S. Ferguson

Multivariate Statistics: A Practical Approach
B. Flury and H. Riedwyl

Readings in Decision Analysis
S. French

Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition
D. Gamerman and H.F. Lopes

Bayesian Data Analysis, Third Edition
A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin

Multivariate Analysis of Variance and Repeated Measures: A Practical Approach for Behavioural Scientists
D.J. Hand and C.C. Taylor

Practical Data Analysis for Designed Practical Longitudinal Data Analysis
D.J. Hand and M. Crowder

Logistic Regression Models
J.M. Hilbe

Richly Parameterized Linear Models: Additive, Time Series, and Spatial Models Using Random Effects
J.S. Hodges

Statistics for Epidemiology
N.P. Jewell

Stochastic Processes: An Introduction, Second Edition
P.W. Jones and P. Smith

The Theory of Linear Models
B. Jørgensen

Principles of Uncertainty
J.B. Kadane

Graphics for Statistics and Data Analysis with R
K.J. Keen

Mathematical Statistics
K. Knight

Introduction to Multivariate Analysis: Linear and Nonlinear Modeling
S. Konishi

Nonparametric Methods in Statistics with SAS Applications
O. Korosteleva

Modeling and Analysis of Stochastic Systems, Second Edition
V.G. Kulkarni

Exercises and Solutions in Biostatistical Theory
L.L. Kupper, B.H. Neelon, and S.M. O'Brien

Exercises and Solutions in Statistical Theory
L.L. Kupper, B.H. Neelon, and S.M. O'Brien

Design and Analysis of Experiments with SAS
J. Lawson

A Course in Categorical Data Analysis
T. Leonard

Statistics for Accountants
S. Letchford

Introduction to the Theory of Statistical Inference
H. Liero and S. Zwanzig

Statistical Theory, Fourth Edition
B.W. Lindgren

Stationary Stochastic Processes: Theory and Applications
G. Lindgren

The BUGS Book: A Practical Introduction to Bayesian Analysis
D. Lunn, C. Jackson, N. Best, A. Thomas, and D. Spiegelhalter

Introduction to General and Generalized Linear Models
H. Madsen and P. Thyregod

Time Series Analysis
H. Madsen

Pólya Urn Models
H. Mahmoud

Randomization, Bootstrap and Monte Carlo Methods in Biology, Third Edition
B.F.J. Manly

Introduction to Randomized Controlled Clinical Trials, Second Edition
J.N.S. Matthews

Statistical Methods in Agriculture and Experimental Biology, Second Edition
R. Mead, R.N. Curnow, and A.M. Hasted

Statistics in Engineering: A Practical Approach
A.V. Metcalfe

Statistical Inference: An Integrated Approach, Second Edition
H. S. Migon, D. Gamerman, and F. Louzada

Beyond ANOVA: Basics of Applied Statistics
R.G. Miller, Jr.

A Primer on Linear Models
J.F. Monahan

Applied Stochastic Modelling, Second Edition
B.J.T. Morgan

Elements of Simulation
B.J.T. Morgan

Probability: Methods and Measurement
A. O'Hagan

Introduction to Statistical Limit Theory
A.M. Polansky

Applied Bayesian Forecasting and Time Series Analysis
A. Pole, M. West, and J. Harrison

Statistics in Research and Development, Time Series: Modeling, Computation, and Inference
R. Prado and M. West

Introduction to Statistical Process Control
P. Qiu

Sampling Methodologies with Applications
P.S.R.S. Rao

A First Course in Linear Model Theory
N. Ravishanker and D.K. Dey

Essential Statistics, Fourth Edition
D.A.G. Rees

Stochastic Modeling and Mathematical Statistics: A Text for Statisticians and Quantitative
F.J. Samaniego

Statistical Methods for Spatial Data Analysis
O. Schabenberger and C.A. Gotway

Bayesian Networks: With Examples in R
M. Scutari and J.-B. Denis

Large Sample Methods in Statistics
P.K. Sen and J. da Motta Singer

Decision Analysis: A Bayesian Approach
J.Q. Smith

Analysis of Failure and Survival Data
P. J. Smith

Applied Statistics: Handbook of GENSTAT Analyses
E.J. Snell and H. Simpson

Applied Nonparametric Statistical Methods, Fourth Edition
P. Sprent and N.C. Smeeton

Data Driven Statistical Methods
P. Sprent

Generalized Linear Mixed Models: Modern Concepts, Methods and Applications
W. W. Stroup

Survival Analysis Using S: Analysis of Time-to-Event Data
M. Tableman and J.S. Kim

Applied Categorical and Count Data Analysis
W. Tang, H. He, and X.M. Tu

Elementary Applications of Probability Theory, Second Edition
H.C. Tuckwell

Introduction to Statistical Inference and Its Applications with R
M.W. Trosset

Understanding Advanced Statistical Methods
P.H. Westfall and K.S.S. Henning

Statistical Process Control: Theory and Practice, Third Edition
G.B. Wetherill and D.W. Brown

Generalized Additive Models: An Introduction with R
S. Wood

Epidemiology: Study Design and Data Analysis, Third Edition
M. Woodward

Experiments
B.S. Yandell

Texts in Statistical Science

Analysis of Categorical Data with R

Christopher R. Bilder
University of Nebraska-Lincoln
Lincoln, Nebraska, USA

Thomas M. Loughin
Simon Fraser University
Surrey, British Columbia, Canada

CRC Press
Taylor & Francis Group
Boca Raton London New York

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

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

Version Date: 20140710

International Standard Book Number-13: 978-1-4987-0676-6 (eBook - PDF)

Contents

Preface

We live in a categorical world! From a positive or negative disease diagnosis to choosing all items that apply in a survey, outcomes are frequently organized into categories so that people can more easily make sense of them. However, analyzing data from categorical responses requires specialized techniques beyond those learned in a first or second course in Statistics. We offer this book to help students and researchers learn how to properly analyze categorical data. Unlike other texts on similar topics, our book is a modern account using the vastly popular R software. We use R not only as a data analysis method but also as a learning tool. For example, we use data simulation to help readers understand the underlying assumptions of a procedure and then to evaluate that procedure's performance. We also provide numerous graphical demonstrations of the features and properties of various analysis methods.

The focus of this book is on the analysis of data, rather than on the mathematical development of methods. We offer numerous examples from a wide rage of disciplines—medicine, psychology, sports, ecology, and others—and provide extensive R code and output as we work through the examples. We give detailed advice and guidelines regarding which procedures to use and why to use them. While we treat likelihood methods as a tool, they are not used blindly. For example, we write out likelihood functions and explain how they are maximized. We describe where Wald, likelihood ratio, and score procedures come from. However, except in Appendix B, where we give a general introduction to likelihood methods, we do not frequently emphasize calculus or carry out mathematical analysis in the text. The use of calculus is mostly from a conceptual focus, rather than a mathematical one.

We therefore expect that this book will appeal to all readers with a basic background in regression analysis. At times, a rudimentary understanding of derivatives, integrals, and function maximization would be helpful, as would a very basic understanding of matrices, matrix multiplication, and finding inverses of matrices. However, the important points and application advice can be easily understood without these tools. We expect that advanced undergraduates in statistics and related fields will satisfy these prerequisites. Graduate students in statistics, biostatistics, and related fields will certainly have sufficient background for the book. In addition, many students and researchers outside these disciplines who possess the basic regression background should find this book useful both for its descriptions and motivations of the analysis methods and for its worked examples with R code.

The book does not require any prior experience with R. We provide an introduction to the essential features and functions of R in Appendix A. We also provide introductory details on the use of R in the earlier chapters to help inexperienced R users. Throughout the book as new R functions are needed, their basic features are discussed in the text and their implementation shown with corresponding output. We focus on using R packages that are provided by default with the initial R installation. However, we make frequent use of other R packages when they are significantly better or contain functionality unavailable in the standard R packages. The book contains the code and output as it would appear in the R Console; we make minor modifications at times to the output only to save space within the book. Code provided in the book for plotting is often meant for color display rather than the actual black-and-white display shown in the print and some electronic editions.

The data set files and R programs that are referenced in each example are available from the book's website, `http://www.chrisbilder.com/categorical`. The programs include code used to create every plot and piece of output that we show. Many of these programs contain code to demonstrate additional features or to perform more detailed and complete analyses than what is presented in the book. We strongly recommend that the book and the website be used in tandem, both for teaching and for individual learning. The website also contains many "extras" that can help readers learn the material. Most importantly, we post videos from one of us teaching a course on the subject. These videos include live, in-class recordings that are synchronized with recordings of a tablet computer screen. Instructors may find these videos useful (as we have) for a blended or flipped classroom setting. Readers outside of a classroom setting may also find these videos especially useful as a substitute for a short-course on the subject.

The first four chapters of the book are organized by type of categorical response variable. Within each of these chapters, we first introduce the measurement type, followed by the basic distributional model that is most commonly used for that type of measurement. We slowly generalize to simple regression structures, followed by multiple regressions including transformations, interactions, and categorical explanatory variables. We conclude each of these chapters with some important special cases. Chapter 5 follows with model building and assessment methods for the response variables in the first four chapters. A final chapter discusses additional topics presented as extensions to the previous chapters. These topics include solutions to problems that are frequently mishandled in practice, such as how to incorporate diagnostic testing error into an analysis, the analysis of data from "choose all that apply" questions, and methods for analyzing data arising under a complex survey sampling design. Many of these topics are broad enough that entire books have been written about them, so our treatment in Chapter 6 is meant to be introductory.

For instructors teaching a one-semester course with the book, we recommend covering most of Chapters 1–5. The topics in Chapter 6 provide supplemental material for readers to learn on their own or to provide an instructor a means to go beyond the basics. In particular, topics from Chapter 6 can make good class projects. This helps students gain experience in teaching themselves extensions to familiar topics, which they will face later in industry or in research.

An extensive set of exercises is provided at the end of each chapter (over 65 pages in all!). The exercises are deliberately variable in scope and subject matter, so that instructors can choose those that meet the particular needs of their own students. For example, some carry out an analysis step by step, while others present a problem and leave the reader to choose and implement a solution. An answer key to the exercises is available for instructors using the book for a course. Details on how to obtain the answer key are available through the book's website.

We could not have written this book without the help and support of many people. First and foremost, we thank our families, and especially our wives, Kimberly and Marie, who put in extra effort on our behalf so that we could reserve time to work on the book. We owe them a huge debt for their support and tolerance, but we are hoping that they will settle for a nice dinner. We thank Rob Calver and his staff at CRC Press for their continued support and encouragement during the writing process. We also thank the hundreds of students who have taken categorical courses from us over the last seventeen years. Their feedback helped us to hone the course material and its presentation to what they are today. We especially thank one of our students, Natalie Koziol, who wrote the MRCV package used in Section 6.4 and made the implementation of those methods available to R users. This book was written in LaTeX through LyX, and we are grateful to the many contributors to these open-source projects. Finally, we need to thank our past and present colleagues and mentors at Iowa State, Kansas State, Oklahoma State, Nebraska, and Simon Fraser Universities who have

both supported our development and brought us interesting and challenging problems to work on.

Christopher R. Bilder and Thomas M. Loughin
Lincoln, NE and Surrey, BC

This page intentionally left blank

Chapter 1

Analyzing a binary response, part 1: introduction

Yes or no. Success or failure. Death or survival. For or against. Binary responses may be the most prevalent type of categorical data. The purpose of this chapter is to show how to estimate and make inferences about a binary response probability and related quantities. We begin in Section 1.1 by examining a homogeneous population where there is one overall probability to be estimated. We generalize this situation in Section 1.2 to the setting where sampled items come from one of two groups.

Throughout Chapter 1 we emphasize the use of R with detailed code explanations. This is done on purpose because we expect that some readers have little R experience beyond the introduction in Appendix A. Future chapters will still emphasize the use of R, but spend less time explaining code.

1.1 One binary variable

1.1.1 Bernoulli and binomial probability distributions

Almost every statistical analysis begins with some kind of statistical model. A statistical model generally takes the form of a probability distribution that attempts to quantify the uncertainty that comes with observing a new response. The model is intended to represent the unknown phenomenon that governs the observation process. At the same time, the model needs to be convenient to work with mathematically, so that inference procedures such as confidence intervals and hypothesis tests can be developed. Selecting a model is typically a compromise between two competing goals: providing a more detailed approximation to the process that generates the data and providing inference procedures that are easy to use.

In the case of binary responses, the natural model is the Bernoulli distribution. Let Y denote a Bernoulli random variable with outcomes of 0 and 1. Typically, we will say $Y = 1$ is a success and $Y = 0$ is a failure. For example, a success would be a basketball free throw attempt that is good or an individual who is cured of a disease by a new drug; a failure would be a free throw attempt that is missed or an individual who is not cured. We denote the probability of success as $P(Y = 1) = \pi$ and the corresponding probability of failure as $P(Y = 0) = 1 - \pi$. The Bernoulli probability mass function (PMF) for Y combines these two expressions into one formula:

$$P(Y = y) = \pi^y(1 - \pi)^{1-y}$$

for $y = 0$ or 1, where we use the standard convention that a capital letter Y denotes the random variable and the lowercase letter y denotes a possible value of Y. The expected value of Y is $E(Y) = \pi$, and the variance of Y is $Var(Y) = \pi(1 - \pi)$.

Often, one observes multiple Bernoulli random variable responses through repeated sampling or *trials* in identical settings. This leads to defining separate random variables for each trial, $Y_1, \ldots, Y_n$, where n is the number of trials. If all trials are identical and independent, we can treat $W = \sum_{i=1}^{n} Y_i$ as a binomial random variable with PMF of

$$P(W = w) = \binom{n}{w} \pi^w (1-\pi)^{n-w} \tag{1.1}$$

for $w = 0, \ldots, n$. The combination function $\binom{n}{w} = n!/[w!(n-w)!]$ counts the number of ways w successes and $n-w$ failures can be ordered. The expected value of W is $E(W) = n\pi$, and the variance of W is $Var(W) = n\pi(1-\pi)$. Notice that the Bernoulli distribution is a special case of the binomial distribution when $n = 1$.

We next show how R can be used to examine properties of the binomial distribution.

Example: Binomial distribution in R (Binomial.R)

The purpose of this example is to calculate simple probabilities using a binomial distribution and to show how these calculations are performed in R. We will be very basic with our use of R in this example. If you find its use difficult, we recommend reading Appendix A before proceeding further.

Consider a binomial random variable counting the number of successes from an experiment that is repeated $n = 5$ times, and suppose that there is a probability of success of $\pi = 0.6$. For example, suppose an individual has this success rate in a particular card game or shooting a basketball into a goal from a specific location. We can calculate the probability of each number of successes, $w = 0, 1, 2, 3, 4, 5$, using Equation 1.1. For example, the probability of 1 success out of 5 trials is

$$P(W = 1) = \binom{5}{1} 0.6^1 (1-0.6)^{5-1} = \frac{5!}{1!4!} 0.6^1 0.4^4 = 0.0768.$$

This calculation is performed in R using the `dbinom()` function:

```
> dbinom(x = 1, size = 5, prob = 0.6)
[1] 0.0768
```

Within the function, the `x` argument denotes the observed value of the binomial random variable (what we are calling w), the `size` argument is the number of trials (n), and the `prob` argument is π. We could have used `dbinom(1, 5, 0.6)` to obtain the same probability as long as the numerical values are in the same order as the arguments within the function (a full list of arguments and their order is available in the help for `dbinom()`). Generally, we will always specify the argument names in our code except with the most basic functions.

We find all of the probabilities for $w = 0, \ldots, 5$ by changing the `x` argument:

```
> dbinom(x = 0:5, size = 5, prob = 0.6)
[1] 0.01024 0.07680 0.23040 0.34560 0.25920 0.07776
```

where `0:5` means the integers 0 to 5 by 1. To display these probabilities in a more descriptive format, we save them into an object and print from a data frame using the `data.frame()` function:

```
> pmf <- dbinom(x = 0:5, size = 5, prob = 0.6)
> save <- data.frame(w = 0:5, prob = round(x = pmf, digits = 4))
> save
  w   prob
1 0 0.0102
2 1 0.0768
3 2 0.2304
4 3 0.3456
5 4 0.2592
6 5 0.0778
```

Note that we could have used different names than `pmf` and `save` for our objects if desired. The `round()` function rounds the values in the `pmf` object to 4 decimal places.

We plot the PMF using the `plot()` and `abline()` functions:

```
> plot(x = save$w, y = save$prob, type = "h", xlab = "w", ylab =
    "P(W=w)", main = "Plot of a binomial PMF for n=5, pi=0.6",
    panel.first = grid(col = "gray", lty = "dotted"), lwd = 3)
> abline(h = 0)
```

Figure 1.1 gives the resulting plot. Appendix A.7.2 describes most of the arguments within `plot()`, so we provide only brief descriptions here. The `x` and `y` arguments specify the x- and y-axis values where we use the `$` symbol to access parts of the `save` data frame. The `type = "h"` argument value specifies that vertical bars are to be plotted from 0 to the values given in the `y` argument. The `main` argument contains the plot title.[1] The `abline()` function plots a horizontal line at 0 to emphasize the bottom of each vertical line.

The simpler specification `plot(x = save$w, y = save$prob, type = "h")` produces a plot similar to Figure 1.1, but our extra arguments make the plot easier to interpret.

Assumptions

The binomial distribution is a reasonable model for the distribution of successes in a given number of trials as long as the process of observing repeated trials satisfies certain assumptions. Those assumptions are:

1. THERE ARE n IDENTICAL TRIALS. This refers to the process by which the trials are conducted. The action that results in the trial and the measurement taken must be the same in each trial. The trials cannot be a mixture of different types of actions or measurements.

2. THERE ARE TWO POSSIBLE OUTCOMES FOR EACH TRIAL. This is generally just a matter of knowing what is measured. However, there are cases where a response measurement has more than two levels, but interest lies only in whether or not one particular level occurs. In this case, the special level can be considered "success" and all remaining levels "failure."

[1] If it is desired to use a plot title with better notation, we could have used `main = expression(paste("Plot of a binomial PMF for ", italic(n) == 5, " and ", italic(pi) == 0.6))` to obtain "Plot of the binomial PMF for $n = 6$ and $\pi = 0.6$." In this code, `expression()` allows us to include mathematical symbols and `paste()` combines these symbols with regular text. Please see Appendix A.7.4 for more information.

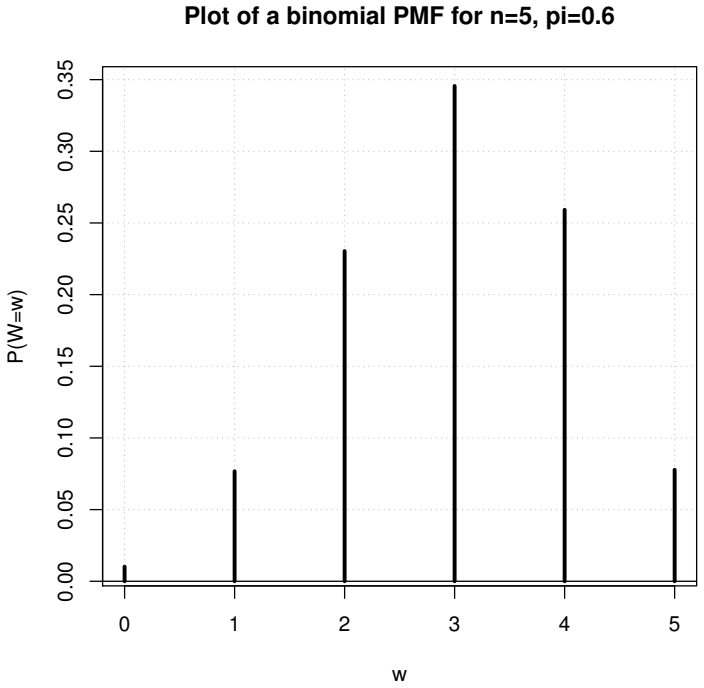

Figure 1.1: PMF for W.

3. The trials are independent of each other. In particular, there is nothing in the conduct of the trials that would cause any subset of trials to behave more similarly to one another. Counterexamples include (a) measuring trials in "clusters," where the units on which success or failure is measured are grouped together somehow before observation, and (b) trials measured in a time series, where trials measured close together in time might react more similarly than those measured far apart.

4. The probability of success remains constant for each trial. This means that all variables that can affect the probability of success need to be held constant from trial to trial. Because these variables are not always known in advance, this can be a very difficult condition to confirm. We often can only confirm that the "obvious" variables are not changing, and then merely assume that others are not as well.

5. The random variable of interest W is the number of successes. Specifically, this implies that we are *not* interested in the order in which successes and failures occur, but rather only in their total counts.

The next two examples detail these assumptions with respect to applications, and they demonstrate how it can be difficult to assure that these assumptions are satisfied.

Example: Field goal kicking

In American and Canadian football, points can be scored by kicking a ball through a target area (goal) at each end of the field. Suppose an experiment is conducted where a placekicker successively attempts five field goal kicks during practice. A success occurs on one trial when the football is kicked over the crossbar and between the two

uprights of the goal posts. A failure occurs when the football does not achieve this result (the football is kicked to the left or right of both uprights or falls short of the crossbar). We want to use these results to estimate the placekicker's true probability of success, so we record how many kicks are successful.

In order to use the binomial distribution here, the experiment needs to satisfy the following conditions:

1. THERE ARE n IDENTICAL TRIALS. In this case, $n = 5$ field goals are attempted. The action is always kicking a football in the same way, and the measurement of success is made the same way each time.

2. THERE ARE TWO POSSIBLE OUTCOMES FOR EACH TRIAL. Each field goal is a success or failure. Notice that we could further divide failures into other categories: too short, long enough but to the left of the uprights, long enough but to the right of the uprights, or some combinations of these. If our interest is only in whether the kick is successful, then differentiating among the modes of failure is not necessary.

3. THE TRIALS ARE INDEPENDENT OF EACH OTHER. Given that the field goals are attempted successively, this may be difficult to satisfy. For example, if one field goal attempt is missed to the left, the kicker may compensate by trying to kick farther to the right on the next attempt. On the other hand, the independence assumption may be approximately satisfied by a placekicker who tries to apply the exact same technique on each trial.

4. THE PROBABILITY OF SUCCESS REMAINS CONSTANT FOR EACH TRIAL. To make sure this is true, the field goals need to be attempted from the same distance under the same surrounding conditions. Weather conditions need to be constant. The same kicker, ball, and goalposts are used each time. We assume that fatigue does not affect the kicker for this small number of attempts. If the attempts occur close together in time, then it may be reasonable to assume that extraneous factors are reasonably constant as well, at least enough so that they do not have a substantial effect on the success of the field goals.

5. THE RANDOM VARIABLE OF INTEREST W IS THE NUMBER OF SUCCESSES. As we will see in Section 1.1.2, in order to estimate the probability of success, we need only to record W (=0, 1, 2, 3, 4, or 5) and not the entire sequence of trial results.

Example: Disease prevalence

The *prevalence* of a disease is the proportion of a population that is afflicted with that disease. This is equivalent to the probability that a randomly selected member of the population has the disease. Many public health studies are performed to understand disease prevalence, because knowing the prevalence is the first step toward solving societal problems caused by the disease. For example, suppose there is concern that a new infectious disease may be transmitted to individuals through blood donations. One way to examine the disease prevalence would be to take a sample a 1,000 blood donations and test each for the disease.

In order to use the binomial distribution here, this setting needs to satisfy the following conditions:

1. THERE ARE n IDENTICAL TRIALS. In this case, $n = 1000$ blood donations are examined. Each blood donation needs to be collected and tested the same way. In

particular, trials would *not* be identical if different diagnostic measures were used on different donations to determine presence of disease.

2. THERE ARE TWO POSSIBLE OUTCOMES FOR EACH TRIAL. Each blood donation is either positive or negative for the disease. Making this determination is not necessarily as straightforward as it may seem. Often, there is a continuous score reported from the results of an assay, such as the amount of an antigen in a specimen, and a threshold or cut-off point is used to make the positive or negative determination. In some instances, multiple thresholds may be used leading to responses such as positive, indeterminate, or negative.

3. THE TRIALS ARE INDEPENDENT OF EACH OTHER. This may be difficult to satisfy completely. For example, if married spouses are included in the sample, then presence of the disease in one spouse's donation may suggest a greater chance that the other spouse's donation will also test positive. Independence can be assured by random sampling from a large population of donations, but may always be in question when any non-random sampling method is used.

4. THE PROBABILITY OF SUCCESS REMAINS CONSTANT FOR EACH TRIAL. Each sampled donation needs to have the same probability of having the disease. This could be unreasonable if there are factors, such as risky behavior, that make donations for certain subpopulations more likely to have the disease than others, and if these subpopulations can be identified in advance. Similarly, if donations are collected over an extended period of time, prevalence may not be constant during the period.

5. THE RANDOM VARIABLE OF INTEREST W IS THE NUMBER OF SUCCESSES. There are $W = 0, \ldots, 1000$ possible positive blood donations. To estimate prevalence, we need to know how many positive donations there are, and not which ones are positive.

The previous examples show that it may be difficult to satisfy all of the assumptions for a binomial model. However, the binomial model may still be used as an approximation to the true model in a given problem, in which case the violated assumptions then would need to be identified in any stated results. Alternatively, if assumptions are not satisfied, there are other models and procedures that can be used to analyze binary responses. In particular, if the probability of success does not remain constant for each trial—for example, if disease probability is related to certain risky behaviors—we may be able to identify and measure the factors causing the variations and then use a regression model for the probability of success (to be discussed in Chapter 2).

Simulating a binomial sample

What does a sample from a binomial distribution look like? Of course, the observed values only can be $0, 1, \ldots, n$. The proportion of observed values that are $0, 1, \ldots, n$ are governed by the PMF and the parameter π within it. The mean and variance of these observed values are also controlled by the PMF and π. These properties are easily derived mathematically using basic definitions of mean and variance. In more complex problems, however, properties of statistics are much harder to derive mathematically.

We show in the next example how to *simulate* a sample using R so that we can check whether theory matches what actually happens. This example will also introduce Monte Carlo computer simulation as a valuable tool for evaluating a statistical procedure. All statistical procedures have assumptions underlying their mathematical framework. Monte

Carlo simulation is especially useful in assessing how well procedures perform when these assumptions are violated. For example, we may want to know if a confidence interval that is designed to work in large samples maintains its stated confidence level when the sample size is small.

A Monte Carlo simulation works by creating a computer version of the population we are studying, sampling from this virtual population in a prescribed way, performing the statistical analysis that we are studying, and measuring how it performs. The details of each step vary from one problem to the next. In all cases we draw a "large" number of samples from the virtual population. In so doing, the *law of large numbers* assures us that the average performance measured across the samples will be close to the true performance of the procedure in this context (a more mathematical definition of the law of large numbers is contained on p. 232 of Casella and Berger, 2002). We follow this prescription in the example below.

Example: Simulation with the binomial distribution in R (Binomial.R)

Below is the code that simulates 1,000 random observations of W from a binomial distribution with $\pi = 0.6$ and $n = 5$:

```
> set.seed(4848)
> bin5 <- rbinom(n = 1000, size = 5, prob = 0.6)
> bin5[1:10]
[1] 3 2 4 1 3 1 3 3 3 4
```

The `set.seed()` function sets a seed number for the simulated observations. Without going into the details behind random number generation, the seed number specification allows us to obtain identical simulated observations each time the same code is executed.[2] The `rbinom()` function simulates the observations, where the `n` argument gives the number of observations (not n as in the number of trials). The `bin5` object contains 1,000 values, where the first 10 of which are printed.

The population mean and variance for W are

$$E(W) = n\pi = 5 \times 0.6 = 3$$

and

$$Var(W) = n\pi(1-\pi) = 5 \times 0.6 \times 0.4 = 1.2.$$

We calculate the sample mean and variance of the simulated observations using the `mean()` and `var()` functions:

```
> mean(bin5)
[1] 2.991
> var(bin5)
[1] 1.236155
```

[2]It is best to not use the same seed number when doing other simulated data examples. A new seed number can be chosen from a random number table. Alternatively, a new seed number can be found by running `runif(n=1)` within R (this simulates one observation from a Uniform(0,1) distribution) and taking the first few significant digits.

The sample mean and variance are very similar to $E(W)$ and $Var(W)$, as expected. If a larger number of observations were simulated, say 10,000, we generally would expect these sample measures to be even closer to their population quantities due to the law of large numbers.

To examine how well the observed frequency distribution follows the PMF, we use `table()` to find the frequencies of each possible response and then use `hist()` to plot a histogram of the relative frequencies:

```
> table(x = bin5)
x
0   1   2   3   4   5
12  84 215 362 244  83
> hist(x = bin5, main = "Binomial with n=5, pi=0.6, 1000 bin.
    observations", probability = TRUE, breaks = c(-0.5:5.5), ylab
    = "Relative frequency")
> -0.5:5.5
[1] -0.5  0.5  1.5  2.5  3.5  4.5  5.5
```

For example, we see that $w = 3$ is observed with a relative frequency of $362/1000 = 0.362$. We had found earlier that $P(W = 3) = 0.3456$, which is very similar to the observed proportion. The histogram is in Figure 1.2, and its shape is very similar to the PMF plot in Figure 1.1. Note that the `probability = TRUE` argument gives the relative frequencies (`probability = FALSE` gives the frequencies, which is the default), and the `breaks` argument specifies the classes for the bars to be -0.5 to 5.5 by 1 (the bars will not be drawn correctly here without specifying `breaks`).

1.1.2 Inference for the probability of success

The purpose of this section is to estimate and make inferences about the probability of success parameter π from the Bernoulli distribution. We start by estimating the parameter using its maximum likelihood estimate, because it is relatively easy to compute and has properties that make it appealing in large samples. Next, confidence intervals for the true probability of success are presented and compared. Many different confidence intervals have been proposed in the statistics literature. We will present the simplest—but worst—procedure first and then offer several better alternatives. We conclude this section with hypothesis tests for π.

For those readers unfamiliar with estimation and inference procedures associated with the likelihood function, we encourage you to read Appendix B first for an introduction. We will reference specific parts of the appendix within this section.

Maximum likelihood estimation and inference

As described in Appendix B, a likelihood function is a function of one or more parameters conditional on the observed data. The likelihood function for π when $y_1, \ldots, y_n$ are observations from a Bernoulli distribution is

$$\begin{aligned} L(\pi|y_1, \ldots, y_n) &= P(Y_1 = y_1) \times \cdots \times P(Y_n = y_n) \\ &= \pi^w (1-\pi)^{n-w}. \end{aligned} \tag{1.2}$$

Alternatively, when we only record the number of successes out of a number of trials, the likelihood function for π is simply $L(\pi|w) = P(W = w) = \binom{n}{w} \pi^w (1-\pi)^{n-w}$. The value of

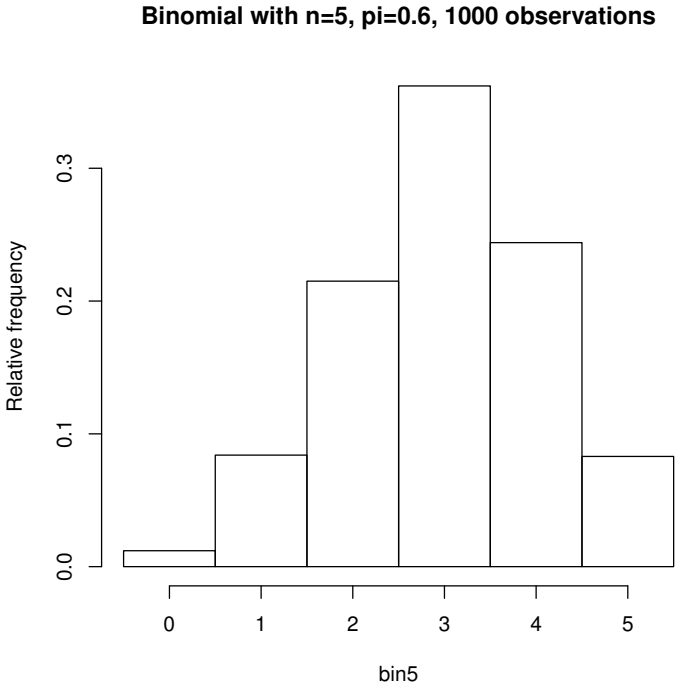

Figure 1.2: Histogram for the observed values of w.

π that maximizes the likelihood function is considered to be the most plausible value for the parameter, and it is called the maximum likelihood estimate (MLE). We show in Appendix B.3 that the MLE of π is $\hat{\pi} = w/n$, which is simply the observed proportion of successes. This is true for both $L(\pi|y_1, \ldots, y_n)$ or $L(\pi|w)$ because the $\binom{n}{w}$ contains no information about π.

Because $\hat{\pi}$ will vary from sample to sample, it is a statistic and has a corresponding probability distribution.[3] As with all MLEs, $\hat{\pi}$ has an approximate normal distribution in a large sample (see Appendix B.3.3). The mean of the normal distribution is π, and the variance is found from

$$\begin{aligned}
\widehat{Var}(\hat{\pi}) &= -E\left\{\frac{\partial^2 \log\left[L(\pi|W)\right]}{\partial \pi^2}\right\}^{-1}\Bigg|_{\pi=\hat{\pi}} \\
&= -E\left\{-\frac{W}{\pi^2} + \frac{n-W}{(1-\pi)^2}\right\}^{-1}\Bigg|_{\pi=\hat{\pi}} \\
&= \left[\frac{n}{\pi} - \frac{n}{1-\pi}\right]^{-1}\Bigg|_{\pi=\hat{\pi}} \\
&= \frac{\hat{\pi}(1-\hat{\pi})}{n}. \qquad (1.3)
\end{aligned}$$

[3] In order for $\hat{\pi}$ to have a probability distribution, it needs to be a random variable. Thus, we are actually using $\hat{\pi} = W/n$ in this case. We could have defined W/n as $\hat{\Pi}$ instead, but this level of formality is unnecessary here. It will be apparent from a statistic's use whether it is a random or observed quantity.

where $\log(\cdot)$ is the natural log function (see Appendix B.3.4). We can write the distribution as $\hat{\pi} \dot{\sim} N(\pi, \widehat{Var}(\hat{\pi}))$ where $\dot{\sim}$ denotes "approximately distributed as." The approximation tends to be better as the sample size grows larger.

Wald confidence interval

Using this normal distribution, we can treat $(\hat{\pi} - \pi)/\widehat{Var}(\hat{\pi})^{1/2}$ as an approximate standard normal quantity (see Appendix B.5). Thus, for any $0 < \alpha < 1$, we have

$$P\left(Z_{\alpha/2} < \frac{\hat{\pi} - \pi}{\sqrt{\widehat{Var}(\hat{\pi})}} < Z_{1-\alpha/2}\right) \approx 1 - \alpha,$$

where Z_a is the a^{th} quantile from a standard normal distribution (e.g., $Z_{0.975}$= 1.96). After rearranging terms and recognizing that $-Z_{\alpha/2} = Z_{1-\alpha/2}$, we obtain

$$P\left(\hat{\pi} - Z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\pi})} < \pi < \hat{\pi} + Z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\pi})}\right) \approx 1 - \alpha.$$

Now, we have an approximate probability that has the parameter π centered between two statistics. When we replace $\hat{\pi}$ and $\widehat{Var}(\hat{\pi})$ with observed values from the sample, we obtain the $(1 - \alpha)100\%$ confidence interval for π

$$\hat{\pi} - Z_{1-\alpha/2}\sqrt{\hat{\pi}(1-\hat{\pi})/n} < \pi < \hat{\pi} + Z_{1-\alpha/2}\sqrt{\hat{\pi}(1-\hat{\pi})/n}.$$

This is the usual interval for a probability of success that is given in most introductory statistics textbooks. Confidence intervals based on the approximate normality of MLEs are called "Wald confidence intervals" because Wald (1943) was the first to show this property of MLEs in large samples.

When w is close to 0 or n, two problems occur with this interval:

1. Calculated limits may be less than 0 or greater than 1, which is outside the boundaries for a probability.
2. When $w = 0$ or 1, $\sqrt{\hat{\pi}(1-\hat{\pi})/n} = 0$ for $n > 0$. This leads to the lower and upper limits to be exactly the same (0 for $w = 0$ or 1 for $w = 1$).

We will discuss additional problems with the Wald interval shortly.

Example: Wald interval (CIpi.R)

Suppose $w = 4$ successes are observed out of $n = 10$ trials. The 95% Wald confidence interval for π is $0.4 \pm 1.96\sqrt{0.4(1-0.4)/10} = (0.0964,\ 0.7036)$, where we use the shorthand notation within parentheses to mean $0.0964 < \pi < 0.7036$. The R code below shows how these calculations are carried out:

```
> w <- 4
> n <- 10
> alpha <- 0.05
> pi.hat <- w/n
> var.wald <- pi.hat*(1-pi.hat)/n
> lower <- pi.hat - qnorm(p = 1-alpha/2) * sqrt(var.wald)
> upper <- pi.hat + qnorm(p = 1-alpha/2) * sqrt(var.wald)
> round(data.frame(lower, upper), 4)
   lower  upper
1 0.0964 0.7036
```

In the code, we use the `qnorm()` function to find the $1 - \alpha/2$ quantile from a standard normal distribution. We can calculate the interval quicker by taking advantage of how R performs vector calculations:

```
> round(pi.hat + qnorm(p = c(alpha/2, 1-alpha/2)) *
    sqrt(var.wald), 4)
[1] 0.0964 0.7036
```

See Appendix A.4 for a similar example.

The confidence interval is quite wide and may not be meaningful for some applications. However, it does give information on a range for π that may be useful in hypothesis testing situations. For example, a test of $H_0 : \pi = 0.5$ vs. $H_a : \pi \neq 0.5$ would not reject H_0 because 0.5 is within this range. If the test was instead $H_0 : \pi = 0.8$ vs. $H_a : \pi \neq 0.8$, there is evidence to reject the null hypothesis. Other ways to perform these tests with a test statistic and a p-value will be discussed shortly.

The inferences for π from the Wald confidence interval rely on the underlying normal distribution approximation for the maximum likelihood estimator. For this approximation to work well, we need a large sample, and, unfortunately, the sample size in the last example was quite small. Furthermore, notice that $\hat{\pi}$ can take on only 11 different possible values in the last example: 0/10, 1/10, ..., 10/10, but a normal distribution is a continuous function. These problems lead the Wald conference interval to be *approximate*, in the sense that the probability that the interval covers the parameter (its *coverage* or *true confidence level*) is not necessarily equal to the stated level $1 - \alpha$. The quality of the approximation varies with n and π, and as we will see later, the Wald interval generally has coverage $< 1 - \alpha$. Such an interval is called a *liberal* interval. On the other hand, an interval with coverage in excess of the stated level is called *conservative*. While this latter property may seem to be a good quality, it can lead to intervals that are quite wide in comparison to others. We want confidence intervals that place the parameter within as narrow a range as possible, while maintaining at least the stated confidence level. If we wanted intervals that had greater coverage, we would have stated a higher confidence level!

There has been a lot of research on finding an interval like this for π, including Agresti and Caffo (2000), Agresti and Min (2001), Borkowf (2006), Brown et al. (2002), Henderson and Meyer (2001), Newcombe (2001), Suess et al. (2006), and Vos and Hudson (2005). Brown et al. (2001) present a thorough review of most competing intervals. We present their recommendations next along with our own thoughts on the best intervals.

Wilson confidence interval

When $n < 40$, Brown et al. (2001) recommend the Wilson interval or the Jeffreys interval because they maintain true confidence levels closer to the stated level than other intervals. The Wilson interval formula is found by examining the test statistic

$$Z_0 = \frac{\hat{\pi} - \pi_0}{\sqrt{\pi_0(1 - \pi_0)/n}},$$

which is a *score test* statistic often used for a test of $H_0 : \pi = \pi_0$ vs. $H_a : \pi \neq \pi_0$, where $0 < \pi_0 < 1$ (see Appendix B.5). The variance in the denominator of Z_0 is computed assuming that the null hypothesis is true, rather than using the unrestricted estimate based on the data. This leads to the advantage that the denominator is not 0 whenever $w = 0$ or n. We can approximate the distribution of Z_0 with a standard normal to obtain $P(-Z_{1-\alpha/2} <$

$Z_0 < Z_{1-\alpha/2}) \approx 1-\alpha$. Treating the approximation as an equality, the Wilson interval contains the set of all possible values of π_0 that satisfy the equation. Conversely, the set of all possible values for π_0 that lead to a rejection of the null hypothesis are outside of the confidence interval. The process of forming an interval from a hypothesis test procedure like this is often referred to as "inverting the test." See also Appendix B.5.2. Because the Wilson interval is based on a score test, it is often referred to as a *score interval* too.

The interval endpoints are found by setting Z_0 equal to $\pm Z_{1-\alpha/2}$, and applying the quadratic formula to solve for π_0. Thus, the $(1-\alpha)100\%$ Wilson interval is

$$\tilde{\pi} \pm \frac{Z_{1-\alpha/2}\sqrt{n}}{n+Z^2_{1-\alpha/2}}\sqrt{\hat{\pi}(1-\hat{\pi})+\frac{Z^2_{1-\alpha/2}}{4n}}, \tag{1.4}$$

where

$$\tilde{\pi} = \frac{w + Z^2_{1-\alpha/2}/2}{n+Z^2_{1-\alpha/2}}$$

can be thought of as an adjusted estimate of π. This interval is named after Wilson (1927) who first proposed finding an interval for π in this manner. Note that the Wilson interval always has limits between 0 and 1.

The Wald and Wilson confidence intervals discussed so far are *frequentist* inference procedures. The "confidence" associated with these types of inference procedures comes about through repeating the process of taking a sample and calculating a confidence interval each time. This leads to the interpretation of

> We would expect $(1-\alpha)100\%$ of all similarly constructed intervals to contain the parameter π

for a $(1-\alpha)100\%$ confidence interval. Alternatively, a commonly used interpretation is

> With $(1-\alpha)100\%$ confidence, the parameter π is between <lower limit> and <upper limit>

where the appropriate lower and upper limits are inserted. Note that a single interval calculated from a sample either does or does not contain the parameter, so it is inappropriate to say it has a *probability* (other than 0 or 1) of containing the parameter. This is a confusing aspect to confidence intervals, causing them to be misinterpreted frequently in practice.

On the other hand, a Bayesian credible interval does have a $(1-\alpha)100\%$ probability of containing the parameter, because parameters are random variables in the Bayesian paradigm. A Jeffreys interval for π, also recommended by Brown et al. (2001), is one such Bayesian interval. We postpone its discussion until Section 6.6, when we describe Bayesian inference procedures in detail.

Agresti-Coull confidence interval

The Wilson interval is our preferred choice for a confidence interval for π. However, Brown et al. (2001) recommend the Agresti-Coull interval (Agresti and Coull 1998) for $n \geq 40$, primarily because it is a little easier to calculate by hand and it more closely resembles the popular Wald interval. The $(1-\alpha)100\%$ Agresti-Coull interval is

$$\tilde{\pi} - Z_{1-\alpha/2}\sqrt{\frac{\tilde{\pi}(1-\tilde{\pi})}{n+Z^2_{1-\alpha/2}}} < \pi < \tilde{\pi} + Z_{1-\alpha/2}\sqrt{\frac{\tilde{\pi}(1-\tilde{\pi})}{n+Z^2_{1-\alpha/2}}}.$$

The interval is essentially the Wald interval where $Z^2_{1-\alpha/2}/2$ successes and $Z^2_{1-\alpha/2}/2$ failures are added to the observed data. Specifically, for $\alpha = 0.05$, this means that about two

successes and two failures are added because $Z_{1-0.05/2} = 1.96 \approx 2$. Similar to the Wald interval, this interval has the undesirable property that it may have limits less than 0 or greater than 1.

Example: Wilson and Agresti-Coull intervals (CIpi.R)

Suppose again $w = 4$ successes are observed out of $n = 10$ trials. For a 95% confidence interval, the adjusted estimate of π is

$$\tilde{\pi} = \frac{w + Z^2_{1-\alpha/2}/2}{n + Z^2_{1-\alpha/2}} = \frac{4 + 1.96^2/2}{10 + 1.96^2} = 0.4278.$$

The 95% Wilson interval limits are

$$\tilde{\pi} \pm \frac{Z_{1-\alpha/2}\sqrt{n}}{n + Z^2_{1-\alpha/2}} \sqrt{\hat{\pi}(1-\hat{\pi}) + \frac{Z^2_{1-\alpha/2}}{4n}}$$
$$= 0.4278 \pm \frac{1.96\sqrt{10}}{10 + 1.96^2} \sqrt{0.4(1-0.4) + \frac{1.96^2}{4 \times 10}}$$

leading to an interval of $0.1682 < \pi < 0.6873$. The 95% Agresti-Coull interval limits are

$$\tilde{\pi} \pm Z_{1-\alpha/2} \sqrt{\frac{\tilde{\pi}(1-\tilde{\pi})}{n + Z^2_{1-\alpha/2}}} = 0.4278 \pm 1.96 \sqrt{\frac{0.4278(1-0.4278)}{10 + 1.96^2}}$$

leading to an interval of $0.1671 < \pi < 0.6884$. Both confidence intervals have limits that are quite similar in this case, but are rather different from the Wald interval limits of (0.0964, 0.7036) that we calculated earlier.

Continuing from the last example, below is how the calculations are performed in R:

```
> p.tilde <- (w + qnorm(p = 1-alpha/2)^2 / 2) / (n + qnorm(p =
    1-alpha/2)^2)
> p.tilde
[1] 0.4277533

> # Wilson C.I.
> round(p.tilde + qnorm(p = c(alpha/2, 1-alpha/2)) * sqrt(n) / (n
    + qnorm(p = 1-alpha/2)^2) * sqrt(pi.hat*(1-pi.hat) + qnorm(p =
    1-alpha/2)^2/(4*n)), 4)
[1] 0.1682 0.6873

> # Agresti-Coull C.I.
> var.ac <- p.tilde*(1-p.tilde) / (n + qnorm(p = 1-alpha/2)^2)
> round(p.tilde + qnorm(p = c(alpha/2, 1-alpha/2)) *
    sqrt(var.ac), 4)
[1] 0.1671 0.6884
```

After calculating $\tilde{\pi}$, we calculate the Wilson and Agresti-Coull intervals through one line of code for each. Note that executing part of a line of code can help highlight how it works. For example, one can execute `qnorm(p = c(alpha/2, 1-alpha/2))` to see that it calculates -1.96 and 1.96.

The `binom.confint()` function from the `binom` package can be used to simplify the calculations. Note that this package is not in the default installation of R, so it needs to be installed before its use (see Appendix A.5 for more information regarding package installation). Below is our use of the function:

```
> library(package = binom)
> binom.confint(x = w, n = n, conf.level = 1-alpha, methods =
    "all")
           method x  n      mean      lower     upper
1  agresti-coull 4 10 0.4000000 0.16711063 0.6883959
2     asymptotic 4 10 0.4000000 0.09636369 0.7036363
3          bayes 4 10 0.4090909 0.15306710 0.6963205
4         cloglog 4 10 0.4000000 0.12269317 0.6702046
5          exact 4 10 0.4000000 0.12155226 0.7376219
6          logit 4 10 0.4000000 0.15834201 0.7025951
7         probit 4 10 0.4000000 0.14933907 0.7028372
8        profile 4 10 0.4000000 0.14570633 0.6999845
9            lrt 4 10 0.4000000 0.14564246 0.7000216
10     prop.test 4 10 0.4000000 0.13693056 0.7263303
11        wilson 4 10 0.4000000 0.16818033 0.6873262
```

The function calculates 11 different intervals for π when the `methods = "all"` argument is used. The first, second, and eleventh intervals are the Agresti-Coull, Wald, and Wilson intervals, respectively. Please see the help for the function for more information on the other intervals. The end-of-chapter exercises discuss some of these in more detail.

Clopper-Pearson confidence interval

The Clopper-Pearson interval (Clopper and Pearson, 1934) is the last confidence interval for π that we will discuss in this section. While Brown et al. (2001) remark that the interval is "wastefully conservative and it is not a good choice for practical use," this interval does have a unique property that the other intervals do not: the true confidence level is always equal to or greater than the stated level. In order to achieve this true confidence level, the interval is usually wider than most other intervals for π.

The interval uses the relationship between the binomial distribution and the beta distribution to achieve its conservative confidence level (see #2.40 on p. 82 of Casella and Berger, 2002 for this distributional relationship). In fact, because the actual or *exact* distribution for W is used, the interval is called an *exact inference* procedure. There are many other exact inference procedures available for statistical problems, and some of these are discussed in Section 6.2.

To review the beta distribution, let V be a beta random variable. The probability density function (PDF) for V is

$$f(v;a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} v^{a-1}(1-v)^{b-1},\ 0 < v < 1 \tag{1.5}$$

where $a > 0$ and $b > 0$ are parameters and $\Gamma(\cdot)$ is the gamma function, $\Gamma(c) = \int_0^\infty x^{c-1}e^{-x}dx$ for $c > 0$. Note that $\Gamma(c) = (c-1)!$ for an integer c. The a and b parameters control the shape of the distribution. The distribution is right-skewed for $a > b$, and the distribution is left-skewed for $a < b$. When $a = b$, the distribution is symmetric about $v = 0.5$. Our

program Beta.R gives a few example plots of the distribution. The α quantile of a beta distribution, denoted by v_α or beta$(\alpha; a, b)$, is found by solving

$$\alpha = \int_0^{v_\alpha} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} v^{a-1}(1-v)^{b-1} dv$$

for v_α.

The $(1-\alpha)100\%$ Clopper-Pearson interval is simply quantiles from two beta distributions:

$$\text{beta}\left(\alpha/2;\ w,\ n-w+1\right) < \pi < \text{beta}\left(1-\alpha/2;\ w+1,\ n-w\right).$$

Due to the restriction $a > 0$, the lower endpoint cannot be computed if $w = 0$. In that case, the lower limit is taken to be 0. Similarly, the upper limit is taken to be 1 whenever $w = n$ due to $b > 0$. Because the remaining beta quantiles lie strictly between 0 and 1, the Clopper-Pearson interval respects the natural boundaries for probabilities.

Often, the Clopper-Pearson interval is given in terms of quantiles from an F-distribution rather than a beta distribution. This comes about through a relationship between the two distributions; see #9.21 on p. 454 of Casella and Berger (2002) for this relationship. Both formulas produce the same limits and beta quantiles are easy to compute, so we omit the F-based formula here.

There are a few variations on the Clopper-Pearson interval. The Blaker interval proposed in Blaker (2000, 2001) also guarantees the true confidence level is always equal to or greater than the stated level. An added benefit is that the interval is no wider than the Clopper-Pearson interval and is often narrower. A disadvantage is that the interval is more difficult to calculate and requires an iterative numerical procedure to find its limits. The CIpi.R program shows how to calculate the interval using the `binom.blaker.limits()` function of the `BlakerCI` package. Another variation on the Clopper-Pearson interval is the mid-p interval. This interval no longer guarantees the true confidence level to be greater than the stated level, but it will be shorter than the Clopper-Pearson interval while performing relatively well with respect to the stated confidence level (Brown et al., 2001). The CIpi.R program shows how to calculate this interval using the `midPci()` function of the `PropCIs` package.

Example: Clopper-Pearson interval (CIpi.R)

Suppose $w = 4$ successes are observed out of $n = 10$ trials again. The 95% Clopper-Pearson interval is $\text{beta}\left(0.025;\ 4,\ 7\right) < \pi < \text{beta}\left(0.975;\ 5,\ 6\right)$. The `qbeta()` function in R calculates these quantiles resulting in an interval $0.1216 < \pi < 0.7376$. Notice that this is the widest of the intervals calculated so far.

Below is the R code used to calculate the interval:

```
> round(qbeta(p = c(alpha/2, 1-alpha/2), shape1 = c(w, w+1),
    shape2 = c(n-w+1, n-w)),4)
[1] 0.1216 0.7376

> binom.confint(x = w, n = n, conf.level = 1-alpha, methods =
    "exact")
  method x  n mean     lower     upper
1  exact 4 10  0.4 0.1215523 0.7376219
```

Within the `qbeta()` function call, the `shape1` argument is a and the `shape2` argument is b. We use vectors within `qbeta()` to find the quantiles. R matches each vector

Table 1.1: Confidence intervals for the hepatitis C prevalence.

Method	Interval	Length
Wald	(0.0157, 0.0291)	0.0134
Agresti-Coull	(0.0165, 0.0302)	0.0137
Wilson	(0.0166, 0.0301)	0.0135
Clopper-Pearson	(0.0162, 0.0302)	0.0140

value to produce the equivalent of separate `qbeta()` function runs for the lower and upper limits. The `binom.confint()` function is used as an alternative way to find the interval where `method = "exact"` gives the Clopper-Pearson interval. Note that this interval was also given earlier when we used `method = "all"`.

Example: Hepatitis C prevalence among blood donors (HepCPrev.R)

Blood donations are screened for diseases to prevent transmission from donor to recipient. To examine how prevalent hepatitis C is among blood donors, Liu et al. (1997) focused on 1,875 blood donations in Xuzhou City, China.[4] They observed that 42 donations tested positive for the antigen produced by the body when infected with the virus. The 95% Wilson interval is $0.0166 < \pi < 0.0301$, where we used the same type of R code as in the previous examples. Thus, with 95% confidence, the hepatitis C antigen prevalence in the Xuzhou City blood donor population is between 0.0166 and 0.0301.

In practice, we would only calculate one of the intervals discussed in this section. For demonstration purposes, Table 1.1 displays additional 95% confidence intervals. Due to the large sample size, we see that the intervals are similar with the Wald interval being the most different from the others. The lengths of the intervals are similar as well with the Clopper-Pearson interval being a little longer than the others.

Tests

When only one simple parameter is of interest, such as π here, we generally prefer confidence intervals over hypothesis tests, because the interval gives a range of possible parameter values. We can typically infer that a hypothesized value for a parameter can be rejected if it does not lie within the confidence interval for the parameter. However, there are situations where a fixed known value of π, say π_0, is of special interest, leading to a formal hypothesis test of $H_0 : \pi = \pi_0$ vs. $H_a : \pi \neq \pi_0$.

With regard to the Wilson interval, it was noted that the score test statistic

$$Z_0 = \frac{\hat{\pi} - \pi_0}{\sqrt{\pi_0(1-\pi_0)/n}},$$

is often used in these situations. When the null hypothesis is true, Z_0 should have approximately a standard normal distribution, where the approximation is generally better for larger samples. The null hypothesis is rejected when an unusual value of Z_0 is observed

[4]The study's main purpose was to examine how well "group testing" would work to estimate overall disease prevalence. See Bilder (2009) for an introduction to group testing.

relative to this distribution, namely something less than $-Z_{1-\alpha/2}$ or greater than $Z_{1-\alpha/2}$. The p-value is a measure of how extreme the test statistic value is relative to what is expected when H_0 is true. This p-value is calculated as $2P(Z > |Z_0|)$ where Z has a standard normal distribution. Note that this test is equivalent to rejecting the null hypothesis when π_0 is outside of the Wilson interval. If desired, the `prop.test()` function can be used to calculate Z_0 (Z_0^2 is actually given) and a corresponding p-value; this is demonstrated in the CIpi.R program. This function also calculates the Wilson interval.

We recommend using the score test when performing a test for π. However, there are alternative testing procedures. In particular, the likelihood ratio test (LRT) is a general way to perform hypothesis tests, and it can be used here to test π (see Appendix B.5 for an introduction). Informally, the LRT statistic is

$$\Lambda = \frac{\text{Maximum of likelihood function under } H_0}{\text{Maximum of likelihood function under } H_0 \text{ or } H_a}.$$

For the specific test of $H_0 : \pi = \pi_0$ vs. $H_a : \pi \neq \pi_0$, the denominator is $\hat{\pi}^w(1-\hat{\pi})^{n-w}$ (using Equation 1.2), because the maximum possible value of the likelihood function occurs when it is evaluated at the MLE. The numerator is $\pi_0^w(1-\pi_0)^{n-w}$ because there is only one possible value of the likelihood function if the null hypothesis is true. The transformed statistic $-2\log(\Lambda)$ turns out to have an approximate χ^2_1 distribution in large samples if the null hypothesis is true. For this test, the transformed statistic can be re-expressed as

$$-2\log(\Lambda) = -2\left\{w\log\left(\frac{\pi_0}{\hat{\pi}}\right) + (n-w)\log\left(\frac{1-\pi_0}{1-\hat{\pi}}\right)\right\}.$$

We reject the null hypothesis if $-2\log(\Lambda) > \chi^2_{1,1-\alpha/2}$, where $\chi^2_{1,1-\alpha/2}$ is the $1-\alpha/2$ quantile from a chi-square distribution with 1 degree of freedom (for example, $\chi^2_{1,0.95} = 3.84$ when $\alpha = 0.05$). The p-value is $P(A > -2\log(\Lambda))$ where A has a χ^2_1 distribution.

An alternative way to calculate a confidence interval for π is to invert the LRT in a similar manner as was done for the Wilson interval (see Exercise 13). This likelihood ratio (LR) interval is automatically calculated by the `binom.confint()` function of the `binom` package, where the `methods = "lrt"` argument value is used. We provide additional code in CIpi.R that shows how to find the interval without this function. The interval is generally harder to compute than the intervals that we recommend in this section and has no advantages over them. LR confidence intervals often are used in some more complicated contexts where better intervals are not available (this will be the case in Chapters 2 to 4). The interval is better than the Wald interval in most problems.

1.1.3 True confidence levels for confidence intervals

As discussed on p. 11, a confidence interval method may not actually achieve its stated confidence level. The reasons for this are explained shortly. Figure 1.3 provides a comparison of the true confidence levels for the Wald, Wilson, Agresti-Coull, and Clopper-Pearson intervals. For each plot, n is 40 and the stated confidence level is 0.95 ($\alpha = 0.05$). The true confidence level (coverage) for each interval method is plotted as a function of π. For example, the true confidence level at $\pi = 0.157$ is 0.8760 for the Wald, 0.9507 for the Wilson, 0.9507 for the Agresti-Coull, and 0.9740 for the Clopper-Pearson intervals, respectively. Obviously, none of these intervals achieve exactly the stated confidence level on a consistent basis. Below are some general conclusions from examining this plot:

- The Wald interval tends to be the farthest from 0.95 the most often. In fact, the true confidence level is often too low for it to be on the plot at extreme values of π.

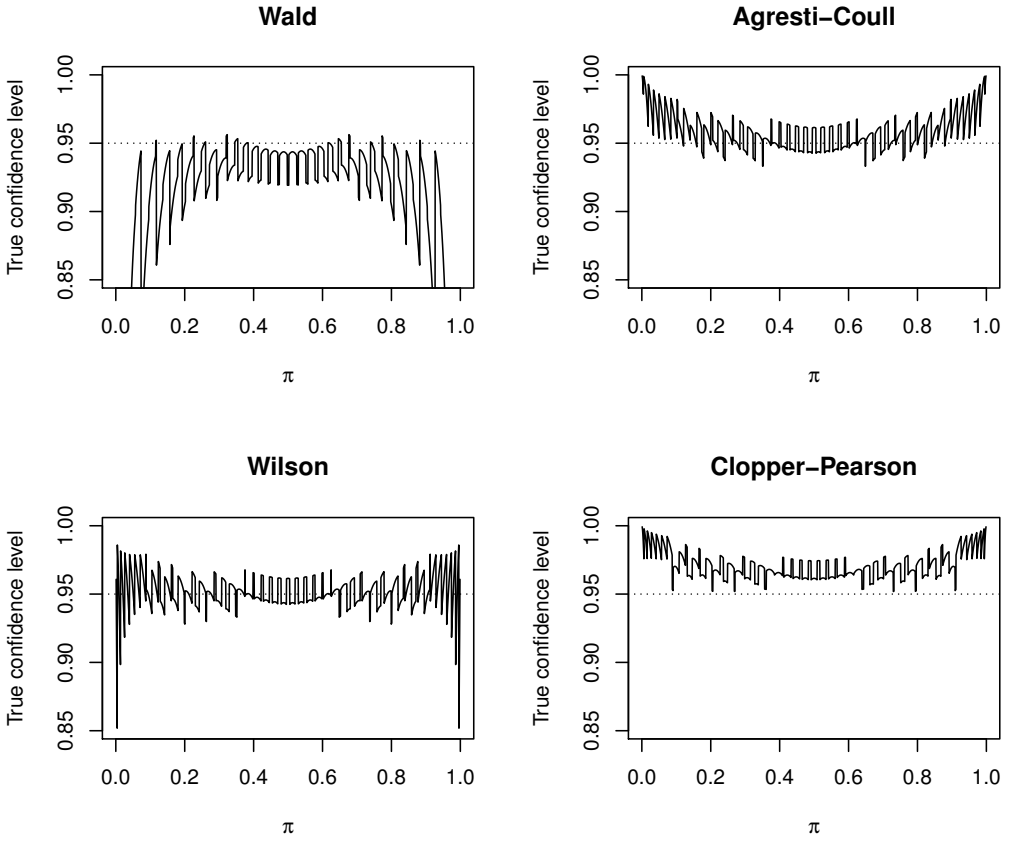

Figure 1.3: True confidence levels with $n = 40$ and $\alpha = 0.05$.

- The Agresti-Coull interval does a much better job than the Wald with its true confidence level usually between 0.93 and 0.98. For values of π close to 0 or 1, the interval can be very conservative.
- The Wilson interval performs a little better than the Agresti-Coull interval with its true confidence level generally between 0.93 and 0.97; however, for very extreme π, it can be very liberal. Note that this performance for extreme π can be improved by changing the lower interval limit to $-\log(1-\alpha)/n$ when $w = 1$ and the upper interval limit to $1 + \log(1-\alpha)/n$ when $w = n - 1$; see p. 112 of Brown et al. (2001) for justification. This small modification was used by the `binom.confint()` function in the past (version 1.0-5 of the package), but it is now no longer implemented as of version 1.1-1 of the package.
- The Clopper-Pearson interval has a true confidence level at or above the stated level, where it is generally oscillating between 0.95 and 0.98. For values of π close to 0 or 1, the interval can be very conservative.

Similar findings can be shown for other values of n and α. The R code used to construct Figure 1.3 is available in the ConfLevel4Intervals.R program, and it will be discussed shortly.

Why do these plots in Figure 1.3 have such strange patterns? It is all because of the discreteness of a binomial random variable. For a given n, there are only $n + 1$ possible

intervals that can be formed, one for each value of $w = 0, 1, \ldots, n$. For a specific value of π, some of these intervals contain π and some do not. Thus, the true confidence level at π, say $C(\pi)$, is the sum of the binomial probabilities for all intervals that do contain π:

$$C(\pi) = \sum_{w=0}^{n} I(w) \binom{n}{w} \pi^w (1-\pi)^{n-w}, \tag{1.6}$$

where $I(w) = 1$ if the interval formed with w contains π, and $I(w) = 0$ if not. Each of these binomial probabilities in Equation 1.6 changes slowly as π changes. As long as we do not move π past any interval limits, the true confidence level changes slowly too. However, as soon as π crosses over an interval limit, a mass of probability is suddenly either added to or subtracted from the true confidence level, resulting in the spikes that appear in all parts of Figure 1.3. We illustrate how to find the true confidence level and when these spikes occur in the next example.

Example: True confidence level for the Wald interval (ConfLevel.R)

We show in this example how to calculate a true confidence level for the Wald interval with $n = 40$, $\pi = 0.157$, and $\alpha = 0.05$. Below is a description of the process:

1. Find the probability of obtaining each possible value of w using the `dbinom()` function with $n = 40$ and $\pi = 0.157$,
2. Calculate the 95% Wald confidence interval for each possible value of w, and
3. Sum up the probabilities corresponding to those intervals that contain $\pi = 0.157$; this is the true confidence level.

Below is the R code:

```
> pi <- 0.157
> alpha <- 0.05
> n <- 40
> w <- 0:n
> pi.hat <- w/n
> pmf <- dbinom(x = w, size = n, prob = pi)
> var.wald <- pi.hat*(1-pi.hat)/n
> lower <- pi.hat - qnorm(p = 1-alpha/2) * sqrt(var.wald)
> upper <- pi.hat + qnorm(p = 1-alpha/2) * sqrt(var.wald)
> save <- ifelse(test = pi>lower, yes = ifelse(test = pi<upper,
    yes = 1, no = 0), no = 0)
> data.frame(w, pi.hat, round(data.frame(pmf, lower, upper),4),
    save)[1:13,]
    w pi.hat    pmf   lower  upper save
1   0  0.000 0.0011  0.0000 0.0000    0
2   1  0.025 0.0080 -0.0234 0.0734    0
3   2  0.050 0.0292 -0.0175 0.1175    0
4   3  0.075 0.0689 -0.0066 0.1566    0
5   4  0.100 0.1187  0.0070 0.1930    1
6   5  0.125 0.1591  0.0225 0.2275    1
7   6  0.150 0.1729  0.0393 0.2607    1
8   7  0.175 0.1564  0.0572 0.2928    1
9   8  0.200 0.1201  0.0760 0.3240    1
10  9  0.225 0.0795  0.0956 0.3544    1
```

```
11 10  0.250 0.0459  0.1158 0.3842    1
12 11  0.275 0.0233  0.1366 0.4134    1
13 12  0.300 0.0105  0.1580 0.4420    0

> sum(save*pmf)
[1] 0.875905
> sum(dbinom(x = 4:11, size = n, prob = pi))
[1] 0.875905
```

The code contains many of the same components that we have seen before. The main difference now is that we are calculating an interval for each possible value of w rather an interval for only one. One new part within the code is the `ifelse()` function. This function does a logical check for whether or not π is within each of the 41 intervals. For example, the second interval is $(-0.0234, 0.0734)$, which does not contain $\pi = 0.157$ so the `save` object has a value of 0 for its second element.

The data frame created at the end puts all of the calculated components together into a table. For example, we see that if $w = 3$, the corresponding interval does not contain π, but at $w = 4$ the corresponding interval does. By examining other parts of the data frame, we see that the intervals for $w = 4$ to 11 all contain $\pi = 0.157$. The probability that a binomial random variable is between 4 and 11 with $n = 40$ and $\pi = 0.157$ is 0.8759, which is the true confidence level. Obviously, the Wald confidence interval does not achieve its stated level of 95%.

Notice that the upper interval limit at $w = 3$ barely does not contain $\pi = 0.157$ and $P(W = 3) = 0.0689$. By using the same code with the change of `pi <- 0.156`, the upper limit at $w = 3$ now does contain $\pi = 0.156$, so that $P(W = 3) = 0.0706$ is included when summing probabilities for the true confidence level. Overall, $w = 3$ to 11 now have confidence intervals that contain $\pi = 0.156$ leading to a true confidence level of 0.9442! This demonstrates what we alluded to earlier as the cause for the spikes in Figure 1.3.

In simple problems like the present one, we can exactly determine the probabilities of each interval that contains a given π, so that the plots like in Figure 1.3 can be made exactly. In other cases, we may have to rely on Monte Carlo simulation. We explore the simulation approach next to enable us to compare an exact true confidence level to one estimated by simulation. This will be helpful later in the text when the simulation method is the only available method of assessment.

Example: Estimated true confidence level for the Wald interval (ConfLevel.R)

Suppose again that $n = 40$, $\pi = 0.157$, and $\alpha = 0.05$. Below is a description of the process to estimate the true confidence level through simulation:

1. Simulate 1,000 samples using the `rbinom()` function with $n = 40$ and $\pi = 0.157$,
2. Calculate the 95% Wald confidence interval for each sample, and
3. Calculate the proportion of intervals that contain $\pi = 0.157$; this is the estimated true confidence level.

Below is the R code:

```
> numb.bin.samples <- 1000  # Binomial samples of size n

> set.seed(4516)
> w <- rbinom(n = numb.bin.samples, size = n, prob = pi)
> pi.hat <- w/n
> var.wald <- pi.hat*(1-pi.hat)/n
> lower <- pi.hat - qnorm(p = 1-alpha/2) * sqrt(var.wald)
> upper <- pi.hat + qnorm(p = 1-alpha/2) * sqrt(var.wald)
> data.frame(lower, upper)[1:10,]
   w pi.hat         lower     upper
1  6  0.150  0.039344453 0.2606555
2  6  0.150  0.039344453 0.2606555
3  7  0.175  0.057249138 0.2927509
4  8  0.200  0.076040994 0.3239590
5  8  0.200  0.076040994 0.3239590
6  6  0.150  0.039344453 0.2606555
7  8  0.200  0.076040994 0.3239590
8  3  0.075 -0.006624323 0.1566243
9  5  0.125  0.022511030 0.2274890
10 4  0.100  0.007030745 0.1929693

> save <- ifelse(test = pi>lower, yes = ifelse(test = pi<upper,
    yes = 1, no = 0), no = 0)
> save[1:10]
[1] 1 1 1 1 1 1 1 0 1 1
> mean(save)
[1] 0.878
```

Again, we are using much of the same code as in the past. The `ifelse()` function is used to check whether $\pi = 0.157$ is within each of the intervals. For example, we see that sample #8 results in $\hat{\pi} = 0.075$ and an interval of (-0.0066, 0.1566), which does not contain 0.157, so the corresponding value of `save` is 0. The mean of all the 0's and 1's in `save` is 0.878. This is our estimated true confidence level for the Wald interval at $n = 40$ and $\pi = 0.157$.

In this relatively simple simulation problem, we already know that the intervals for $w = 4, \ldots, 11$ contain $\pi = 0.157$ while the others do not. To see that the simulation is, indeed, estimating $P(4 \leq W \leq 11)$, the `table()` function is used to calculate the number of times each w occurs:

```
> counts <- table(w)
> counts
w
 1   2   3   4   5   6   7   8   9  10  11
 8  35  64 123 147 165 172 123  76  46  26
12  13
11   4
> sum(counts[4:11])/numb.bin.samples
[1] 0.878
```

For example, there were 64 out of the 1,000 observations that resulted in a $w = 3$. This is very similar to the $P(W = 3) = 0.0689$ that we obtained for the past example. Summing up the counts for $w = 4, \ldots, 11$ and dividing by 1000, we obtain the same estimate of 0.878 for the true confidence level.

The estimate of the true confidence level here is almost the same as the actual true confidence level found in the previous example. Due to using a large number of samples, the law of large numbers ensures that this will happen. We could even go as far as finding a confidence interval for the true confidence level! For this case, we have 878 "successes" out of 1,000 "trials." A 95% Wilson interval for the true confidence level itself is (0.8563, 0.8969), which happens to contain 0.8759, the known true confidence level.

Figure 1.3 provides the true confidence levels for $\pi = 0.001, ..., 0.999$ by 0.0005, where we use straight lines to fill in the missing confidence levels between plotting points at levels of π not used. In order to calculate all of these 1,997 different confidence levels for a particular interval, we repeat the same code as before, but now for each π by using a "for loop" within R. The next example illustrates this process.

Example: True confidence level plot (ConfLevel4Intervals.R, ConfLevelWaldOnly.R)

With $n = 40$ and $\alpha = 0.05$, we calculate the true confidence levels for the Wald interval using the following code:

```
> alpha <- 0.05
> n <- 40
> w <- 0:n
> pi.hat <- w/n
> pi.seq <- seq(from = 0.001, to = 0.999, by = 0.0005)

> # Wald
> var.wald <- pi.hat*(1-pi.hat)/n
> lower.wald <- pi.hat - qnorm(p = 1-alpha/2) * sqrt(var.wald)
> upper.wald <- pi.hat + qnorm(p = 1-alpha/2) * sqrt(var.wald)

> # Save true confidence levels in a matrix
> save.true.conf <- matrix(data = NA, nrow = length(pi.seq), ncol
    = 2)

> # Create counter for the loop
> counter <- 1

> # Loop over each pi
> for(pi in pi.seq) {
    pmf <- dbinom(x = w, size = n, prob = pi)
    save.wald <- ifelse(test = pi>lower.wald, yes = ifelse(test =
        pi<upper.wald, yes = 1, no = 0), no = 0)
    wald <- sum(save.wald*pmf)
    save.true.conf[counter,] <- c(pi, wald)
    # print(save.true.conf[counter,])
    counter <- counter+1
}

> plot(x = save.true.conf[,1], y = save.true.conf[,2], main =
    "Wald", xlab = expression(pi), ylab = "True confidence level",
    type = "l", ylim = c(0.85,1))
> abline(h = 1-alpha, lty = "dotted")
```

We create a vector `pi.seq` which is a sequence of numbers from 0.001 to 0.999 by 0.0005. The `for(pi in pi.seq)` function code (often referred to as a "for loop") instructs R to take one π value out of `pi.seq` at a time. The code enclosed by braces then finds the true confidence level for this π. The `save.true.conf` object is a matrix that is created to have 1,997 rows and 2 columns. At first, all of its values are initialized to be `"NA"` within R. Its values are updated then one row at a time by inserting the value of π and the true confidence level. Finally, the `counter` object allows us to change the row number of `save.true.conf` within the loop.[5]

After the for loop, we use use the `plot()` function to plot the value of π on the x-axis and the true confidence level on the y-axis using the appropriate columns of `save.true.conf`. The `type = "l"` argument instructs R to construct a line plot where each π and true confidence level pair is connected by a line. The `abline()` function draws a horizontal dotted line at 0.95, which is the stated confidence level. Please see the upper left plot in Figure 1.3 for the final result. In order to construct all four plots in Figure 1.3, we insert the code for the other three intervals into the braces of the loop. We also add three columns to the `save.true.conf` matrix and use additional calls to the `plot()` function. Please see ConfLevel4Intervals.R for the code.

The `binom` package in R also can be used to calculate the true confidence levels. The `binom.coverage()` function calculates the true confidence level for one π at a time, and the `binom.plot()` function plots the true confidence levels over a set of different values of π. Examples of using these functions are in the programs for this example. Note that we purposely demonstrated the calculations without `binom.coverage()` first, because convenient functions like it are not available for other situations examined elsewhere in the textbook.

1.2 Two binary variables

We consider now the situation when the same Bernoulli trial is measured on units that can be classified into groups. The simplest such case is when a population consists of two groups, such as females and males, fresh- and salt-water fish, or American and foreign companies. Below are two examples with a binary response on trials that form two groups.

Example: Larry Bird's free throw shooting

A free throw is a shot in basketball where the shooter can shoot freely (unopposed by another player) from a specific location on the court. The shot is either made (a success) or missed (a failure). Most often during a National Basketball Association (NBA) game, free throws are shot in pairs. This means a free throw shooter has one attempt and then subsequently has a second attempt no matter what happens on the first.

The former NBA player Larry Bird was one of the most successful at making free throws during his career with a success rate of 88.6%. By comparison, the NBA

[5]If desired, the call to the `print()` function can be uncommented to see the progress during the loop. If this is done, it is best to turn off the "buffered output" in R: select Misc > Buffered output from the R main menu.

Table 1.2: Larry Bird's free throw outcomes; data source is Wardrop (1995).

		Second		
		Made	Missed	Total
First	Made	251	34	285
	Missed	48	5	53
	Total	299	39	338

Table 1.3: Salk vaccine clinical trial results; data source is Francis et al. (1955, p. 25).

	Polio	Polio free	Total
Vaccine	57	200,688	200,745
Placebo	142	201,087	201,229
Total	199	401,775	401,974

average during this time was about 75% (`http://www.basketball-reference.com`). Bird's outstanding success rate is among his many achievements, for which he has been recognized as one of the 50 greatest players in the history of the NBA (`http://www.nba.com/history/players/50greatest.html`). During the 1980-81 and 1981-82 NBA seasons, the outcomes from Bird's free throw attempts shot in pairs were recorded, and a summary is shown in Table 1.2. For example, Bird made both his first and second attempts 251 times. Also, Bird made the first attempt, but then subsequently missed the second attempt 34 times. Overall, he made the first attempt 285 times without regard to what happened on the second attempt. In total, Bird shot 338 pairs of free throw pairs during the season.

Basketball fans and commentators often speculate that the results of a second free throw might depend on the results of the first. For example, if a shooter misses the first attempt, will disappointment or determination lead to altering his/her approach for the second attempt? If so, then we should see that the probability of success on the second attempt is different depending on whether the first attempt was made or missed. Thus, the two groups in this problem are formed by the results of the first attempt, and the Bernoulli trials that we observe are the results of the second attempt. Given the data in Table 1.2, we will investigate if the second attempt outcome is dependent on what happens for the first attempt.

Example: Salk vaccine clinical trial

Clinical trials are performed to determine the safety and efficacy of new drugs. Frequently, the safety and efficacy responses are categorical in nature; for example, the efficacy response may be simply whether a drug cures or does not cure a patient of a disease. In order to ensure that a new drug is indeed better than doing nothing (patients sometimes get better without intervention), it is essential to have a control group in the trial. This is achieved in clinical trials by randomizing patients into two groups: new drug or control. The control group is often a placebo, which is administered just like the new drug but contains no medication.

One of the most famous and largest clinical trials ever performed was in 1954. Over 1.8 million children participated in the clinical trial to determine the effectiveness of the polio vaccine developed by Jonas Salk (Francis et al., 1955). While the actual design of the trial sparked debate (Brownlee, 1955; Dawson, 2004), we forgo this discussion and focus on the data obtained from the randomized, placebo-controlled portion of

Table 1.4: Probability and observed count structures for two independent binomial random variables.

		Response		
		1	2	
Group	1	π_1	$1-\pi_1$	1
	2	π_2	$1-\pi_2$	1

		Response		
		1	2	
Group	1	w_1	$n_1 - w_1$	n_1
	2	w_2	$n_2 - w_2$	n_2
		w_+	$n_+ - w_+$	n_+

the trial. The data, given in Table 1.3, show that 57 out of the 200,745 children in the vaccine group developed polio during the study period, as opposed to 142 out of the 201,229 children in the placebo group. The question of interest for the clinical trial was "Does the vaccine help to prevent polio?" We will develop comparison measures in this section to answer this question.

1.2.1 Notation and model

The model and notation follow those used for a single binomial random variable in Section 1.1. We start by considering two separate Bernoulli random variables, Y_1 and Y_2, one for each group. The probabilities of success for the two groups are denoted by π_1 and π_2, respectively. We observe n_j trials of Y_j leading to w_j observed successes, $j = 1, 2$.[6] We replace a subscript with "+" to denote a sum across that subscript. Thus, $n_+ = n_1 + n_2$ is the total number of trials, and $w_+ = w_1 + w_2$ is the total number of observed successes. This notation is depicted in Table 1.4. The table on the right side of Table 1.4 is called a *two-way contingency table*, because it gives a listing of all possible cross-tabulations of two categorical variables. We cover more general forms of contingency tables in Chapters 3 and 4.

We denote the random variable representing the number of successes in group j by W_j and write its PMF as

$$P(W_j = w_j) = \binom{n_j}{w_j} \pi_j^{w_j} (1-\pi_j)^{n_j - w_j}, \; w_j = 0, 1, \ldots, n_j, \; j = 1, 2.$$

We assume that the two random variables Y_1 and Y_2 are independent, so that the outcome of one cannot affect the outcome of the other. In the Salk vaccine clinical trial, for example, this means that children assigned to receive vaccine cannot pass on immunity or disease to children in the placebo group and vice versa. This assumption is critical in what follows, and so this model is referred to as the *independent binomial model*. When independence is not satisfied, then other models need to be used that account for dependence between the random variables (e.g., see Section 1.2.6 for handling paired data).

When we want to simulate data from this model, we will use R code like what is shown below. This sampling procedure will be important shortly when we use these simulated counts to evaluate statistical inference procedures, like confidence intervals, to determine if they work as expected.

[6] More formally, we could define y_{ij} as the observed value for the i^{th} independent trial in the j^{th} group. This leads to $w_j = \sum_{i=1}^{n_j} y_{ij}$.

Example: Simulate counts in a contingency table (SimContingencyTable.R)

Consider the case of $\pi_1 = 0.2$, $\pi_2 = 0.4$, $n_1 = 10$, and $n_2 = 10$. The R code below shows how to simulate one set of counts for a contingency table.

```
> pi1 <- 0.2
> pi2 <- 0.4
> n1 <- 10
> n2 <- 10

> set.seed(8191)
> w1 <- rbinom(n = 1, size = n1, prob = pi1)
> w2 <- rbinom(n = 1, size = n2, prob = pi2)

> c.table <- array(data = c(w1, w2, n1-w1, n2-w2), dim = c(2,2),
    dimnames = list(Group = c(1,2), Response = c(1, 2)))
> c.table
     Response
Group 1 2
    1 1 9
    2 3 7
> c.table1[1,1]  # w1
[1] 1
> c.table1[1,2]  # n1-w1
[1] 9
> c.table1[1,]  # w1 and n1-w1
1 2
1 9
> sum(c.table1[1,])  # n1
[1] 10
```

Similar to Section 1.1, we use the `rbinom()` function to simulate values for w_1 and w_2. To form the contingency table (what we name `c.table`), we use the `array()` function. Its `data` argument contains the counts within the contingency table. These counts are concatenated together using the `c()` function. Notice that the data are entered by columns $(w_1, w_2, n_1 - w_1, n_2 - w_2)$. The `dim` argument specifies the contingency table's dimensions as (number of rows, number of columns), where the `c()` function is used again. Finally, the `dimnames` argument gives names for the row and column measurements. The names are given in a *list* format, which allows for a number of objects to be linked together (please see Appendix A.7.3 for more on lists if needed). In this case, the objects are `Group` and `Response` that contain the levels of the rows and columns, respectively.

For this particular sample, $w_1 = 1$, $n_1 - w_1 = 9$, $w_2 = 3$, and $n_2 - w_2 = 7$. We access these values from within the contingency table by specifying a row and column number with `c.table`. For example, `c.table[1,2]` is equal to $n_1 - w_1$. We omit a column or row number within `[ ]` to have a whole row or column, respectively, displayed. Summed counts are found by using the `sum()` function with the appropriate counts.

If we wanted to repeat this process, say, 1,000 times, the `n` argument of the `rbinom()` functions would be changed to 1,000. Each contingency table would need to be formed separately using the `array()` function. The corresponding program for this example provides the code.

Likelihood and estimates

The main interests in this problem are estimating the probabilities of success, π_1 and π_2, for each group and comparing these probabilities. Maximum likelihood estimation again provides a convenient and powerful solution. Because Y_1 and Y_2 are independent, so too are W_1 and W_2. The likelihood function formed by independent random variables is just the product of their respective likelihood functions. Hence, the likelihood function is $L(\pi_1, \pi_2|w_1, w_2) = L(\pi_1|w_1) \times L(\pi_2|w_2)$. Maximizing this likelihood over π_1 and π_2 results in the "obvious" estimates $\hat{\pi}_1 = w_1/n_1$ and $\hat{\pi}_2 = w_2/n_2$, the sample proportions in the two groups. In other words, when the random variables are independent, each probability is estimated using only the data from its own group.

Example: Larry Bird's free throw shooting (Bird.R)

The purpose of this example is to estimate the probability of successes using a contingency table structure in R. If the Bernoulli trial results are already summarized into counts as in Table 1.2, then a contingency table is created in R using the `array()` function:

```
> c.table <- array(data = c(251, 48, 34, 5), dim = c(2,2),
    dimnames = list(First = c("made", "missed"), Second =
    c("made", "missed")))
> c.table
        Second
First     made missed
made     251     34
missed    48      5

> list(First = c("made", "missed"), Second = c("made", "missed"))
$First
[1] "made"   "missed"
$Second
[1] "made"   "missed"
```

Because the levels of row and column variables have names, we use these names within the `dimnames` argument.

The estimates of the probability of successes (or sample proportions) are found by taking advantage of how R performs calculations:

```
> rowSums(c.table)  # n1 and n2
  made missed
   285     53

> pi.hat.table <- c.table/rowSums(c.table)
> pi.hat.table
        Second
First          made     missed
  made   0.8807018 0.11929825
  missed 0.9056604 0.09433962
```

The `rowSums()` function find the sum of counts in each row to obtain n_1 and n_2. By taking the contingency table of counts divided by these row sums, we obtain $\hat{\pi}_1$ and $\hat{\pi}_2$ in the first column and $1 - \hat{\pi}_1$ and $1 - \hat{\pi}_2$ in the second column. Notice that R does

this division correctly taking the counts in the first (second) row of `c.table` divided by the first (second) element of `rowSums(c.table)`.

Data are often available as measurements on each trial, rather than as summarized counts. For example, the Larry Bird data would originally have consisted of 338 unaggregated pairs of first and second free throw results. Thus, the data might have appeared first as[7]

```
> head(all.data2)
   first second
1   made   made
2 missed   made
3   made missed
4 missed missed
5   made   made
6 missed   made
```

In this example, `first` represents the group and `second` is the trial response. All 338 observations are stored in a data frame named `all.data2` within the corresponding program for this example, and we print the first 6 observations using the `head()` function. We call this data format the *raw data* because it represents how the data looked before being processed into counts.

To form a contingency table from the raw data, we can use the `table()` or `xtabs()` functions:

```
> bird.table1 <- table(all.data2$first, all.data2$second)
> bird.table1
        made missed
made     251     34
missed    48      5
> bird.table1[1,1]  # w1
[1] 251

> bird.table2 <- xtabs(formula = ~ first + second, data =
    all.data2)
> bird.table2
        second
first     made missed
  made     251     34
  missed    48      5
> bird.table2[1,1]  # w1
[1] 251
```

In both cases, the functions produce a contingency table that is saved into an object to allow parts of it to be accessed as before. Note that the "xtabs" name comes about through an abbreviation of *crosstabulations*, which is a frequently used term to describe the joint summarization of multiple variables.

The estimated probability that Larry Bird makes his second free throw attempt is $\hat{\pi}_1 = 0.8807$ given that he makes the first and $\hat{\pi}_2 = 0.9057$ given he misses the first. In this sample, the probability of success on the second attempt is larger when the

[7]The actual order of Larry Bird's free throw results are not available. We present this as a hypothetical ordering to emulate what may have occurred.

first attempt is missed rather than made. This is somewhat counterintuitive to many basketball fans' perceptions that a missed first free throw should lower the probability of success on the second free throw. However, this is only for one sample. We would like to generalize to the population of all free throw attempts by Larry Bird. In order to make this generalization, we need to use statistical inference procedures. We discuss these next.

1.2.2 Confidence intervals for the difference of two probabilities

A relatively easy approach to comparing π_1 and π_2 can be developed by taking their difference $\pi_1 - \pi_2$. The corresponding estimate of $\pi_1 - \pi_2$ is $\hat{\pi}_1 - \hat{\pi}_2$. Each success probability estimate has a probability distribution that is approximated by a normal distribution in large samples: $\hat{\pi}_j \dot{\sim} N(\pi_j, \widehat{Var}(\hat{\pi}_j))$, where $\widehat{Var}(\hat{\pi}_j) = \hat{\pi}_j(1-\hat{\pi}_j)/n_j$, $j = 1, 2$. Because linear combinations of normal random variables are themselves normal random variables (Casella and Berger, 2002, p. 156), the probability distribution for $\hat{\pi}_1 - \hat{\pi}_2$ is approximated by $N(\pi_1 - \pi_2, \widehat{Var}(\hat{\pi}_1 - \hat{\pi}_2))$, where $\widehat{Var}(\hat{\pi}_1 - \hat{\pi}_2) = \hat{\pi}_1(1-\hat{\pi}_1)/n_1 + \hat{\pi}_2(1-\hat{\pi}_2)/n_2$.[8] This distribution forms the basis for a range of inference procedures.

The easiest confidence interval to form for $\pi_1 - \pi_2$ uses the normal approximation for $\hat{\pi}_1 - \hat{\pi}_2$ directly to create a Wald interval:

$$\hat{\pi}_1 - \hat{\pi}_2 \pm Z_{1-\alpha/2}\sqrt{\frac{\hat{\pi}_1(1-\hat{\pi}_1)}{n_1} + \frac{\hat{\pi}_2(1-\hat{\pi}_2)}{n_2}}.$$

Unfortunately, the Wald interval for $\pi_1 - \pi_2$ has similar problems with achieving the stated level of confidence as the Wald interval for π. We will investigate this shortly.

Due to these problems, a number of other confidence intervals for $\pi_1 - \pi_2$ have been proposed. Inspired by the good general performance of the Agresti-Coull interval for a single probability, Agresti and Caffo (2000) investigated various intervals constructed as Wald-type intervals on data with different numbers of added successes and failures. They found that adding one success and one failure for each group results in an interval that does a good job of achieving the stated confidence level. Specifically, let

$$\widetilde{\pi}_1 = \frac{w_1 + 1}{n_1 + 2} \text{ and } \widetilde{\pi}_1 = \frac{w_2 + 1}{n_2 + 2}$$

be the amended estimates of π_1 and π_2. Notice that unlike $\tilde{\pi}$ for the Agresti-Coull interval, the $\widetilde{\pi}_1$ and $\widetilde{\pi}_2$ estimates do not change when the confidence level changes. The $(1-\alpha)100\%$ Agresti-Caffo confidence interval for $\pi_1 - \pi_2$ is

$$\widetilde{\pi}_1 - \widetilde{\pi}_2 \pm Z_{1-\alpha/2}\sqrt{\frac{\widetilde{\pi}_1(1-\widetilde{\pi}_1)}{n_1 + 2} + \frac{\widetilde{\pi}_2(1-\widetilde{\pi}_2)}{n_2 + 2}}.$$

Overall, we recommend the Agresti-Caffo method for general use.

Other confidence interval methods though have been developed analogous to the single-parameter case discussed in Section 1.1.2. There is a score interval based on inverting the test statistic for a score test of $H_0 : \pi_1 - \pi_2 = d$ (i.e., determining for what values d

[8] This is an application of the following result: $Var(aU + bV) = a^2Var(U) + b^2Var(V) + 2abCov(U, V)$, where U and V are random variables and a and b are constants. If U and V are independent random variables, then $Cov(U, V) = 0$. See p. 171 of Casella and Berger (2002).

of $\pi_1 - \pi_2$ that the null hypothesis is not rejected). This turns out to be a considerably more difficult computational problem than in the single-parameter case, because H_0 does not actually specify the values of π_1 and π_2. Exercise 24 discusses how this interval is calculated. Similarly, there is a two-group Bayesian credible interval similar to Jeffreys method, but this involves distributions that are not as simple to compute as the standard normal. Details of this interval are available in Agresti and Min (2005a).

Example: Larry Bird's free throw shooting (Bird.R)

The purpose of this example is to calculate a confidence interval for the difference in the second free throw success probabilities given the first attempt outcomes. Continuing the code from earlier, we obtain the following:

```
> alpha <- 0.05
> pi.hat1 <- pi.hat.table[1,1]
> pi.hat2 <- pi.hat.table[2,1]

> # Wald
> var.wald <- pi.hat1*(1-pi.hat1) / sum(c.table[1,]) +
    pi.hat2*(1-pi.hat2) / sum(c.table[2,])
> pi.hat1 - pi.hat2 + qnorm(p = c(alpha/2, 1-alpha/2)) *
    sqrt(var.wald)
[1] -0.11218742  0.06227017

> # Agresti-Caffo
> pi.tilde1 <- (c.table[1,1] + 1) / (sum(c.table[1,]) + 2)
> pi.tilde2 <- (c.table[2,1] + 1) / (sum(c.table[2,]) + 2)
> var.AC <- pi.tilde1*(1-pi.tilde1) / (sum(c.table[1,]) + 2) +
    pi.tilde2*(1-pi.tilde2) / (sum(c.table[2,]) + 2)
> pi.tilde1 - pi.tilde2 + qnorm(p = c(alpha/2, 1-alpha/2)) *
    sqrt(var.AC)
[1] -0.10353254  0.07781192
```

The 95% Wald interval is $-0.1122 < \pi_1 - \pi_2 < 0.0623$, and the 95% Agresti-Caffo interval is $-0.1035 < \pi_1 - \pi_2 < 0.0778$. The intervals are somewhat similar with the Wald interval being shifted to the left of the Agresti-Caffo interval. Using the Agresti-Caffo interval, with 95% confidence, the difference in the second free throw success probabilities given the outcome of the first is between -0.1035 and 0.0778. Because this interval contains 0, we cannot detect a change in Bird's probability of a successful second free throw following made and missed first attempts. This means that either there is no difference, or there is a difference, but it was not detected in this sample. The latter situation could be caused by either bad luck (an unusual sample) or too small of a sample size.

The same confidence intervals can be obtained in other ways. First, we can use the following code when the data are not already within R via the `array()` function:

```
> w1 <- 251
> n1 <- 285
> w2 <- 48
> n2 <- 53
> alpha <- 0.05
> pi.hat1 <- w1/n1
> pi.hat2 <- w2/n2
```

```
> var.wald <- pi.hat1*(1-pi.hat1) / n1 +  pi.hat2*(1-pi.hat2) / n2
> pi.hat1 - pi.hat2 + qnorm(p = c(alpha/2, 1-alpha/2)) *
    sqrt(var.wald)
[1] -0.11218742  0.06227017
```

Second, the `prop.test()` function shown later gives the Wald confidence interval as part of its output (also used in Section 1.1.2). Finally, the `wald2ci()` function in the `PropCIs` package also calculates the Wald and Agresti-Caffo confidence intervals. Please see the corresponding program for example code.

To find a true confidence level for one of these confidence intervals, the joint probability distribution for all possible combinations of (W_1, W_2) is needed. This distribution is just the product of two binomial probabilities because these random variables are independent. For a given n_1 and n_2, there are $(n_1+1)(n_2+1)$ possible observed combinations of (w_1, w_2), and a confidence interval can be computed for each of these combinations. For set values of π_1 and π_2, some of these intervals contain $\pi_1 - \pi_2$ and some do not. Thus, the true confidence level at π_1 and π_2, $C(\pi_1, \pi_2)$, is the sum of the joint probabilities for all intervals that do contain $\pi_1 - \pi_2$:

$$C(\pi_1, \pi_2) = \sum_{w_2=0}^{n_2} \sum_{w_1=0}^{n_1} I(w_1, w_2) \binom{n_1}{w_1} \pi_1^{n_1} (1-\pi_1)^{n_1-w_1} \binom{n_2}{w_2} \pi_2^{w_2} (1-\pi_2)^{n_2-w_2}$$

where the indicator function $I(w_1, w_2)$ is 1 if the corresponding interval contains $\pi_1 - \pi_2$ and $I(w_1, w_2)$ is 0 otherwise. Calculation details are given in the next example.

Example: True confidence levels for the Wald and Agresti-Caffo intervals (ConfLevelTwoProb.R)

The true confidence level for the Wald interval can be found in a similar manner as discussed in Section 1.1.3. Consider the case of $\alpha = 0.05$, $\pi_1 = 0.2$, $\pi_2 = 0.4$, $n_1 = 10$, and $n_2 = 10$. To find all possible combinations of (w_1, w_2), we use the `expand.grid()` function, which finds all possible combinations of the arguments (separated by commas) within its parentheses. We repeat this same process to find all possible values of $(\hat{\pi}_1, \hat{\pi}_2)$ and $P(W_1 = w_1,\ W_2 = w_2)$. Below is the R code:

```
> alpha <- 0.05
> pi1 <- 0.2
> pi2 <- 0.4
> n1 <- 10
> n2 <- 10

> # All possible combinations of w1 and w2
> w.all <- expand.grid(w1 = 0:n1, w2 = 0:n2)

> # All possible combinations of pi^_1 and pi^_2
> pi.hat1 <- (0:n1)/n1
> pi.hat2 <- (0:n2)/n2
> pi.hat.all <- expand.grid(pi.hat1 = pi.hat1, pi.hat2 = pi.hat2)

> # Find joint probability for w1 and w2
> prob.w1 <- dbinom(x = 0:n1, size = n1, prob = pi1)
> prob.w2 <- dbinom(x = 0:n2, size = n2, prob = pi2)
```

```
> prob.all <- expand.grid(prob.w1 = prob.w1, prob.w2 = prob.w2)
> pmf <- prob.all$prob.w1*prob.all$prob.w2

> # P(W1 = w1, W2 = w2)
> head(data.frame(w.all, pmf = round(pmf,4)))
  w1 w2    pmf
1  0  0 0.0006
2  1  0 0.0016
3  2  0 0.0018
4  3  0 0.0012
5  4  0 0.0005
6  5  0 0.0002
```

For example, the probability of observing $P(W_1 = 1,\ W_2 = 0) = 0.0016$. Using these probabilities, we calculate the true confidence level for the interval:

```
> var.wald <- pi.hat.all[,1] * (1-pi.hat.all[,1]) / n1 +
    pi.hat.all[,2] * (1-pi.hat.all[,2]) / n2
> lower <- pi.hat.all[,1] - pi.hat.all[,2] - qnorm(p = 1-alpha/2)
    * sqrt(var.wald)
> upper <- pi.hat.all[,1] - pi.hat.all[,2] + qnorm(p = 1-alpha/2)
    * sqrt(var.wald)
> save <- ifelse(test = pi1-pi2 > lower, yes = ifelse(test =
    pi1-pi2 < upper, yes = 1, no = 0), no = 0)
> sum(save*pmf)
[1] 0.9281274
> data.frame(w.all, round(data.frame(pmf, lower, upper),4),
    save)[1:15,]
   w1 w2    pmf   lower  upper save
1   0  0 0.0006  0.0000 0.0000    0
2   1  0 0.0016 -0.0859 0.2859    0
3   2  0 0.0018 -0.0479 0.4479    0
4   3  0 0.0012  0.0160 0.5840    0
5   4  0 0.0005  0.0964 0.7036    0
6   5  0 0.0002  0.1901 0.8099    0
7   6  0 0.0000  0.2964 0.9036    0
8   7  0 0.0000  0.4160 0.9840    0
9   8  0 0.0000  0.5521 1.0479    0
10  9  0 0.0000  0.7141 1.0859    0
11 10  0 0.0000  1.0000 1.0000    0
12  0  1 0.0043 -0.2859 0.0859    1
13  1  1 0.0108 -0.2630 0.2630    1
14  2  1 0.0122 -0.2099 0.4099    1
15  3  1 0.0081 -0.1395 0.5395    0
```

All possible Wald intervals are calculated, and the `ifelse()` function is used to check if $\pi_1 - \pi_2 = 0.2 - 0.4 = -0.2$ is within each interval. The last data frame shows the first 15 intervals with the results from this check. The probabilities corresponding to the intervals that contain -0.2 are summed to produce the true confidence level of 0.9281, which is less than the stated level of 0.95.

We can also calculate the true confidence level while holding one of the probabilities constant and allowing the other to vary. Figure 1.4 gives a plot where $\pi_2 = 0.4$ and π_1 varies from 0.001 to 0.999 by 0.0005. We exclude the R code here because it is

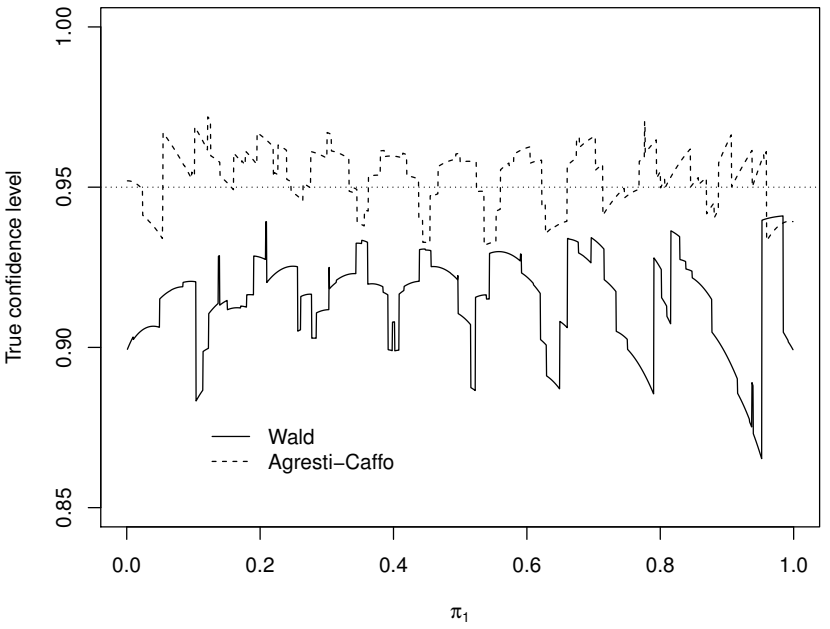

Figure 1.4: True confidence levels with $n_1 = 10$, $n_2 = 10$, $\pi_2 = 0.4$, and $\alpha = 0.05$.

quite similar to what was used in Section 1.1.3, where now we use the `for()` function to loop over the different values of π_1. Both the Agresti-Caffo and Wald lines are drawn on the plot simultaneously by using the `plot()` function first for the Wald true confidence levels and then using the `lines()` function for the Agresti-Caffo true confidence levels. The `legend()` function places the legend on the plot. Please see the program corresponding to this example for the code.

Figure 1.4 shows that the Wald interval never achieves the true confidence level! The Agresti-Caffo interval has a true confidence level between 0.93 and 0.97. We encourage readers to change the pre-set value for π_2 in the program to examine what happens in other situations. For example, the Wald interval is extremely liberal and the Agresti-Caffo interval is extremely conservative for small π_1 when $\pi_2 = 0.1$ with $n_1 = 10$, $n_2 = 10$, and $\alpha = 0.05$.

We can also allow π_2 to vary by the same increments as π_1 in order to produce a three-dimensional plot with the true confidence level on the third axis. The R code is in the corresponding program to this example. Two `for()` function calls—one loop for π_2 and one loop for π_1—are used within the code. Once all true confidence levels are found, the `persp3d()` function from the `rgl` package of R produces an interactive three-dimensional plot. Using the left and right mouse buttons inside the plot window, the plot can be rotated and zoomed in, respectively. Figure 1.5 gives separate plots for the Wald (left) and Agresti-Caffo (right) intervals. For both plots, a plane is drawn at the 0.95 stated confidence level. We can see that the Wald interval never achieves the stated confidence level, but the Agresti-Caffo interval does a much better job. We encourage the reader to construct these plots in order to see the surfaces better through rotating them.

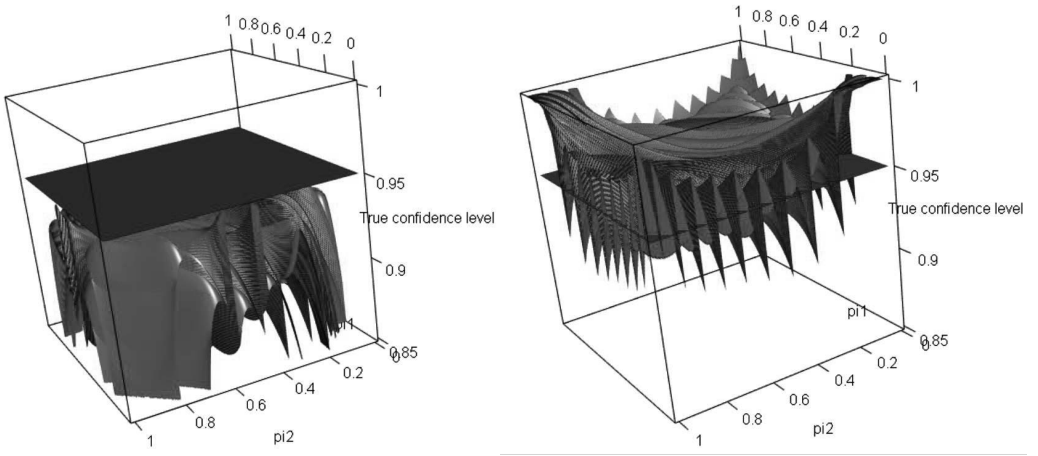

Figure 1.5: True confidence levels with $n_1 = 10$, $n_2 = 10$, and $\alpha = 0.05$. The left-side plot is for the Wald interval, and the right-side plot is for the Agresti-Caffo interval. Note that true confidence levels less than 0.85 are excluded from the Wald interval plot.

These true confidence level calculations can also be made through Monte Carlo simulation. We provide an example showing how this is done in our corresponding program.

1.2.3 Test for the difference of two probabilities

A formal test of the hypotheses $H_0 : \pi_1 - \pi_2 = 0$ vs. $H_a : \pi_1 - \pi_2 \neq 0$ can be conducted, again in several ways. A Wald test uses a test statistic

$$Z_W = \frac{\hat{\pi}_1 - \hat{\pi}_2}{\sqrt{\frac{\hat{\pi}_1(1-\hat{\pi}_1)}{n_1} + \frac{\hat{\pi}_2(1-\hat{\pi}_2)}{n_2}}},$$

and compares this statistic against the standard normal distribution. Note that the denominator contains the estimated variance of $\hat{\pi}_1 - \hat{\pi}_2$ without regard to the null hypothesis.

Probability distributions for test statistics are generally computed assuming that null hypothesis is true. In the present context, that means that the two group probabilities are equal, and so a better estimated variance than what is in Z_W can be computed by assuming that $\pi_1 = \pi_2$. Notice that this condition implies that Y_1 and Y_2 have the same distribution. Thus, W_1 and W_2 are both counts of successes from the same Bernoulli random variable, and therefore w_1 and w_2 can be combined to represent w_+ successes in n_+ trials. Let $\bar{\pi} = w_+/n_+$ be the estimated probability of success when the null hypothesis is true. Then it can be shown that $\widehat{Var}(\hat{\pi}_1 - \hat{\pi}_2) = \bar{\pi}(1-\bar{\pi})(1/n_1 + 1/n_2)$. This leads to a test based on comparing the statistic

$$Z_0 = \frac{\hat{\pi}_1 - \hat{\pi}_2}{\sqrt{\bar{\pi}(1-\bar{\pi})(1/n_1 + 1/n_2)}}$$

to a standard normal distribution. This is the score test.

A more general procedure that is used for comparing observed counts to estimated expected counts from any hypothesized model is the Pearson chi-square test. This particular test is often used to perform hypothesis tests in a contingency table setting, and we will spend much more time discussing it in Section 3.2. The model

that is implied by our null hypothesis is a single binomial with n_+ trials and probability of success $\bar{\pi}$. The test statistic is formed by computing (observed count − estimated expected count)^2/(estimated expected count) over *all* observed counts, meaning here both successes and failures in the two groups. The estimated expected number of successes in group j under the null hypothesis model is $n_j\bar{\pi}$, and similarly the expected number of failures is $n_j(1-\bar{\pi})$. Thus, the Pearson chi-square test statistic is

$$X^2 = \sum_{j=1}^{2}\left(\frac{(w_j - n_j\bar{\pi})^2}{n_j\bar{\pi}} + \frac{(n_j - w_j - n_j(1-\bar{\pi}))^2}{n_j(1-\bar{\pi})}\right). \tag{1.7}$$

This can be simplified to

$$X^2 = \sum_{j=1}^{2}\frac{(w_j - n_j\bar{\pi})^2}{n_j\bar{\pi}(1-\bar{\pi})}.$$

The X^2 statistic has a distribution that is approximately χ^2_1 when n_1 and n_2 are large and when the null hypothesis is true. If the null hypothesis is false, the observed counts tend not to be close to what is expected when the null hypothesis is true; thus, large values of X^2 relative to the χ^2_1 distribution lead to a rejection of the null hypothesis. It can be shown that the Pearson chi-square and score test results are identical for this setting, because $X^2 = Z_0^2$ (see Exercise 25) and the χ^2_1 distribution is equivalent to the distribution of a squared standard normal random variable (e.g., $Z^2_{0.975} = \chi^2_{1,0.95} = 3.84$). For those readers unfamiliar with the latter result, please see p. 53 of Casella and Berger (2002) for a derivation if desired.

A LRT can also be conducted. The test statistic can be shown to be

$$\begin{aligned}-2\log(\Lambda) = -2\Bigg[&w_1\log\left(\frac{\bar{\pi}}{\hat{\pi}_1}\right) + (n_1 - w_1)\log\left(\frac{1-\bar{\pi}}{1-\hat{\pi}_1}\right) + w_2\log\left(\frac{\bar{\pi}}{\hat{\pi}_2}\right)\\ &+(n_2 - w_2)\log\left(\frac{1-\bar{\pi}}{1-\hat{\pi}_2}\right)\Bigg]\end{aligned} \tag{1.8}$$

where we take $0 \times \log(\infty) = 0$ by convention. The null hypothesis is rejected if $-2\log(\Lambda) > \chi^2_{1,1-\alpha}$.

For all of these tests, the use of the standard normal or chi-squared distribution is based on large-sample approximations. The tests are asymptotically equivalent, meaning that they will give essentially the same results in very large samples. In small samples, however, the three test statistics can have distributions under the null hypothesis that are quite different from their approximations. Larntz (1978) compared the score, the LRT, and three other tests in various small-sample settings and found that the score test clearly maintains its size better than the others.[9] Thus, the score test is recommended here, as it was for testing the probability from a single group.

Example: Larry Bird's free throw shooting (Bird.R)

The purpose of this example is to show how to perform the score test, Pearson chi-square test, and LRT in R. We can use the `prop.test()` function to perform the score and Pearson chi-square tests:

[9]The size of a testing procedure is the probability that it rejects the null hypothesis when the null hypothesis is true. A test that holds the correct size is one that rejects at a rate equal to the type I error level of α.

```
> prop.test(x = c.table, conf.level = 0.95, correct = FALSE)

  2-sample test for equality of proportions without continuity
      correction

data:  c.table
X-squared = 0.2727, df = 1, p-value = 0.6015
alternative hypothesis: two.sided
95 percent confidence interval:
-0.11218742  0.06227017
sample estimates:
   prop 1    prop 2
0.8807018 0.9056604
```

The argument value for `x` is the contingency table. Alternatively, we could have assigned `x` a vector with w_1 and w_2 within it and used a new argument `n` with a vector value of n_1 and n_2 (see corresponding program for an example). The `correct = FALSE` argument value guarantees that the test statistic is calculated as shown by Z_0; otherwise, a *continuity correction* is applied to help ensure that the test maintains its size at or below α.[10]

The output gives the test statistic value as $Z_0^2 = 0.2727$ and a p-value of $P(A > 0.2727) = 0.6015$, where A has a χ^2_1 distribution. The decision is to not reject the null hypothesis. The conclusion is that there is not a significant change in Bird's second free throw success percentage over the possible outcomes of the first attempt. Note that the `chisq.test()` function and the `summary.table()` method function also provide ways to perform the Pearson chi-square test. We will discuss these functions in Section 3.2.

The code for the LRT is shown below:

```
> pi.bar <- colSums(c.table)[1]/sum(c.table)
> log.Lambda <- c.table[1,1] * log(pi.bar / pi.hat.table[1,1]) +
    c.table[1,2] * log((1-pi.bar) / (1-pi.hat.table[1,1])) +
    c.table[2,1] * log(pi.bar / pi.hat.table[2,1]) + c.table[2,2]
    * log((1-pi.bar) /(1-pi.hat.table[2,1]))
> test.stat <- -2*log.Lambda
> crit.val <- qchisq(p = 0.95, df = 1)
> p.val <- 1-pchisq(q = test.stat, df = 1)
> round(data.frame(pi.bar, test.stat, crit.val, p.val, row.names
    = NULL), 4)
  pi.bar test.stat crit.val p.val
1 0.8846    0.2858   3.8415 0.593
```

Under the null hypothesis, the estimate of the probability of success parameter is found using the sum of the counts in the first column of `c.table` divided by the total sample size, and the result is put into `pi.bar`. The code for the `log.Lambda` object shows how to convert most of Equation 1.8 into the correct R syntax. The transformed test

[10]Note that the test statistic Z_0 is a discrete random variable. Modifications to Z_0 (or any other test statistic that is a discrete random variable) called *continuity corrections* are sometimes made, and they can be helpful when using a continuous distribution to approximate a discrete distribution. These corrections often lead to very conservative tests (i.e., reject the null hypothesis at a rate less than α when the null hypothesis is true), so they are not often used. Alternative procedures are discussed in Section 6.2.

statistic is $-2\log(\Lambda) = 0.2858$, and the p-value is $P(A > 0.2858) = 0.5930$. These are nicely printed using the `data.frame()` function, where the `row.names = NULL` argument value prevents the printing of an errant row name. The overall conclusion is the same as for the score test. Note that the test statistic and p-value could have been calculated a little more easily using the `assocstats()` function of the `vcd` package, and this function also gives the Pearson chi-square test statistic as well. We show how to use this function in the corresponding program to this example.

We conclude this example with a few additional notes:

- Notice that the success probability conditioning on the first free throw being missed was subtracted from the success probability conditioning on the first free throw being made. This is especially important to know if a one-side hypothesis test was performed or if 0 was outside of a confidence interval. For example, many basketball fans think that a missed first free throw has a negative impact on the second free throw outcome. If the lower limit of the interval had been positive (i.e., the whole interval is above 0), it would have confirmed this line of thinking with respect to Larry Bird.

- As with other applications of statistics, care needs to be taken when interpreting the results with respect to the population. For example, suppose we wanted to make some claims regarding all of Larry Bird's past, present, and future free throw pairs when the data was collected. Strictly speaking, a random sample would need to be taken from this entire population to formally make statistical inferences. Random samples are often not possible in a sports setting, as in our example where we have data from the 1980-1 and 1981-2 NBA seasons. Inference on a broader population of free throws may or may not be appropriate. For example, Larry Bird's free throw shooting success rate may have changed from year to year due to practice or injuries.

- In addition to the sampling problem, Larry Bird's career concluded in 1992. Therefore, the population data may be obtainable, and we could actually calculate population parameters such as π_1 and π_2. Statistical inference would not be necessary then.

1.2.4 Relative risks

The problem with basing inference on $\pi_1 - \pi_2$ is that it measures a quantity whose meaning changes depending on the sizes of π_1 and π_2. For example, consider two hypothetical scenarios where the probability of disease is listed for two groups of people, say for smokers (π_1) and for nonsmokers (π_2):

1. $\pi_1 = 0.51$ and $\pi_2 = 0.50$
2. $\pi_1 = 0.011$ and $\pi_2 = 0.001$.

In both cases $\pi_1 - \pi_2 = 0.01$. But in the first scenario, an increase of 0.01 due to smoking is rather small relative to the already sizable risk of disease in the nonsmoking population. On the other hand, scenario 2 has smokers with 11 times the chance of disease than nonsmokers. We need to be able to convey the relative magnitudes of these changes better than differences allow.

In this instance, a preferred way to compare two probabilities is through the *relative risk*, $RR = \pi_1/\pi_2$ (assuming $\pi_2 \neq 0$). For the example above, $RR = 0.011/0.001 = 11.0$ for

the second scenario meaning that smokers are *11 times as likely* to have the disease than nonsmokers. Alternatively, we could say that smokers are 10 times *more* likely to have the disease than nonsmokers. On the other hand, for the first scenario, $RR = .51/.50 = 1.02$, indicating that smokers are just 2% more likely (or 1.02 times as likely) to have the disease. Notice that when $\pi_1 = \pi_2$, $RR = 1$.

These numerical values are based on population probabilities. To obtain an MLE for RR, we can make use of the invariance property of MLEs described in Appendix B.4 that allows us to substitute the observed proportions for the probabilities, $\widehat{RR} = \hat{\pi}_1/\hat{\pi}_2$, assuming $\hat{\pi}_2 \neq 0$. It is this estimate that is often given in news reports that state risks associated with certain factors such as smoking or obesity.

Because $\widehat{RR}$ is an MLE, inference can be carried out using the usual procedures. It turns out, however, that the normal approximation is rather poor for MLEs that are ratios, especially when the estimate in the denominator may have non-negligible variability as is the case here. Therefore, inference based on a normal approximation for $\widehat{RR}$ is not recommended. However, the normal approximation holds somewhat better for $\log(\widehat{RR}) = \log(\hat{\pi}_1) - \log(\hat{\pi}_2)$—the MLE for $\log(RR)$— so inference is generally carried out on the log scale. The variance estimate for $\log(\widehat{RR})$ can be derived by the delta method (Appendix B.4.2) as

$$\widehat{Var}(\log(\widehat{RR})) = \frac{1-\hat{\pi}_1}{n_1\hat{\pi}_1} + \frac{1-\hat{\pi}_2}{n_2\hat{\pi}_2} = \frac{1}{w_1} - \frac{1}{n_1} + \frac{1}{w_2} - \frac{1}{n_2}.$$

A $(1-\alpha)100\%$ Wald confidence interval for the population relative risk is found by first computing the confidence interval for $\log(\pi_1/\pi_2)$,

$$\log\left(\frac{\hat{\pi}_1}{\hat{\pi}_2}\right) \pm Z_{1-\alpha/2}\sqrt{\frac{1}{w_1} - \frac{1}{n_1} + \frac{1}{w_2} - \frac{1}{n_2}}.$$

The exponential transformation is then used to find the Wald interval for the relative risk itself:

$$\exp\left[\log\left(\frac{\hat{\pi}_1}{\hat{\pi}_2}\right) \pm Z_{1-\alpha/2}\sqrt{\frac{1}{w_1} - \frac{1}{n_1} + \frac{1}{w_2} - \frac{1}{n_2}}\right],$$

where $\exp(\cdot)$ is the inverse of the natural log function ($b = \exp(a)$ is equivalent to $a = \log(b)$). When w_1 and/or w_2 are equal to 0, the confidence interval cannot be calculated. One ad-hoc adjustment is to add a small constant, such as 0.5, to the 0 cell count and the corresponding row total. For example, if $w_1 = 0$, replace w_1 with 0.5 and n_1 with $n_1 + 0.5$. Exercise 32 will investigate how well the interval achieves its stated confidence level.

Example: Salk vaccine clinical trial (Salk.R)

The purpose of this example is to calculate the estimated relative risk and the confidence interval for the population relative risk in order to determine the effectiveness of the Salk vaccine in preventing polio. Below is the R code used to enter the data into an array and to perform the necessary calculations:

```
> c.table <- array(data = c(57, 142, 200688, 201087), dim =
    c(2,2), dimnames = list(Treatment = c("vaccine", "placebo"),
    Result = c("polio", "polio free")))
> c.table
          Result
Treatment polio polio free
  vaccine    57     200688
  placebo   142     201087
```

```
> pi.hat.table <- c.table/rowSums(c.table)
> pi.hat.table
          Result
Treatment        polio polio free
  vaccine 0.0002839423  0.9997161
  placebo 0.0007056637  0.9992943

> pi.hat1 <- pi.hat.table[1,1]
> pi.hat2 <- pi.hat.table[2,1]

> round(pi.hat1/pi.hat2, 4)
[1] 0.4024
> round(1/(pi.hat1/pi.hat2), 4)  # inverted
[1] 2.4852

> alpha <- 0.05
> n1 <- sum(c.table[1,])
> n2 <- sum(c.table[2,])

> # Wald confidence interval
> var.log.rr <- (1-pi.hat1)/(n1*pi.hat1) +
    (1-pi.hat2)/(n2*pi.hat2)
> ci <- exp(log(pi.hat1/pi.hat2) + qnorm(p = c(alpha/2,
    1-alpha/2)) * sqrt(var.log.rr))
> round(ci, 4)
[1] 0.2959 0.5471
> rev(round(1/ci, 4))  # inverted
[1] 1.8278 3.3792
```

Defining index 1 to represent the vaccine group and 2 the placebo group, we find $\widehat{RR} = 0.40$. The estimated probability of contracting polio is only 0.4 times (or 40%) as large for the vaccine group than for placebo. Notice that we used the word *estimated* with this interpretation because we are using parameter estimates. The confidence interval is $0.30 < RR < 0.55$. Therefore, with 95% confidence, we can say that the vaccine reduces the *population* risk of polio by 45-70%. Also, notice the exponential function is calculated using `exp()` in R (for example, `exp(1)` is 2.718).

The ratio for the relative risk is often arranged so that $\widehat{RR} \geq 1$. This allows for an appealing interpretation that the group represented in the numerator has a risk that is, for example, "11 times as large" as the denominator group. This can be easier for a target audience to appreciate than the alternative, "0.091 times as large". However, the application may dictate which group should be the numerator regardless of the sample proportions. This was the case for the Salk vaccine example, where it is natural to think of the vaccine in terms of its risk reduction.

Also, relative risk is generally a more useful measure than the difference between probabilities when the probabilities are fairly small. It is of limited use otherwise. For example, if $\pi_2 = 0.8$, then the maximum possible relative risk is $1/0.8$=1.25. It is therefore useful to have an alternative statistic for comparing probabilities that is applicable regardless of the sizes of the probabilities. This is part of the motivation for the next measure.