

Springer Texts in Statistics

Series Editors:

G. Casella

S.E. Fienberg

I. Olkin

For further volumes:

<http://www.springer.com/series/417>

Mary Kathryn Cowles

Applied Bayesian Statistics

With R and OpenBUGS Examples



Springer

Mary Kathryn Cowles
Department of Statistics
and Actuarial Science
University of Iowa
Iowa City, Iowa, USA

ISSN 1431-875X

ISBN 978-1-4614-5695-7

ISBN 978-1-4614-5696-4 (eBook)

DOI 10.1007/978-1-4614-5696-4

Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2012951150

© Springer Science+Business Media New York 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Brendan, Lucy, and Donald.

Preface

I have taught a course called “Bayesian Statistics” at the University of Iowa every academic year since 1998–1999. This book is intended to fit the goals and audience addressed by my course. The “Course Objectives” section of my syllabus reads:

Through hands-on experience with real data from a variety of applications, students will learn the basics of designing and carrying out Bayesian analyses, and interpreting and communicating the results. Students will learn to use software packages including R and OpenBUGS to fit Bayesian models.

The course is intended to be intensely practical, focussing on building understanding of the concepts and procedures required to perform Bayesian analysis of real data to answer real questions. Emphasis is given to such issues as determining what data is needed to address a particular question; choosing an appropriate probability distribution for sample data; quantifying already-existing knowledge in the form of a prior distribution on model parameters; verifying that the posterior distribution will be proper if improper prior distributions are used; and when and how to specify hierarchical models. Interpretation and communication of results are stressed, including differences from, and similarities to, classical approaches to the same problems.

WinBUGS and OpenBUGS currently are the dominant software in applied use of Bayesian methods. I have chosen to introduce OpenBUGS as the primary data analysis software in this textbook because, unlike WinBUGS, OpenBUGS is undergoing continuing development and has versions that run natively under Linux and Macintosh operating systems as well as Windows. Although some background is provided on the Markov chain Monte Carlo sampling procedures employed by WinBUGS and OpenBUGS, the emphasis is on those tasks that a *user* must carry out correctly for reasonably trustworthy inference. These include using appropriate tools to assess whether and when a sampler has converged to the target distribution, deciding how many iterations are needed for acceptable accuracy in estimation, and how to report results of a Bayesian analysis conducted with OpenBUGS. Caveats about the fallibility of convergence diagnostics are emphasized.

Students of different levels and disciplines take the course, including: undergraduate mathematics and statistics majors; master's students in statistics, biostatistics, statistical genetics, educational testing and measurement, and engineering; and PhD students in economics, marketing, psychology, and geography as well as the previously listed fields. In addition, several practicing statisticians employed by the University of Iowa and American College Testing (ACT) have taken the course.

The goal of the course, and of this book, is to provide an introduction to Bayesian principles and practice that is clear, useful, and unintimidating to motivated students even if they do not have an advanced background in mathematics and probability. I emphasize intuitive insight without sacrificing mathematical correctness. Prerequisites are one or two semesters of calculus-based probability and mathematical statistics (at least at the Hogg and Tannis level) and one or two semesters of classical statistical methods, including linear regression (David Moore's *Basic Practice of Statistics* level). Elementary integral and differential calculus is occasionally used in lectures and homework. Linear algebra is not required.

Coralville, Iowa

Mary Kathryn Cowles

Contents

1	What Is Bayesian Statistics?	1
1.1	The Scientific Method (But It Is Not Just for Science...)	1
1.2	A Bit of History	2
1.3	Example of the Bayesian Method: Does My Friend Have Breast Cancer?	3
1.3.1	Quantifying Uncertainty Using Probabilities	3
1.3.2	Models and Prior Probabilities	5
1.3.3	Data	6
1.3.4	Likelihoods and Posterior Probabilities	7
1.3.5	Bayesian Sequential Analysis	8
1.4	Calibration Experiments for Assessing Subjective Probabilities...	8
1.5	What Is to Come?	10
	Problems	11
2	Review of Probability	13
2.1	Review of Probability	13
2.1.1	Events and Sample Spaces	13
2.1.2	Unions, Intersections, Complements	14
2.1.3	The Addition Rule	15
2.1.4	Marginal and Conditional Probabilities	15
2.1.5	The Multiplication Rule	17
2.2	Putting It All Together: Did Brendan Mail the Bill Payment?	17
2.2.1	The Law of Total Probability	17
2.2.2	Bayes' Rule in the Discrete Case	19
2.3	Random Variables and Probability Distributions	20
	Problems	21
3	Introduction to One-Parameter Models: Estimating a Population Proportion	25
3.1	What Proportion of Students Would Quit School If Tuition Were Raised 19%: Estimating a Population Proportion	25

3.2	The First Stage of a Bayesian Model	25
3.2.1	The Binomial Distribution for Our Survey	26
3.2.2	Kernels and Normalizing Constants	27
3.2.3	The Likelihood Function	27
3.3	The Second Stage of the Bayesian Model: The Prior	28
3.3.1	Other Possible Prior Distributions	29
3.3.2	Prior Probability Intervals	31
3.4	Using the Data to Update the Prior: The Posterior Distribution ...	32
3.5	Conjugate Priors	34
3.5.1	Computing the Posterior Distribution with a Conjugate Prior	34
3.5.2	Choosing the Parameters of a Beta Distribution to Match Prior Beliefs	35
3.5.3	Computing and Graphing the Posterior Distribution	38
3.5.4	Plotting the Prior Density, the Likelihood, and the Posterior Density	38
3.6	Introduction to R for Bayesian Analysis	38
3.6.1	Functions and Objects in R	39
3.6.2	Summarizing and Graphing Probability Distributions in R	42
3.6.3	Printing and Saving R Graphics	44
3.6.4	R Packages Useful in Bayesian Analysis	44
3.6.5	Ending a Session	46
	Problems	46
4	Inference for a Population Proportion	49
4.1	Estimation and Testing: Frequentist Approach	49
4.1.1	Maximum Likelihood Estimation	49
4.1.2	Frequentist Confidence Intervals	51
4.1.3	Frequentist Hypothesis Testing	52
4.2	Bayesian Inference: Summarizing the Posterior Distribution	54
4.2.1	The Posterior Mean	54
4.2.2	Other Bayesian Point Estimates	55
4.2.3	Bayesian Posterior Intervals	57
4.3	Using the Posterior Distribution to Test Hypotheses	59
4.4	Posterior Predictive Distributions	61
	Problems	63
5	Special Considerations in Bayesian Inference	67
5.1	Robustness to Prior Specifications	67
5.2	Inference Using Nonconjugate Priors	69
5.2.1	Discrete Priors	69
5.2.2	A Histogram Prior	71
5.3	Noninformative Priors	72
5.3.1	Review of Proper and Improper Distributions	72
5.3.2	A Noninformative Prior for the Binomial Likelihood	73

5.3.3	Jeffreys Prior	73
5.3.4	Verifying the Propriety of the Posterior Distribution When Using an Improper Prior	77
	Problems	78
6	Other One-Parameter Models and Their Conjugate Priors	81
6.1	Poisson	81
6.2	Normal: Unknown Mean, Variance Assumed Known	81
6.2.1	Example: Mercury Concentration in the Tissue of Edible Fish	82
6.2.2	Parametric Family for Likelihood	83
6.2.3	Likelihood for μ Assuming that Population Variance Is Known	85
6.2.4	Sufficient Statistics	85
6.2.5	Finding a Conjugate Prior for μ	86
6.2.6	Updating from Prior to Posterior in the Normal–Normal Case	86
6.2.7	Specifying Prior Parameters	88
6.2.8	Mercury in Fish Tissue	89
6.2.9	The Jeffreys Prior for the Normal Mean	92
6.2.10	Posterior Predictive Density in the Normal–Normal Model	93
6.3	Normal: Unknown Variance, Mean Assumed Known	94
6.3.1	Conjugate Prior for the Normal Variance, μ Assumed Known	95
6.3.2	Obtaining the Posterior Density	96
6.3.3	Jeffreys Prior for Normal Variance, Mean Assumed Known	97
6.4	Normal: Unknown Precision, Mean Assumed Known	97
6.4.1	Inference for the Variance in the Mercury Concentration Problem	98
	Problems	99
7	More Realism Please: Introduction to Multiparameter Models	101
7.1	Conventional Noninformative Prior for a Normal Likelihood with Both Mean and Variance Unknown	102
7.1.1	Example: The Mercury Concentration Data	104
7.2	Informative Priors for μ and σ^2	106
7.3	A Conjugate Joint Prior Density for the Normal Mean and Variance	106
7.3.1	Example: The Mercury Contamination Data	108
7.3.2	The Standard Noninformative Joint Prior as a Limiting Form of the Conjugate Prior	109
	Problems	110

8	Fitting More Complex Bayesian Models: Markov Chain Monte Carlo	111
8.1	Why Sampling-Based Methods Are Needed	111
8.1.1	Single-Parameter Model Example	111
8.1.2	Numeric Integration	113
8.1.3	Monte Carlo Integration	118
8.2	Sampling-Based Methods	120
8.2.1	Independent Sampling	120
8.3	Introduction to Markov Chain Monte Carlo Methods	123
8.3.1	Markov Chains	123
8.3.2	Markov Chains for Bayesian Inference	124
8.4	Introduction to OpenBUGS and WinBUGS	125
8.4.1	Using OpenBUGS for the Problem of Estimating a Binomial Success Parameter	126
8.4.2	Model Specification	127
8.4.3	Data and Initial Values Files	127
8.4.4	Running the Model	128
8.4.5	Assessing Convergence in OpenBUGS	133
8.4.6	Posterior Inference Using OpenBUGS	138
8.4.7	OpenBUGS for Normal Models	140
8.5	Exercises	144
9	Hierarchical Models and More on Convergence Assessment	147
9.1	Specifying Bayesian Hierarchical Models Example: A Better Model for the College Softball Player's Batting Average	147
9.1.1	The First Stage: The Likelihood	148
9.1.2	The Second Stage: Priors on the Parameters That Appeared in the Likelihood	149
9.1.3	The Third Stage: Priors on Any Parameters That Do Not Already Have Them	150
9.1.4	The Joint Posterior Distribution in Hierarchical Models	150
9.1.5	Higher-Order Hierarchical Models	151
9.2	Fitting Bayesian Hierarchical Models	151
9.3	Estimation Based on Hierarchical Models	153
9.3.1	Prediction from Hierarchical Models	154
9.4	More on Convergence Assessment in WinBUGS/OpenBUGS	156
9.4.1	The Brooks Gelman and Rubin Diagnostic	158
9.4.2	Convergence in the Hierarchical Softball Example with a Vague Prior	162
9.5	Other Hierarchical Models	167
9.5.1	Hierarchical Normal Means	167
9.6	Directed Graphs for Hierarchical Models	170
9.6.1	Parts of a DAG	170

9.7	*Gibbs Sampling for Hierarchical Models	171
9.7.1	Deriving Full Conditional Distributions	172
9.8	Recommendations for Using MCMC to Fit Bayesian Models	174
9.8.1	How Many Chains	174
9.8.2	Initial Values	174
9.8.3	General Advice	175
9.9	Exercises	175
10	Regression and Hierarchical Regression Models	179
10.1	Review of Linear Regression	179
10.1.1	Centering the Covariate	180
10.1.2	Frequentist Estimation in Regression	180
10.1.3	Example: Mercury Deposited by Precipitation Near the Brule River in Wisconsin	181
10.2	Introduction to Bayesian Simple Linear Regression	187
10.2.1	Standard Noninformative Prior	187
10.2.2	Bayesian Analysis of the Brule River Mercury Concentration Data	189
10.2.3	Informative Prior Densities for Regression Coefficients and Variance	192
10.3	Generalized Linear Models	192
10.4	Hierarchical Normal Linear Models	194
10.4.1	Example: Estimating the Slope of Mean Log Mercury Concentration Throughout North America Using Data from Multiple MDN Sites	195
10.4.2	Stages of a Hierarchical Normal Linear Model	195
10.4.3	Univariate Formulation of the Second Stage	196
10.4.4	Bivariate Formulation of the Second Stage	196
10.4.5	Third Stage: Univariate Formulation	197
10.4.6	Third Stage: Bivariate Formulation	197
10.4.7	The Wishart Density	198
10.5	WinBUGS Examples for Hierarchical Normal Linear Models	199
10.5.1	Example with Univariate Formulation at Second and Third Stages	200
10.5.2	Example with Bivariate Formulation at Second and Third Stages	202
	Problems	204
11	Model Comparison, Model Checking, and Hypothesis Testing	207
11.1	Bayes Factors for Model Comparison and Hypothesis Testing	207
11.1.1	Bayes Factors in the Simple/Simple Case	207
11.1.2	Interpreting a Bayes Factor	210
11.1.3	The Bayes Factor in More General Models	210
11.2	Bayes Factors and Bayesian Hypothesis Testing	212
11.2.1	Obtaining Posterior Probabilities from WinBUGS/OpenBUGS	214
11.2.2	Bayesian Viewpoint on Point Null Hypotheses	215

11.3 The Deviance Information Criterion	216
11.4 Posterior Predictive Checking	219
11.5 Exercises	223
Tables of Probability Distributions	225
References	227
Index	231

Chapter 1

What Is Bayesian Statistics?

1.1 The Scientific Method (But It Is Not Just for Science...)

In almost every field of human activity, people use data to further their learning and to guide decision-making and action. The following steps, paraphrased from [Berry \(1996\)](#), have been described as “the scientific method.” However, they are equally appropriate for use by a biologist seeking to better understand the behavior of monarch butterflies, the marketing director of a grocery store chain determining where to open a new store, or a new university graduate deciding whether to accept a particular job offer.

1. Define the question or problem to be addressed.
2. Assess the relevant information already available. Decide whether it is sufficient for the purpose at hand.
 - a. If yes, draw appropriate conclusions, make appropriate decisions, and take appropriate action.
 - b. If no, proceed to step 3.
3. Determine what additional information is needed and design a study or experiment to attempt to obtain it.
4. Carry out the study designed in step 3.
5. Use the data obtained in step 4 to update what was previously known. Return to step 2.

Statistics is central to steps 2, 3, and 5. *Bayesian* statistics is particularly well suited to steps 2 and 5, because it provides a quantitative framework for representing current knowledge and for rationally integrating new information.

1.2 A Bit of History

One could argue that the Great Fire of London in 1666 sparked the philosophy and methods that now are called Bayesian statistics. Destroying over 13,000 houses, 89 churches, and dozens of public buildings, the Great Fire led to the rise of insurance protection as we understand it today. The following year, one Nicholas Barbon opened an office to insure buildings, and in 1680, he established the first full-fledged fire insurance company in England. By the early eighteenth century, the idea of life insurance as well as property insurance was taking hold in England. However, lack of adequate vital statistics and of probability theory led to the failure of many early life insurers.

Enter Thomas Bayes. Born in London in 1702, Bayes became an ordained Presbyterian minister by profession and a mathematician and scientist by avocation. He applied his mind to the questions urgently raised by the insurers and laid out his resulting theory of probability in his *Essay towards solving a problem in the doctrine of chances*. After Bayes' death in 1761, his friend Richard Price sent the paper to the Royal Society of London. The paper was published in the *Philosophical Transactions of the Royal Society of London* in 1764.

Bayes' conclusions were accepted enthusiastically by Pierre-Simon Laplace and other contemporary leading probabilists. However, George Boole questioned them in his 1854 treatise on logic called *Laws of Thought*. Bayes' method became controversial in large part because mathematicians and scientists did not yet know how to treat prior probabilities (a topic that we will deal with throughout this book!). In the first half of the twentieth century, a different approach to statistical inference arose, which has come to be called the *frequentist* school. However, Bayesian thinking continued to progress with the works of Bruno de Finetti in Italy, Harold Jeffreys and Dennis Lindley in England, Jimmy Savage in the USA, and others.

Until about 1990, the application of Bayesian methods to statistical analysis in real-world problems was very limited because the necessary mathematical computations could be done analytically only for very simple models. In the early 1990s, the increasing accessibility of powerful computers, along with the development of new computing algorithms for fitting Bayesian models, opened the door to the use of Bayesian methods in complex, real-world applications. The subsequent explosion of interest in Bayesian statistics has led not only to extensive research in Bayesian methodology but also to the use of Bayesian methods to address pressing questions in diverse application areas such as astrophysics, weather forecasting, health-care policy, and criminal justice.

Today, Bayesian statistics is widely used to guide learning and decision-making in business and industry as well as in science. For example, software using Bayesian analysis guides Google's driverless robotic cars ([McGrayne 2011a](#)), and Bayesian methods have attained sufficiently wide acceptance in medical research that, in 2006, the United States Food and Drug Administration (FDA) put into place a set of guidelines for designing clinical trials of medical devices using Bayesian methods ("Guidance for the Use of Bayesian Statistics in Medical Device

Clinical Trials”, <http://www.fda.gov/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm071072.htm>). McGrayne (2011b) offers a lively introduction to the history and current status of Bayesian statistics.

1.3 Example of the Bayesian Method: Does My Friend Have Breast Cancer?

The National Cancer Institute recommends that women aged 40 and above should have mammograms every one to two years. A mammogram produces x-ray images of tissues and structure inside the breast and may help detect and identify cancerous tumors, benign cysts, and other breast conditions. A mammogram administered to a woman who has no signs or symptoms associated with breast cancer is called a “screening mammogram.”

A friend of mine recently was referred by her physician for her first screening mammogram. My friend does not have a family history of breast cancer, and before being referred for the screening mammogram, she had given no thought whatsoever to breast cancer as something that could conceivably happen to her. However, as the date of the mammogram approached, she began to wonder about her chances of being diagnosed with breast cancer. She was at step 1 of the scientific method—she had defined a question that she wanted to address. In other words, she was *uncertain* about her status with respect to breast cancer and wanted to learn more about it.

In the next sections, we will couch my friend’s learning process in the framework of the simplest possible application of Bayes’ rule within the scientific method. We will introduce the notion of using probabilities to quantify knowledge or uncertainty and of using data to update such probabilities in a rational way.

1.3.1 *Quantifying Uncertainty Using Probabilities*

In science, business, and daily life, people quantify uncertainty in the form of probabilities. The weather forecaster says there is a 30% probability of precipitation today; the seismologist says that there is a 21% chance of a major earthquake along the San Andreas fault by the year 2032; a doctor may tell a cancer patient that she has a 50% chance of surviving for 5 years or longer. Two different interpretations of *probability* are in common use.

1.3.1.1 The Long-Run Frequency Interpretation of Probability

In previous statistics or math classes, you undoubtedly have encountered the long-run frequency interpretation of the probability of an event. For example, Moore (2007, page 248) says:

The probability of any outcome of a random phenomenon is the proportion of the times the outcome would occur in a very long series of repetitions.

Coin flipping is an obvious example of this interpretation of probability. Saying that the probability of a fair coin coming up heads is 0.5 means that we expect to get a head about half the time if we flip the coin a huge number of times under exactly the same conditions.

Although this interpretation is useful (and the *frequentist* or *classical* approach to statistics is based on it), it has serious shortcomings. Trying to use the long-run frequency interpretation as a *definition* of probability results in a circular argument; see [Woodworth \(2004, page 25\)](#) for a summary of the mathematical issue. From a more intuitive standpoint, the frequency interpretation is limited to situations in which a sequence of repeatable experiments is possible (or at least imaginable). No frequency interpretation is possible for probabilities of many kinds of events about which we would like to quantify uncertainty. For example, in the winter of 2007–2008, editorial writers were assessing the probability that the United States economy was headed for a major recession. Thousands of homes purchased in the previous several years were in foreclosure, and some mortgage companies had gone bankrupt due to bad loans. The price of oil was over \$100 a barrel. Although everyone certainly wanted to know the probability that a recession was coming, obviously, the question could not be couched as the proportion of the time that countries facing exactly the economic, social, and political conditions that then existed in the USA would go into recession.

1.3.1.2 Subjective Probability

The subjective interpretation of probability is:

A probability of an event or of the truth of a statement is a number between 0 and 1 that quantifies a particular person's subjective opinion as to how likely that event is to occur (or to have already occurred) or how likely the statement is to be true.

This interpretation of probability clearly is not limited to repeatable events. Note that the subjective interpretation was of “a” probability, not “the” probability, of an event or statement. Not only may different people have different subjective probabilities regarding the same event, but the same person's subjective probability is likely to change as more information becomes available. (As we will see shortly, these updates to a person's subjective probability are where the mathematical identity called Bayes' rule comes in.)

Some people object to admitting that there is any place for subjectivity in science. However, that does not make it any less true that two different scientists can look at the same data (experimental results, observational results, or whatever) and come to different conclusions because of their previously acquired knowledge of the subject.

Here is a hypothetical example in which your own life experience and knowledge of the world might lead you to different conclusions from identical experimental results obtained from different applications. Suppose that I tell you that I have carried out a study consisting of 6 trials. Each trial has only two possible outcomes, which I choose to call “success” and “failure.” I believe that the probability of

success was the same for each trial and that the six trials were independent. I tell you that the outcome was six successes in six trials, and I ask you to predict the outcome if I carry out a seventh trial of my experiment.

After you think about this for a while, I offer to tell you what the experiment was. I explain that I randomly selected six dates from the 2011 calendar year, and on each of those dates, I looked out the window at 11:00 a.m. If I saw daylight, I recorded a success; if it was completely dark outside, I recorded a failure. (By the way, I live in Iowa, not in the extreme northerly or southerly latitudes.) Does this information affect your prediction regarding the outcome of a seventh trial?

Suppose that, instead, I explained my study as follows. Over the last 10 years, a family in my neighborhood has had six children. Each time a new baby was born, I asked the mother whether it was a boy or a girl. If the baby was a girl, I recorded a success; if a boy, I recorded a failure. (Obviously the choice of which gender to designate a “success” and which a “failure” is completely arbitrary!) Now the mother is pregnant again. Would your assessment of whether the seventh trial is likely to be another success be different in this case compared to the previous case of observing daylight?

1.3.1.3 Properties of Probabilities

To help my friend assess her chances of being diagnosed with breast cancer, we must recall two of the elementary properties of probability. Regardless of which interpretation of probability is being used, these must hold:

- Probabilities must not be negative. If A is any event and $P(A)$ denotes “the probability that A occurs,” then

$$P(A) \geq 0$$

- All possible outcomes of an experiment or random phenomenon, taken together, must have probability 1.

1.3.2 Models and Prior Probabilities

For my friend, there are two possible true states of the world:

1. She has breast cancer.
2. She does not have breast cancer.

We may refer to these statements as *models* or *hypotheses*—statements about a certain aspect of the real world, which could predict observable data.

Before obtaining any data specifically about her own breast cancer status, it would be rational for my friend to figure that her chance of being discovered to

Table 1.1 Models and prior probabilities

Model	Prior probability
Breast cancer	0.0045
No breast cancer	0.9955

have breast cancer is similar to that of a randomly selected person in the population of all women who undergo screening mammograms. Her physician tells her that published studies (Poplack et al. 2000) have shown that, among women who have a screening mammogram, the proportion who are diagnosed with breast cancer within 1 year is about 0.0045 (4.5 per 1,000).

Therefore, my friend assigns the following *prior probabilities* to the two models (Table 1.1).

Note that at this point, my friend has carried out step 2 in the scientific method from Sect. 1.1. Prior probabilities refer to probabilities assessed *before* new data are gathered in step 3.

1.3.3 Data

By actually having the mammogram, my friend will go on to step 3: She will collect *data*. Although there are several specific possible results of a mammogram, they may be grouped into just two possible outcomes: A “positive” mammogram indicates possible or likely cancer and results in a recommendation of further diagnostic procedures, and a “negative” mammogram does not give any evidence of cancer or any need for further procedures. We will use the notation $D+$ ($D-$) to represent the event that a person has (does not have) breast cancer and $M+$ ($M-$) to indicate the event that the person has a positive (negative) mammogram result.

The probabilities of the two possible mammogram outcomes are different depending on whether a person has breast cancer or not—that is, depending on which model is correct. In this case, these probabilities are properties of the particular test being used. A perfect screening test would always have a positive result if the person had the disease and a negative result if not. That is, for a perfect test, $P(M+|D+)$ would equal 1, and $P(M+|D-)$ would equal 0. (In this standard notation for conditional probability, the vertical bar is read “given” or “conditional on.”) However, perfect tests generally do not exist. A study (Poplack et al. 2000) of tens of thousands of people who received screening mammograms found that, if a person does have breast cancer (i.e., has a confirmed diagnosis of breast cancer within 1 year after the mammogram), then the probability that the screening mammogram will be positive is about 0.724. That is, $P(M+|D+) = 0.724$. This is called the *sensitivity* of a screening mammogram. Furthermore, for screening mammograms, $P(M+|D-) = 0.027$. The probabilities of a negative test under the two models are $P(M-|D-) = 0.973$ and $P(M-|D+) = 0.276$.

We can summarize all of this in Table 1.2. [These and subsequent tables are similar in structure to tables in Albert (1997).]

Table 1.2 Models, prior probabilities, and conditional probabilities of outcomes

Model	Prior Probabilities	$P(M+ \mid \text{Model})$	$P(M- \mid \text{Model})$
Breast cancer	0.0045	0.724	0.276
No breast cancer	0.9955	0.027	0.973

Table 1.3 Bayes' rule to go from prior probabilities to posterior probabilities

Model	Prior Probabilities	Likelihood for M+	Prior \times Likelihood	Posterior Probabilities
Breast cancer	0.0045	0.724	0.0033	0.107
No breast cancer	0.9955	0.0274	0.0273	0.893

1.3.4 Likelihoods and Posterior Probabilities

Bayes' rule is the mathematical rule for using data to update one's probabilities about models in a rational, systematic way. The *likelihood* associated with a particular model is the probability of the observed data under that model. Bayes' rule combines the prior probabilities with the likelihood to compute *posterior probabilities* for the respective models. Posterior probabilities are available only *after* the data are observed.

In its simplest form, Bayes' rule states:

$$P(\text{Model} \mid \text{Data}) \propto P(\text{Model}) \times P(\text{Data} \mid \text{Model})$$

or in words, the posterior probability of a model is proportional to the prior probability times the likelihood. Again, the symbol “ \mid ” is read “given” or “conditional on.” The symbol “ \propto ” means “is proportional to.”

My friend's mammogram came out positive (M+). Naturally, she was frightened and suspected that this meant she had breast cancer. However, she put the observation of the positive mammogram into the table and continued with the calculations for applying Bayes' rule.

The entries in the “Prior \times likelihood” column are not probabilities. Since “Breast cancer” and “No breast cancer” are the only possible states of the world, their probabilities must sum to 1. Bayes' rule says the posterior probabilities are *proportional* to, rather than *equal* to, the product of the prior probabilities and the likelihoods. To convert the products to probabilities, we must *normalize* them—divide each one by the sum of the two (0.0306 in this example). The result is shown in the “Posterior probabilities” column of Table 1.3.

The information from the mammogram increased my friend's probability that she had breast cancer by a factor of 24. Although the actual probability was still fairly small (a bit over 1/10), it was far too large to ignore. At her doctor's recommendation, my friend underwent a procedure called stereotactic core needle biopsy (SCNB).

Table 1.4 Updated posterior probabilities

Model	Prior Probabilities	Likelihood for S-	Prior \times Likelihood	Posterior Probabilities
Breast cancer	0.107	0.11	0.012	0.014
No breast cancer	0.893	0.94	0.839	0.986

1.3.5 Bayesian Sequential Analysis

SCNB is a procedure in which mammographic methods are used to guide the placement of a hollow needle into the suspicious part of the breast. Cells are drawn out through the needle and are examined under a microscope. If cancerous cells are observed, the test is considered positive (S+) and surgery is indicated. If no abnormal cells are observed, the test is negative (S−). According to [Bauer et al. \(1997\)](#), the sensitivity, or $Pr(S+|D+)$, of SCNB is 0.89 while the specificity, or $Pr(S−|D−)$, is 0.94. It is possible for even a highly skilled microbiologist to mistakenly identify normal cells as cancerous, leading to a probability of a false positive of $Pr(S+|D−) = 0.06$ (this is 1—specificity). Similarly, the probability of a false negative is $Pr(S−|D+) = 0.11$, which is 1—sensitivity.

In deciding to undergo SCNB, my friend was returning to step 2 of the scientific method, determining that the current information was insufficient and that she must obtain more data. We will assume that, conditional on her true disease status, the results from the two different tests are independent. This probably is a reasonable assumption in this setting. (We will revisit this topic in the next chapter when we discuss independence in more detail.)

After the mammogram, my friend's current probabilities were the *posterior* probabilities based on the mammogram. These became her *prior* probabilities with respect to the new data obtained from the SCNB. A Bayesian analysis in which accumulating information is incorporated over time, with the posterior probabilities from one step becoming the prior probabilities for the next step, is called a Bayesian sequential analysis.

The SCNB procedure on my friend showed no cancerous cells. The use of Bayes' theorem to update her probabilities regarding her disease status is shown in Table 1.4.

After incorporating the data from the SCNB, the posterior probability that my friend had breast cancer was only 0.014. Her doctor recommended no immediate surgery or other treatment but scheduled her for a follow-up mammogram in a year.

1.4 Calibration Experiments for Assessing Subjective Probabilities

Sometimes, we need to quantify our subjective probability of an event occurring in order to make a decision or take an action. For example, suppose that you have

been offered a job as a statistician with a marketing firm in Cincinnati. In order to decide whether to accept the job and move to Cincinnati, you wish to quantify your subjective probability of the event that you would like the job and would like Cincinnati.

A tool used to assess a person's degree of belief that a particular event will happen (or that it has already happened if the person is not in a position to know that) is the *calibration experiment*. This is a hypothetical random experiment, the possible outcomes of which are *equally likely in the opinion of the person whose subjective probability is being quantified*. For example, imagine that I promise to buy you dinner if you correctly call a coin flip. If you have no preference for heads or tails, then for you, the two possible outcomes are equally likely. Calibration experiments may be useful if either the person is not knowledgeable or comfortable with probability or the person is uncertain as to his or her opinion about the event.

The central idea of a calibration experiment to assess someone's subjective probability of an event is that the person is offered a hypothetical choice between two games that provide a chance at winning a desirable prize: a realization of the calibration experiment with known probability of success or through the occurrence of the event of interest. Which game the person chooses brackets his or her subjective probability within a particular interval. The games are imaginary—no prizes actually change hands. Based on the person's choice at one step, the calibration experiment is adjusted, and a new pair of games offered.

The following example illustrates the procedure. You are thinking about the representation of women among university faculty in the mathematical and physical sciences, and you begin to wonder how many women are on the faculty of the Department of Physics at Florida State University (FSU). Let's use the symbol A to represent the event that the FSU physics department has more than two female faculty members and $P_s(A)$ to represent your subjective probability that this event has occurred (i.e., that the statement is true). (If you attend FSU or have knowledge of its physics faculty, please think of another example for yourself—and don't give away the answer to your classmates!)

The calibration experiment will be flipping one or more coins, which *you* believe to be fair (i.e., each coin has a 50/50 chance of coming up heads when flipped). We don't need to try to determine the exact value of $P_s(A)$ —perhaps it will be accurate enough if we can produce an interval no wider than 0.125 (1/8) that contains this subjective probability. The monetary prizes mentioned in the games are purely imaginary (sorry!).

For step 1, in assessing $P_s(A)$, you are given a choice of the following two (hypothetical!) games through which you may try to win \$100.00:

- Game 1 A neutral person will flip one coin. I will pay you \$100 if the coin comes up heads. I will pay you nothing otherwise.
- Game 2 I will pay you \$100 if the physics department at FSU has more than two female faculty. I will pay you nothing if it has two or fewer.

If you choose Game 1, then I conclude that

$$0 < P_s(A) < 0.5$$

Now, the most efficient way to construct the steps in assessing subjective probability (i.e., the method that produces the shortest interval in the fewest steps) is to design the calibration experiment at each step so that the probability of winning through the calibration experiment is the midpoint of the interval produced at the previous step. (If you are familiar with the notion of a *binary search*, you will recognize this strategy.) Accordingly, I will set up step 2 in assessing $P_s(A)$ so that your chance of winning the \$100 prize through coin flipping is 0.25. Here is the choice of games at step 2:

- Game 1 A neutral person will flip two fair coins. I will pay you \$100 if both coins come up heads. I will pay you nothing otherwise.
- Game 2 I will pay you \$100 if the physics department at FSU has more than two female faculty. I will pay you nothing if it has two or fewer.

If you choose Game 2, then I conclude that your

$$0.25 < P_s(A) < 0.50$$

Since this interval is wider than our goal of 0.125, we need to proceed to step 3. In Exercise 1.5, you will verify that the calibration experiment in Game 1 below gives you a 0.375 probability (the midpoint of the interval from step 2) of winning the (imaginary) money. Here are the choices for step 3:

- Game 1 A neutral person will flip three fair coins. I will pay you \$100 if exactly two of them come up heads. I will pay you nothing otherwise.
- Game 2 I will pay you \$100 if the physics department at FSU has more than two female faculty. I will pay you nothing if it has two or fewer.

If you choose Game 1, then I conclude that

$$0.25 < P_s(A) < 0.375$$

We have succeeded in finding an interval of width 1/8 that traps your subjective probability of event A. If we hadn't set a target of this interval width, we could keep on going with more steps until it became too difficult for you to choose between two offered games.

In this example, we could just go to the Internet and look up the FSU physics department to find out whether there are more than two women faculty members. However, because the prizes are just imaginary, calibration experiments can be used to assess subjective probabilities of events, the occurrence of which cannot be verified.

1.5 What Is to Come?

This chapter has provided a brief introduction to the scientific method, subjective probability, and how the Bayesian approach uses data to update people's knowledge. The example of my friend's mammogram and subsequent follow-up, in which there

were only two possible states of the world and two possible data outcomes for each test, illustrated the simplest form of Bayes' rule.

In the next chapter, we will review basic concepts of probability. In the remainder of the book, we will build on these foundations in order to understand Bayesian modeling and inference for increasingly realistic problems.

Problems

1.1. Give an example of an event, the probability of which would be useful to know but for which the long-frequency interpretation of probability is not applicable.

1.2. What if the result of the mammogram instead had been negative? Write out a table like Table 1.3 for a negative mammogram. What is the posterior probability that my friend will be diagnosed with breast cancer in this case?

1.3. What if the results of both the mammogram and the SCNB had been positive? Write out a table like Table 1.4 for a positive SCNB. What is the posterior probability that my friend has breast cancer in this case?

1.4. Understanding the tables

- (a) In Tables 1.1–1.3, must the numbers in the “Prior probability” column sum to 1, or is that just a coincidence? Explain briefly.
- (b) In Tables 1.2–1.3, should the numbers in the “Likelihood for +” column sum to 1? Why is this necessary or not necessary?

1.5. Suppose that three fair coins are flipped independently. What is the probability that exactly two of them come up heads?

1.6. Find a partner who is willing to play a question-and-answer game with you. The person may be an adult or a child, a friend, a spouse, or a classmate—whoever. (This can work well and be fun for a child, by the way.) Your mission is to use a calibration experiment to assess your partner's subjective probability of some event or of the truth of some statement. Think of an event or statement (call it D) that the person will understand but will be unlikely to know for sure whether it will happen (or has happened) in the case of an event or whether it is true in the case of a statement. For a young child, you might choose something like “Your teacher will be wearing red tomorrow” or “Your soccer team will win its next game.” For an adult, you might choose something like “The population of South Carolina is larger than the population of Monaco.” In any case, use a method like that described in Sect. 1.4 to assess his or her $P_S(D)$. Design your questions so as to bracket $P_S(D)$ in an interval of width $1/8$ in the fewest possible steps.

Your answer to this problem should describe the person briefly (“my daughter,” “my roommate,” etc.), identify the event or statement, and detail all of your questions and the person's responses. In addition, report the interval produced at each step.

Chapter 2

Review of Probability

2.1 Review of Probability

In this section, we will review some basic principles of probability and introduce terminology and notation that will be used throughout the book. In the process, we will derive Bayes' rule for discrete events, which we have already applied in Chap. 1.

2.1.1 Events and Sample Spaces

Statisticians and probabilists use the term *event* to refer to any outcome or set of outcomes of a random phenomenon. Events are the basic elements to which probability can be applied. Capital letters near the beginning of the alphabet are the conventional symbols for events.

For example, suppose the random phenomenon consists of drawing a patient at random from a huge database of patients insured by a health maintenance organization (HMO). To draw an item at random from a set of items means to draw in such a way that every item in the set has the same probability of being the one drawn. A common way of saying this is that all the items are *equally likely* to be drawn. In the case of drawing a record from a database of thousands of records, drawing “at random” would be done by a computer and would actually be based on “pseudorandom” numbers, which, although not truly random, behave closely enough to true randomness to serve this purpose adequately.

Let's define event A as an outcome of the random draw such that the patient we draw is under 6 years of age. As in Chap. 1, we will denote the probability of event A as $P(A)$.

The *sample space* is the set of all possible outcomes of a random phenomenon. A commonly used symbol for it is S . If a random phenomenon occurs, one of the outcomes in S has to happen. Thus,

$$P(S) = 1.$$

2.1.2 Unions, Intersections, Complements

The *intersection* of two events A and B is the event “both A and B ,” which is represented by $A \cap B$. For example, if event B is the event that the patient we draw weighs at least 150 pounds, then $A \cap B$ is the event that the patient we draw is under 6 years of age and weighs at least 150 pounds.

The *union* of two events A and B is the event “either A or B or both,” or, in symbols, $A \cup B$. In our patients-in-the-database example with A and B defined as above, $A \cup B$ is the event that the patient we draw either is under 6 years of age, or weighs at least 150 pounds, or both.

A set of events is said to be *exhaustive* if, taken together, the events encompass the entire sample space. Using the notation \dots to mean “continuing in like manner,” we may write this in symbols as: A set of events A_1, A_2, A_3, \dots is *exhaustive* if

$$A_1 \cup A_2 \cup A_3 \cup \dots = S$$

Continuing the patients-in-the-database example, let’s define A_1 as the event that the patient we draw is under 6 years of age, A_2 as the event that the patient is at least 6 years but under 21 years of age, and A_3 as the event that the patient is at least 21 years old. Since, if we do draw a patient, one of these three events must occur, they are exhaustive.

The *complement* of an event A is the event “everything else that could possibly happen except A ,” notated A^C or \bar{A} . Clearly, $A \cup \bar{A} = S$.

The *null event*, represented by the symbol \emptyset , is an event that can never happen.

Two events A and B are *disjoint* or equivalently *mutually exclusive* if they cannot occur together. If A and B are mutually exclusive, then $A \cap B = \emptyset$. This is the case, for example, if A is the event that the patient we draw from the database is under 6 years of age, and B is the event that the patient is 6–11 years of age. A set of more than two events are mutually exclusive if no two of them can happen together. For example, the three events A_1 , A_2 , and A_3 defined a few paragraphs back are mutually exclusive as well as exhaustive.

2.1.3 The Addition Rule

The addition rule of probability states that if two events A and B are mutually exclusive, then the probability that one or the other happens is just the sum of the probabilities of each event individually:

$$P(A \cup B) = P(A) + P(B)$$

Since any event A and its complement \bar{A} are mutually exclusive by definition, and $A \cup \bar{A}$ is the sample space S , one implication of the addition rule is $P(\bar{A}) = 1 - P(A)$.

2.1.4 Marginal and Conditional Probabilities

For any two events A and B , the *conditional probability* of B given A or $P(B|A)$, is the probability that event B will occur given that we already know that event A has occurred.

The *marginal probability* of an event is the probability of the event without conditioning on the occurrence or nonoccurrence of any other events. When we have spoken of $P(A)$, $P(B)$, etc., those have all been marginal probabilities.

To obtain an algebraic formula for a conditional probability, we can begin with the *multiplicative rule* of probability, which says that the probability of the intersection of two events is the product of the marginal probability of the first and the conditional probability of the second times the first:

$$P(A \cap B) = P(A)P(B|A)$$

$$P(A \cap B) = P(B)P(A|B)$$

Thus, if $P(A) \neq 0$, the conditional probability of event B given event A may be calculated as

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad P(A) > 0 \quad (2.1)$$

To illustrate these concepts, we return to the patients-in-the-database example. Imagine that there were 10,000 patients in the database, and recall that our computer software is able to draw a patient using a procedure that gives each patient in the database an equal probability (1/10,000) of being the one drawn. Thus, the probability of drawing a patient with a particular characteristic is simply the proportion of patients in the database that have that characteristic. Table 2.1 cross-tabulates the 10,000 by age category and weight category.

Table 2.1 Patients cross-tabulated by age and weight

	< 150 pounds	≥ 150 pounds	Total
under 6	798	2	800
≥ 6	4,702	4,498	9,200
Total	5,500	4,500	10,000

Recall that we have been letting A denote the event that the patient that we draw is under 6 years of age. Then $P(A)$ —the marginal probability of event A —is $\frac{800}{10,000} = 0.08$. Similarly, with B defined as the event that the patient we draw weighs over 150 pounds, the marginal probability of event B is $P(B) = \frac{4,500}{10,000} = 0.45$.

Now imagine that we have drawn our patient and have observed that the age given in his record is 5 years. Thus, we know event A has occurred. We have not yet looked at the weight recorded in the record, and we want to determine the probability that the patient weighs at least 150 pounds, given that we already know he is under 6 years of age—that is, we want to evaluate $P(B|A)$. Since we know event A has occurred, we can restrict our attention to the first row of the table and can take the number of patients in that row who weigh at least 150 pounds divided by the row total: $P(B|A) = \frac{2}{800} = 0.0025$.

To verify that formula (2.1) says the same thing, we find the probability of drawing a patient who both is less than 6 years of age and weighs at least 150 pounds: $P(A \cap B) = \frac{2}{10,000} = 0.0002$. Then

$$\begin{aligned}
 P(B|A) &= \frac{P(A \cap B)}{P(A)} \\
 &= \frac{0.0002}{0.08} \\
 &= 0.0025
 \end{aligned}$$

2.1.4.1 Independence

Two events are *independent* if the occurrence (or nonoccurrence) of one of them does not affect the probability that the other one occurs. That is, events A and B are independent if

$$\begin{aligned}
 P(A|B) &= P(A) \\
 P(B|A) &= P(B)
 \end{aligned}$$

In the database represented in Table 2.1, obviously events A and B are not independent! The conditional probability that a patient will weigh at least 150 pounds given that he is under 6 years of age ($P(B|A) = 0.0025$) is far from equal to the marginal probability that a randomly selected patient will weigh at least 150 pounds ($P(B) = 0.45$).

Table 2.2 Patients cross-tabulated by eye color and weight

	< 150 pounds	≥ 150 pounds	Total
Green eyes	440	360	800
Not green	5,060	4,140	9,200
Total	5,500	4,500	10,000

2.1.5 The Multiplication Rule

There is a special form of the multiplicative rule of probability for *independent events*. If A and B are independent, then

$$P(A \cap B) = P(A)P(B)$$

Table 2.2 gives a different cross-tabulation of the patients in the database, this time by eye color (green or not green) and weight category.

Of course the marginal probability of drawing a patient who weighs at least 150 pounds is unchanged— $P(B) = \frac{4,500}{10,000} = 0.45$. Let C represent the event that a patient drawn at random has green eyes. What about $P(B|C)$ —the conditional probability that the patient drawn weighs at least 150 pounds, given that it is known that the patient has green eyes. Well, according to Table 2.2, the marginal probability of drawing a patient with green eyes is $P(C) = \frac{800}{10,000} = 0.08$. Furthermore, the probability of drawing a patient who both has green eyes and weighs at least 150 pounds is $P(B \cap C) = 360/10,000 = 0.036$. Thus, the conditional probability $P(B|C) = \frac{P(B \cap C)}{P(C)} = \frac{0.036}{0.08} = 0.45 = P(B)$. Thus, for this situation, events B and C are independent.

2.2 Putting It All Together: Did Brendan Mail the Bill Payment?

2.2.1 The Law of Total Probability

The law of total probability comes into play when you wish to know the marginal (unconditional) probability of some event, but you only know its probability under some conditions. Here's an example. I have asked my husband Brendan to mail our credit card bill payment. We will incur a late fee if the payment isn't received at the payment center within three days. I want to calculate $P(A)$, the probability of the event that the payment gets there on time. I believe that $P(M)$, the probability that Brendan will remember to mail the letter today or tomorrow, is 0.60.

Furthermore, I believe that if he mails the letter today or tomorrow, the probability that the postal service will deliver it to the addressee within the next

3 days is 0.95. This is a conditional probability:

$$P(A|M) = 0.95$$

I believe there's only 1 chance in 10,000 that the letter will get there somehow if Brendan forgets to mail it. (Maybe he drops it on the sidewalk and a kind passerby picks it up and puts it in a mailbox.) In any case, I believe that

$$P(A|\bar{M}) = 0.0001$$

In preparing to use the law of total probability to find $P(A)$, note that for any events A and M , the event A is the union of two events: the intersection of A with M and the intersection of A with \bar{M} . In symbols,

$$A = (A \cap M) \cup (A \cap \bar{M})$$

Since events $(A \cap M)$ and $(A \cap \bar{M})$ are disjoint, the addition rule applies:

$$P(A) = P(A \cap M) + P(A \cap \bar{M})$$

Applying the multiplication rule to both terms on the right hand side yields

$$P(A) = P(A|M)P(M) + P(A|\bar{M})P(\bar{M})$$

For the example

$$\begin{aligned} P(A) &= (0.95)(0.60) + (0.0001)(0.40) \\ &= 0.57004 \end{aligned}$$

Uh-oh, there's only a little better than a 50/50 chance that the bill payment will arrive on time.

2.2.1.1 Generalized Law of Total Probability

The law of total probability may be generalized to the situation in which there are more than two different conditions under which the event of interest could occur. If M_1, M_2, M_3, \dots are mutually exclusive and exhaustive events, then

$$P(A) = P(A|M_1)P(M_1) + P(A|M_2)P(M_2) + P(A|M_3)P(M_3) + \dots$$

2.2.2 Bayes' Rule in the Discrete Case

My prior probability that Brendan would remember to mail the bill payment was $P(M) = 0.60$. The data is that the payment actually arrived within 3 days! Bayes' rule calculates my posterior probability that Brendan mailed the payment given the data, that is,

$$P(M|A)$$

when we know $P(A|M)$, $P(A|\bar{M})$, and $P(M)$.

By the definition of conditional probability,

$$P(M|A) = \frac{P(M \cap A)}{P(A)}$$

Using the multiplication rule to expand the numerator and the law of total probability to expand the denominator gives Bayes' rule:

$$P(M|A) = \frac{P(A|M)P(M)}{P(A|M)P(M) + P(A|\bar{M})P(\bar{M})}$$

For the example, this is

$$\begin{aligned} P(M|A) &= \frac{0.95(0.60)}{0.95(0.60) + (0.0001)(0.40)} \\ &= \frac{0.57}{0.57004} \\ &= 0.99993 \end{aligned}$$

Thus, given that the payment arrived on time, it is almost certain that Brendan remembered to mail it.

2.2.2.1 Generalized Bayes' Rule

Corresponding to the generalized law of total probability, generalized Bayes' rule holds if event A could happen conditional on one of a number of different other events, M_1, M_2, M_3, \dots . To apply generalized Bayes' rule, we must know the conditional probabilities $P(A|M_1)$, $P(A|M_2)$, etc., as well as the marginal probabilities $P(M_1)$, $P(M_2)$, etc. After the event A has occurred, we want to assess the conditional probability of one of the events M_j , $P(M_j|A)$.

If M_1, M_2, M_3, \dots are mutually exclusive and exhaustive events, then

$$P(M_j|A) = \frac{P(A|M_j)P(M_j)}{P(A|M_1)P(M_1) + P(A|M_2)P(M_2) + P(A|M_3)P(M_3) + \dots}$$

2.3 Random Variables and Probability Distributions

As further preparation for Bayesian statistics, we need to review notions of probability beyond those applicable to discrete events. A *random variable* may be informally defined as a function that assigns one real number to each outcome in the sample space of a random phenomenon. The outcomes in the sample space of the original random phenomenon may or may not be numeric to begin with. Capital letters near the end of the alphabet are the conventional symbols for random variables, with the corresponding lower case letters being used to represent specific numeric values they can take on.

For a simple example, consider a coin toss, in which the sample space consists of the two possible outcomes, head and tail. We may define a random variable, say X , which takes on the value 1 if the outcome of a flip is a head, and the value 0 if the outcome is a tail. If we flip the coin and get a head, we express the outcome in terms of a value of the random variable as $x = 1$.

Similarly, our random experiment might consist of drawing a household at random from among all the households in Johnson County, Iowa, and recording the number of people living in the household. The sample space of this random experiment is numeric, consisting of the integers 1, 2, \dots . In cases like this, in which the sample space is numeric, often statisticians don't explicitly differentiate between the sample space of the random experiment and the space of the associated random variable. Instead, they may say something like "Define the random variable Y as the number of people in the randomly selected household." In these examples, X and Y are *discrete random variables* because each has a discrete set of possible values that it can take on. The set of possible values that a random variable can assume is called the *space* of the random variable. The space of the random variable X in the coin flip example is $\{0, 1\}$, and the space of the random variable Y in the household example is $\{1, 2, \dots\}$.

By contrast, a *continuous random variable* is not restricted to a discrete set of possible values, but instead may take on any value in a continuum. For example, we might define the random variable W as the height of a woman drawn at random from among female first year students at the University of Iowa. Although we might choose to measure this random variable only to the nearest quarter inch, in fact, the underlying actual height could take on any value within a wide interval. That is, the space of the random variable W is this interval. The exact endpoints of the interval, which in this example might be on the order of 4 feet and 7 feet, are not critical in understanding the concept of continuous random variables.

Probability distributions describe the behavior of random variables. The probability distribution of a discrete random variable identifies the possible values the variable can take on and associates a numeric probability with each. Each of these probabilities must be between 0 and 1, and the probabilities of all the possible values in the space of the random variable must sum to 1. One way of presenting the probability distribution of a discrete random variable is by means of a table that lists each possible value in the space of the random variable, along with its associated probability. For the fair coin toss example, such a table for the random variable X would look like

x	$\Pr(X=x)$
0	0.5
1	0.5

The probability distributions of certain discrete random variables can be described succinctly using a *probability mass function* or *pmf*—a mathematical function that can be evaluated at each value in the space of the random variable to yield the probability that the random variable takes on that value. We will make use of several families of pmfs, including the binomial and Poisson, in future chapters.

Describing the behavior of continuous random variables is more subtle. Because there are an infinite number of individual values in any interval, the probability that a continuous random variable takes on any specific numeric value is zero. Instead, we talk about the probability that a continuous random variable takes on a value in a particular interval of interest. Such a probability is expressed as the integral of an appropriate *probability density function* or *pdf* over that interval.

We will study pdfs in much more detail in future chapters and will make extensive use of several families of pdfs, including the beta, the gamma, and the normal families.

Problems

2.1. An experiment consists of flipping a fair coin three times independently. Let event A be “at least one of the results is a head.” Let event B be “all three results are the same.”

- List the sample space of the experiment.
- List the outcomes in B , and find $P(B)$.
- List the outcomes in $A \cap B$, and find $P(A \cap B)$.
- List the outcomes in $A \cup B$, and find $P(A \cup B)$.
- Are the events A and B independent? Show why or why not.

2.2. This problem is based on problems 4.8 and 5.17 in [Berry \(1996\)](#). Suppose the following statements are true for Minneapolis on January 15:

1. The probability that it snows both today and tomorrow is 0.2.
2. The probability that it snows today but not tomorrow is 0.1.
3. The probability that it snows tomorrow but not today is 0.1.
4. The probability that it snows neither today nor tomorrow is 0.6.

Find the probability that:

1. It will snow today.
2. It will snow either today or tomorrow or both.
3. It will snow tomorrow.
4. It will snow tomorrow, given that it has snowed today.

2.3. This problem is based on an example on pages 9–11 of [Gelman et al. \(2004\)](#). Hemophilia is a rare hereditary bleeding disorder caused by a defect in genes that control the body's production of blood-clotting factors. It occurs almost exclusively in males. However, women may be carriers of the hemophilia gene. Female carriers of the hemophilia gene usually show no physical symptoms of hemophilia. A son born of a woman who is a hemophilia carrier and a man who does not have hemophilia has a 0.5 probability of inheriting hemophilia from his mother. A son born of a woman who is not a carrier and a man who does not have hemophilia has zero probability of inheriting hemophilia.

Danielle is a young married woman. Her husband does not have hemophilia. Because Danielle's mother is known to be a carrier of hemophilia, there is a 0.5 probability that Danielle inherited a hemophilia gene from her mother and is also a carrier. We may consider two possible “models”: Danielle is a carrier, and Danielle is not a carrier.

Danielle gives birth to three sons. None of them are identical twins, and we will consider their hemophilia outcomes to be independent conditional on her carrier status. For each of the sons, we will define a random variable Y_i that takes on the value 1 if the son has hemophilia and 0 if he does not.

(a) Use the information given above to determine

- (1) Prior probabilities for the two possible “models” (carrier and not carrier) evaluated *before* Danielle gives birth to her first son
- (2) Two different sets of likelihood probabilities, one for each model

Using the notation above, for the i th son, $y_i = 0$ indicates the son is not affected by hemophilia; $y_i = 1$ indicates that he is affected. For each model, you will need

$$Pr(y_i = 0 | model)$$

and

$$Pr(y_i = 1 | model)$$

(b) Now you learn that the three outcomes are:

$$y_1 = 0$$

$$y_2 = 1$$

$$y_3 = 0$$

Do a sequential Bayesian analysis in which you compute the posterior probability that the woman is a carrier using the data from each son one at a time. For each step, use Bayes' rule and make a table with columns for model, prior probabilities, likelihood given observed data, product, and posterior probabilities.

(c) Also answer the following questions:

- (1) What was the posterior probability that the woman was a carrier after the first son's status became known?
- (2) Did the posterior probability change based on the data from the second son? Why or why not?
- (3) Did the posterior probability change based on the data from the third son? Why or why not?

2.4. The following facts described the students who took my Bayesian statistics class in a recent year:

- 35% of the students were statistics grad students.
 - 25% of the students were biostatistics grad students.
 - 40% of the students were undergraduates or grad students from other departments.
 - 60% of the statistics grad students were women.
 - 75% of the biostatistics grad students were women.
 - 40% of the students in other categories were women.
- (a) You drew a student at random from my class list for that year. What is the probability that the student you drew is a woman?
- (b) Suppose that the student you drew was a woman. What is the probability that she was **not** a statistics grad student and **not** a biostatistics grad student, given that she was a woman?

2.5. Identify each of the random variables below as discrete or continuous, and specify its space.

- (a) A die as described in Problem 2.1 is rolled. Define a random variable Y as the number on the face that comes up.
- (b) A county in the state of Nebraska is selected at random. Define a random variable X as the number of homicides reported in that county in the year 2011.
- (c) A nutritionist is studying the effect of a dietary supplement on the growth rate of baby rats. She gives this supplement daily to 150 newborn rats and records their weights at birth and at 30 days of age. Let the random variable W represent the weight change in grams of a baby rat from birth to age 30 days.

Chapter 3

Introduction to One-Parameter Models: Estimating a Population Proportion

3.1 What Proportion of Students Would Quit School If Tuition Were Raised 19%: Estimating a Population Proportion

On March 15, 2002, the *Iowa City Press Citizen* carried an article about the intended 19% tuition increase to go into effect at the University of Iowa (UI) for the next academic year. Let's revisit that time and suppose that you wish to send the regents and the state legislature some arguments against this idea. To support your argument, you would like to tell the regents and legislators what proportion of current UI students are likely to quit school if tuition is raised that much.

Your research question is as follows: What is the unknown *population parameter* π —the proportion in the entire population of UI students who would be likely to quit school if tuition is raised 19%?

You do not have the time or resources to locate and interview all 28,000+ students, so you cannot evaluate π exactly. Instead, you will pick a simple random sample of $n = 50$ students from the student directory and ask each of them whether she or he would be likely to quit school if tuition were raised by 19%. You wish to use your sample data to estimate the population proportion π and to determine the amount of uncertainty in your estimate.

3.2 The First Stage of a Bayesian Model

When planning to use data to estimate population parameters, the statistician must identify a probability distribution from which the data to be collected may plausibly be considered to be drawn. This probability model forms the first stage of the Bayesian model.

3.2.1 The Binomial Distribution for Our Survey

Before you select the students in your sample and interview them, you can regard each student's potential response as a Bernoulli random variable. A *Bernoulli* or *binary* random variable can take on one of only two values. One value is arbitrarily called a “success” (numerical representation 1), the other a “failure” (numerical representation 0). We'll choose to call a “yes” answer a success.

The unknown population proportion π in Sect. 3.1 is also the probability that a randomly selected student from this population would answer “yes.” Because we know nothing about the people in your sample except that they will come from the student directory, it is reasonable to assume that they all have the same probability of saying yes to your question—namely, the population proportion π . That is, we assume that the observations will be *exchangeable*. If we knew more about each student in the sample (e.g., whether she or he is a senior or a freshmen, how wealthy she or he is), this assumption would not be reasonable.

We also will assume, because you will draw a simple random sample, that the responses from the individual students are *independent*. This would not be reasonable if you chose 25 pairs of roommates or sets of siblings, or used any other sampling scheme such that the responses of any subset of the sample would be expected to be more similar to each other than to those of other members of the sample.

Define a random variable Y as the count of the number of successes in your sample. Y meets the definition of a *binomial random variable*—it is the count of the number of successes in n -independent Bernoulli trials, all with the same success probability. We can write

$$Y \sim \text{Binomial}(n, \pi)$$

What are the possible values of Y ? (Of course, you won't find out the value that Y takes on in your own survey until you actually draw the $n = 50$ students and interview them.)

If we knew π , we could use the binomial probability mass function to compute the probability of obtaining any one of the possible values y that the random variable Y could take on in our sample:

$$p(y|\pi) = \binom{n}{y} \pi^y (1 - \pi)^{n-y}, \quad y = 0, 1, \dots, n \quad (3.1)$$

Here $\binom{n}{y}$ (pronounced “ n choose y ”) means the number of possible combinations of y items that could be chosen out of n total items. It is shorthand for $\frac{n!}{y!(n-y)!}$.

We are thinking of the expression in (3.1) as the probability mass function of Y —a probability distribution on the possible values of Y in random samples of size n drawn from a population in which the parameter π has some specified value.

For example, if we magically knew that $\pi = 0.1$, then the probability of getting $y = 4$ yesses among the respondents in a random sample of 50 students would be

$$p(Y = 4 | \pi = 0.1) = \binom{50}{4} 0.1^4 0.9^{46} = 0.181$$

3.2.2 Kernels and Normalizing Constants

In Bayesian statistics, we often want to distinguish between the *kernel* of a function and the *normalizing constant*. The kernel includes all terms that will change in value for different values of the variable of interest. When considering (3.1) as the probability mass function for a random variable Y , we must consider all terms that contain a y as part of the kernel, because these will change with the different possible values of Y that are plugged in. Thus, every term in (3.1) is part of the kernel.

3.2.3 The Likelihood Function

But we don't know π .

Instead, *after* you interview the 50 students, you will know how many said yes. That is, you will know which value y the random variable Y actually took on. Suppose this number turns out to be $y = 7$. We will want to use this information to *estimate* π . In this case, we may change perspective and regard the expression in (3.1) as a function of the unknown parameter π given the now known (fixed) data value y . When viewed in this way, the expression is called the *likelihood function*. If y were 7, it would look like

$$L(\pi) = \binom{50}{7} \pi^7 (1 - \pi)^{43}, \quad 0 < \pi < 1 \quad (3.2)$$

We could compute this likelihood for different values of π . Intuitively, values of π that give larger likelihood evaluations are more consistent with the observed data.

Figure 3.1 is a graph of the likelihood function for a binomial sample with 7 successes in 50 trials. The range of possible values of π —the interval $(0,1)$ —is on the x axis, and the curve represents evaluation of (3.2) over this interval.

Note that the interpretation of kernel versus constant differs for (3.1) versus (3.2). Now we are interested in evaluating the likelihood (3.2) for changing values of π , not of y . Consequently, the $\binom{50}{7}$ is now considered just a constant, and only the $\pi^7(1 - \pi)^{43}$, which varies with π , is the kernel. That is, we can write

$$L(\pi; y) \propto \pi^7 (1 - \pi)^{43}, \quad 0 < \pi < 1$$

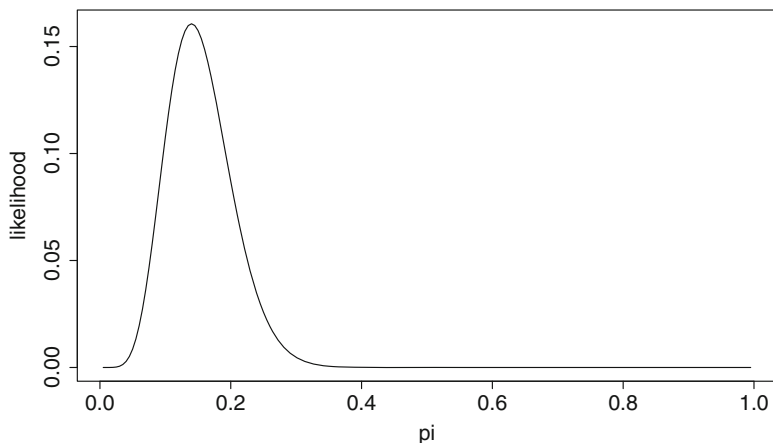


Fig. 3.1 Binomial likelihood function; 7 successes in 50 trials

or more generically

$$L(\pi; y) \propto \pi^y (1 - \pi)^{n-y}, \quad 0 < \pi < 1 \quad (3.3)$$

3.3 The Second Stage of the Bayesian Model: The Prior

To carry out a Bayesian analysis to learn about the unknown population proportion π , we need to assess our previous knowledge or belief about π *before* we observe the data from the survey.

As you might guess after reading Sect. 1.3.1.2, the Bayesian approach to expressing prior knowledge about a population parameter is to put a probability distribution on the parameter—that is, to treat the unknown population parameter *as if* it were a random variable. Note that this does not mean that Bayesians believe that the value of the parameter of interest is a moving target that varies in a random way. It simply provides a mathematical way of describing what is already known about the parameter (recall Step 2 of the scientific method in Sect. 1.1).

Because it is a proportion, the parameter π hypothetically could take on any value in the interval $(0, 1)$, although most of us realize that some ranges of values are much more likely than others. Because π can take on any of a continuum of values, we quantify our knowledge or belief most appropriately by means of a *probability density function*. This is different from the problems in Sect. 1.3.2, which involved a discrete set of models, to each of which we assigned a prior probability.

A person who has little or no knowledge about university students might consider all values in $(0, 1)$ equally plausible before seeing any data. A *uniform* density on $(0, 1)$ describes this belief (or state of ignorance!) mathematically and graphically (Fig. 3.2):

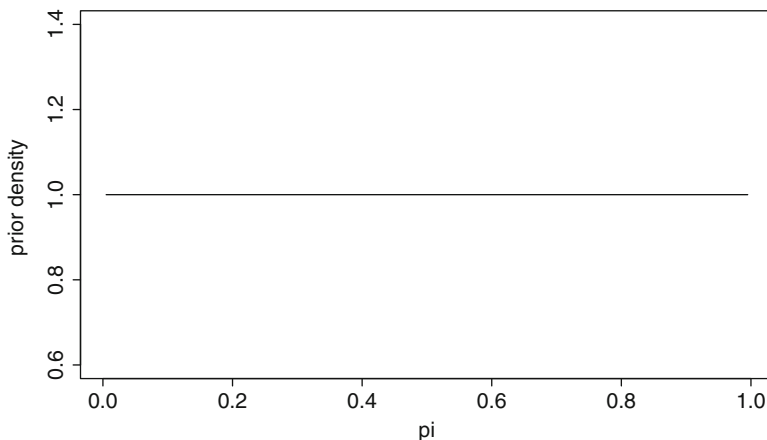


Fig. 3.2 Uniform density on $(0, 1)$

$$\pi \sim U(0, 1)$$

$$p(\pi) = 1, \quad 0 < \pi < 1$$

This continuous uniform distribution is called a “vague” or “noninformative” prior. It says that if we pick any two intervals within $(0, 1)$ that are of equal width—say $(0.2, 0.31]$ and $(0.85, 0.96]$ —there is equal probability that π lies in each of them.

3.3.1 Other Possible Prior Distributions

If a person has knowledge or belief regarding the value of π , his or her prior will be informative. We will look at only a few of the innumerable kinds of prior distributions that could be used. For example, Fig. 3.3 shows two different possible priors expressing the belief that π most likely lies between 0.1 and 0.25.

Figure 3.4 depicts an example of a histogram prior. Such a prior distribution is constructed by first dividing the support of the parameter into intervals. Above each interval, one then draws a bar, the area of which represents the prior probability that the parameter falls into the interval. For a valid histogram prior, the areas of all the bars must sum to one. Often subject-matter experts who are not experts in probability and statistics are more comfortable trying to describe their knowledge about a parameter through a histogram prior than through some of the other, more mathematical, types of priors we will discuss. The histogram prior in Fig. 3.4 represents the prior belief that the probability that π lies in the interval $[0, 0.1)$ is 0.25, in $[0.1, 0.2)$ is 0.5, etc. It places no probability mass on values of π greater than 0.4.

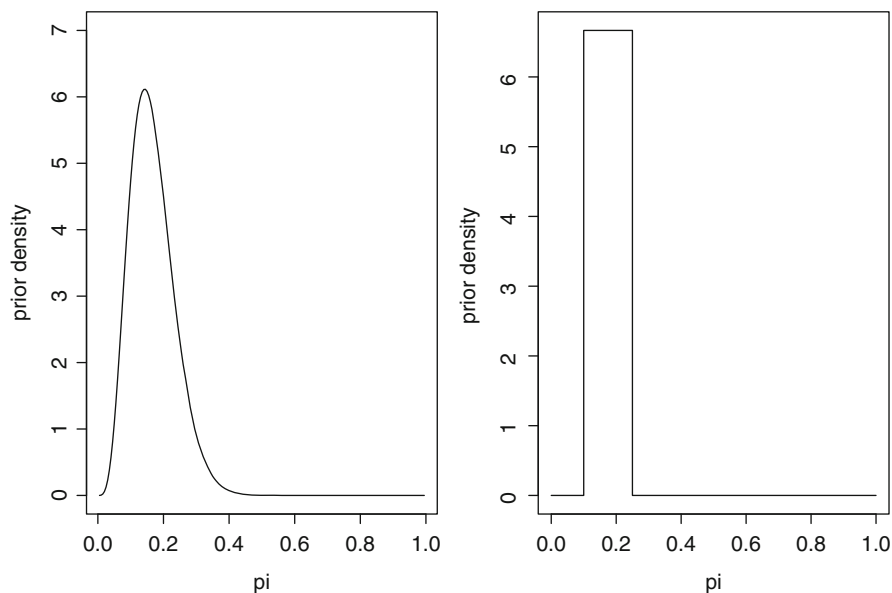


Fig. 3.3 Two densities with most of their mass on $(0.1, 0.25)$

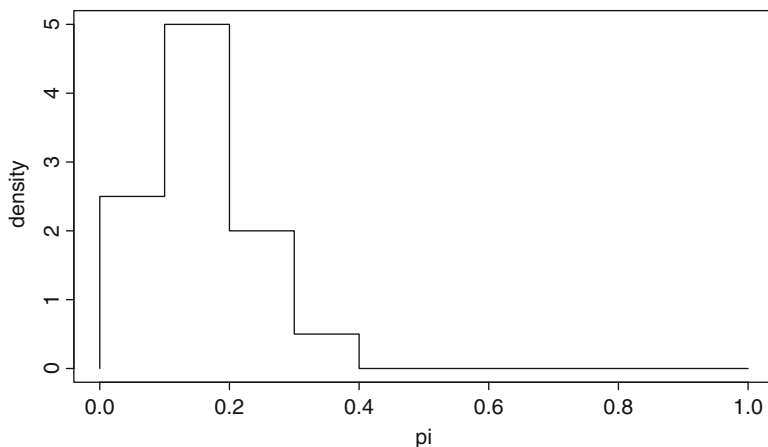


Fig. 3.4 A histogram prior

All of the prior distributions we have mentioned so far treat the unknown parameter π as if it were a *continuous* random variable. In some applications, even though the parameter of interest may in reality take on any value over a continuum, if very exact inference is not required, a discrete prior may be simpler to work with and may adequately express available prior information. For example, it probably wouldn't make any difference to us (or presumably to the Regents) if the true

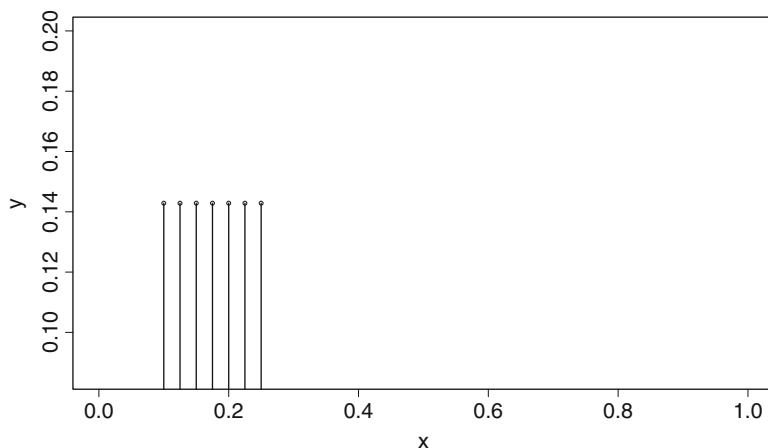


Fig. 3.5 A discrete uniform prior on π

population proportion of students who would quit school was 0.18 or 0.20 or 0.22—all of those are around $1/5$, far too large a fraction of the student body to lose. In such a situation, a discrete prior distribution, which simply selects certain specific possible values of the parameter of interest and puts a “lump” of prior probability mass on each, might be used. For example, the prior distribution depicted in Fig. 3.5 is a discrete uniform distribution; it specifies seven possible values of π and places equal prior probability on each:

$$p(\pi) = \frac{1}{7}, \pi = 0.1, 0.125, 0.15, 0.175, 0.20, 0.225, 0.25$$

Of course, the probabilities assigned to all the distinct parameter values under a discrete prior must sum to 1.

3.3.2 Prior Probability Intervals

One way of summarizing a prior probability distribution is in terms of an interval that traps a specified proportion of the prior probability mass. The most commonly considered prior probability intervals are 95% central intervals. If we say that the interval $[l, u]$ is a 95% central prior probability interval for π , we mean that the prior mass on values less than l is 0.025 and on values greater than u is also 0.025; the remaining 0.95 prior probability is on the interval $[l, u]$. Another way to say the same thing is that l is the 0.025 quantile of the probability distribution, and u is the 0.975 quantile. Central prior intervals with other probability levels are also used.

Figuring out the endpoints of the 95% central prior interval under the continuous uniform prior in Fig. 3.2 is particularly easy. As shown in Fig. 3.6, since the height

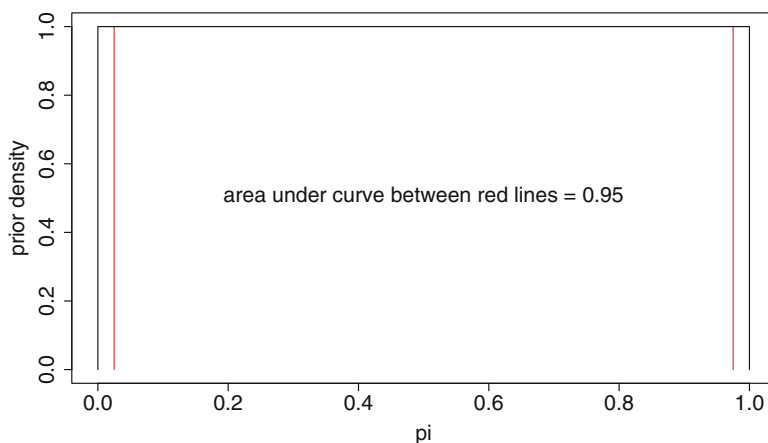


Fig. 3.6 The 95% central prior interval under a uniform prior on π

of the density is 1 over $(0,1)$, the area under the density above $(0.025, 0.975)$ is 0.95, so $(0.025, 0.975)$ is the 95% central interval with this prior. The 90% central interval would be 0.05, 0.95, etc.

In Sect. 3.6.2, we will learn to use functions in the statistical software called R to determine quantiles (and other characteristics) of other densities.

When using a discrete prior probability mass function, a prior “interval” is actually a set of consecutive discrete values of the random variable. Often such a set cannot be constructed so as to have exactly the desired total probability mass. In such cases, the smallest “interval” that traps *at least* the required probability mass is used. For example, suppose that we wanted a 95% prior interval based on the discrete prior in Fig. 3.5. The total prior mass on all seven specified possible values of π is 1.0. But if we omit either the smallest or the largest value (0.1 or 0.25), the total probability mass on the remaining six values is $6/7 = 0.857$, so the interval doesn’t cover points with enough total probability. Thus, the best 95% probability interval available with this prior is the set $\{0.1, 0.125, 0.15, 0.175, 0.20, 0.225, 0.25\}$. This would also be the best 90% prior interval.

3.4 Using the Data to Update the Prior: The Posterior Distribution

The possible ways of choosing a prior for an unknown proportion are endless. In Sect. 3.5, we will consider one common way of thinking about this question and will look at the effects of different prior specifications.

For the moment, let’s see what happens if we use the “noninformative” continuous uniform prior for our analysis.

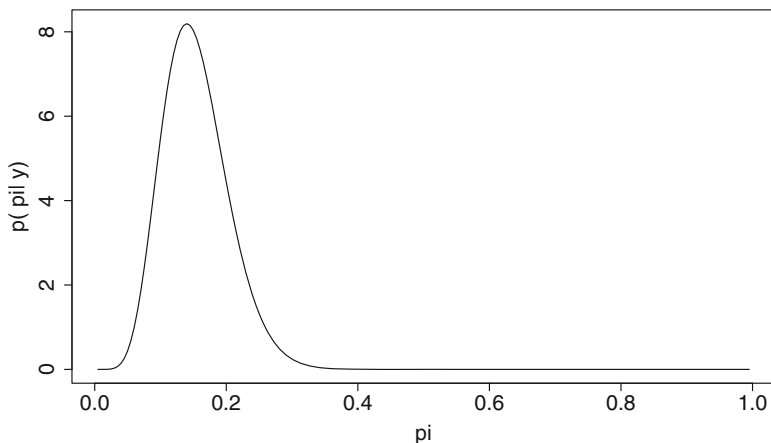


Fig. 3.7 Posterior density $p(\pi|y)$ with uniform prior

At last, you may select your sample and conduct your survey! It turns out that 7 of the 50 students say they would quit school if tuition were raised 19%.

Thus, the *sample* proportion of yesses in your observed data is $\hat{\pi} = \frac{7}{50} = 0.14$. Note that this is the value of π at which the likelihood function in Fig. 3.1 attained its maximum.

You have already applied Bayes' theorem in Sect. 1.3.4 to use data to update from prior probabilities to posterior probabilities in the case of discrete models or events. Bayes' theorem for probability density functions analogously states that the posterior density function is proportional to the prior density times the likelihood function. For the population proportion π ,

$$p(\pi|data) \propto p(\pi) L(\pi; y)$$

For the quitting-school example and binomial likelihood, combining the prior density on π with the likelihood in (3.3) yields

$$p(\pi|y) \propto p(\pi) \pi^y (1 - \pi)^{n-y}, \quad 0 < \pi < 1$$

Having chosen the uniform prior, $p(\pi) = 1$, $0 < \pi < 1$, and having observed $y = 7$ "successes" out of $n = 50$ people surveyed

$$p(\pi|y) \propto 1 \times \pi^7 (1 - \pi)^{43}, \quad 0 < \pi < 1$$

We can graph this function to see what the posterior density $p(\pi|y)$ looks like (Fig. 3.7).

Note that the mode (highest peak) is at $\pi = 0.14$, and most of the area under the curve is above values of π in the interval (0.05, 0.35). Section 3.5 will show how to make more specific inferences about π .

3.5 Conjugate Priors

A common way to construct a prior distribution is to designate that the prior is a member of a particular parametric family of densities. One then chooses the parameters of the prior density so as to reflect as closely as possible his or her beliefs about the unknown parameter being studied. When possible, it is very convenient analytically to choose the prior from a parametric family that has the same functional form as the likelihood function. Such a prior is called a *conjugate prior*.

Regarding the binomial example, recall that π must lie between 0 and 1, and note how the parameter π appears in the binomial likelihood (3.2). There π is raised to a nonnegative power, and $(1 - \pi)$ also is raised to a nonnegative power. The *beta* family of densities matches this treatment of π in the two required ways: The beta density has support on the interval $(0, 1)$; furthermore, in a beta density, the random variable and one minus the random variable each appear raised to a nonnegative power.

A *beta* family of densities, with fixed parameters α and β and with the *random variable* called π would be written as follows:

$$\pi \sim \text{Beta}(\alpha, \beta)$$

or

$$\begin{aligned} p(\pi) &= \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} \pi^{\alpha-1} (1 - \pi)^{\beta-1} \\ &\propto \pi^{\alpha-1} (1 - \pi)^{\beta-1}, \quad 0 < \pi < 1 \end{aligned} \quad (3.4)$$

Here the term $\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)}$ is a normalizing constant, because it does not change for different values of π . The kernel of the beta density is $\pi^{\alpha-1} (1 - \pi)^{\beta-1}$.

3.5.1 Computing the Posterior Distribution with a Conjugate Prior

Recall the relationship of the posterior distribution to the prior and likelihood:

$$p(\pi|y) \propto p(\pi) L(\pi; y)$$

So in the case of a beta prior and a binomial likelihood,

$$\begin{aligned} p(\pi|y) &\propto \pi^{\alpha-1} (1 - \pi)^{\beta-1} \pi^y (1 - \pi)^{n-y} \\ &= \pi^{\alpha+y-1} (1 - \pi)^{\beta+n-y-1} \end{aligned}$$

This is the kernel of another beta density!

$$p(\pi|y) = \text{Beta}(\alpha + y, \beta + n - y)$$

Since the beta family of priors is conjugate for the binomial likelihood, the posterior distribution is also beta—a member of the same parametric family as the prior distribution. We will encounter many other pairings in which a particular family of densities is conjugate for a particular likelihood family. In each case, the resulting posterior density will be in the same family as the prior. This is the implication of conjugacy.

3.5.2 *Choosing the Parameters of a Beta Distribution to Match Prior Beliefs*

Here are several ways to think about choosing the parameters of a beta distribution to express prior beliefs or knowledge about an unknown proportion:

Strategy 1: Graph some beta densities until you find one that matches your beliefs.

Strategy 2: Note that a $\text{beta}(\alpha, \beta)$ prior is equivalent to the information contained in a previously observed dataset with $\alpha - 1$ successes and $\beta - 1$ failures. (To see this, compare the binomial likelihood in (3.3) with the beta density in (3.4), considering each as a function of π .)

Strategy 3: Solve for the values of α and β that yield:

- The desired mean (The mean of a $\text{beta}(\alpha, \beta)$ density is $\frac{\alpha}{\alpha + \beta}$).
- The desired *equivalent prior sample size*, which for a $\text{beta}(\alpha, \beta)$ prior is $\alpha + \beta - 2$. When you use this method, you are saying that your knowledge about π is as strong as if you'd seen a previous sample consisting of $\alpha - 1$ successes and $\beta - 1$ failures.

Strategy 4: Choose values of α and β that produce a prior probability interval that reflects your belief about π .

Strategy 5: Think about what you would expect to see in the data that you are going to collect, and choose a prior density for π that would make such future data likely. We will postpone consideration of this strategy until after we have discussed predictive distributions in Sect. 4.4.

The new data *must not* be used in any way in constructing the prior density! We'll see shortly why that would make inference invalid.

We will apply the first four strategies to the quitting-school-because-of-rising-tuition example. We are attempting to construct a reasonable prior *before* we see the results of the actual survey of 50 UI students. (Forget Sect. 3.4—you have not yet collected your data!)

We wish to use any relevant data available before we do our survey. Suppose that we read that such a survey has already been taken at Iowa State University (if you're

not from Iowa, you will just have to imagine the degree of rivalry between UI and ISU!) in which:

- 50 students were interviewed.
- 10 said they would quit school; 40 said they would not.

By strategy 2, this might suggest a $\text{beta}(11, 41)$ prior. However, we need to be very cautious here because the sample on which the prior distribution is to be based was not drawn from the same population (UI students) in which we are interested and from which we will draw our sample. The question is whether ISU students might be different from UI students in ways that would be likely to affect their probability of dropping out in response to a tuition increase—that is, is a sample of ISU students *exchangeable* with a sample of UI student for our purposes? Some relevant facts are that the UI has a medical school and a law school, which ISU does not, while ISU has a veterinary school and a College of Agriculture, which UI does not. At the UI, the majority of the student body (53.5%) are women, whereas at ISU, only 43.5% of students are women. Undergraduate tuition is a bit higher at the UI (\$6,544 for Iowa residents, \$20,658 for nonresidents) than at ISU (\$6,360 and \$17,350) for the 2008–2009 year. These and other factors suggest that, for drawing inference about UI students, the information in a sample of ISU students is not equivalent to the information contained in a sample of the same number of UI students. Thus, the $\text{beta}(11, 41)$ prior most likely is not appropriate.

On the other hand, there presumably is some relevant information for us in the ISU survey. We want to make use of that, but give it less weight than we would give to 50 observations from the UI population. One of many valid approaches to specifying a prior in this case is to say that we want a prior mean of 0.2, the same as the sample proportion $\hat{\pi}_{ISU}$ from the ISU data, but an “equivalent prior sample size” (remember, for a beta prior that is $\alpha - 1 + \beta - 1$) that is smaller than 50. One possibility is to look at the graphs of several different beta distributions, all with the same mean 0.2 but with smaller and smaller equivalent prior sample sizes, and seek one that matches our beliefs. The first plot in Fig. 3.8 represents a prior belief that the unknown value of the population parameter π is close to 0.2; this belief is as strong as if we had seen a previous sample from the population of interest (UI students) of size 48 with 9 successes and 39 failures. The fourth plot also represents a prior belief that π is near 0.2, but in this case, the belief is only as strong as if we had seen a previous (hypothetical!) sample of size 4.25 with 0.25 successes and 4 failures.

Note that when the sum of the two parameters of the beta distribution is larger as in the first plot, the density curve is fairly concentrated (tall and thin) over an interval near 0.2. The smaller the sum ($\alpha + \beta$), the more the beta prior spreads out the probability over larger intervals within (0,1)—that is, the larger the variance, or uncertainty, in the prior density.

We can also consider whether the central prior intervals produced by any of these prior densities match our knowledge. Figure 3.9 gives the numeric endpoints of these intervals and shows them on the density plots. As we should have expected, the smaller the parameter values, the wider the corresponding prior interval.

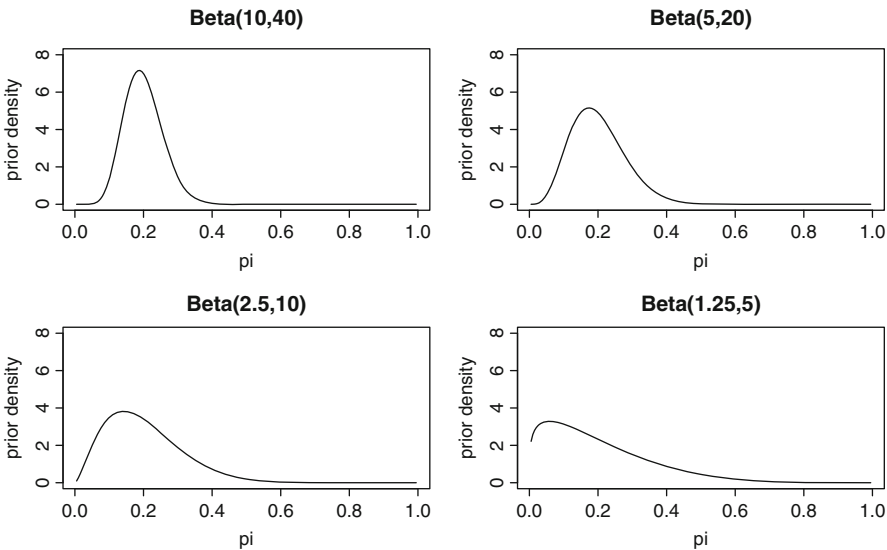


Fig. 3.8 Beta densities with different parameter values

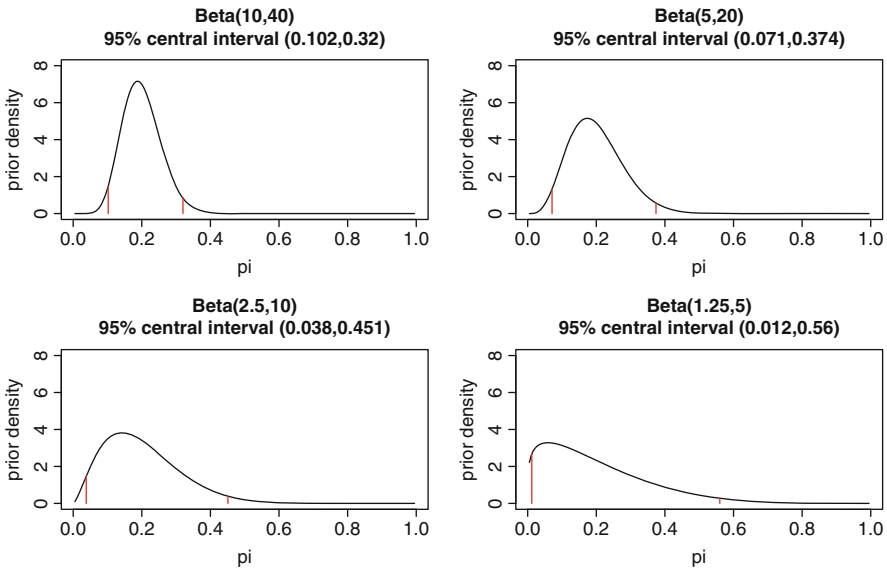


Fig. 3.9 95% prior intervals for beta densities with different parameter values

3.5.3 Computing and Graphing the Posterior Distribution

Suppose you chose the $\text{beta}(10, 40)$ prior because it best represented your beliefs. You then gathered your own data on $n = 50$ UI students, and, as we found out in Sect. 3.2.3, you got $y = 7$ “successes” and $n - y = 43$ “failures.” Then your posterior distribution of π given your beta prior and the new data is

$$p(\pi|y) \propto \pi^{\alpha+y-1} (1-\pi)^{\beta+n-y-1} \\ \pi^{17-1} (1-\pi)^{83-1}$$

This is the kernel of a $\text{beta}(17, 83)$ density.

If the $\text{beta}(1.25, 5)$ prior better represented my previous knowledge, then my posterior distribution for π , given my prior and your data, would be a $\text{beta}(8.25, 48)$.

3.5.4 Plotting the Prior Density, the Likelihood, and the Posterior Density

The plots in Fig. 3.10 show the result of combining the data ($y = 7$ successes and $n - y = 43$ failures) with each of the four beta prior densities from Fig. 3.8 to produce four different posterior densities for π . In these plots, the likelihood function has been normalized (multiplied by the factor that causes the area under the curve to be 1) to make it comparable to the prior and posterior densities.

Note that in all cases, the posterior density is more concentrated than either the prior density or the likelihood. This makes sense: When we combine our previous knowledge with the additional information in the current data, our knowledge becomes more precise than when we consider either one of the two sources alone. The posterior density always is in some sense a compromise between the prior density and the likelihood. The less informative the prior is (in the case of a beta prior, the smaller its parameter values), the more the data, as expressed in the likelihood, dominates the posterior density. We will make these notions much more precise in Sect. 4.2.1.

3.6 Introduction to R for Bayesian Analysis

R (R Core Development Team 2008) is a computer language and environment for statistical analysis and graphics. All of the graphics and many of the numeric results in this book have been produced using R.

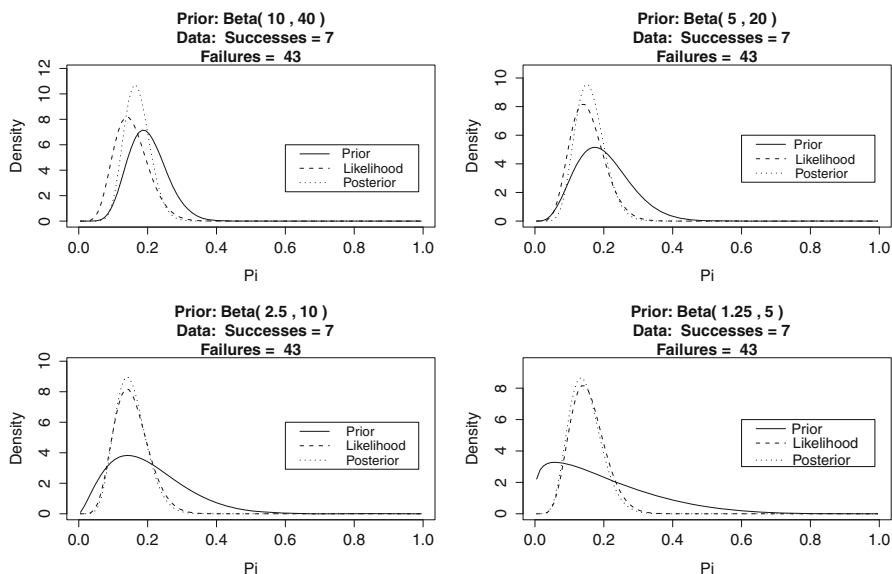


Fig. 3.10 Prior densities, normalized likelihoods, and resulting posterior densities

R is available for Windows, Linux, and Macintosh operating systems and may be downloaded for free from the web site: www.cran.r-project.org. Excellent manuals and other documentation can be viewed or downloaded at the same site. An additional recommended resource for using R for Bayesian analysis is [Albert \(2007\)](#).

R has hundreds of built-in functions for plotting and analyzing data, summarizing probability distributions, performing mathematical calculations, and many other purposes. In addition, R is a programming language, so you can write functions of your own. Furthermore, special-purpose add-on libraries of functions called “packages” have been developed, tested, and contributed by users. These can be downloaded and installed along with R to extend its capabilities.

3.6.1 Functions and Objects in R

I am assuming that you have access to a computer on which R is installed and that you know how to start R. You will be confronted by a window with prompt

```
>
```

If you type something after the prompt and press “Enter,” R will interpret it. For example, if you type

```
> 4 + 6
```

R will respond

```
[1] 10
```

The [1] is there because R thinks of numbers in vectors; the 10 is the first (and only!) element in the vector.

We can use R's sequence operator, the colon, to quickly instruct R to display a vector of consecutive integers:

```
> 3:55
[1] 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
    22 23 24 25 26 27
[26] 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
    47 48 49 50 51 52
[51] 53 54 55
```

Here R showed us the index of the element of the vector that appeared at the beginning of each row of the display.

So far R has just displayed output on the screen. Nothing has been saved in R's memory. We can use R's assignment operator—an arrow typed in two characters, a less-than sign, and a hyphen—to assign values to named objects. The following creates an object called `myvect` and stores a sequence of integers to it.

```
> myvect <- 3:55
>
```

It looks as if nothing has happened—we just got another prompt when we pressed “Enter.” However, if we now enter the name of the object, R will display its value.

```
> myvect
[1] 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    19 20 21 22 23 24 25 26 27
[26] 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
    44 45 46 47 48 49 50 51 52
[51] 53 54 55
```

Now we can use built-in R functions to carry out all kinds of operations on this named vector of numbers. To run an R function, type its name followed by parentheses. Usually you will need to put some values, called *arguments*, inside the parentheses to tell the function what to operate on. For example, `sd` is the name of the R function to calculate standard deviations:

```
> sd(myvect)
[1] 15.44345
```

We can also use the concatenation operator, `c`, to create vectors with entries that we specify:

```
> vect2 <- c(3.2, 1.97, -6.45)
> vect2
[1] 3.20 1.97 -6.45
```

R functions are also objects. If we type the name of the function without parentheses, R displays the content of the function itself:

```
> sd
function (x, na.rm = FALSE)
{
  if (is.matrix(x))
    apply(x, 2, sd, na.rm = na.rm)
  else if (is.vector(x))
    sqrt(var(x, na.rm = na.rm))
  else if (is.data.frame(x))
    sapply(x, sd, na.rm = na.rm)
  else sqrt(var(as.vector(x), na.rm = na.rm))
}
<environment: namespace:stats>
```

R has a built-in help facility. You can type the keyword `help` followed by the name of a function in parentheses, and R will display documentation for the function. For example, if you type

```
> help(sd)
```

a window will open containing the following:

```
sd(stats) R Documentation
```

```
Standard Deviation
```

```
Description
```

```
This function computes the standard deviation of the
  values in x.
```

```
If na.rm is TRUE then missing values are removed
  before
```

```
computation proceeds. If x is a matrix or a data
  frame, a vector
```

```
of the standard deviation of the columns is returned.
```

```
Usage
```

```
sd(x, na.rm = FALSE)
```

```
Arguments
```

```
x a numeric vector, matrix or data frame.
```

```
na.rm logical. Should missing values be removed?
```

In the “Usage” section, we are told that the `sd` function has two arguments. The user *must* provide a value for the first argument, *x*. However, the second argument

Table 3.1 Probability distributions in R

Distribution	R abbreviation
Beta	beta
Cauchy	cauchy
Chisquare	chisq
Exponential	exp
F	f
Gamma	gamma
Logistic	logis
lognormal	lnorm
Normal	norm
t	t
Uniform	unif
Weibull	weibull
Poisson	pois
Binomial	binom

Table 3.2 R functions for each probability distribution

Prefix	Function	Example
d	Density or probability mass function	dbeta, dbinom
p	Cumulative density or cumulative probability mass function	pbeta, pbinom
q	Quantiles	qbeta, qbinom
r	Random sample generation	rbeta, rbinom

has a default value of FALSE, which the function will use if the user doesn’t specify a second argument.

3.6.2 Summarizing and Graphing Probability Distributions in R

R has built-in functions for extracting characteristics of the following probability distributions shown in Table 3.1:

For each of the distributions, four functions are available, which differ only by their first letter shown in Table 3.2:

You can get detailed documentation on all four of the functions for any of the distributions by entering “help(name of one of the functions)”. For example,

```
help(dgamma)
```

will produce documentation on all four of the functions for the gamma distribution.

The quantile functions can be used to calculate the endpoints of probability intervals. For example, I used the qbeta function to calculate the prior probability intervals Sect. 3.5.2. Here is code to obtain the endpoints of a 95% central probability interval under a beta(10,40) prior:

```
> qbeta( c(0.025,0.975) , 10, 40)
[1] 0.1024494 0.3202212
```

We can use the density functions in combination with R's `plot` function to get density plots. At minimum, the `plot` function requires a vector of x coordinates and a vector of y -coordinates as its arguments. An optional `type` argument can be used to tell it to plot a line instead of points. Enter `help(plot)` for further details on the `plot` function.

The following code creates a vector x of 100 regularly spaced points on the interval $[0.005, 0.995]$, creates a vector y with the evaluations of the $\text{beta}(10, 40)$ density at the values in x , and plots the pairs as a line plot:

```
x <- seq(0.005, 0.995, length=100)
y <- dbeta(x, 10, 40)
plot(x, y, type="l")
```

If we want to be fancier, we can specify the labels for the x and y axes and a main title for the plot:

```
plot(x, y, type="l", xlab=expression(pi),
     ylab="Density",
     main = "Beta(10, 40)")
```

When plotting beta densities, I always restrict the x -axis to the interval $[0.005, 0.995]$ because some beta densities become unbounded at 0 and/or 1. (Try entering `qbeta(c(0, 1), 0.5, 0.5)`).

What about a density, such as the normal or gamma, that has unbounded support? We can use the relevant quantile function in R to choose an appropriate range of x values over which to plot such a density. Here's code to plot a normal density with mean 20 and standard deviation 5 over the interval between its 0.005 and 0.995 quantiles.

```
x <- seq( qnorm( 0.005, 20, 5 ), qnorm( 0.995, 20, 5 ),
        length = 100)
y <- dnorm( x, 20, 5)
plot( x, y, type="l")
```

3.6.2.1 Parameterizations in Statistical Software

Different statistical software packages use different parameterizations for probability distributions. *It is your responsibility as the user of statistical software to check the documentation (or experiment with the software) to find out which parameterization is being used.* For example, much statistical software expects the normal distribution to be specified in terms of the mean and variance, whereas R expects the mean and standard deviation—check `help(dnorm)`. WinBUGS and OpenBUGS—statistical software for fitting Bayesian models that we will employ extensively in this book—use still another parameterization. If you are thinking in terms of one parameterization but your software is using another, all of your results will be wrong.

3.6.3 *Printing and Saving R Graphics*

If you are running R under Windows, when you have completed a plot, you can use R's File menu to print it or to save it to disk in your choice of graphics file format. It is also possible to copy an R graph into a Word document. Right click on the plot window; then select "Copy as bitmap." Then paste into an open Word document.

If you are running R under Linux, you can use the `dev.print` function to print graphics. To save a graphic to disk, first create the graphic on the screen so you have the exact code. Then run the `pdf` function or the `postscript` function to direct the output to a file of the desired format. Then rerun the code to produce the graphic. Note: You can use the up-arrow key on your keyboard to step back through previously run lines of R code and re-execute them. If you are running R under Linux, use the built-in R help to learn how to use these functions.

3.6.4 *R Packages Useful in Bayesian Analysis*

The functions in R are grouped into components called *packages*. Those that provide the essential functionality in R (math, statistics, probability, basic graphics, etc.) are included in the basic R installation and are automatically loaded whenever R is run. Hundreds of additional packages have been developed to perform specialized functions that not all users need. A listing of these packages, with brief descriptions and download access to their documentation, is available at the Comprehensive R Archive Network website, <http://cran.r-project.org>.

If you are using R on a Windows or Macintosh computer with Internet access (and you are authorized to install software on the computer), the easiest way to download and install contributed packages is to use the packages pull-down menu right in R. If you are running R under Linux, you can use the `install.packages` function.

To see a list of the packages that have been loaded and so are currently available in your R session, enter

```
library()
```

To load a package that has been installed on your system but is not automatically made active whenever R is started, use the name of the package as the argument of the `library` function. For example,

```
library(survival)
```

would load the `survival` package (assuming that it was installed on the computer).

In this session, we will use an R package that has been written specifically to help in learning Bayesian statistical concepts and elementary procedures. Later on in the course, we will use other R packages that perform more sophisticated Bayesian data analyses.

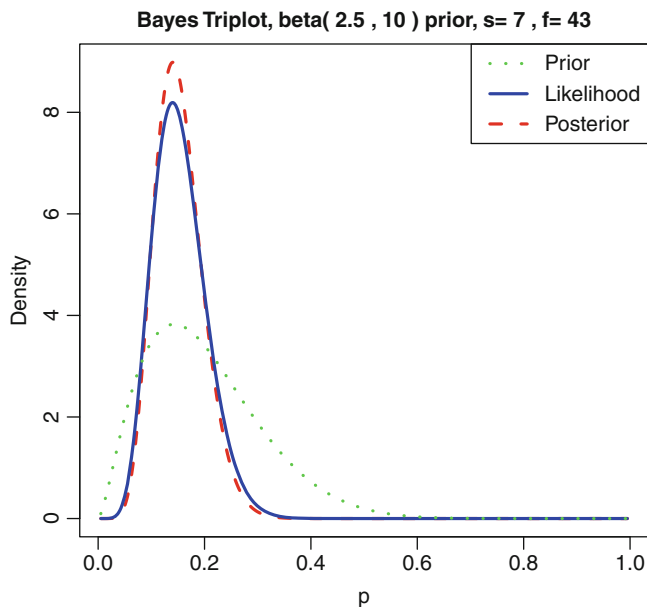


Fig. 3.11 Output of triplot function in LearnBayes package

3.6.4.1 LearnBayes

The LearnBayes package (Albert 2008) was written by Jim Albert to accompany his textbook *Bayesian Computation with R* (Albert 2007). Releases 2.0 and higher of LearnBayes include a function called `triplot`, which makes plots similar to those in Fig. 3.10. Assuming that the LearnBayes package is installed on the computer on which you are working, the following code will load the package and get documentation of the `triplot` function:

```
library(LearnBayes)
help(triplot, package="LearnBayes")
```

Note that the function requires two arguments: a vector of length two giving the parameters of the beta prior and a vector of length two giving the number of successes and number of failures in the observed data. An optional third argument may be provided to control placement of the legend in the plot area. The following function call will create the plot for the $\text{beta}(2.5, 10)$ prior and the survey data with 7 yesses and 43 noes:

```
triplot( c(2.5, 10), c(7,43) )
```

The result is in Fig. 3.11.

To move the legend from the default position (upper right corner of the plot) to the middle, we could add the third argument as follows:

```
triplot( c(2.5, 10), c(7,43), "center")
```

3.6.5 Ending a Session

R maintains what it calls the “workspace,” in which it stores any functions or other objects that have been created during a given session. If you exit from R without saving the workspace image, then any new work you have done will be lost. If you are working on a lab computer and wish to save your work to a flash drive or other external storage device so that you can reload it into R on another computer later, use the “File” menu (in Windows) or the `chpwd()` function (in Linux) to *change the working directory* to the desired device before leaving R.

To exit from R, enter

```
> q()
```

You will be asked whether you want to save the workspace image. If you respond “No,” then any new objects created during this session will be lost. If you are working on your own computer, and you want all objects you created to remain associated with R, choose “Yes.”

Problems

3.1. For the beta density with parameters $\alpha = 2$ and $\beta = 7$, do the following:

1. Referring to Table A.2, calculate the mean and mode as functions of the parameters.
2. Use an R function to determine the median and a 90% central interval.
3. Plot the density.

3.2. To see some of the different shapes that beta densities may take on, plot each of the following densities:

1. $\text{Beta}(0.5, 0.5)$
2. $\text{Beta}(10.2, 1.5)$
3. $\text{Beta}(1.5, 10.2)$
4. $\text{Beta}(100, 62)$

3.3. The uniform distribution is a special case of the beta distribution.

1. What are the numeric values of its parameters? That is,

$$U(0, 1) = \text{Beta}(?, ?)$$

2. What is the equivalent prior sample size for a $U(0, 1)$ prior?

3.4. [This problem is loosely based on examples from Chap. 4 of [Albert \(1997\)](#)]. You are the assistant coach of the women’s softball team at a Midwestern college. The head coach has asked you to assess a new first year player who is joining the

team. As a high school student, she was at bat 120 times and got 40 hits. You wish to estimate θ , her underlying true probability of getting a hit in any at bat as a college-level player.

1. Specify a beta prior that seems appropriate to capture your knowledge or uncertainty about θ before the new player plays in any college-level games. Use any information you have that seems important—her high school record, anything you know about college-level women’s softball, etc. Use R functions as needed. Explain in a few sentences (supplemented with plots and/or R output) how you chose the values of α and β .

There is no one right answer here—I want to see how you think about this and what procedure you use.

2. Specify a beta prior that you think might reflect the player’s mother’s beliefs about θ . This may be similar to, or quite different from, your prior. Again, justify your choice with graphical or numeric R output.
3. Suppose the player now plays eight college-level games, has thirty at bats, and gets 5 hits. Thus, the data are

$$y = 5, \quad n = 30$$

We will use a binomial likelihood for these data. This requires the assumption that, conditional on θ , each at bat is an independent Bernoulli trial with success probability θ . There are several reasons why independence might actually *not* be a reasonable assumption in this problem. Give one.

Note: For our present purposes, we’ll use a binomial likelihood anyway. We’ll come up with a better model when we talk about hierarchical models later in the semester.

4. Obtain the following characteristics of the posterior distribution $p(\theta|y)$
 - a. Name of posterior distribution and its parameter values
 - b. Posterior density plot
 - c. A plot showing the prior density, the likelihood, and the posterior density, all on the same axes

based on the data from the college-level games under each of three priors:

- a. Your prior from part 1
- b. The mother’s prior from part 2
- c. A uniform (noninformative) prior

This problem will be continued at the end of Chap. 4.

Chapter 4

Inference for a Population Proportion

Statistical inference is drawing conclusions about an entire population based on data in a sample drawn from that population. From both frequentist and Bayesian perspectives, there are three main goals of inference: *estimation*, *hypothesis testing*, and *prediction*. Estimation and hypothesis testing deal with drawing conclusions about unknown and unobservable population parameters.

Prediction is estimating the values of potentially observable but currently unobserved quantities. For example, we might want to predict the number of “yesses” in a future survey of 50 UI students. Prediction in statistical inference isn’t restricted to predicting future observations, however. It may refer to estimating values that have already occurred but were not measured. For example, we may want to use values of acid rain deposition measured from rain gauges at specific sites to predict acid rain deposition at other locations that have no rain gauges.

Before investigating how the Bayesian uses the posterior distribution of a population parameter to make inference, we will review the approach usually undertaken by frequentists so that we are ready to make comparisons.

4.1 Estimation and Testing: Frequentist Approach

4.1.1 Maximum Likelihood Estimation

When trying to estimate the unknown value of a population parameter, the frequentist statistician, like the Bayesian, begins by specifying the distribution of the data, given the unknown parameter(s). In the case of our data consisting of independent yes/no responses to the survey questions, this will be the binomial probability mass function, first given in (3.1) and repeated here:

$$p(y|\pi) = \binom{n}{y} \pi^y (1 - \pi)^{n-y}, \quad y = 0, 1, \dots, n$$

Then the statistician switches perspective and views the same expression as a function of the unknown parameter, given known data values. The frequentist does *not* treat the parameter as if it were a random variable and does not specify a prior distribution to summarize other information not contained in the current dataset.

One goal of frequentist estimation is to obtain a *point estimate* of a population parameter. The point estimate may be thought of as the best single-number guess of the value of the population parameter, based solely on the current data. The most commonly used method of frequentist point estimation is maximum likelihood estimation—finding the value of the parameter that would give the largest possible evaluation of the likelihood. Intuitively, this is the value of the parameter that would have made the observed data the most likely.

A maximum likelihood *estimate* is the numeric value calculated for a particular dataset. A maximum likelihood *estimator* is the formula for calculating maximum likelihood estimates for a given form of the likelihood. When the likelihood is a continuous function of the unknown parameter, as is the case with the binomial likelihood, we can use calculus to find the maximum likelihood estimator. It is usually easier to do this for the log of the likelihood instead of the likelihood in its original form. Since the log transformation is monotonic, the same value will maximize both. The steps should be familiar: Take the first derivative of the log likelihood with respect to the parameter; set the first derivative equal to 0 and solve for the parameter in terms of the data; verify that the second derivative of the log likelihood is negative at this point (i.e., that we have a maximum rather than a minimum—a minimum likelihood estimator wouldn't be very useful!). We'll go through the process for the binomial likelihood here, in part, because the same calculus steps will be needed later on when we study Jeffreys priors.

The log of the binomial likelihood is

$$l(\pi) = \log \binom{n}{y} + y \log(\pi) + (n - y) \log(1 - \pi),$$

$$0 < \pi < 1$$

The first derivative with respect to π is

$$\frac{dl(\pi)}{d\pi} = \frac{y}{\pi} - \frac{n - y}{1 - \pi}$$

Setting this equal to 0 and solving for π show that the sample proportion is the maximum likelihood estimator of the population proportion:

$$\hat{\pi} = \frac{y}{n}$$

In our example, the m.l.e. is $\hat{\pi} = \frac{7}{50} = 0.14$.

We can verify that this is a maximum rather than a minimum by taking the second derivative of the log likelihood:

$$\frac{d^2 l(\pi)}{d\pi^2} = -\frac{y}{\pi^2} - \frac{n-y}{(1-\pi)^2}$$

Evaluating this at $\hat{\pi} = \frac{y}{n}$ gives $\frac{-n^3}{y(n-y)}$, which is strictly negative (although undefined unless there are at least 1 success and 1 failure in the sample). Yes, we have a maximizer.

4.1.2 Frequentist Confidence Intervals

In addition to obtaining a point estimate, the frequentist needs some measure of how good the point estimate is. The estimate, based on a limited sample, is very unlikely to be exactly equal to the true population parameter.

Recall that the frequentist approach to statistics is so named because it is based on the long-run frequency interpretation of probability. The way that the frequentist addresses the question of how good the point estimate is likely to be is by asking another question: What would happen if we drew many, many, many random samples, all of the same size, from the same population, and calculated the sample proportion for each one? Different samples would be likely to produce different numbers of “yesses,” and thus different sample proportions $\hat{\pi}$. The *sampling distribution* of an estimator is the distribution of the values it would take on over all possible random samples of the chosen size drawn from the population of interest.

Frequentist confidence intervals are intervals that we have some reason to believe contains the true value of the unknown population parameter. Frequentist procedures for calculating confidence intervals depend on data values in the sample at hand. Therefore, different samples will produce not only different point estimates but different confidence intervals as well. The fundamental idea behind frequentist confidence intervals is that the procedure performs as claimed under repeated sampling. For example, the procedure for computing intervals with confidence level 95% will produce an interval that actually does contain the true parameter value when applied to 95% of simple random samples drawn from the population of interest.

We have only one sample. We have no way of knowing for sure whether our particular sample is one of the lucky 95% of all possible samples that do produce an interval that contains the true value of π , or whether we instead by bad luck got one of the 5% of samples that produce 95% confidence intervals that don’t contain the true value.

R includes several built-in functions that can be used for frequentist estimation and hypothesis testing regarding population proportions. The `binom.test` function can be used to compute the maximum likelihood estimate $\hat{\pi}$ and a confidence interval for the population proportion π . The syntax is

```
binom.test( number of successes, total sample size,
            conf.level )
```

To get the point estimate and a 90% confidence interval for π based on our survey data with 7 success in 50 trials, we would enter

```
binom.test( 7, 50, conf.level=.90 )
```

The output is

```
Exact binomial test
```

```
data: 7 and 50
number of successes = 7, number of trials = 50,
p-value = 2.099e-07
alternative hypothesis: true probability of success
is not equal to 0.5
90 percent confidence interval:
 0.0675967 0.2469352
sample estimates:
probability of success
              0.14
```

Thus, $\hat{\pi} = 0.14$, and the frequentist is 90% confident that the true value of π is in the interval (0.0676, 0.2469).

This is very different from saying that the probability that π is in the interval is 0.90. π has some fixed value. If that value is 0.07, it's in the interval. If it's 0.06, it isn't. Once we have selected our sample and computed numeric values of the interval endpoints, there's no more probability involved.

4.1.3 Frequentist Hypothesis Testing

Hypothesis testing is appropriate when different courses of action would be taken given different values of an unknown population parameter. In our student survey example, if we had evidence that the population proportion of all UI students who would quit school because of a 19% tuition increase was very small, we would not want to take our argument to the regents. On the other hand, if we were convinced that the proportion was more substantial, say larger than 10%, we might indeed want to go before the regents to argue against the tuition increase.

Recall that a statistical hypothesis is a statement about an unknown population parameter. Hypothesis tests involve setting up two such statements, which are mutually exclusive.

Thus, we might want to test the following hypotheses regarding π :

$$H_0 : \pi \leq 0.1$$

$$H_a : \pi > 0.1$$

The frequentist uses the *p-value* to evaluate the evidence in the data against the null hypothesis H_0 and in favor of the alternative H_a . The concept of the p-value again is rooted in the question of what would happen under repeated sampling, specifically, *assuming that H_0 is true*, if we draw many, many simple random samples, what is the probability of getting a sample with as much evidence against H_0 as our actual sample has, or even more. That is, the p-value most definitely is *not* the probability that H_0 is true. Instead, it is based on the assumption that H_0 is true and states a probability about data results under repeated sampling. The smaller the p-value, the less likely it would have been to draw a sample like ours if H_0 were true. Thus, small p-values indicate that the data is inconsistent with the null hypothesis.

The R function `binom.test` can be used to test hypotheses about a population proportion. To test $H_0 : \pi \leq 0.1$ versus $H_a : \pi > 0.1$, we would enter

```
binom.test( 7, 50, p = .1, alternative="greater")
```

and the output is

```
Exact binomial test
```

```
data: 7 and 50
number of successes = 7, number of trials = 50,
p-value = 0.2298
alternative hypothesis: true probability of success
is greater than 0.1
95 percent confidence interval:
 0.0675967 1.0000000
sample estimates:
probability of success
      0.14
```

The critical part of the output for the hypothesis test is that the p-value is 0.2298. Since the null hypothesis says that π is small (less than or equal to 0.1), large numbers of successes in the sample would provide evidence against the null. In this case, “as much evidence against the null as our sample provides, or more” means 7 or more successes. Hence, this p-value is the probability of getting a random sample of size 50 with 7 or more successes, if the true value of π is 0.1. Thus, we would have had between a one-in-five and a one-in-four chance of getting a random sample of size 50 with at least 7 “yesses” even if the population proportion of yesses was as small as 0.1.

4.2 Bayesian Inference: Summarizing the Posterior Distribution

All Bayesian inference is based on the posterior distribution, which contains all the current information about the unknown parameter. Although a plot of the posterior density gives a full graphical description, numeric summaries of the posterior are needed as well.

4.2.1 The Posterior Mean

The mean of the posterior distribution is often used as the Bayesian *point estimate* of a parameter. For a beta prior and binomial likelihood, the posterior mean is

$$E(\pi|y) = \frac{\alpha + y}{\alpha + y + \beta + n - y} = \frac{\alpha + y}{\alpha + \beta + n}$$

In our example, with the beta(10, 40) prior

$$E(\pi|y) = \frac{17}{100} = 0.17$$

For a beta prior and binomial likelihood, the posterior mean is always *between* the prior mean and the value $\frac{y}{n}$ computed from the current data.

In our example, the prior mean was 0.20, and $\frac{y}{n} = 0.14$. In our example, if we instead had used the beta(1.25, 5) prior

$$E(\pi|y) = \frac{8.25}{56.25} = 0.147$$

More specifically, the posterior mean is a weighted average of the prior mean and the m.l.e. $\hat{\pi}$. If we denote the posterior mean by μ_{post} , then

$$\begin{aligned} \mu_{post} &= \frac{\alpha + y}{\alpha + \beta + n} \\ &= w \frac{\alpha}{\alpha + \beta} + (1 - w) \frac{y}{n} \end{aligned}$$

where $w = \frac{\alpha + \beta}{\alpha + \beta + n}$. Bayesians refer to this weighted averaging as “shrinkage” —in calculating the posterior mean, the observed sample proportion $\hat{\pi}$ is “shrunk” toward the prior mean $\frac{\alpha}{\alpha + \beta}$. The weight placed on the prior mean $\frac{\alpha}{\alpha + \beta}$ is proportional to the sum of the two prior parameters, and the weight placed on the m.l.e. is proportional to the sum of the number of successes and number of failures in the observed data.

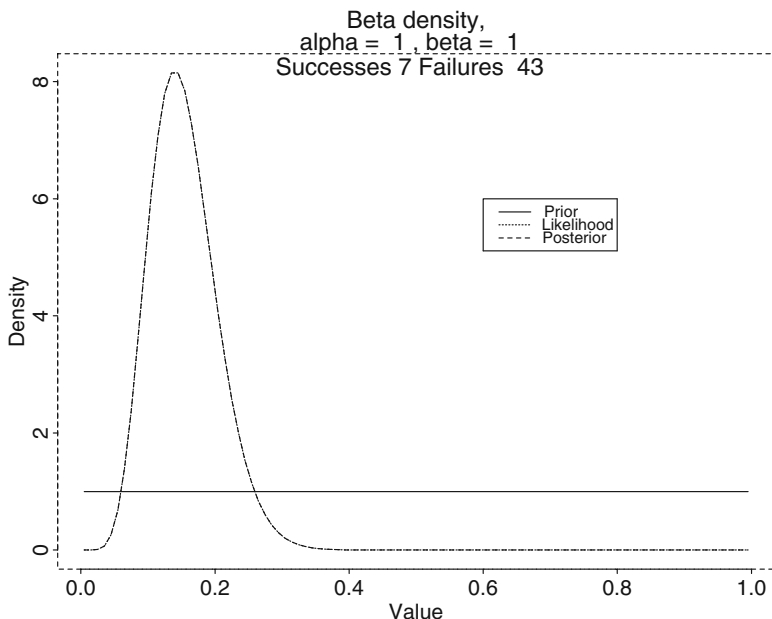


Fig. 4.1 Uniform prior and binomial likelihood

4.2.2 Other Bayesian Point Estimates

The posterior median and posterior mode are sometimes used instead of the posterior mean as Bayesian point estimates. We will investigate different point estimates in the context of the posterior distribution produced with a $U(0, 1)$ prior and a binomial likelihood:

$$p(\pi|y) = \text{Beta}(1 + y, 1 + n - y) \\ \propto \pi^y (1 - \pi)^{n-y},$$

which is proportional to the likelihood, as we noted in Sect. 3.4 and illustrated in Fig. 4.1.

The posterior mean is *not* equal to the m.l.e. $\hat{\pi}$. From Table A.2, observe that the *mode* of a $\text{Beta}(\alpha, \beta)$ distribution is $\frac{\alpha-1}{\alpha+\beta-2}$. Thus, with a uniform prior, the mode of the posterior distribution given above is $\frac{y}{n} = \hat{\pi}$. Again, this makes sense: If the posterior density is proportional to the likelihood, then the same value will maximize both—that is, the posterior mode equals the m.l.e.

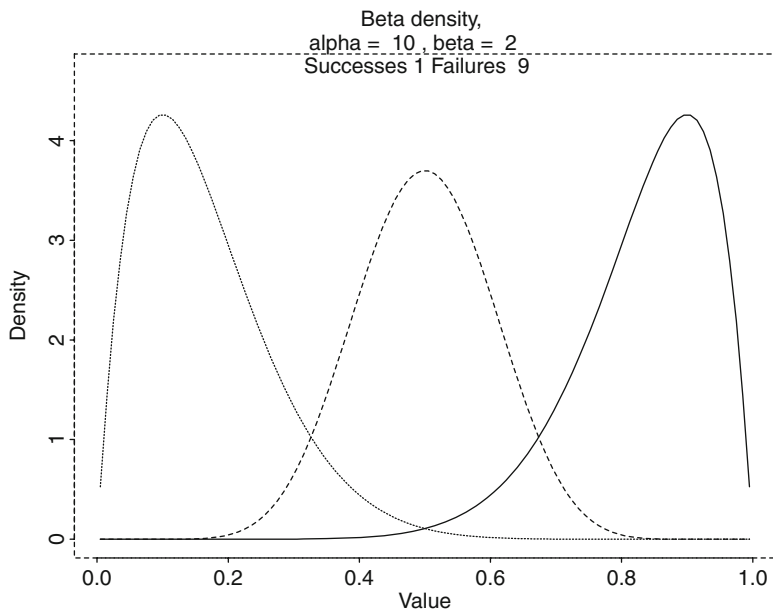


Fig. 4.2 Illustration of conflict between the prior and the data

4.2.2.1 The Posterior Variance

The posterior variance is one summary of the spread of the posterior distribution. The larger the posterior variance, the more uncertainty we still have about the parameter, even after learning from the current data. See Table A.2 for the formula for the variance of a random variable with a beta distribution. For an informative beta prior and a binomial likelihood, the posterior variance is almost always smaller than the prior variance. This makes intuitive sense: When we add the information contained in the current data, we have more precise knowledge (less uncertainty) than what was expressed in the prior alone.

In our school-quitting example, with the uniform prior, the prior variance = $\frac{1}{12} = 0.083$, and posterior variance = 0.00246. If we instead used the beta(10, 40) prior, the prior variance = 0.00314 and the posterior variance = 0.00140. As we would expect, the posterior variance is smaller with the informative beta(10,40) prior than with the noninformative uniform prior.

The exceptional cases, when the posterior variance is not smaller than the prior variance, occur when the prior and the data are in direct conflict. Figure 4.2 illustrates a worst-case scenario of this type. The beta prior density has mean 0.1, while the sample proportion in the data is 0.9. The posterior density is a compromise between the prior and the likelihood, but the posterior variance is not smaller than the prior variance.

4.2.3 Bayesian Posterior Intervals

Intervals called “credible sets” also are used as numeric posterior summaries. There are two commonly used kinds.

4.2.3.1 Equal-Tail Posterior Credible Sets

You have already encountered equal-tail prior intervals in Sect. 3.3.2. The same logic can be applied to produce a posterior central interval with a specified probability of containing the true parameter value. For example, the endpoints of a 95% equal-tail credible set are the 0.025 and the 0.975 quantiles of the posterior distribution. We can use built-in R functions to calculate them. For our quitting-school problem with the $\text{beta}(10,40)$ prior, the posterior density was $\text{beta}(17, 83)$, and the `qbeta` function in R can be used as follows:

```
> qbeta( c(0.025, 0.975), 17, 83 )
[1] 0.1033333 0.2491463
```

This interval is shown graphically in Fig. 4.3.

If we had instead used a uniform prior, so that the posterior was $\text{beta}(8,44)$, then the 95% equal-tail credible set would be

```
> qbeta( c(0.025, 0.975), 8, 44)
[1] 0.07024083 0.26255154
```

This interval is shown in Fig. 4.4.

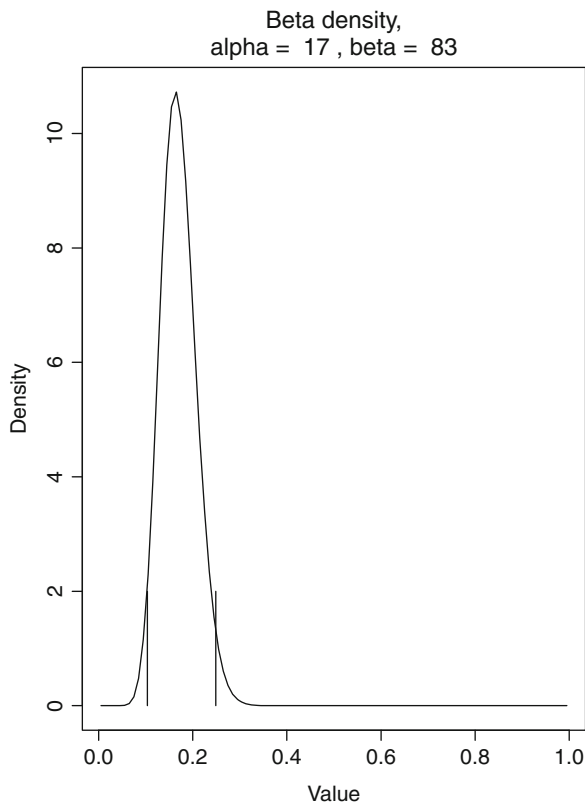
Note that this interval is wider than that obtained with the $\text{beta}(10, 40)$ prior.

The meaning of “equal tail” is that the same amount of area under the posterior density curve is excluded from the interval on the low end as on the high end. Equal-tail credible sets are the most commonly reported Bayesian posterior intervals. (If an author doesn’t identify what kind of posterior interval is being reported, you can assume that it is equal tail.) Equal-tail intervals are easy to compute and easy to understand. Their disadvantage is that, unless the posterior density is symmetric and unimodal, there may be points outside the interval that have higher posterior density than some points inside the interval. Figure 4.5 shows an extreme example. The posterior density depicted is bimodal with widely separated modes. The equal-tail credible set includes the region around 5, where the posterior density is much smaller than it is immediately outside the interval.

4.2.3.2 Highest Posterior Density Regions

The other kind of Bayesian posterior interval is the *highest posterior density region*, or HPD region. The posterior density at any point *inside* such an HPD region is greater than the density at any point *outside* it. The HPD region also is the shortest possible interval trapping the desired probability. HPD regions are preferable to

Fig. 4.3 Vertical lines indicate endpoints of 95% equal-tail posterior credible set; beta(10,40) prior



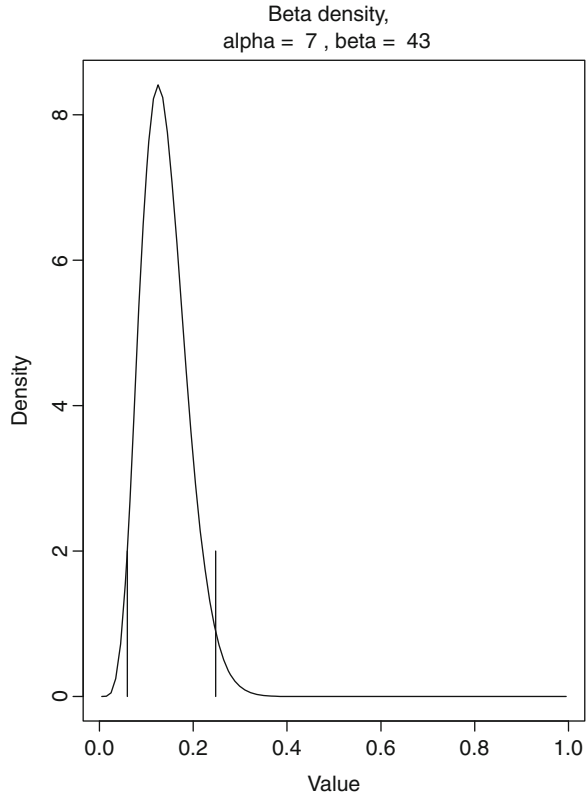
equal-tail credible sets when the posterior is highly skewed or multimodal. However, they are generally difficult to compute.

The intuition behind the computation of an HPD region is as follows. Suppose that we want a 95% posterior probability region. We begin by placing a horizontal line just touching the posterior density curve at its mode. We then slide the line downwards toward the x-axis until it cuts the density curve at points such that the area under the density curve between these points is exactly 0.95. As illustrated in Fig. 4.6, the HPD region may not even be an interval. In this case, it is the union of two disjoint intervals—the intervals where the horizontal dashed line lies below the curve; the low-probability region in the middle is not part of the HPD region.

4.2.3.3 Interpretation of Bayesian Intervals

Recall that the posterior distribution represents our updated subjective probability distribution for the unknown parameter. Thus, for us, the interpretation of the 95% credible set is that the probability that the true π is in that interval is 0.95. For example, if the beta(10,40) had been a true representation of our prior beliefs or

Fig. 4.4 Vertical lines indicate endpoints of 95% equal-tail posterior credible set; uniform prior



knowledge about the parameter π , then after seeing our survey data, we would believe that

$$P(0.103 < \pi < 0.249) = 0.95$$

This is precisely the kind of statement that the frequentist cannot make about confidence intervals. This is one of the implications of the difference between the long-run frequency interpretation of probability and the subjective interpretation of probability.

4.3 Using the Posterior Distribution to Test Hypotheses

We now revisit our test of the following hypotheses, this time from a Bayesian perspective:

$$H_0 : \pi \leq 0.1$$

$$H_1 : \pi > 0.1$$

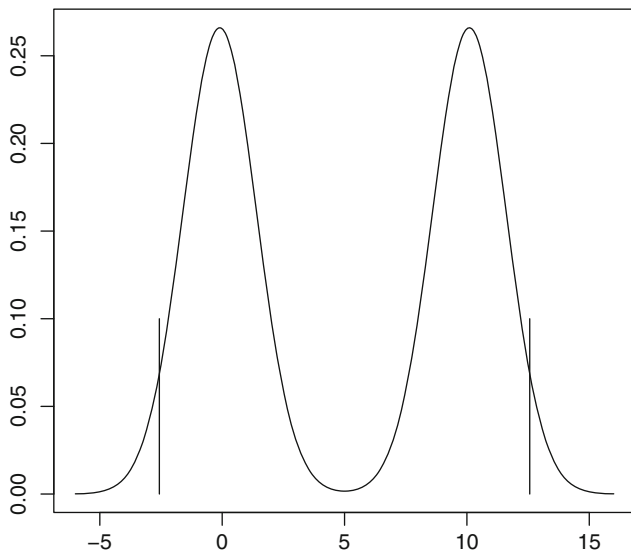


Fig. 4.5 Equal-tail credible set for a bimodal density

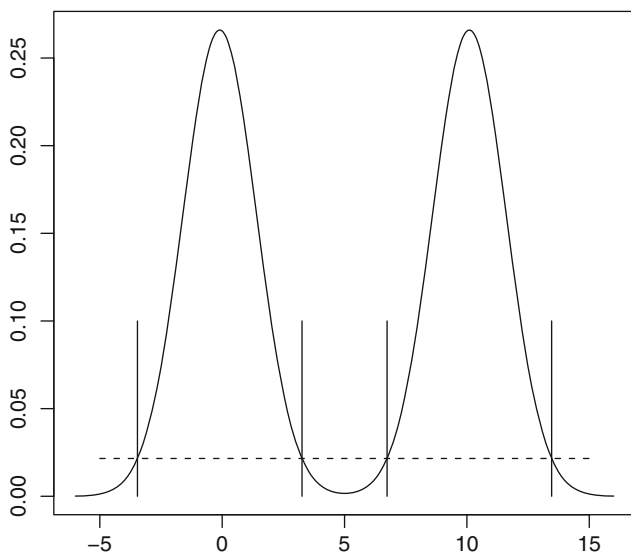


Fig. 4.6 HPD region for a bimodal density is two disjoint intervals—the interval between the two left more vertical lines and the one between the two right more vertical lines

We simply need the posterior probabilities of these two ranges of values for π . Suppose that the $\text{beta}(10, 40)$ had been our true prior, so our posterior distribution is $\text{beta}(17, 83)$. We can use a built-in R function to obtain $P(\pi \leq 0.1 \mid y)$.

```
> pbeta(.1, 17, 83)
[1] 0.01879825
```

With this prior, we would conclude that $P(\pi \leq 0.1 \mid y) = 0.019$.

If we instead had used the uniform prior, so our posterior was $\text{beta}(8, 44)$,

```
> pbeta(.1, 8, 44)
[1] 0.1329079
```

With this prior, we would conclude that $P(\pi \leq 0.1 \mid y) = 0.133$. Note that different people, approaching the question with different prior information, will end up with different (subjective) posterior probabilities on H_0 . Different people also will have different views on how small $P(\pi \leq 0.1 \mid y)$ has to be in order for it to be appropriate to go before the regents.

The interpretation of a Bayesian posterior probability is totally different from that of a frequentist p-value. Recall that a frequentist p-value is the probability, evaluated under the assumption that the null hypothesis is true, of drawing a random sample that contained as much evidence against the null as, or more than, the dataset we actually have. A frequentist p-value cannot be interpreted as the probability that the null hypothesis is true.

4.4 Posterior Predictive Distributions

In many studies, the research question of interest is predicting values of a future sample from the same population. Statisticians speak of *estimating* unobservable population parameters but of *prediction* values of potentially observable, but not yet observed, quantities.

For example, suppose we are considering interviewing another sample of 50 UI students in the hope of getting more evidence to present to the regents, and we would like to get an idea of how it is likely to turn out before we go to the trouble of doing so!

More generally, we are considering a new sample of sample size n^* and want to estimate the probability of some particular number y^* of successes in this sample. If, based on the already-completed study, we actually knew the true value of the population proportion π , we'd just use the binomial probability:

$$p(y^* \mid \pi) = \binom{n^*}{y^*} \pi^{y^*} (1 - \pi)^{n^* - y^*}, \quad y^* = 0, \dots, n^*$$

But of course, we still have uncertainty about π even after observing the original sample, so this won't work. We need the probability of getting y^* successes in a future sample given the information in our current data y , not given some particular value of π . Recall that all of our current knowledge about π is contained in the posterior distribution obtained using the original survey. Thus, we must integrate the binomial probability mass function for y^* given π over the posterior density of

π . Thus, the *posterior predictive probability* of getting some particular value of y^* in a future sample of size n^* is

$$p(y^* | y) = \int_0^1 p(y^* | \pi) p(\pi | y) d\pi, \quad y^* = 0, \dots, n$$

where y denotes the data from the original sample and $p(\pi | y)$ is the posterior distribution based on that sample. This is particularly easy to compute if $n^* = 1$, in which case the probability of getting 1 success is π , so

$$\begin{aligned} Pr(y^* = 1 | y) &= \int_0^1 Pr(y^* = 1 | \pi) p(\pi | y) d\pi \\ &= \int_0^1 \pi p(\pi | y) d\pi \\ &= E(\pi | y) \end{aligned}$$

because, by definition, the expected value of a random variable is obtained by integrating the random variable over its density. This is just the posterior mean of π . If we had used the $\text{beta}(10, 40)$ prior, resulting in the posterior density being $\text{beta}(17, 83)$, then $Pr(y^* = 1 | y) = \frac{17}{17+83} = 0.17$.

In general, if a Bayesian analysis has been done to estimate a population proportion π , using a $\text{beta}(\alpha, \beta)$ prior and a dataset with y successes in a sample of size n , then the posterior density $p(\pi | y)$ is $\text{beta}(\alpha_{post}, \beta_{post})$, where $\alpha_{post} = \alpha + y$ and $\beta_{post} = \beta + n - y$, and the predictive probability of getting y^* successes in a future sample of size n^* is

$$\begin{aligned} p(y^* | y) &= \int_0^1 \binom{n^*}{y^*} \pi^{y^*} (1 - \pi)^{n^* - y^*} \frac{\Gamma(\alpha_{post} + \beta_{post})}{\Gamma(\alpha_{post})\Gamma(\beta_{post})} \pi^{\alpha_{post} - 1} (1 - \pi)^{\beta_{post} - 1} d\pi \\ &= \binom{n^*}{y^*} \frac{\Gamma(\alpha_{post} + \beta_{post})}{\Gamma(\alpha_{post})\Gamma(\beta_{post})} \int_0^1 \pi^{y^* + \alpha_{post} - 1} (1 - \pi)^{n^* - y^* + \beta_{post} - 1} d\pi \end{aligned}$$

The expression inside the integral is the kernel of yet another beta density— $\text{beta}(y^* + \alpha_{post}, n^* - y^* + \beta_{post})$ —so we can easily figure out to what it will integrate. Remember that the normalizing constant of a density makes the density integrate to 1 over its entire support. Thus, the unnormalized density must integrate to $\frac{1}{\text{normalizing constant}}$. So the posterior predictive probability will be

$$p(y^* | y) = \binom{n^*}{y^*} \frac{\Gamma(\alpha_{post} + \beta_{post})}{\Gamma(\alpha_{post})\Gamma(\beta_{post})} \frac{\Gamma(y^* + \alpha_{post})\Gamma(n^* - y^* + \beta_{post})}{\Gamma(\alpha_{post} + \beta_{post} + n^*)} \quad (4.1)$$

The following property of gamma functions can be used to simplify (4.1):

$$\Gamma(x+1) = x\Gamma(x)$$

for any $x > 0$.

The function `pbetap` in the R package `LearnBayes` calculates predictive probabilities. The arguments are a vector containing α and β parameters of the beta posterior distribution based on the current data, the size of the future sample, and a vector of the numbers of successes in the future sample for which we want probabilities. For example, suppose our posterior distribution based on our existing survey data is $\text{beta}(8, 44)$ and we are planning a new survey with sample size $n^* = 25$. We can get the probabilities of obtaining 3, 4, 5, or 6 “yesses” in that future sample by entering

```
library(LearnBayes)
pbetap( c( 8, 44), 25, 3:6 )
```

The output is

```
0.18597889 0.17310343 0.13631895 0.09376436
```

We could make the output more readable by assigning the results of `pbetap` to an object and then displaying the requested numbers of successes and their corresponding predictive probabilities in two columns:

```
pprobs <- pbetap( c( 8, 44), 25, 3:6 )
cbind(3:6, pprobs)
```

```
      pprobs
[1,] 3 0.18597889
[2,] 4 0.17310343
[3,] 5 0.13631895
[4,] 6 0.09376436
}
```

Problems

4.1. This is a continuation of Problem 3.4 in Chap. 3.

1. Obtain the following characteristics of the posterior distribution $p(\theta|y)$
 - a. Posterior mean and mode
 - b. 95% posterior interval for θ
 - c. Posterior probability that $\theta > 0.25$

based on the data from the college-level games under each of three priors:

- a. Your prior
- b. The mother's prior
- c. A uniform prior

4.2. During the severe floods in the Midwest in the summer of 2008, the adjacent towns of Iowa City and Coralville in Johnson County, Iowa, were hit hard. Despite sustained efforts at sandbagging throughout the community, hundreds of homes, businesses, churches, and university buildings were damaged or destroyed. Major roads and bridges were closed for weeks. Less than a year later, with parts of both towns still recovering from the flood, a vote was held on a proposal to impose a local sales tax of one cent on the dollar to pay for flood-prevention and flood-mitigation projects. A few days before the actual vote, a local newspaper reported in its online edition, The Gazette Online (<http://www.gazetteonline.com>), on May 2, 2009:

“The outcome of Tuesday’s local-option sales tax election in Johnson County appears too close to call, based on results from a Gazette Communications poll of voters.

The telephone survey of 327 registered voters in Johnson County, conducted April 27–29, shows 40% in favor of the 4-year 1% sales tax. . .”

(Forty percent of 327 respondents is 131.)

A member of a local organization called “Ax the Tax” claims that this means that under half of all registered voters in the county support the local-option sales tax. She would like to use the sample survey data of the newspaper to test the two hypotheses:

$$H_0 : \pi \geq 0.5$$

$$H_a : \pi < 0.5$$

where π represents the proportion of all Johnson County registered voters who support the sales tax.

1. A frequentist method of testing these hypotheses is based on the p-value. The p-value is the probability of observing the sample result obtained, or something more extreme, if indeed exactly half of the registered voters in Johnson County supported the sales tax; that is,

$$p - \text{value} = Pr(y \leq 131 | \pi = 0.5)$$

where y is a binomial random variable with sample size $n = 327$ and success probability $\pi = 0.5$. Compute the p-value for this example (use of an R function will make this easy). If this probability is small, then one concludes that there is significant evidence in support of hypothesis $H_a : \pi < 0.5$.

2. Now, consider a Bayesian approach to testing these hypotheses. Suppose that a uniform prior is assigned to π . Find the posterior distribution of π and use it to compute the posterior probabilities of H_0 and H_1 .

4.3. Referring to the previous problem, again suppose that a uniform prior is placed on the proportion π , and that from a random sample of 327 voters, 131 support the sales tax. Also suppose that the newspaper plans on taking a new survey of 20 voters. Let y^* denote the number in this new sample who support the sales tax.

1. Find the posterior predictive probability that $y^* = 8$.
2. Find the 95% posterior predictive interval for y^* . Do this by finding the predictive probabilities for each of the possible values of y^* and ordering them from largest probability to smallest. Then add the most probable values of y^* into your probability set one at a time until the total probability exceeds 0.95 for the first time.

Chapter 5

Special Considerations in Bayesian Inference

5.1 Robustness to Prior Specifications

In statistics, an inference is described as *robust* if it is not affected substantially by changing the assumptions used in drawing it. Frequentists as well as Bayesians must worry about robustness of their inferences. One concern for both camps is that assumptions regarding the form of the likelihood can affect inference. For example, suppose that our data are scores obtained by 100 undergraduates on a calculus exam, and that we wish to use these data to estimate the mean of the scores that would have been obtained if all undergraduates who took calculus I that year had taken this exam. We might get very different estimates under each of the following assumptions:

1. The 100 scores are independent draws from a normal distribution.
2. The population of scores is likely to include extreme outliers, so the 100 scores are independent draws from a t distribution with 5 degrees of freedom.
3. Students who took the same calculus class are likely to have more similar scores than students in different classes; thus, rather than treating the 100 scores as independent, the likelihood must account for correlation within classes.

Furthermore, frequentists make assumptions about characteristics of the population of interest when they perform power and sample-size calculations during the design of a study.

In addition to all the robustness issues encountered by frequentist statisticians, the Bayesian must consider whether inference is robust to different prior specifications. Would different choices of parametric family for the prior lead to different inference? How sensitive is inference to different values of prior parameters within a particular parametric family of priors?

A valuable tool in addressing these kinds of questions is a *sensitivity analysis*—an explicit comparison of important characteristics of the posterior distribution obtained under all plausible prior distributions under consideration.

Table 5.1 Posterior summaries for binomial likelihood, data = 17 successes and 43 failures, under different prior specifications

Prior density	$Pr(\pi > 0.10 y)$	$E(\pi y)$	95% equal tail credible set
$U(0, 1)$	0.867	0.154	(0.070, 0.263)
$Beta(1.25, 5)$	0.842	0.147	(0.068, 0.249)
$Beta(2.5, 10)$	0.884	0.152	(0.075, 0.250)
$Beta(5, 20)$	0.937	0.160	(0.087, 0.250)
$Beta(10, 40)$	0.981	0.170	(0.103, 0.249)

Returning to the example of the quitting-school survey from Chap. 6, Table 5.1 presents a *sensitivity analysis* of the effects of the five different beta prior specifications that we entertained on posterior inference about π .

Note that whether inference is “affected substantially” by changes in assumptions is a subjective determination that depends upon the primary purpose of the analysis. Usually a statistical study will involve many different analyses of a particular dataset, but one research question will be of the greatest importance. For example, in a clinical trial comparing two treatment regimens for a particular disease, many different variables are measured and recorded on each patient, but one of them is designated as the *primary endpoint*, and a particular statistical procedure performed on this variable is the *primary analysis* of the study.

Since the main purpose of the quitting-school survey is to make a compelling argument to the Regents that a substantial proportion of students would drop out if tuition were increased, we might designate the test of the hypothesis that $\pi > 0.10$ as our primary analysis. In that case, I would conclude that the analysis was sensitive (not robust) to the different prior specifications, because I would consider a 98% chance that $\pi > 0.10$ is much more convincing than an 84% chance (and I suspect that most Regents would agree).

On the other hand, if we had determined that the primary purpose of our analysis was to obtain a point estimate and 95% credible set for π , we instead might conclude that the analysis was robust to the different prior specifications since these posterior summaries are quite similar under all five priors.

If a sensitivity analysis indicates that important aspects of inference are very sensitive to different prior specifications, the appropriate course of action for the Bayesian is to report the results obtained under all plausible prior distributions, thus enabling the consumer of the data analysis to draw his or her own conclusions. Since it may be infeasible to repeat a complex analysis with many different prior specifications (and reporting too many different results may simply confuse the reader or listener), reporting results obtained with a small number of carefully-chosen prior densities is standard practice. Often these will be the prior that the study investigators actually believe, and a noninformative prior (such as the uniform prior used for a binomial likelihood). Another choice (recommended by Spiegelhalter et al. (1993) in the context of clinical trials comparing a new treatment with standard treatment) is to compare results with an enthusiastic prior (consistent with the belief that the new treatment is clearly superior to the old treatment) to those

produced with a skeptical prior (expressing the belief that the new treatment is no better than the old one). Other approaches to assessing robustness in clinical trials are presented in [Greenhouse and Wasserman \(1995\)](#) and [Carlin and Sargent \(1996\)](#).

5.2 Inference Using Nonconjugate Priors

As discussed in Sect. 3.3.1, the choice of prior distributions for any given Bayesian model is unlimited. While conjugate priors are convenient for simple models, they may not adequately represent prior beliefs. Furthermore, for most complex real-world models, no conjugate family exists. Bayes' rule for updating from prior to posterior applies with nonconjugate priors as well. Let's examine how that works in the context of our binomial model.

5.2.1 Discrete Priors

Recall from Sect. 3.3.1 that a discrete prior distribution may be used even for an inherently continuous-valued parameter like the success probability π . For people without a background in probability (i.e., for most people including Regents!), discrete priors are easier to understand than probability densities.

In the survey problem, suppose we chose to put all prior probability on a discrete set of values such as

$$P(\pi = p) = 0.1, \quad p = 0.02, 0.06, 0.10, 0.14, 0.18, 0.22, 0.26, 0.30, 0.34, 0.38 \quad (5.1)$$

The probability distribution in (5.1) is a discrete uniform prior, because the same probability mass, 0.1, is assigned to each specified possible value of π . Any set of probabilities that matched our prior beliefs about π could be used, as long as they summed to 1.

In this case, the generic form of Bayes' rule—*posterior* \propto *prior* \times *likelihood*—takes the following form:

$$P(\pi = p|y) \propto \begin{cases} 0.1 \times p^y (1-p)^{n-y} & p = 0.02, 0.06, 0.10, 0.14, 0.18, 0.22, 0.26, 0.30, 0.34, 0.38 \\ 0 \times p^y (1-p)^{n-y} & \text{otherwise} \end{cases}$$

Obviously, regardless of the likelihood, the posterior probability will be 0 for all values of π except those given positive prior mass. More generally, a discrete prior will always give rise to a discrete posterior distribution.

Table 5.2 Computing posterior probabilities with a discrete prior

p	Prior $P(\pi = p)$	Likelihood $p^y(1 - p)^{n-y}$	Product	Posterior $P(\pi = p y)$
0.02	0.1	5.369e-13	5.369e-14	0.0001
0.06	0.1	1.957e-10	1.957e-11	0.0399
0.10	0.1	1.078e-09	1.078e-10	0.2196
0.14	0.1	1.608e-09	1.608e-10	0.3276
0.18	0.1	1.205e-09	1.205e-10	0.2455
0.22	0.1	5.715e-10	5.715e-11	0.1164
0.26	0.1	1.913e-10	1.913e-11	0.0390
0.30	0.1	4.776e-11	4.776e-12	0.0097
0.34	0.1	9.136e-12	9.136e-13	0.0019
0.38	0.1	1.353e-12	1.353e-14	0.0003

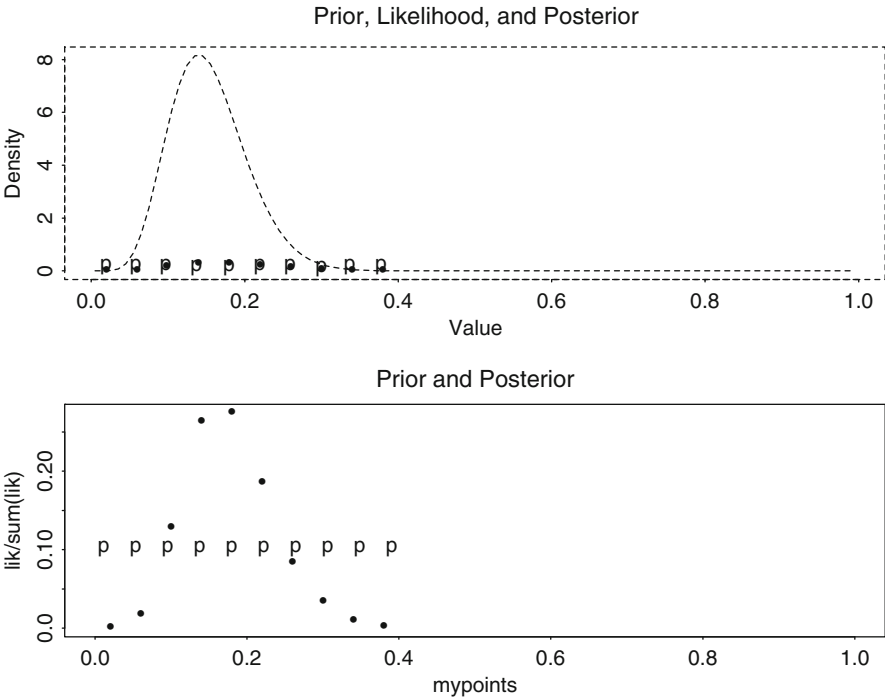


Fig. 5.1 Posterior probability mass function with discrete prior

The final step in calculating posterior probabilities is to normalize them so they sum to one. With $y = 7$ successes and $n = 50$, the calculation of the posterior probabilities for each possible value of π proceeds as in Table 5.2 [which is similar in structure to tables in Albert (1997)].

Figure 5.1 depicts this calculation graphically. In the upper panel, the p s represents the discrete prior, the smooth curve the likelihood, and the dots the posterior probabilities. The lower panel is a detail view of the bottom part of the upper graph; it makes it possible to see the prior and posterior mass points clearly.

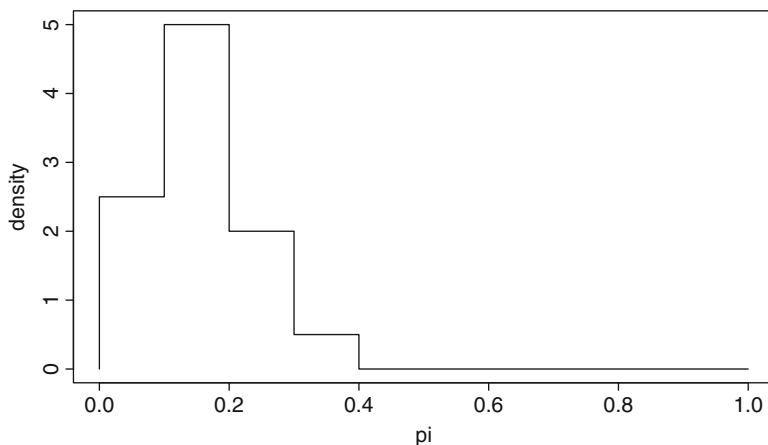


Fig. 5.2 A histogram prior

5.2.2 A Histogram Prior

In Sect. 3.3.1, we mentioned histogram priors, which, like discrete priors, may be easier for nonstatisticians to understand and specify. Other advantages of histogram priors are that they do not require any parametric assumptions and provide great flexibility in specifying prior beliefs.

To construct a histogram prior for a success probability, one begins by dividing the interval $(0,1)$ into predefined, nonoverlapping subintervals. Making the subintervals of equal length simplifies the process, but if some other partition works better for the particular problem, that's fine. The next step is to assign a probability to each interval in accordance with one's prior belief that the population proportion lies in that interval.

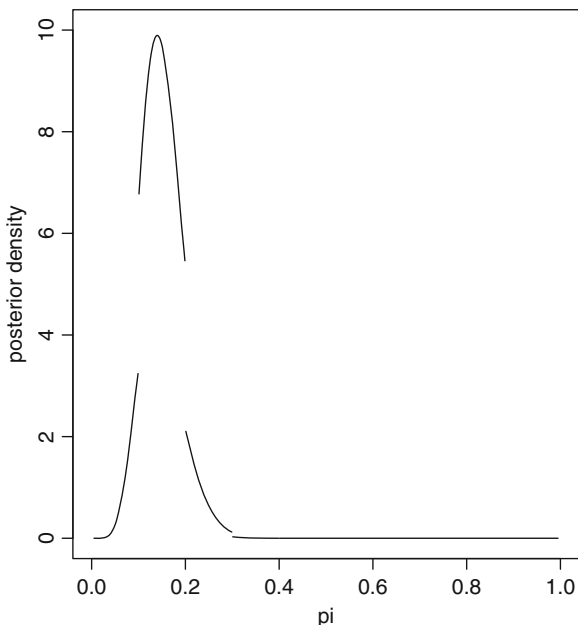
For example, someone might define a histogram prior for π in the quitting-school example as follows:

Interval	Prior probability
$(0, 0.1]$	0.25
$(0.1, 0.2]$	0.50
$(0.2, 0.3]$	0.20
$(0.3, 0.4]$	0.05
> 0.4	0.00

Figure 5.2 represents this histogram prior graphically.

As always, the posterior distribution will be calculated by multiplying these prior probabilities times the binomial likelihood and normalizing the result. Because of the discontinuities in the histogram prior, the graph of the posterior density in Fig. 5.3 also has discontinuities.

Fig. 5.3 Posterior distribution using histogram prior



5.3 Noninformative Priors

Noninformative prior distributions are useful in Bayesian analysis when we want inference to be unaffected by information apart from the current data. We have already mentioned that an analysis using a noninformative prior should be included in a sensitivity analysis so as to evaluate the influence of other priors on posterior inference.

Noninformative priors also are appropriate when we truly have little previous knowledge compared to the information contained in the new data. For example, we would not bother to carry out a difficult and expensive scientific experiment unless we thought it was going to increase our knowledge substantially. In such cases, we expect and want the likelihood to dominate the prior.

For many statistical models, there are several different choices of priors that carry little information.

5.3.1 Review of Proper and Improper Distributions

Recall that a probability density is valid only if it integrates to one over the support of the random variable. Any strictly positive function that integrates to a positive, finite number over a specified support can be *normalized* so that it integrates to one and becomes a valid density. Sometimes it is useful to treat a function as if it were

an unnormalized density even if its integral over the indicated support is not finite. When this is done, the function is referred to as an *improper* density. For example,

$$p(\sigma) = \frac{1}{\sigma}, \quad 0 < \sigma < \infty$$

In an effort to introduce the least possible amount of external information into a Bayesian analysis, improper densities sometimes are used as priors. This must be done with great care. Using a proper prior guarantees that the posterior also will be proper, but an improper prior may produce an improper posterior. If the *posterior density* is improper, it doesn't exist, so no valid inference can come out of it. Thus, if you choose to use an improper prior, you must verify that the resulting posterior is proper. This is an extremely important point, and throughout this book, we will examine how to do this verification for different kinds of models.

5.3.2 A Noninformative Prior for the Binomial Likelihood

We have already encountered one choice of noninformative prior density for the binomial likelihood, $U(0,1)$. Compared to other probability density functions, the uniform density is relatively easy for nonstatisticians to understand, and it is intuitively obvious that a uniform prior does not favor any particular range of possible values of the parameter π .

5.3.3 Jeffreys Prior

5.3.3.1 Invariance Under Transformation

A disadvantage of the uniform prior is that it is not “invariant under transformations”—that is, if we needed inferences about some transformation of π instead of π itself, performing the mathematically appropriate adjustment to the uniform prior would not produce a corresponding noninformative prior on the transformed parameter.

For example, the *logit* transformation of the binomial success probability π is used in epidemiological case-control studies, as well as in logistic regression (which we will study in Sect. 10.3). Letting ϕ denote the logit of π , the logit transformation is given as:

$$\phi = g(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$$

It is a one-to-one function, so it has an inverse:

$$\pi = g^{-1}(\phi) = \frac{\exp(\phi)}{1 + \exp(\phi)}$$

The binomial probability mass function can be written in terms of ϕ :

$$p(y|\phi) = \binom{n}{y} \left(\frac{\exp(\phi)}{1 + \exp(\phi)}\right)^y \left(\frac{1}{1 + \exp(\phi)}\right)^{n-y}$$

Such a process of writing an equivalent form of the distribution of the data after transforming the parameter(s) is called *reparameterization*. In different problems, reparameterization can have many different benefits—improving interpretability of parameters, simplifying computation of frequentist maximum likelihood estimators and confidence intervals, and improving the performance of some algorithms for sampling-based Bayesian inference.

If we reparameterize the likelihood, the prior must be adjusted appropriately. Since we are treating the parameter in the likelihood as if it were a random variable, this is a *transformation-of-variables* problem. Recall the transformation of variables procedure from your mathematical statistics class: If $y = g(x)$ is a one-to-one transformation of x , then $x = g^{-1}(y)$. Let $p_X(x)$ denote the density function of x . Then the density function of y , $p_Y(y)$ is

$$p_Y(y) = p_X(g(y)) \left| \frac{dx}{dy} \right|$$

If we transform the uniform prior on the binomial parameter π into a prior on $\phi = \text{logit}(\pi)$ (Exercise 5.3), the result is the logistic density:

$$p_\phi(\phi) = \frac{\exp(\phi)}{[1 + \exp(\phi)]^2}, \quad -\infty < \phi < \infty \quad (5.2)$$

This is a symmetric, bell-shaped density centered at 0 that resembles a t-distribution.

A systematic approach to deriving noninformative priors that are invariant to transformation was one of innumerable contributions to Bayesian statistics by Sir Harold Jeffreys (1891–1989), a British mathematician, statistician, geophysicist, and astronomer. In [Jeffreys \(1946\)](#), he proposed a procedure that has come to be known as *Jeffreys prior*. It is based on the Fisher information.

5.3.3.2 The Fisher Information

First, recall the Fisher information, a quantity used by frequentists in computing the asymptotic variance of maximum likelihood estimators. Let $p(y|\theta)$ denote the probability density function of a realization of a random variable Y given the unknown parameter θ . The British statistician Sir R.A. Fisher (1890–1962) defined the information about a parameter provided by an experiment as

$$I(\theta|y) = -E \frac{\partial^2(\log(p(y|\theta)))}{\partial \theta^2}$$

The expectation is taken over possible values of y for fixed θ . Since the information is an *expectation*, it depends on the *distribution* of Y , not on any observed value, y .

Since the log-likelihood $\log(L(\theta|y))$ differs from $\log(p(y|\theta))$ only by a constant, all their derivatives are equal. Thus, the information can equivalently be defined as

$$I(\theta|y) = -E \frac{\partial^2 \log(L(\theta|y))}{\partial \theta^2}$$

If there are n independent observations $\mathbf{y} = y_1, y_2, \dots, y_n$, then the probability densities multiply and the log-likelihoods add. Thus, the Fisher information becomes

$$I(\theta|\mathbf{y}) = -E \frac{\partial^2(\log(L(\theta|\mathbf{y})))}{\partial \theta^2} = n I(\theta|y)$$

Finally, as is shown in any mathematical statistics text,

$$I(\theta|y) = E \left(\frac{\partial \log(L(\theta|y))}{\partial \theta} \right)^2$$

What happens to the Fisher information if we transform the unknown parameter θ to $\phi = g(\theta)$? Then

$$\frac{\partial \log(L(\phi|y))}{\partial \phi} = \frac{\partial \log(L(\theta|y))}{\partial \theta} \frac{\partial \theta}{\partial \phi}$$

Squaring and taking expectations over values of y (note that $\frac{\partial \theta}{\partial \phi}$ does not depend on y), we get

$$I(\phi|y) = I(\theta|y) \left(\frac{\partial \theta}{\partial \phi} \right)^2$$

5.3.3.3 Jeffreys' Recipe

Clearly,

$$\sqrt{I(\phi|y)} = \sqrt{I(\theta|y)} \left| \frac{\partial \theta}{\partial \phi} \right|$$

That is, the square root of the Fisher information is invariant to transformations in the following sense. Suppose we specify a likelihood using a particular form of the parameter (which we'll call θ), derive the Fisher information, and take the square root, obtaining $\sqrt{I(\theta|y)}$. Then we decide we want to work with a transformation of θ , namely, ϕ . The square root of the Fisher information for ϕ can be computed from $\sqrt{I(\theta|y)}$ by multiplying by the Jacobian $\frac{\partial \theta}{\partial \phi}$. *The result will be exactly the same as if we'd started by specifying the (reparameterized) likelihood in terms of ϕ and had calculated the square root of the Fisher information for ϕ .*

To obtain this invariance property for prior densities, Jeffreys proposed using the square root of the Fisher information as a prior:

$$p(\theta) \propto \sqrt{I(\theta|y)}$$

The resulting expression is called the Jeffreys prior for the given likelihood.

The Jeffreys prior is probably the most commonly used noninformative prior in Bayesian practice. It depends on the form of the likelihood but not on the current observed data. Compared to other possible noninformative priors, it has the *invariance property*: no matter what scale we choose for measuring the unknown parameter, the same prior results when the parameter is transformed to any other scale.

However, the Jeffreys prior has disadvantages as well. For some families of likelihoods (e.g., Cauchy), the Fisher information does not exist. For multiparameter models, there is disagreement about exactly how to derive Jeffreys priors. Jeffreys prior for some likelihood families is improper, necessitating extra care in using it in Bayesian analysis (see Sect. 5.3.4).

5.3.3.4 Example: Jeffreys Prior for the Binomial Likelihood

To derive the Jeffreys prior for the binomial likelihood, we begin with the log likelihood

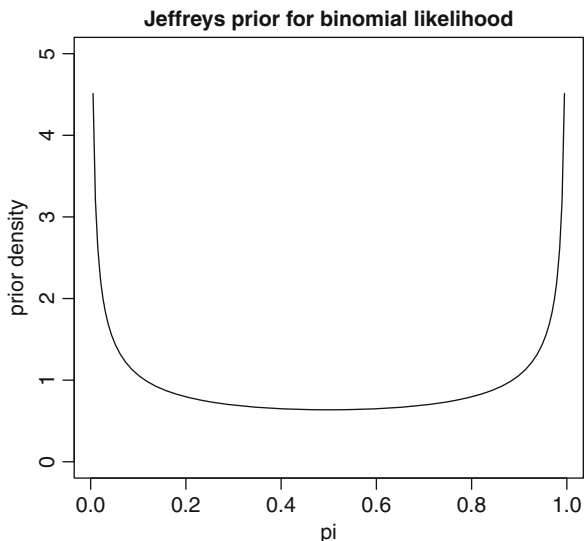
$$\log(L(\pi|y)) = y \log \pi + (n - y) \log(1 - \pi) + \text{constant}$$

The second derivative of the log likelihood is

$$\frac{\partial^2 \log(L(\pi|y))}{\partial \pi^2} = -\frac{y}{\pi^2} - \frac{n - y}{(1 - \pi)^2}$$

Taking the expectation of this expression with respect to y is easy because y appears only linearly (no functions of y like y^2 or e^y that complicate taking

Fig. 5.4 Density plot of Jeffreys prior for the binomial success probability



expectations). Thus, we can just plug in $E(y|\pi)$ for y . If $y \sim \text{Binomial}(n, \pi)$, then $E(y|\pi) = n\pi$. Thus,

$$I(\pi|y) = \frac{n}{\pi(1-\pi)}$$

Taking the square root and removing the constant n gives

$$p(\pi) \propto \pi^{-\frac{1}{2}} (1-\pi)^{-\frac{1}{2}}, \quad 0 < \pi < 1$$

We recognize this density as $\text{Beta}(\frac{1}{2}, \frac{1}{2})$. Thus, in the case of the binomial likelihood, the Jeffreys prior is a member of the conjugate family. As shown in Fig. 5.4, the plot of this density is bathtub-shaped. Although the visual impression may be that this prior contains more information than the uniform prior, in fact the variance is larger ($\frac{1}{8}$ versus $\frac{1}{12}$) and the equivalent prior sample size smaller (1 versus 2, 0 versus 1, or even -1 versus 0, depending on which definition of equivalent prior sample size for the beta distributions you use) for the Jeffreys prior than the uniform prior. Thus, Jeffreys prior for π is even less informative than the uniform prior.

5.3.4 Verifying the Propriety of the Posterior Distribution When Using an Improper Prior

So far, we have discussed two choices of noninformative prior densities for the binomial likelihood: $\text{Uniform}(0,1)$ and $\text{Beta}(\frac{1}{2}, \frac{1}{2})$.

A third version is sometimes used:

$$p(\pi) \propto \pi^{-1}(1 - \pi)^{-1}, \quad 0 < \pi < 1$$

This looks like the kernel of a beta density, except it would be $Beta(0, 0)$. This is an improper density, because both parameters must be strictly positive in order for a Beta density to be proper. This prior may be appealing because it yields the mle, $\frac{y}{n}$, as the posterior mean. However, since it is improper, there is no guarantee that the posterior density is proper, and if the posterior is improper, none of its characteristics (including the posterior mean) exist! Thus, if you choose to use this prior, you must make sure that your data have the necessary properties to produce a proper posterior density. Recall that for a binomial likelihood and a beta prior with parameters α and β , the posterior density $p(\pi|y)$ is $Beta(\alpha + y, \beta + n - y)$. If both α and β are 0, then $p(\pi|y)$ is $beta(y, n - y)$. In order for this to be a proper beta density, both y and $n - y$ must be strictly positive; that is, there must be at least one success and one failure in the data.

Problems

5.1. This is a continuation of Problem 4.1 in Chap. 4. Perform two additional Bayesian analyses of the softball player data, one with Jeffreys prior and one with the improper $Beta(0, 0)$ prior. Obtain the same posterior summaries as in Problem 4.1. Verify that the posterior density obtained with the improper prior is proper.

5.2. Do a sensitivity analysis to assess the sensitivity of your inference in Problems 4.1 and 5.1 to the five prior densities. Comment on whether or not your inference is robust.

5.3. Verify analytically that a uniform prior on π induces the density in (5.2) as the prior on $\phi = \text{logit}(\pi)$.

5.4. Suppose that a trucking company owns a large fleet of well-maintained trucks and assume that breakdowns appear to occur at random times. The president of the company is interested in learning about the daily rate λ at which breakdowns occur. (Realistically, each truck would have a breakdown rate that depends possibly on its type, age, condition, driver, usage, etc. The breakdown rate for the whole company can be viewed as the sum of the breakdown rates of the individual trucks.) For a given value of the rate parameter λ , it is known that the number of breakdowns y on a particular day has a Poisson distribution with mean λ :

$$p(y|\lambda) = \frac{e^{-\lambda} \lambda^y}{y!}, \quad y = 0, 1, 2, \dots$$

1. Suppose that one observes the number of truck breakdowns for n consecutive days—denote these numbers by y_1, \dots, y_n . If one assumes that these are

exchangeable measurements (conditionally independent given λ), find the joint probability distribution of y_1, \dots, y_n .

2. The numbers of breakdowns for 5 days are recorded to be 2, 5, 1, 0, and 3. Find the likelihood function $L(\lambda)$ of the rate parameter λ for these observations. Graph this function. (You may either use R or do it “by hand” by calculating the likelihood for the values $R = 0.1, 0.5, 1, 2, 4, 8$, and 16 and connecting the points with a smooth curve.)
3. Use calculus to find the mle of λ . Then use the `poisson.test` function in R to confirm the mle and to obtain a 95% frequentist confidence interval for λ .

5.5. The president of the company has some knowledge about the location of the Poisson rate parameter λ based on the observed number of breakdowns from previous years. His prior beliefs about λ are represented in the following:

$$p(\lambda) \propto \lambda^3 \exp(-2\lambda), \quad \lambda > 0$$

1. Is this prior a member of a particular parametric family? If so, what family and what are the prior parameters?
2. Plot this prior density, either using software or by picking a few values at which to evaluate it as in the previous problem. Based on the plot, describe the president’s prior beliefs about the rate parameter λ .
3. Write out the mathematical form of the unnormalized posterior density. Identify its parametric family and parameters.
4. Find the posterior mean and 95% central credible set for λ based on this posterior.
5. Was the president’s prior from a conjugate family for the Poisson likelihood? How could you tell?

5.6. A new employee of the trucking firm wishes to learn about the breakdown rate. She does not have the previous information available to the president, so she wishes to assign a noninformative prior density.

1. Derive the Jeffreys prior that goes with the Poisson likelihood.
2. Compute the resulting posterior distribution using this prior and the data given earlier for the 5 days.
3. Find the posterior mean and 95% central credible set for this posterior.
4. Compare the Bayesian point and interval estimates with the classical estimates found in the first exercise.
5. Compare the new employee’s estimates with the president’s estimates.
6. Suppose the employee has a personal rule that she would not drive for a company whose fleet of trucks had a daily breakdown rate > 2 . Based on her analysis with the Jeffreys prior, what is her posterior probability that $\lambda > 2$ for this company?
7. Contrast this with the president’s posterior probability that $\lambda > 2$.

5.7. Show that if a $Beta(0,0)$ prior is used with a binomial likelihood, the posterior mean $E(\pi|y)$ is equal to the frequentist mle, $\frac{y}{n}$.

Chapter 6

Other One-Parameter Models and Their Conjugate Priors

6.1 Poisson

You encountered the Poisson distribution in problems at the end of the previous chapter. The Poisson distribution is useful when the random variable is a count of the number of rare events occurring per unit time, unit volume, unit distance, etc. For example, the number of new cases of rhabdomyosarcoma (a rare form of cancer) occurring in Johnson County, Iowa, each year might be represented as a Poisson random variable. So might the number of flaws in each 1,000 feet of yarn produced by a spinning machine. A Poisson random variable can take on only nonnegative integer values.

In order for the Poisson distribution to be appropriate, there is a constant average rate at which the events occur, and the numbers of events in disjoint intervals (different years, different segments of yarn, etc.) must be independent. In our examples, this implies that if there were an unusually large number of new rhabdomyosarcoma cases in Johnson County in one particular year, that would not affect the probability distribution for the number of new cases in the following year. Thus, the Poisson distribution would not be appropriate for counts of a contagious disease.

6.2 Normal: Unknown Mean, Variance Assumed Known

So far, we have been considering discrete data—binary responses to survey questions and integer counts of rare events. Thus, the distributions of the random variables of interest, and the resulting likelihoods, have been probability mass functions. Now we will begin to consider cases in which the data are realizations of *continuous* random variables, which are described by probability density functions (pdfs).

As you know, there are many parametric families of continuous pdfs. Both frequentists and Bayesians must use care in choosing the density that is likely to best describe the population of values from which their sample data is going to be drawn.

The normal (also called the Gaussian) density is one of the most commonly used pdfs, and I am sure you are familiar with its bell-shaped density curve. The normal density is a good model for data when the random variable is continuous-valued, the distribution of values in the population is likely to be symmetric around a single mode, and the tails of the distribution are not heavy. It is a good choice for many variables that are measurements on living things, like weights, body temperatures, or heart rates of a species of mammals. The normal density is not appropriate for variables for which the population distribution is likely to be skewed, such as household incomes.

You should be familiar with the normal probability density function, here shown for a random variable Y from a normal density with mean μ and population variance σ^2 :

$$Y \sim N(\mu, \sigma^2)$$

$$p(y|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right), \quad -\infty < y < \infty$$

6.2.1 Example: Mercury Concentration in the Tissue of Edible Fish

You probably are aware that the US Food and Drug Administration has recommended that pregnant women, nursing mothers, and young children avoid eating certain types of fish and limit their consumption of others. In addition, state and local governments sometimes issue advisories to limit consumption of fish caught in particular local rivers or lakes. Both of these kinds of advisories occur because some fish and shellfish contain high levels of mercury, which can harm the developing nervous systems of fetuses, infants, and young children.

Mercury is a chemical element that occurs naturally in the environment and can also be emitted into the air by certain industrial processes. Rain washes mercury out of the air and deposits it on the ground, from which it can run off into lakes and rivers. Bacteria in the water convert elemental mercury into methyl mercury (meHG), the form of mercury that has neurotoxic effects. The bacteria are eaten by plankton, which are eaten by small fish, which in turn are eaten by larger fish. With each ascending level of the food chain, the concentration of meHG increases, so that the concentration of meHG in the tissues of large fish may be thousands of times as high as the concentration of mercury in the water. The human body can eliminate meHG only slowly. If a woman ingests meHG from fish at a higher rate than her body can eliminate it, the level of meHG builds up in her tissues. Methyl mercury

can cross the blood–brain barrier. Thus, high levels of mercury in the tissue of a pregnant woman or nursing mother can be transmitted to her infant’s brain through her blood or breast milk.

In this book, we will use Bayesian methods to examine the issue of mercury in fish from two perspectives. In the present chapter, we will study the levels of mercury found in samples of fish tissue, and in a future chapter, we will apply Bayesian modeling to investigate mercury deposition from rainfall in the continental United States.

Our first example dataset includes mercury concentrations in parts per million (ppm) measured on 21 tissue samples from common carp caught at a particular site on the Des Moines River in Madison County, Iowa. The data are taken from a database of over 100,000 fish tissue mercury records collected by the Environmental Mercury Mapping, Modeling, and Analysis (EMMMA) project of the US Geological Survey (<http://emmma.usgs.gov/datasets.aspx>). Since data on concentrations of chemicals often are right skewed, the log transformation frequently is used to symmetrize their distributions. We will follow that practice with this dataset.

According to the Natural Resources Defense Council (<http://www.nrdc.org/health/effects/mercury/guide.asp>) the concentrations of mercury in fish tissue fall into the following categories:

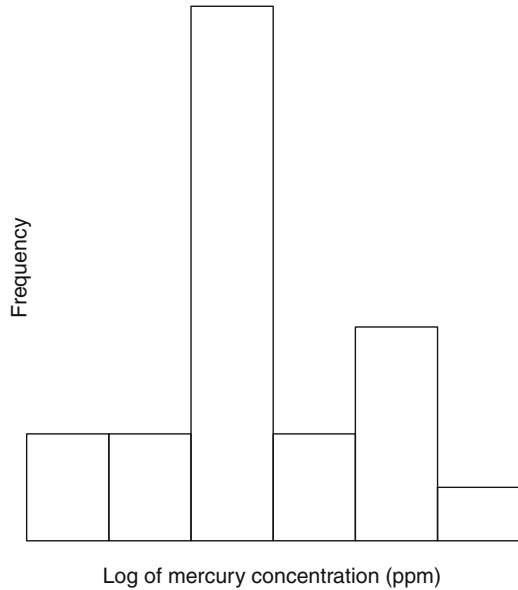
- Least mercury: Less than 0.09 parts per million (-2.41 on the log scale)
- Moderate mercury: From 0.09 to 0.29 parts per million (-2.41 to -1.24 on the log sale)
- High mercury: From 0.29 to 0.49 parts per million (-1.24 to -0.71 on the log scale)
- Highest mercury: More than 0.49 parts per million (more than -0.71 on the log scale)

We wish to estimate the mean μ of log-transformed mercury concentration in the tissue of the population of all fish caught in the Des Moines River at the location represented by our data. In addition, we wish to estimate the probability that μ falls into each of these four categories.

6.2.2 *Parametric Family for Likelihood*

Since we plan to use the normal density as the distribution of the observed data, we should check whether the sample values look like draws from a normal population. However, we don’t want to actually look at the numeric values of the data until after we have specified the prior because we don’t want to let the current data influence our prior in any way. Figure 6.1 is a histogram of the log-transformed concentrations—without showing the actual range of values covered by the data. Although the histogram is not perfectly symmetrical and bell shaped, for a sample of only 21 observations, it is about as close as real data gets.

Fig. 6.1 Histogram of log mercury concentrations in fish tissue from Des Moines River



When, as here, there are more than one observation in the dataset, we must begin constructing our Bayesian model by specifying the joint distribution of all the data. The issue of *exchangeability*, which we have met in previous chapters, arises again. Are we comfortable with the assumption that the observations in the dataset are random draws from the same normal distribution? If we don't have information that would lead us to expect some observations, or groups of observations, to be systematically different from others, then the assumption of exchangeability is reasonable.

If we consider observations in a sample exchangeable, we typically specify their joint distribution by treating the observations as conditionally independent given one or more shared parameters. That means that we can write the joint density simply as the product of the densities of the individual observed values:

$$\begin{aligned}
 p(y_1, y_2, \dots, y_n | \mu, \sigma^2) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right) \\
 &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{\sum_{i=1}^n (y_i - \mu)^2}{2\sigma^2}\right) \\
 &= \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{2\sigma^2}\right) \exp\left(-\frac{n(\bar{y} - \mu)^2}{2\sigma^2}\right)
 \end{aligned}
 \tag{6.1}$$

Exchangeability is sometimes defined as “invariance to permutations of the indices.” It is easy to see what this means in this case. The indices are the subscripts on the y ’s. If for some reason we decided to swap y_2 with y_9 , the product in (6.1) would be unchanged because the same μ and σ^2 are involved in the terms for all they’s.

In Problem 6.1, you will show how the last line in (6.1) was obtained.

6.2.3 Likelihood for μ Assuming that Population Variance Is Known

We will perform our first analysis of the mercury concentration data under an unrealistic assumption: that the exact numeric value of the population variance σ^2 is known. Of course this is impossible. We could not know the exact value of σ^2 unless we had measured every fish that had ever swum the Des Moines River in Madison County, and if we had done that, we would also know the exact value of μ and would not need to use a sample to draw inference! However, studying Bayesian analysis with a normal likelihood as if it were a one-parameter problem (with only μ unknown) is a worthwhile learning experience, because such models will form the building blocks of more complex and realistic models and of the computational methods for fitting them.

If σ^2 is assumed to be a known constant, then (6.1) may be viewed as a likelihood for the only unknown parameter, μ :

$$L(\mu|\mathbf{y}) \propto \exp\left(-\frac{n(\bar{y} - \mu)^2}{2\sigma^2}\right) \quad (6.2)$$

That is, the expression in (6.2) is viewed as a function of μ , for a fixed (observed) value of \bar{y} and a fixed (unrealistically assumed known) value of σ^2 . The other terms in (6.1) do not contain μ , so with respect to a likelihood for μ , they are just constants and can be dropped.

6.2.4 Sufficient Statistics

Note that \bar{y} appears in the likelihood instead of all the individual values y_i from each observation. Recall that a *statistic* is a number that can be calculated from sample data just by arithmetic. We do not need to know the values of any unknown parameters to calculate a statistic. \bar{y} is a statistic. When, as in this case, a statistic contains all the information in the data that is useful in estimating the unknown parameter of interest, the statistic is called a *sufficient statistic*. Using sufficient statistics when they exist makes Bayesian computation much easier.

6.2.5 Finding a Conjugate Prior for μ

Just as was the case when we were dealing with a binomial likelihood and the unknown parameter was the success probability π , there are an infinite number of ways of specifying a prior for the unknown mean μ of a normal distribution. However, a conjugate prior simplifies posterior calculations, so we will identify the parametric family that is conjugate to the likelihood for a normal mean and see whether there is a member of the conjugate family that adequately expresses our prior information.

In seeking a family of densities that is conjugate to the normal likelihood, we are looking for a density in which the random variable appears in the same functional form as μ appears in the normal likelihood. Note that in (6.2), we can reverse the positions of μ and \bar{y} without changing the value of the expression at all:

$$\exp\left(-\frac{n(\bar{y}-\mu)^2}{2\sigma^2}\right) = \exp\left(-\frac{n(\mu-\bar{y})^2}{2\sigma^2}\right) \quad (6.3)$$

So we are looking for a density in which the random variable appears in the same form as μ appears on the right hand side of (6.3). But the right hand side of (6.3) is the kernel of a normal density! So the normal density is the conjugate prior for μ in the normal likelihood when σ^2 is assumed known.

If we express the likelihood in terms of the sufficient statistic \bar{y} , then we can write the Bayesian normal model as

$$\begin{aligned} \bar{y} \mid \mu, \sigma^2 &\sim N\left(\mu, \frac{\sigma^2}{n}\right) \\ \mu &\sim N(\mu_0, \sigma_0^2) \end{aligned}$$

In Sect. 6.2.7 we will discuss how to select numeric values for the prior parameters μ_0 and σ_0^2 to express our prior knowledge about μ .

Recall that the important implication of conjugacy is that the posterior density will be in the same parametric family as the prior. Thus, in the normal prior/normal likelihood case, the posterior density $p(\mu|\bar{y})$ will also be normal.

6.2.6 Updating from Prior to Posterior in the Normal–Normal Case

I have emphasized that the same density can be parameterized in different ways. It turns out that in Bayesian statistics, writing the normal density in terms of the mean and *precision* instead of the mean and variance simplifies calculation of the posterior density for μ .

6.2.6.1 Precisions

The precision is the inverse of the variance. The more spread out a distribution is (larger variance), the less precise it is (smaller precision). When reading Bayesian literature or using Bayesian software, you must always make sure whether normal distributions are parameterized in terms of the variance or the precision.

Rewriting our likelihood and prior using precisions yields:

$$\begin{aligned}\bar{y} \mid \mu, \tau^2 &\sim N\left(\mu, \frac{n}{\tau^2}\right) \\ \mu &\sim N\left(\mu_0, \frac{1}{\tau_0^2}\right)\end{aligned}$$

where $\tau^2 = \frac{1}{\sigma^2}$ and $\tau_0^2 = \frac{1}{\sigma_0^2}$.

6.2.6.2 The Posterior Density

Bayes' rule applies here in the usual way: the posterior density is proportional to the prior times the likelihood. Thus,

$$\begin{aligned}p(\mu \mid \mathbf{y}) &\propto \frac{\sqrt{n\tau}}{\sqrt{2\pi}} \exp\left(-\frac{n\tau^2(\mu - \bar{y})^2}{2}\right) \frac{\tau_0}{\sqrt{2\pi}} \exp\left(-\frac{\tau_0^2(\mu - \mu_0)^2}{2}\right) \\ &\propto \exp\left(-\frac{n\tau^2(\mu - \bar{y})^2}{2} - \frac{\tau_0^2(\mu - \mu_0)^2}{2}\right) \\ &\propto \exp\left(-\frac{(n\tau^2 + \tau_0^2)\mu^2 + 2\mu(n\tau^2\bar{y} + \tau_0^2\mu_0)}{2}\right) \\ &= \exp\left(-\frac{(n\tau^2 + \tau_0^2)(\mu^2 - 2\mu\frac{n\tau^2\bar{y} + \tau_0^2\mu_0}{(n\tau^2 + \tau_0^2)})}{2}\right) \\ &\propto \exp\left(-\frac{(n\tau^2 + \tau_0^2)(\mu - \frac{n\tau^2\bar{y} + \tau_0^2\mu_0}{(n\tau^2 + \tau_0^2)})^2}{2}\right)\end{aligned}\tag{6.4}$$

The last line in (6.4) is the kernel of a normal density:

$$\mu \mid \mathbf{y} \sim N\left(\frac{n\tau^2\bar{y} + \tau_0^2\mu_0}{n\tau^2 + \tau_0^2}, \frac{1}{n\tau^2 + \tau_0^2}\right)$$

The posterior mean is a weighted average of \bar{y} and the prior mean. The weights are proportional to the respective *precisions*, $n\tau^2$ and τ_0^2 .

Equivalent Prior Sample Size

Determining the equivalent prior sample size is easy in this simplified normal model with the data precision τ^2 assumed known. Since n real data observations have weight proportional to $n\tau^2$, we may think of the prior precision as

$$\tau_0^2 = n_0 \tau^2$$

Thus, the prior contains the same amount of information as $n_0 = \frac{\tau_0^2}{\tau^2}$ observations.

Posterior Precision and Posterior Variance

Similarly, the posterior precision is the sum of the precisions from the prior and the likelihood. This makes intuitive sense. A density with a larger precision reflects more information (less uncertainty) about the random variable. The posterior density combines the information from both the prior and the current data, so it contains more information than either of them taken separately. Thus, the posterior precision should be larger than either the prior precision or the precision of \bar{y} .

The posterior variance of μ is the inverse of the precision:

$$\text{Var}(\mu|y) = \frac{1}{n\tau^2 + \tau_0^2} = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2} \quad (6.5)$$

6.2.7 Specifying Prior Parameters

We saw in Sect. 3.5.2 that there were many strategies for picking the parameter values for a beta prior to go with a binomial likelihood. Similar approaches work for specifying the parameters of a normal prior for a normal mean. Often we will have some degree of knowledge about where the normal population is centered, so choosing the mean of the prior distribution for μ usually is less difficult than picking the prior variance (or precision). Workable strategies include:

1. Graph normal densities with different variances until you find one that matches your prior information
2. Identify an interval which you believe has 95% probability of trapping the true value of μ , and find the normal density that produces it
3. Quantify your degree of certainty about the value of μ in terms of equivalent prior sample size

6.2.8 Mercury in Fish Tissue

6.2.8.1 Specifying the Prior Parameters

To complete our Bayesian model, we need to choose numeric values for the parameters μ_0 and τ_0^2 of the normal prior on μ . Recall that our data are log-transformed concentrations of mercury in fish tissue and that the units of the untransformed observations were parts per million. I am not an expert on mercury contamination of fish in Iowa rivers, but I do have a source of information.

First, the Iowa Department of Natural Resources web page says that fish caught in Iowa are generally safe to eat and that only occasionally are advisories issued due to mercury levels. Therefore, I expect that fish caught in the Des Moines River will fall into the lowest category of mercury concentrations given in Sect. 6.2.1. On the log scale, the upper bound of that category is -2.41 . Based on these facts, my best guess for the population mean of log-transformed mercury concentrations is -2.45 , so I will use that for the prior mean μ_0 .

Now we need the prior precision, τ_0^2 . Recall that we are assuming that we magically know the exact value of the population variance (and therefore of the population precision) of log-transformed mercury concentrations in all fish from the Des Moines River. Suppose we know that $\tau^2 = 2.5$. Now I don't have a very strong belief in my choice of -2.45 for the prior mean; I am only as confident as I would be if I had seen three previous tissue samples from the Des Moines River. Thus, my equivalent prior sample size is $n_0 = 3$. I should set $\tau_0^2 = n_0 \tau^2 = 3(2.5) = 7.5$. Thus, my prior is

$$\mu \sim N(-2.45, 1/7.5)$$

where 7.5 is the prior *precision*. I will verify that this setting matches my prior knowledge by looking at the 95% prior interval for μ produced by this specification. Here are R code and output:

```
> qnorm( c(0.025, 0.975), -2.45, sqrt( 1/7.5) )
[1] -3.165678 -1.734322
```

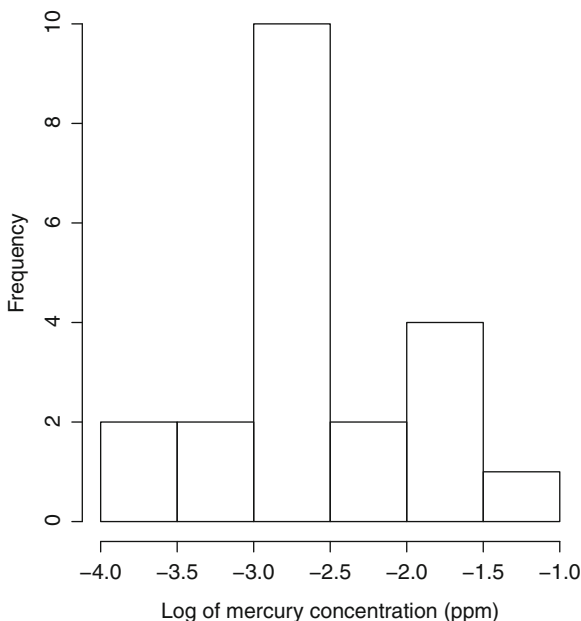
Since it is easier to think about concentrations than about log-concentrations, I'll transform the interval endpoints back to the concentration scale:

```
> exp(qnorm( c(0.025, 0.975), -2.45, sqrt( 1/7.5) ))
[1] 0.04218554 0.17651978
```

Hmm, those endpoints seem too high. I'll try again with a Normal $(-2.75, 1/7.5)$ density.

```
> exp(qnorm( c(0.025, 0.975), -2.75, sqrt( 1/7.5) ))
[1] 0.03125182 0.13076907
```

Fig. 6.2 Histogram of log mercury concentrations in fish tissue from Des Moines River



The statement that there is 95% probability that $\exp(\mu)$ is in the interval (0.031, 0.131) is consistent with my prior belief, so I will use the Normal $(-2.75, 1/7.5)$ density as my prior for μ .

6.2.8.2 Computing the Posterior Density

Now that we have completed our prior specification, we finally can look at the data. Figure 6.2 is the histogram with x-axis labels included:

Furthermore, the sample mean $\bar{y} = -2.563$ and the sample size $n = 21$.

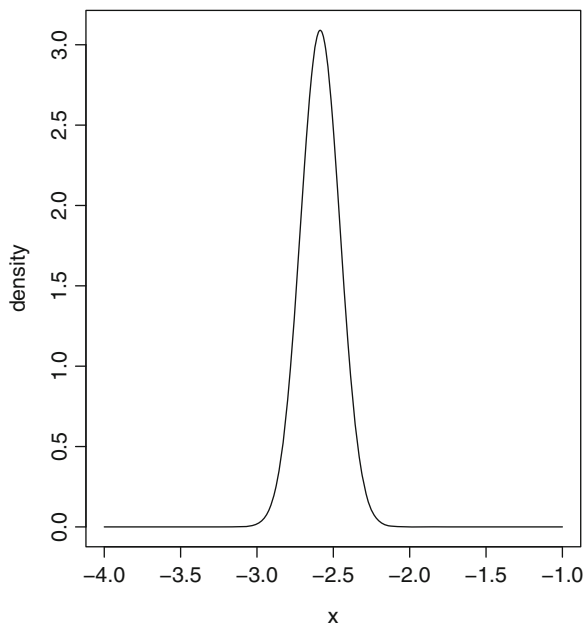
Therefore, the posterior precision is $21(2.5) + 7.5 = 60$, and the posterior mean is $\frac{21(2.5)(-2.563) + 7.5(-2.75)}{60} = -2.586$. That is,

$$\mu|\bar{y} \sim N(-2.586, 1/60)$$

This density is shown in Fig. 6.3.

Note that $p(\mu|\bar{y})$ and $p(\mu|\mathbf{y})$ are equivalent and both notations are correct, since the sufficient statistic \bar{y} contains all the information regarding μ from the entire vector of data values \mathbf{y} .

Fig. 6.3 Posterior density for μ



6.2.8.3 Using the Posterior Density to Perform Inference

As in all Bayesian analyses, the posterior density contains all of our current information about the parameter of interest and will provide the basis for all inference. We stated in Sect. 6.2.1 that we wanted to estimate the population mean μ and to determine the probabilities that μ lies in each of the four mercury-contamination categories specified by the Natural Resources Defense Council. We have already calculated a Bayesian point estimate of μ , the posterior mean:

$$E(\mu|\bar{y}) = -2.586$$

We can complete the estimation procedure by finding a 95% equal-tail posterior credible set for μ :

```
> qnorm( c(0.025, 0.975), -2.586, sqrt(1/60) )
[1] -2.839030 -2.332970
```

Thus, a person who agreed with my prior, after seeing the current data, would believe that there is 95% probability that μ is in the interval $(-2.839, -2.333)$.

We can exponentiate the endpoints of the interval to obtain the corresponding 95% interval on the original (not log-transformed) scale:

```
> exp(qnorm( c(0.025, 0.975), -2.586, sqrt(1/60) ))
[1] 0.05848235 0.09700723
```

Note that when we use a monotonic transformation such as the log transformation, we can obtain *quantiles* on the original scale by applying the reverse transformation to quantiles obtained on the transformed scale. We *cannot* do the same thing with means. A well-known inequality in mathematics called Jensen's inequality (Jensen 1906), applied to the special case of the log function, states

$$\log(E(Y)) \neq E(\log(Y))$$

The `pnorm` function in R can help us find the posterior probabilities that μ lies in each of the categories defined by the NRDC.

```
> pnorm( c(-2.41, -1.24, -0.71), -2.586, sqrt(1/60) )
[1] 0.9136045 1.0000000 1.0000000
```

Thus, $Pr(\mu < -2.41|\bar{y}) = 0.914$ and $Pr(-2.41 < \mu < -1.24)$ is about 0.086. The posterior probability that μ is in either of the highest two categories of log concentration is close to zero—good news for consumers of fish caught in the Des Moines River!

6.2.9 The Jeffreys Prior for the Normal Mean

It can be shown that the Jeffreys prior for the normal mean when σ^2 is assumed known is

$$p(\mu) \propto 1, \quad -\infty < \mu < \infty$$

This obviously is an improper prior, since the integral $\int_{-\infty}^{\infty} 1 d\mu$ is not finite. It can be thought of as the limiting case of an $N(\mu_0, \sigma_0^2)$ density as the prior variance σ_0^2 goes to ∞ . Equivalently, it is the limit of an $N(\mu_0, \tau_0^2)$ density as the prior precision τ_0^2 goes to 0. As both interpretations make clear, it contains no prior information at all. To say the same thing a third way, since the prior precision $\tau_0^2 = 0$, the equivalent prior sample size n_0 must also be zero: this prior contains the same amount of information as a data sample with 0 observations!

When the Jeffreys prior is combined with a normal likelihood, in the posterior density calculations in (6.4), all the terms that are multiples of τ_0^2 drop out, leaving the posterior density (expressed in terms of its precision):

$$\mu|\bar{y} \sim N\left(\bar{y}, \frac{n}{\tau^2}\right) \quad (6.6)$$

That is, the posterior looks just like the likelihood, with the roles of μ and \bar{y} reversed. In this case, the Bayesian posterior mean will equal the frequentist maximum likelihood estimate, and the endpoints of frequentist confidence intervals for each confidence level will be the same as those of Bayesian credible sets with the same posterior probability level. Of course, the interpretations of the two types of intervals will be different.

If we had used a Jeffreys prior in our analysis of the mercury concentration data, the resulting posterior density would have been

$$\mu|\bar{y} = -2.563 \sim N\left(-2.563, \frac{1}{52.5}\right)$$

Note that the posterior precision is smaller this time than when we used an informative prior. As expected, the posterior credible sets (on the log scale and original scale, respectively) also will be wider:

```
> qnorm( c(0.025, 0.975), -2.563, sqrt(1/52.5) )
[1] -2.833501 -2.292499
> exp(qnorm( c(0.025, 0.975), -2.563, sqrt(1/52.5) ))
[1] 0.05880663 0.10101369
```

In addition, the posterior probabilities that μ lies in each of the four intervals would have been 0.866, 0.134, 0, and 0, respectively.

6.2.10 *Posterior Predictive Density in the Normal–Normal Model*

Suppose we wish to predict concentrations of mercury in future samples of tissue from fish caught in the Des Moines River. We encountered the concept of a posterior predictive distribution in the context of our survey data example when we wished to predict the number of “yesses” in a future survey sample. In that case, in which the data were realizations of a binomial random variable, the posterior predictive distribution was discrete and provided a probability for each of the possible numbers of successes in the future sample.

In our current problem, the data are realizations of a continuous-valued random variable, so the posterior predictive distribution will be a density rather than a set of probabilities. However, the same logic is used to obtain it. Since we are assuming that the population of log-transformed mercury concentration values is normal with variance known to be 2.5 log ppm and if μ were known, the density of any possible future value would be

$$p(y_{new}|\mathbf{y}, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \quad (6.7)$$

Since μ is not known exactly, and all of our knowledge about it is contained in the posterior density $p(\mu|\mathbf{y})$, we must integrate (6.7) over the posterior density to obtain the posterior predictive density:

$$p(y_{new}|\mathbf{y}) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(\mu-\mu_1)^2}{2\sigma_1^2}\right) d\mu \quad (6.8)$$

where μ_1 and σ_1^2 , respectively, denote the posterior mean and variance of μ as given in (6.4) and (6.5). If we do the integral, we find that the posterior predictive distribution is normal, with mean equal to the posterior mean of μ and variance equal to the sum of the posterior variance of μ and the (supposedly known) population variance σ^2 , that is,

$$y_{new}|\mathbf{y} \sim N(\mu_1, \sigma_1^2 + \sigma^2) \quad (6.9)$$

This conclusion also makes intuitive sense. The larger the variance in a distribution, the more uncertainty it reflects. Our uncertainty about a future individual value of y includes all the uncertainty we have about the value of the population mean μ and all the variability between individual members of the population.

We can use (6.9) to obtain the posterior predictive density of the log mercury concentration in future tissue samples of fish from the Des Moines River. Let's continue the analysis with the Jeffreys prior, begun in Sect. 6.2.9. Rewriting the posterior density in terms of its variance, we have

$$\mu|\bar{y} \sim N\left(-2.563, \frac{1}{52.5} = 0.019\right)$$

Furthermore, in the population of individual tissue samples, the variance of log mercury concentration is assumed known to be $\frac{1}{2.5} = 0.4$.

Therefore, the posterior predictive density for a future measurement y_{new} is

$$y_{new}|\mathbf{y} \sim N(-2.563, 0.419)$$

The `qnorm` function in R will give us a 95% posterior predictive interval for a new observation:

```
> qnorm( c(0.025, 0.975), -2.563, sqrt(0.419) )
[1] -3.831689 -1.294311
```

Thus, based on a Bayesian analysis using the noninformative Jeffreys prior, we would say that there is 95% probability that the log concentration in a future individual tissue sample will lie between -3.83 and -1.29 . Again, we can exponentiate these interval endpoints to get the interval on the original (not log-transformed) scale:

```
> exp(qnorm( c(0.025, 0.975), -2.563, sqrt(0.419) ))
[1] 0.02167298 0.27408659
```

6.3 Normal: Unknown Variance, Mean Assumed Known

Usually when we use a normal likelihood in either a frequentist or a Bayesian analysis, the unknown parameter of primary interest is the population mean μ . However, sometimes interest centers instead on the spread of the population

distribution—that is, its variance σ^2 . Quality control in industry is a real-world setting in which assessing variance is crucial. For example, if a manufacturing plant produces bullets for a particular caliber of gun, not only the mean diameter of the bullets must be correct but also the variance in bullet diameters must be sufficiently small in order for all bullets to fit and fire correctly.

As we did with the normal mean, we first will study Bayesian inference for the normal variance under the unrealistic assumption that the other parameter (in this case μ) is a known number. This “single-parameter” model, too, is an important building block of realistic models that we will encounter later.

Recall the joint distribution of n observations modeled as conditionally independent draws from a normal population with known mean μ and unknown variance σ^2 :

$$\begin{aligned} p(y_1, \dots, y_n | \mu, \sigma^2) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(y_i - \mu)^2}{2\sigma^2} \right] \\ &\propto \frac{1}{(\sigma^2)^{\frac{n}{2}}} \exp \left[-\frac{\sum_{i=1}^n (y_i - \mu)^2}{2\sigma^2} \right] \end{aligned}$$

The *sufficient statistic* for σ^2 is the single number $\frac{\sum (y_i - \mu)^2}{n}$. We can rewrite the joint distribution using this sufficient statistic, represented by the symbol v , this way:

$$p(y | \sigma^2) \propto \frac{1}{(\sigma^2)^{\frac{n}{2}}} \exp \left[-\frac{nv}{2\sigma^2} \right]$$

Again, our next step in inference is to change our perspective to one in which the data have been observed, so the sufficient statistic has a fixed, known value, and we wish to evaluate the expression as a function of changing values of the unknown parameter σ^2 . The likelihood of σ^2 is

$$L(\sigma^2; \mathbf{y}) \propto \frac{1}{(\sigma^2)^{\frac{n}{2}}} \exp \left[-\frac{nv}{2\sigma^2} \right], \quad 0 < \sigma^2 < \infty \quad (6.10)$$

6.3.1 Conjugate Prior for the Normal Variance, μ Assumed Known

In preparation for a conjugate Bayesian analysis, we must identify the parametric family of prior densities for σ^2 . We need a density for a random variable with support on the positive real line, in which the random variable appears in the same functional form as in (6.10)—in a denominator raised to a power and again in the denominator of an exponent. Before reading any further, see whether you can find such a family in Tables A.1, A.2, or A.3

The conjugate family is inverse gamma. A conjugate prior for σ^2 would be

$$p(\sigma^2) = \frac{\beta^\alpha}{\Gamma\alpha} \frac{1}{(\sigma^2)^{\alpha+1}} \exp\left(-\frac{\beta}{\sigma^2}\right), \quad 0 < \sigma^2 < \infty \quad (6.11)$$

There are several strategies for specifying the parameters of an inverse gamma prior density to express our knowledge or opinion about a normal variance. We could:

1. Plot inverse gamma densities with different parameter values until we found one that matched our prior knowledge.
2. Decide on appropriate numeric values for the mean and variance of the prior distribution for σ^2 and solve the expressions for mean of an inverse gamma density and variance of an inverse gamma density for α and β (See table of distributions in the appendix).
3. Use an R function to find values of α and β that produce a prior probability interval that matches our prior knowledge.

Strategy 3 is a little trickier with inverse gamma densities than with the other densities that we have studied so far, because R does not have a built-in set of functions for inverse gamma densities. (Some R packages do offer such functions, but we can do what we need here without them). As you will prove in Problem 6.3, if X and Y are two random variables such that $0 < X, Y < \infty$ and $Y = \frac{1}{X}$, if $X \sim \text{Gamma}(\alpha, \beta)$, then $Y \sim \text{Inverse Gamma}(\alpha, \beta)$, $0 < X, Y < \infty$.

We know that R *does* have functions for gamma random variables. Furthermore, for any strictly positive values x and y , the inverse function is monotonic: if $x < y$, then $\frac{1}{x} > \frac{1}{y}$. The same principle concerning monotonic transformations and endpoints that we used in Sect. 6.2.8.3 helps us here. Specifically, suppose we want the 0.025 and 0.975 quantiles of an IG(3, 6) density. These will be the inverses of the 0.975 and 0.025 quantiles of the Gamma(3, 6) density, obtained with R as follows:

```
> 1 / qgamma( c(0.975, 0.025), 3, 6)
[1] 0.8304857 9.6981903
```

6.3.2 Obtaining the Posterior Density

As always, the posterior density will be proportional to the prior times the likelihood:

$$\begin{aligned} p(\sigma^2|\mathbf{y}) &\propto \frac{1}{(\sigma^2)^{\alpha+1}} \exp\left(-\frac{\beta}{\sigma^2}\right) \times \frac{1}{(\sigma^2)^{\frac{n}{2}}} \exp\left[-\frac{nv}{2\sigma^2}\right], \quad 0 < \sigma^2 < \infty \\ &\propto \frac{1}{(\sigma^2)^{\frac{n}{2}+\alpha+1}} \exp\left[-\frac{1}{\sigma^2}\left(\frac{nv}{2} + \beta\right)\right], \quad 0 < \sigma^2 < \infty \end{aligned} \quad (6.12)$$

Sure enough, this is the kernel of another inverse gamma density:

$$\sigma^2 | \mathbf{y} \sim IG\left(\alpha + \frac{n}{2}, \beta + \frac{nv}{2}\right) \quad (6.13)$$

The form of the posterior density reveals another strategy for specifying an inverse gamma prior, involving the equivalent prior sample size. Clearly the α parameter in the prior is analogous to the data sample size n divided by 2. Similarly, β from the prior corresponds to n times the average squared distance of the data values y_i from the supposedly known value μ . Thus, we may think of the parameters of an inverse gamma prior for a normal variance this way:

$$\sigma^2 \sim IG\left(\frac{n_0}{2}, \frac{n_0 \sigma_0^2}{2}\right) \quad (6.14)$$

contains the same information as if we had seen a previous real data sample of size n_0 in which the average squared distance of the observations from μ was σ_0^2 .

6.3.3 Jeffreys Prior for Normal Variance, Mean Assumed Known

It can be shown that the Jeffreys prior for a normal variance is

$$p(\sigma^2) \propto \frac{1}{\sigma^2}, \quad 0 < \sigma^2 < \infty \quad (6.15)$$

It is an improper prior, since $\int_0^\infty \frac{1}{\sigma^2} d\sigma^2$ is not finite. This is the limit of an inverse gamma prior as both parameters go to 0.

An inverse gamma density is proper only if both of its parameters are strictly positive. The Jeffreys prior can be used for inference regarding a normal variance only if, when combined with the likelihood, the posterior produced is a proper inverse gamma. Consider (6.13) in the case in which α and β from the prior were both equal to 0. The data would have to fill two requirements in order for both parameters in the posterior inverse gamma density to be strictly positive: the sample size n would have to be at least 1, so that $\frac{n}{2} > 0$, and at least one observed data point y_i in the dataset would have to have value not equal to the known μ so that $v = \frac{\sum_{i=1}^n (y_i - \mu)^2}{n} > 0$. Needless to say, these criteria are met in virtually all datasets.

6.4 Normal: Unknown Precision, Mean Assumed Known

Since the precision parameter in a normal density is just the inverse of the variance, if we do inference on one, we can easily derive the corresponding inference for the other. Specifically, if

$$\tau^2 = \frac{1}{\sigma^2},$$

and the prior on σ^2 is specified as

$$\sigma^2 \sim IG(\alpha, \beta)$$

then the equivalent prior induced on τ^2 is

$$\tau^2 \sim G(\alpha, \beta)$$

and the resulting posterior density for τ^2 is

$$\tau^2 | \mathbf{y} \sim G\left(\alpha + \frac{n}{2}, \beta + \frac{nv}{2}\right)$$

6.4.1 Inference for the Variance in the Mercury Concentration Problem

Suppose now that we magically knew that the mean μ of log-transformed mercury concentration in tissue from fish caught in the Des Moines River was $-2.5 \log$ ppm, and we wanted to use our data to infer about the population variance σ^2 . We would have had to define our prior on σ^2 before seeing that data. Perhaps an expert on mercury contamination told us that he was 95% sure that the variance was between 0.25 and 0.75. To find the parameters of an inverse gamma density for which those are the 0.025 and 0.975 quantiles, respectively, I can use trial and error with the `qgamma` function. I know that the mean of an inverse gamma density is $\frac{\beta}{\alpha-1}$, so I will begin with an α and a β for which that ratio is around 0.5. After some experimentation, I arrive at:

```
> 1/qgamma( c(0.975, 0.025), 13.3, 5.35)
[1] 0.2506642 0.7492485
```

So my prior on σ^2 is $IG(13.3, 5.35)$. Now in the real data, $n = 21$ and $v = 0.371$. Thus, the posterior density obtained is

$$\sigma^2 | \mathbf{y} \sim IG(23.8, 9.25)$$

The posterior mean $E(\sigma^2 | \mathbf{y}) = 0.406$ and a 95% equal tail posterior credible set is

```
> 1/qgamma( c(0.975, 0.025), 23.8, 9.25)
[1] 0.2699072 0.6078543
```

Problems

6.1. We used the following identity in deriving the likelihood for the mean μ of a normal distribution. Verify that it is true.

$$\sum_{i=1}^n (y_i - \mu)^2 = \sum_{i=1}^n (y_i - \bar{y})^2 + n (\bar{y} - \mu)^2$$

6.2. The observed weights (in grams) of 20 pieces of candy randomly sampled from candy-making machines in a certain production area are as follows:

46 58 40 47 47 53 43 48 50 55 49 50 52 56 49
54 51 50 52 50

Assume that weights of this type of candy are known to follow a normal distribution, and that the mean weight of candies produced by machines in this area is known to be 51 g. We are trying to estimate the variance, which we will now call θ .

1. What is the conjugate family of prior distributions for a normal variance (not precision) when the mean is known?
2. Suppose previous experience suggests that the expected value of θ is 12 and the variance of θ is 4. What parameter values are needed for the prior distribution to match these moments?
3. What is the posterior distribution $p(\theta|y)$ for these data under the prior from the previous step?
4. Find the posterior mean and variance of θ .
5. Comment on whether the assumptions of known mean or known variance are likely to be justified in the situation in Problem 6.1.

6.3. Consider two random variables X and Y , $0 < X, Y < \infty$, where $Y = \frac{1}{X}$. Show that if $X \sim \text{Gamma}(\alpha, \beta)$, then $Y \sim \text{Inverse Gamma}(\alpha, \beta)$, $0 < X, Y < \infty$.

Chapter 7

More Realism Please: Introduction to Multiparameter Models

Real-world problems nearly always require statistical models with more than one unknown quantity. However, usually only one, or a few, parameters or predictions are of substantive interest. Our analysis of mercury concentrations in fish tissue provides a simple, but nevertheless typical, example. We may be primarily interested in the population mean of log mercury concentration, but of course we don't really know the value of the population variance σ^2 . Therefore, in a realistic model, we must treat σ^2 as an unknown parameter along with μ .

Frequentists often refer to unknown parameters that are not of substantive interest as “nuisance parameters.” Bayesian statistics provides a sound mathematical framework for handling them and appropriately quantifying the uncertainty about the parameters of interest that is induced by our lack of knowledge about the other unknown model parameters.

Bayesian analysis seeks the *posterior marginal* distribution of the parameter or parameters of interest—that is, the distribution of those parameters conditional only on the observed data (not on any other unknown parameters). In the example of the normal model, the posterior marginal density of μ is $p(\mu|\mathbf{y})$.

The general Bayesian approach is to obtain the *joint posterior* distribution of all unknown quantities in the model and then to integrate out the one(s) in which we are not interested.

As our example, let's reconsider Bayesian analysis of sample data drawn from a normal population, this time realistically admitting that we don't know either the population mean or the population variance. In this case, we will need to specify a joint prior on both of the unknown parameters.

7.1 Conventional Noninformative Prior for a Normal Likelihood with Both Mean and Variance Unknown

Suppose we have no prior information or that we want our analysis to depend only on the current data. We need to construct a noninformative joint prior density for μ and σ^2 .

The standard noninformative prior in this case arises by considering μ and σ^2 a priori independent. A priori independence may be a reasonable assumption here. It means that if we had prior knowledge about the center of the population distribution (μ), that wouldn't tell us anything about the spread of the population distribution (represented by σ^2), and conversely, prior information about the spread wouldn't tell us anything about the center.

Recall that, if two random variables are independent, then their joint density is simply the product of their individual marginal densities. Thus, the standard noninformative prior that we are seeking is simply the product of the standard noninformative priors for μ when σ^2 is assumed known and for σ^2 when μ is assumed known:

$$p(\mu, \sigma^2) \propto \frac{1}{\sigma^2}, \quad -\infty < \mu < \infty, 0 < \sigma^2 < \infty \quad (7.1)$$

You will recognize $\frac{1}{\sigma^2}$ as the Jeffreys prior on σ^2 when μ is assumed known. Multiplying the expression in (7.1), there is an invisible “1,” which is the flat prior on μ . Note that this joint prior distribution is improper. If we use it, we must make sure that the data have the necessary properties to produce a proper joint posterior distribution. With this prior and a normal likelihood, the data must consist of at least two observations, and at least two observed data values must be unequal. Needless to say, most datasets satisfy these minimal requirements.

With observed data vector \mathbf{y} , the joint posterior is proportional to

$$p(\mu, \sigma^2 | \mathbf{y}) \propto p(\mu, \sigma^2) \times p(\mathbf{y} | \mu, \sigma^2) \quad (7.2)$$

$$\begin{aligned} &\propto \frac{1}{\sigma^2} \times \frac{1}{(\sigma^2)^{\frac{n}{2}}} \exp \left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2 \right) \\ &= \frac{1}{(\sigma^2)^{\frac{n}{2}+1}} \exp \left(-\frac{1}{2\sigma^2} \left[\sum_{i=1}^n (y_i - \bar{y})^2 + n(\bar{y} - \mu)^2 \right] \right) \\ &= \frac{1}{(\sigma^2)^{\frac{n}{2}+1}} \exp \left(-\frac{1}{2\sigma^2} [(n-1)s^2 + n(\bar{y} - \mu)^2] \right) \end{aligned} \quad (7.3)$$

where s^2 is the sample variance of the y_i s, calculated as $s^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$. As in the normal model with σ^2 assumed known, \bar{y} is the sufficient statistic for μ . Here s^2 is the sufficient statistic for the unknown σ^2 .

Since our primary interest is in the population mean μ , we must integrate σ^2 out of the joint posterior density to obtain the posterior marginal density of μ given \mathbf{y} . Note that the expression in (7.2) is unnormalized. Before we do the integration, we want to find the normalizing constant of $p(\mu, \sigma^2 | \mathbf{y})$. This identity from conditional probability will get us there:

$$p(\mu, \sigma^2 | \mathbf{y}) = p(\mu | \sigma^2, \mathbf{y}) p(\sigma^2 | \mathbf{y})$$

So if we can find $p(\mu | \sigma^2, \mathbf{y})$ and $p(\sigma^2 | \mathbf{y})$ (including their normalizing constants), then their product is the normalized joint posterior. Let's do the second component first. We simply need to integrate μ out of (7.2) and identify the result as the kernel of a density that we know:

$$\begin{aligned} p(\sigma^2 | \mathbf{y}) &\propto \int_{-\infty}^{\infty} \frac{1}{(\sigma^2)^{\frac{n+1}{2}+1}} \exp\left(-\frac{1}{2\sigma^2} [(n-1)s^2 + n(\bar{y} - \mu)^2]\right) d\mu \\ &\propto \frac{1}{(\sigma^2)^{\frac{n+1}{2}}} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right) \int_{-\infty}^{\infty} \frac{1}{\sigma} \exp\left(-\frac{n(\mu - \bar{y})^2}{2\sigma^2}\right) d\mu \\ &\propto \frac{1}{(\sigma^2)^{\frac{n+1}{2}}} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right) \sqrt{\frac{2\pi}{n}} \\ &\propto \frac{1}{(\sigma^2)^{\frac{n+1}{2}}} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right) \end{aligned} \quad (7.4)$$

We immediately recognize the last line of (7.4) as the kernel of an inverse gamma density with parameters $\frac{n-1}{2}$ and $\frac{(n-1)s^2}{2}$. Therefore, its normalizing constant has to be $\frac{\left(\frac{(n-1)s^2}{2}\right)^{\frac{n-1}{2}}}{\Gamma(\frac{n-1}{2})}$.

To obtain the conditional posterior distribution of μ given σ^2 and \mathbf{y} , we use what we already know [from expression (6.6) in Chap. 6] about the posterior distribution of μ with *known* variance and a flat prior on μ :

$$\mu | \sigma^2, \mathbf{y} \sim N\left(\bar{y}, \frac{\sigma^2}{n}\right)$$

So the normalized joint posterior distribution is the product:

$$\begin{aligned} p(\mu, \sigma^2 | \mathbf{y}) &= \frac{\left(\frac{(n-1)s^2}{2}\right)^{\frac{n-1}{2}}}{\Gamma(\frac{n-1}{2})} \frac{1}{(\sigma^2)^{\frac{n+1}{2}}} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right) \times \\ &\quad \frac{\sqrt{n}}{\sqrt{2\pi}\sigma} \exp\left(-\frac{n(\mu - \bar{y})^2}{2\sigma^2}\right) \end{aligned} \quad (7.5)$$

Now the *marginal* posterior distribution of μ can be obtained by direct integration:

$$\begin{aligned}
 p(\mu \mid \mathbf{y}) &= \int \frac{\left(\frac{(n-1)s^2}{2}\right)^{\frac{n-1}{2}}}{\Gamma\left(\frac{n-1}{2}\right)} \frac{1}{(\sigma^2)^{\frac{n+1}{2}}} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right) \frac{\sqrt{n}}{\sqrt{2\pi}\sigma} \exp\left(-\frac{n(\mu - \bar{y})^2}{2\sigma^2}\right) \\
 d\sigma^2 &= \frac{\left(\frac{(n-1)s^2}{2}\right)^{\frac{n-1}{2}}}{\Gamma\left(\frac{n-1}{2}\right)} \frac{\sqrt{n}}{\sqrt{2\pi}} \int \frac{1}{(\sigma^2)^{\frac{n+1}{2}}} \exp\left(-\frac{(n-1)s^2 + n(\mu - \bar{y})^2}{2\sigma^2}\right) d\sigma^2 \\
 &= \frac{\left(\frac{(n-1)s^2}{2}\right)^{\frac{n-1}{2}}}{\Gamma\left(\frac{n-1}{2}\right)} \frac{\sqrt{n}}{\sqrt{2\pi}} \frac{\Gamma\left(\frac{n}{2}\right)}{\left[\frac{(n-1)s^2 + n(\mu - \bar{y})^2}{2}\right]^{\frac{n}{2}}} \\
 &= \frac{\left(\frac{(n-1)s^2}{2}\right)^{\frac{n-1}{2}}}{\Gamma\left(\frac{n-1}{2}\right)} \frac{\sqrt{n}}{\sqrt{2\pi}} \frac{\Gamma\left(\frac{n}{2}\right)}{\left(\frac{(n-1)s^2}{2}\right)^{n/2} \left[1 + \frac{1}{(n-1)} \left(\frac{\mu - \bar{y}}{s/\sqrt{n}}\right)^2\right]^{n/2}} \\
 &= \frac{\Gamma\left(\frac{n}{2}\right)}{\Gamma\left(\frac{n-1}{2}\right) \sqrt{(n-1)\pi s} / \sqrt{n} \left[1 + \frac{1}{(n-1)} \left(\frac{\mu - \bar{y}}{s/\sqrt{n}}\right)^2\right]^{n/2}}
 \end{aligned} \tag{7.6}$$

Compare the last line of (7.6) to the pdf of a student's t distribution given in Table A.2. Yes, this is a t distribution with

- mean \bar{y}
- scale parameter $\frac{s^2}{n}$
- degrees of freedom $n - 1$

Recall that there is a whole family of student's t distributions. All members are symmetric and bell shaped like the normal density, but t densities are less peaked and more spread out than normal densities. The larger the degrees of freedom of a t density, the more similar in shape it is to the normal.

7.1.1 Example: The Mercury Concentration Data

We now reanalyze the data on log-transformed mercury concentrations. Although estimating the population mean μ is our primary aim, we recognize that the variance σ^2 is also unknown. We first carry out the Bayesian analysis using the improper noninformative prior described in the previous section.

In this dataset, the sample mean $\bar{y} = -2.563$, the sample variance $s^2 = 0.385$, and the sample size $n = 21$. Thus, the marginal posterior density of μ is

$$\mu|\mathbf{y} \sim t_{20}(-2.563, 0.018)$$

The posterior mean is $E(\mu|\mathbf{y}) = \bar{y} = -2.563$. The 95% equal tail posterior credible set based on the posterior t distribution can be obtained using the `qt` function in R, which gives the quantiles of a t distributions with mean 0 and scale parameter 1:

```
> qt(c(0.025, 0.975), 20)
[1] -2.085963 2.085963
```

The required credible set is calculated by multiplying each of these quantiles by the square root of the scale parameter and adding the mean:

```
> -2.5629 + qt(c(0.025, 0.975), 20) * sqrt(0.018)
[1] -2.845341 -2.280459
```

Notice that this posterior credible set is wider than the one we calculated in Sect. 6.2.9 when assuming that σ^2 was known. Our uncertainty about σ^2 in the present reanalysis is reflected appropriately in greater uncertainty about μ .

We expect that the current analysis with the standard noninformative joint prior on μ and σ^2 will give results that match a frequentist analysis. We can verify this by using the R function `t.test` to calculate the frequentist point estimate and 95% confidence interval:

```
> t.test(logiowafish)
```

```
One Sample t-test
```

```
data: logiowafish
t = -18.9193, df = 20, p-value = 3.11e-14
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -2.845479 -2.280328
sample estimates:
mean of x
-2.562904
```

Yes, the results agree up to three digits to the right of the decimal point. The differences are due to my having rounded along the way in the Bayesian analysis.

Note that this Bayesian analysis was *not* conjugate. The joint prior density was the product of a normal density times an inverse gamma density (improper limiting forms of each), but the joint posterior density was not a product of those two families. Yes, the marginal posterior density of σ^2 was inverse gamma, but the marginal posterior density of μ was t , not normal.

7.2 Informative Priors for μ and σ^2

When prior information is available about the unknown mean and variance parameters of a normal population, we will wish to incorporate it into the Bayesian analysis. Common practice is to assume a priori independence between μ and σ^2 and to specify the joint prior as the product of a proper normal prior on μ and a proper inverse gamma prior on σ^2 .

In this case, neither of the marginal posterior densities $p(\mu|\mathbf{y})$ and $p(\sigma^2|\mathbf{y})$ will be of a standard parametric form. The joint posterior density $p(\mu, \sigma^2|\mathbf{y})$ most certainly does not factor into the product of a normal density times and inverse gamma density. Thus, this is not a conjugate analysis.

The term *semi-conjugate* is often used for a prior specification such as this, in which the prior specified for each unknown model parameter would have been conjugate if all other model parameters were assumed known, but the entire joint prior is not conjugate.

Because the marginal posterior densities of interest are not of known parametric families, obtaining posterior means, credible sets, or other numeric summaries of interest is not as straightforward as it was. Simulation-based methods of Bayesian model fitting enable us to perform Bayesian inference for nonconjugate models. These methods are the topic of the next chapter.

7.3 A Conjugate Joint Prior Density for the Normal Mean and Variance

Interestingly, a conjugate prior does exist for the normal likelihood with both mean and variance parameters unknown. It relies on the somewhat counterintuitive notion that the precision of our prior knowledge about the population mean parameter μ depends on the value of the unknown population variance σ^2 . Now that simulation-based methods are conveniently available for Bayesian model fitting, this joint prior density is not commonly used for inference in simple models with a normal likelihood. However, it is worth knowing about because of both its historical use and its current usefulness as a building block in more complex models.

The conjugate joint prior density—the *normal-inverse gamma density*—is constructed as the product of a marginal inverse gamma prior density on σ^2 and a conditional normal density for μ given σ^2 :

$$\begin{aligned} p(\mu, \sigma^2) &= p(\sigma^2)p(\mu|\sigma^2) \\ &= IG(\alpha, \beta) \times N(\mu_0, \frac{1}{\kappa}\sigma^2) \end{aligned} \tag{7.7}$$

The four prior parameters that must be specified are α and β from the inverse gamma prior on σ^2 , the mean μ_0 of the prior density on μ , and κ . The conditional

prior variance of μ given the unknown parameter σ^2 is σ^2 divided by the prior parameter κ .

In pdf format, the normal-inverse gamma prior looks like this:

$$p(\mu, \sigma^2) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{1}{(\sigma^2)^{\alpha+1}} \exp\left(-\frac{\beta}{\sigma^2}\right) \times \frac{\sqrt{\kappa}}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{\kappa(\mu - \mu_0)^2}{2\sigma^2}\right) \quad (7.8)$$

Once the current data, n observations y_1, y_2, \dots, y_n with sample mean \bar{y} , have been observed, of course the posterior density is proportional to the joint prior times the normal likelihood. If we factor the likelihood in the same way as in (7.2), we can derive the posterior as follows:

$$\begin{aligned} p(\mu, \sigma^2 | \mathbf{y}) &\propto \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{1}{(\sigma^2)^{\alpha+1}} \exp\left(-\frac{\beta}{\sigma^2}\right) \times \frac{\sqrt{\kappa}}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{\kappa(\mu - \mu_0)^2}{2\sigma^2}\right) \times \\ &\quad \frac{1}{(\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{(n-1)s^2}{2\sigma^2}\right) \frac{\sqrt{n}}{\sigma} \exp\left(-\frac{n(\bar{y} - \mu)^2}{2\sigma^2}\right) \\ &\propto \frac{1}{(\sigma^2)^{\alpha+\frac{n}{2}+1}} \exp\left(-\frac{1}{\sigma^2} \left(\beta + \frac{(n-1)s^2}{2}\right)\right) \times \\ &\quad \frac{\sqrt{\kappa}}{\sigma} \exp\left(-\frac{\kappa(\mu - \mu_0)^2 + n(\mu - \bar{y})^2}{2\sigma^2}\right) \end{aligned} \quad (7.9)$$

Now let's do a little algebra with the numerator of the last expression inside an exponent in (7.9):

$$\begin{aligned} \kappa(\mu - \mu_0)^2 + n(\mu - \bar{y})^2 &= \kappa(\mu^2 - 2\mu\mu_0 + \mu_0^2) + n(\mu^2 - 2\mu\bar{y} + \bar{y}^2) \\ &= (\kappa + n)\mu^2 - 2\mu(\kappa\mu_0 + n\bar{y}) + \kappa\mu_0^2 + n\bar{y}^2 \\ &= (\kappa + n) \left[\mu^2 - \frac{2\mu(\kappa\mu_0 + n\bar{y})}{\kappa + n} + \left(\frac{\kappa\mu_0 + n\bar{y}}{\kappa + n} \right)^2 \right] \\ &\quad - \frac{(\kappa\mu_0 + n\bar{y})^2}{\kappa + n} + \kappa\mu_0^2 + n\bar{y}^2 \\ &= (\kappa + n) \left(\mu - \frac{\kappa\mu_0 + n\bar{y}}{\kappa + n} \right)^2 + \frac{\kappa n(\bar{y} - \mu_0)^2}{\kappa + n} \end{aligned} \quad (7.10)$$

Note that we got from line 2 to line 3 by in (7.10) completing the square—specifically, by adding and subtracting $\frac{(\kappa\mu_0 + n\bar{y})^2}{\kappa + n}$. Now finally we can integrate the result from (7.10) into (7.9) to obtain

$$p(\mu, \sigma^2 | \mathbf{y}) \propto \frac{1}{(\sigma^2)^{\alpha + \frac{n}{2} + 1}} \exp \left[-\frac{1}{\sigma^2} \left(\beta + \frac{(n-1)s^2}{2} + \frac{\kappa n (\bar{y} - \mu_0)^2}{2(\kappa + n)} \right) \right] \times \\ \frac{\sqrt{\kappa}}{\sigma} \exp \left[-(\kappa + n) \left(\frac{\left(\mu - \frac{\kappa \mu_0 + n \bar{y}}{\kappa + n} \right)^2}{2\sigma^2} \right) \right] \quad (7.11)$$

Ah hah, we can recognize the joint posterior density in (7.11) as normal-inverse gamma:

$$\mu, \sigma^2 | \mathbf{y} \sim IG \left(\alpha + \frac{n}{2}, \beta + \frac{(n-1)s^2}{2} + \frac{\kappa n (\bar{y} - \mu_0)^2}{2(\kappa + n)} \right) \times \\ N \left(\frac{\kappa \mu_0 + n \bar{y}}{\kappa + n}, \frac{\sigma^2}{\kappa + n} \right) \quad (7.12)$$

Indeed, the normal-inverse gamma prior density was conjugate for the normal likelihood, since the resulting posterior density is in the same family as the prior.

Recall that the parameters of the inverse gamma density can be thought of as $\alpha = \frac{n_0}{2}$ and $\beta = \frac{n_0 \sigma_0^2}{2}$. With that reparameterization, the joint posterior density becomes

$$\mu, \sigma^2 | \mathbf{y} \sim IG \left(\frac{n_0}{2} + \frac{n}{2}, \frac{n_0 \sigma_0^2}{2} + \frac{(n-1)s^2}{2} + \frac{\kappa n (\bar{y} - \mu_0)^2}{2(\kappa + n)} \right) \times \\ N \left(\frac{\kappa \mu_0 + n \bar{y}}{\kappa + n}, \frac{\sigma^2}{\kappa + n} \right) \quad (7.13)$$

It is clear that n_0 is the equivalent prior sample size with respect to the variance, and κ is the equivalent prior sample size with respect to the mean μ .

By an integration process analogous to that in (7.6), we can find that the posterior marginal density $p(\mu | \mathbf{y})$ is a t density with:

- mean $\frac{\kappa \mu_0 + n \bar{y}}{\kappa + n}$
- scale parameter $\frac{n_0 \sigma_0^2 + (n-1)s^2 + \frac{\kappa n (\bar{y} - \mu_0)^2}{(\kappa + n)}}{(\kappa + n)(n_0 + n)}$
- degrees of freedom $n + n_0$

7.3.1 Example: The Mercury Contamination Data

Let's use the same prior information that we had in Chap. 6 regarding mercury contamination in fish, but this time we will represent it by the conjugate joint prior. Our best guess for the mean (on the log scale) was -2.75 , and we had about as

much belief in that value as if we had seen three previous observations. Thus, for the conjugate prior, μ_0 will be -2.75 , and κ will be 3.

In our analysis with just the variance unknown, we specified an inverse gamma prior with parameters $\alpha = 13.3$ and $\beta = 5.35$. In the joint conjugate setting, this would be equivalent to $n_0 = 27$ and $\sigma_0^2 = 0.402$.

Recall that the sample mean in the fish concentration data is $\bar{y} = -2.563$, the sample variance $s^2 = 0.385$, and the sample size $n = 21$. Combining these data values with the specified prior parameters yields the following posterior marginal densities:

$$\begin{aligned}\sigma^2|\mathbf{y} &\sim IG(23.8, 9.24) \\ \mu|\mathbf{y} &\sim t_{48}(-2.586, 0.0162)\end{aligned}\tag{7.14}$$

For σ^2 , the posterior mean is $\frac{9.24}{23.8-1} = 0.405$, and the 95% equal tail credible set can be found using R in the same way as we did in Sect. 6.4.1:

```
> 1/qgamma( c(0.975, 0.025), 23.8, 9.24)
[1] 0.2696154 0.6071971
```

The posterior mean of μ , $E(\mu|\mathbf{y})$, is -2.586 log units. We can use R to find the 95% credible set as follows:

```
> -2.586 + qt( c(0.025, 0.975), 48) * sqrt( 0.0162)
[1] -2.841912 -2.330088
```

This credible set is a bit narrower than the one we calculated when using the standard noninformative joint prior in Sect. 7.1.1. This is because this time we incorporated additional information through an informative prior.

7.3.2 The Standard Noninformative Joint Prior as a Limiting Form of the Conjugate Prior

The standard noninformative prior discussed in Sect. 7.1 actually is an improper limiting form of the conjugate joint prior, in which α goes to $-1/2$ and both β and κ go to 0. Equivalently, n_0 goes to -1 , and σ_0^2 and κ go to 0. With $\kappa = 0$, the value of μ_0 can be arbitrary. When thought of in this way (rather than as the product of an improper inverse gamma prior on σ^2 times an *independent* improper normal prior on μ), the conventional noninformative joint prior can be considered a conjugate prior.

Problems

7.1. Reanalyze the candy-weight data from Problem 6.2. This time, acknowledge that both μ and σ^2 are unknown, and use the standard noninformative prior given in (7.1).

1. Find $p(\mu|\mathbf{y})$, the posterior marginal density of μ . Name it; give the values of its parameters; and use it to find the posterior mean and 95% credible set for μ .
2. Suppose that the population variance σ^2 was the parameter of primary interest. Find its posterior marginal density. Give numeric values of the posterior mean $E(\sigma^2|\mathbf{y})$ and the 95% posterior credible set for σ^2 .

7.2. Repeat Problem 7.1 using the conjugate prior from Sect. 7.3. Use the same inverse gamma prior for σ^2 that you used in Problem 6.2. For the conditional normal prior on μ , specify a mean of 51 and an equivalent prior sample size of 10.

1. Find the joint posterior density as a product of two densities. Give numeric values of their parameters.
2. Find $p(\mu|\mathbf{y})$, the posterior marginal density of μ . Name it; give the values of its parameters; and use it to find the posterior mean and 95% credible set for μ .
3. Suppose that the population variance σ^2 was the parameter of primary interest. Find its posterior marginal density. Give numeric values of the posterior mean $E(\sigma^2|\mathbf{y})$ and the 95% posterior credible set for σ^2 .

7.3. Show that the normal-inverse gamma prior produces the conventional noninformative prior if n_0 goes to -1 and κ and σ_0^2 both go to 0.

7.4. Fill in the details of the integration in (7.6). Be sure to explain how line 3 was obtained from line 2.

Chapter 8

Fitting More Complex Bayesian Models: Markov Chain Monte Carlo

So far we have been dealing primarily with simple, conjugate Bayesian models for which it was possible to perform exact posterior inference analytically. In more realistic and complex Bayesian models, such analytical calculations generally are not feasible. This chapter introduces the sampling-based methods of fitting Bayesian models that have transformed Bayesian statistics over the last 20 years.

8.1 Why Sampling-Based Methods Are Needed

As we already have discussed, the goals of Bayesian analysis are to make inference about unknown model parameters and to make predictions about unobserved data values. The computationally challenging aspects of these tasks involve *integration*. In this section, we will see why integration is an essential element in Bayesian analysis and will investigate the limitations of some popular methods of integration.

8.1.1 Single-Parameter Model Example

The challenge of integration arises even in single-parameter models with non-conjugate priors—for example, the model with a histogram prior on a binomial success parameter π discussed in Sect. 5.2.2. Let's see what is required to carry out the following standard inferential procedures: Plot the posterior density $p(\pi|y)$ as in Fig. 5.3, calculate the posterior mean $E(\pi|y)$, and obtain posterior predictive probabilities for the number of successes in a future sample of size 20.

8.1.1.1 Plotting the Posterior Density

As long as we can write down the likelihood and the prior(s) in mathematical form, we always can obtain an expression *proportional to* the resulting posterior distribution. (This is just Bayes' theorem, and it is true regardless of how many parameters there are in the model.) However, in order to plot a density, we need the normalizing constant, so that the area under our density plot will be 1. When the prior is nonconjugate and the posterior density is not of a recognizable family, the normalizing constant must be obtained by integration: We must find out to what numeric value the unnormalized density integrates, and then the normalizing constant is just its inverse.

In the example from Sect. 5.2.2, the required integral is

$$\int_0^1 \pi^7 (1 - \pi)^{43} p(\pi) d\pi \quad (8.1)$$

where the histogram prior $p(\pi)$ is as given below. The third column shows the normalized prior densities, such that the areas of the histogram bars sum to 1.

Interval	Prior probability	Prior density
(0, 0.1]	0.25	2.5
(0.1, 0.2]	0.50	5.0
(0.2, 0.3]	0.20	2.0
(0.3, 0.4]	0.05	0.5
> 0.4	0.00	0.0

In this case, doing the integral analytically is possible (but tedious, since the integrand is a 44-term polynomial). Since the prior density is 0 for $\pi > 0.4$, the posterior will also be 0 there, and hence, the integral needs to be evaluated only over (0.0, 0.4).

8.1.1.2 Calculating the Posterior Mean

Remember that the posterior mean of a continuous-valued parameter θ in a one-parameter Bayesian model is calculated as

$$E(\theta|y) = \int_a^b \theta p(\theta|y) d\theta$$

where $p(\theta|y)$ is the *normalized posterior density* and a and b are the limits of its support. Thus, *two integrations* are required: one to obtain the normalizing constant of the posterior density and the other to calculate the expectation.

8.1.1.3 Calculating Posterior Predictive Probabilities

Similarly, as discussed in Sect. 4.4, posterior predictive probabilities are calculated by integrating the likelihood over the posterior density, so two integrations are required.

8.1.2 Numeric Integration

When analytic integration isn't feasible, *numeric integration*—deterministic computer algorithms for approximating integrals—sometimes can produce acceptably accurate numeric results. I will give only a brief introduction here. Consult a textbook on numeric analysis, such as Chap. 4 of [Burden and Faires \(2011\)](#), for more details.

Probably the simplest numeric algorithm for one-dimensional integrals is the *composite midpoint rule*. Suppose we want to approximate the integral

$$\int_a^b f(x)dx$$

where (a, b) is a finite interval and f has a bounded first derivative on (a, b) . The composite midpoint rule involves dividing the interval (a, b) into n subintervals of equal length $h = \frac{b-a}{n}$. Denote the endpoints of these intervals by $x_i = a + ih$ for $i = 0, \dots, n$, so that the i th interval is (x_{i-1}, x_i) . Then the function is evaluated at the midpoint of each subinterval, and a little rectangle of that height is constructed above the subinterval. The integral is approximated by the sum of the areas of the n little rectangles:

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=1}^n f\left(\frac{x_{i-1} + x_i}{2}\right)$$

In general, larger numbers of subintervals produce more accurate approximations.

The composite midpoint rule is the simplest member of a group of numeric integration algorithms called Newton–Cotes algorithms, all of which require evaluating the integrand at equally spaced points. Other numeric integration algorithms exist that choose points at which to evaluate the integrand in such a way as to approximate the integral with greater accuracy while requiring fewer function evaluations. One of these algorithms, called *Gaussian quadrature*, was very popular in Bayesian statistics in the 1960s. It is implemented in the R function `integrate`.

8.1.2.1 Using Numeric Integration in the Single-Parameter Example

We have to be careful with integration in the example with a histogram prior for a binomial success parameter. The discontinuities in the prior produce discontinuities

in the unnormalized posterior as well. The problem can be avoided by performing a separate numeric integration over each of the four intervals over which the unnormalized posterior is continuous—(0,0.1), (0.1,0.2), (0.2,0.3), and (0.3,0.4)—and adding up the results. (Since the prior is 0 for $\pi > 0.4$, we already know that the posterior is 0 there as well, so we don't have to bother with that part of the integral.)

Let's begin by defining an R function to calculate the unnormalized posterior given a value of π and the applicable prior density evaluation at that value of π :

```
unnormpost <- function(pi, pr)
{
  like <- pi^7 * (1-pi)^43
  like * pr
}
```

Here's what this function produces for $\pi = 0.15$, for which the prior value is 5:

```
> unnormpost( .15, 5 )
[1] 7.88175e-09
```

We can provide more than one value of π , and `unnormpost` will evaluate the unnormalized posterior for all of them. In the code below, `mypis` is a vector of 10 values, all in the interval (0.1,0.2). In fact, it is a vector of the midpoints of 10 subintervals of width 0.01. The same value of the prior—5—goes with all of them.

```
> mypis <- seq(0.105, 0.195, by = .01)
> mypis
[1] 0.105 0.115 0.125 0.135 0.145 0.155 0.165 0.175
    0.185 0.195
> unnormpost(mypis, 5)
[1] 5.966033e-09 6.956896e-09 7.650439e-09
    7.998324e-09 7.999981e-09
[6] 7.693614e-09 7.142924e-09 6.423269e-09
    5.609825e-09 4.769086e-09
```

Now that we know how to use the `unnormpost` function, we can set up to use it to approximate the normalizing constant using the composite midpoint rule. Note that the output of the `unnormpost` function provides the heights of the little rectangles. To approximate the integral over (0.1,0.2), all we need to do is multiply each height by the subinterval width and add up the areas.

```
> mypis <- seq(0.105, 0.195, by = .01)
> heights <- unnormpost(mypis, 5)
> sum( heights * 0.01)
[1] 6.821039e-10
```

Here is an R function to calculate the entire integral by performing the above procedure for each of the component intervals and accumulating the total.

```

function()
{
  endpoints <- c(0, 0.1, 0.2, 0.3, 0.4)
  prior <- c(2.5, 5.0, 2.0, 0.5)
  h <- 0.001          # width of subintervals

  integral <- 0       # initialize variable to accumulate
                      # total integral

  for( i in 1:4)
  {
    mypis <- seq( endpoints[i] + h/2, endpoints[i+1]
      - h/2, by=h)
    heights <- unnormpost(mypis, prior[i])
    integral <- integral + sum( heights * h)
  }
  integral
}

```

The numeric result is:

```
[1] 8.126965e-10
```

Figure 8.1 shows the results of using this procedure multiple times to approximate the integral in (8.1) using different numbers of subintervals ranging from 10 to 1,000. (The numbers of subintervals shown span the whole interval (0,1), not each of the intervals over which the separate numeric integrations are performed.) The true area under the curve and the approximated area based on the numeric integration are shown in the legends. As the number of subintervals used for the composite midpoint rule increases, the approximation gets closer and closer to the true integral value.

We can plot the normalized posterior this way (Fig. 8.2):

```

plot(pi, prior * like / integral, type="l",
  ylab="normalized posterior")

```

Now that we have the normalizing constant for the posterior, we can use a second numeric integration to approximate the posterior mean.

$$\begin{aligned}
 E(\pi|y) &= \int_0^1 \pi p(\pi|y) d\pi \\
 &= \frac{1}{8.126965e-10} \int_0^1 \pi \pi^7 (1-\pi)^{43} p(\pi) d\pi \\
 &= \frac{1}{8.126965e-10} \int_0^1 \pi^8 (1-\pi)^{43} p(\pi) d\pi
 \end{aligned} \tag{8.2}$$

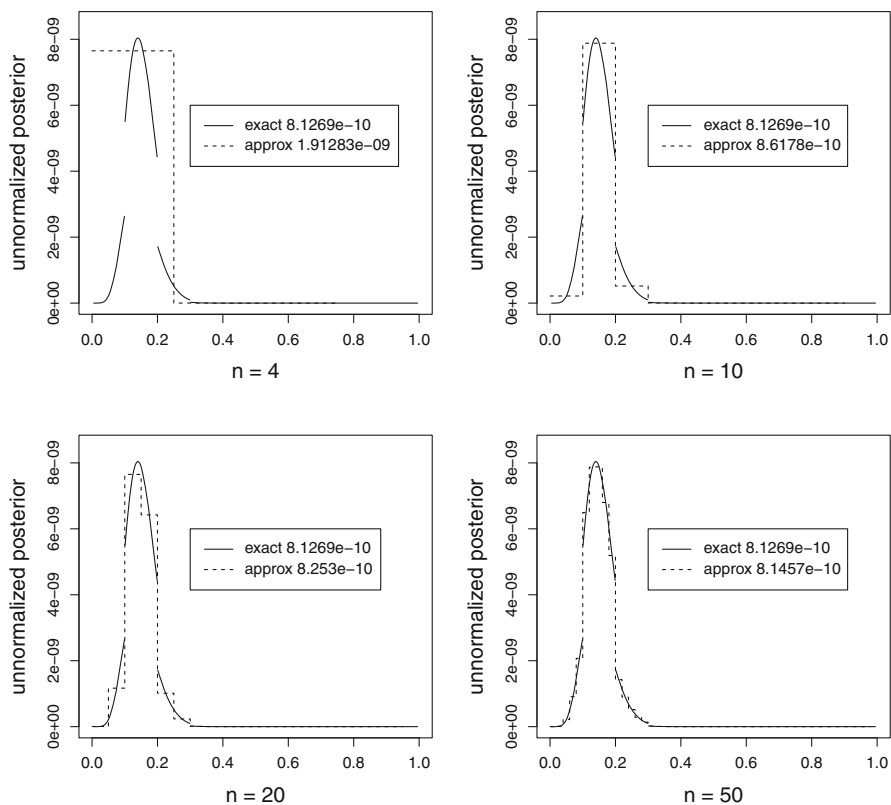


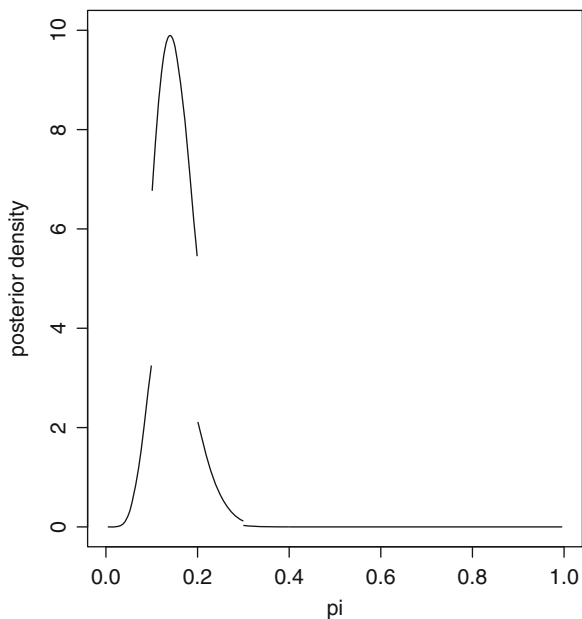
Fig. 8.1 Applying the composite midpoint rule to approximate an integral. The *solid line* is the true curve. The *dashed line* is the approximation based on n rectangles spanning (0,1)

So a slight change to our previous R code for the composite midpoint rule will do the trick.

```
function()
{
  expec <- function(p, pr)
  {
    hold <- p^8 * (1-p)^43      # Note p^8 instead of p^7
    pr * hold
  }

  normconst <- 8.126965e-10    # normalizing constants

  endpoints <- c(0, 0.1, 0.2, 0.3, 0.4)
  prior <- c(2.5, 5.0, 2.0, 0.5)
```

Fig. 8.2 The normalized posterior

```

h <- 0.001          # width of subintervals

integral <- 0        # initialize variable to accumulate
                      total integral

for( i in 1:4)
{
  mypis <- seq( endpoints[i] + h/2, endpoints[i+1]
    - h/2, by=h)
  heights <- expec(mypis, prior[i])
  integral <- integral + sum( heights * h)/normconst
  # divide by normalizing const
}
integral
}

```

The approximate posterior mean is:

```

> postmean
[1] 0.1493313

```

We could obtain very similar results by Gaussian quadrature using R's `integrate` function.

8.1.2.2 Limitations of Numeric Integration

Obviously numeric integration is a very useful tool in Bayesian analysis (and innumerable other fields). However, its applicability in Bayesian analysis is limited by a phenomenon popularly called “the curse of dimensionality.” To understand this phrase, consider two situations. Suppose first that we have a model with three parameters, say θ_1 , θ_2 , θ_3 . To obtain the normalizing constant of the three-dimensional joint posterior density, we would need to perform a three-dimensional numeric integration of the unnormalized posterior:

$$\int \int \int p(\mathbf{y}|\theta_1, \theta_2, \theta_3)\pi(\theta_1, \theta_2, \theta_3)\pi(\theta_1, \theta_2, \theta_3)d\theta_1d\theta_2d\theta_3$$

To apply the multidimensional composite midterm rule to approximate this integral, we would divide the support along each dimension into subintervals, so that the whole three-dimensional support was divided into little rectangular solids. If we used 10 subintervals in each dimension, we would have $10^3 = 1,000$ little rectangular solids in which to evaluate the integrand. If we needed greater accuracy, we might use 100 subintervals in each dimension, resulting in $100^3 = 1,000,000$ integrand evaluations. Unless the integrand is very complicated, this would be a reasonable task using a contemporary desktop computer. However, suppose that our model had not three but 3,000 parameters. We would need to perform a 3,000-dimensional integration to get the normalizing constant and a 2,999-dimensional integration to get the mean of the marginal posterior density of any individual parameter of interest. If we tried to use the composite midterm rule with 10 subintervals in each dimension, the computer would have to perform $10^{3,000}$ (i.e., 1 with 3,000 zeroes after it!) function evaluations. The computation would take way too long to be practical, and most likely would produce an inaccurate result to boot.

Bayesian models with tens or hundreds of thousands of parameters are in common use today in application areas such as weather forecasting, climate modeling, marketing, and econometrics. Although there are numeric integration algorithms that scale better to higher dimensions than the composite midpoint rule, even they become infeasible before we even reach twenty dimensions. Thus, numeric integration is hardly an adequate solution to the integration issues in realistic Bayesian models.

8.1.3 Monte Carlo Integration

Numeric integration methods are deterministic—given the same inputs, a particular numeric integration algorithm will produce the same results every time. Another broad class of computer-based integration methods is called Monte Carlo integration—methods based on (pseudo-) random sampling from probability distributions. To approximate an integral, numeric integration requires evaluating

the integrand at a fixed set of points, whereas Monte Carlo integration involves evaluating the integrand at a randomly generated set of points.

Let's redo the above example of the binomial likelihood and histogram prior, this time using Monte Carlo integration. To approximate the integral of the unnormalized posterior within the interval (0.1, 0.2), we need to generate random values of π from the uniform density on this interval, evaluate the unnormalized posterior density at each one, take the average, and multiply by the width of the interval.

```
> mypiss <- runif( 100, 0.1, 0.2 )
> heights <- unnormpost(mypiss, 5)
> mean(heights) * 0.1
[1] 6.754542e-10
```

Because the input values `mypiss` are randomly generated, we will get a slightly different answer each time we run this code.

Here is a function to calculate the entire integral by using Monte Carlo integration within each interval:

```
> mcinteg <- function()
{
  endpoints <- c(0, 0.1, 0.2, 0.3, 0.4)
  prior <- c(2.5, 5.0, 2.0, 0.5)
  nrand <- 100      # number of random points in each
                    # interval
  integral <- 0     # initialize variable to accumulate
                    # total integral

  for( i in 1:4)
  {
    mypiss <- runif(nrand, endpoints[i], endpoints[i+1])
    heights <- unnormpost(mypiss, prior[i])
    integral <- integral + mean( heights ) * (endpoints
      [i+1] - endpoints[i])
  }
  integral
}
```

And here is the result of running it:

```
> mcinteg()
[1] 8.267407e-10
```

The result is close to that obtained with numeric integration in the previous section.

An excellent introduction to Monte Carlo integration is provided in Chap. 3 of [Robert and Casella \(2010\)](#).

Monte Carlo integration has limitations similar to those of numeric integration. Just as the midpoint rule may require a large number of subintervals (and therefore of function evaluations) for accurate approximation of a complicated integral,

Monte Carlo integration may require evaluating the function at a large number of sampled values. For multidimensional integrals, uniform random samples of coordinates in the support of the function are required. In even moderate dimensions (5 or 6), the number of such sets of coordinates required for reasonable accuracy may be in the tens or hundreds of thousands or more. If the function is complicated and slow to evaluate, computing time becomes infeasible. Needless to say, Monte Carlo integration is not the solution to fitting complex, high-dimensional Bayesian models.

8.2 Sampling-Based Methods

Because of the limitations of analytic, numeric, and Monte Carlo integration, Bayesian statisticians turned to sampling-based methods of fitting Bayesian models. The idea is to draw samples from the joint posterior distribution of unknown quantities in the model. We know how to use samples to estimate characteristics of distributions (we are statisticians after all!)—sample means to estimate theoretical means, empirical quantiles to estimate theoretical quantiles, etc.

8.2.1 Independent Sampling

For some simple models, drawing independent samples from the posterior distribution of unknown parameters is straightforward. To assess whether the independent-sampling method works, we will revisit two examples that we previously have approached analytically.

8.2.1.1 One-Parameter Model: The Survey Regarding Quitting School

Recall the example from Chap. 3 in which the parameter of interest was a binomial success probability π and we used a conjugate beta prior, resulting in a beta posterior. When the prior was uniform ($\text{Beta}(1,1)$) and the data were 7 successes in 50 trials, the posterior density $p(\pi)$ was $\text{Beta}(8,44)$. Recalling that the mean of a $\text{Beta}(\alpha, \beta)$ density is $\frac{\alpha}{\alpha+\beta}$, we used R to obtain the posterior mean $E(\pi|y)$ to be 0.154 and a 95% posterior credible set to be (0.070, 0.263):

```
> 8 / (8+44)
[1] 0.1538462
> qbeta( c(0.025, 0.975), 8, 44)
[1] 0.07024083 0.26255154
```

Now let's try drawing samples from the posterior density of π and using them to estimate the same posterior summaries:


```

> postsamp <- rbeta( 50, 8, 44 )
> mean(postsamp)
[1] 0.1505567
> quantile( postsamp, c(0.025, 0.975))
      2.5%      97.5%
0.08326764 0.22745452

```

Hmm, the estimate of the mean isn't bad, but the quantiles needed for the 95% credible set are estimated poorly. That was with 50 samples drawn from the posterior. Let's see what happens with a larger sample:

```

> postsamp <- rbeta( 500, 8, 44 )
> mean(postsamp)
[1] 0.1550475
> quantile( postsamp, c(0.025, 0.975))
      2.5%      97.5%
0.0726649 0.2424132

```

That's better! We can improve the accuracy by drawing a still larger sample from the posterior:

```

> postsamp <- rbeta( 5000, 8, 44 )
> mean(postsamp)
[1] 0.1532853
> quantile( postsamp, c(0.025, 0.975))
      2.5%      97.5%
0.06823506 0.26647570

```

8.2.1.2 Two Parameter Model: Normal-Inverse Gamma Model for the Mercury Contamination Data

In Sect. 7.3.1, we used a normal-inverse gamma model for the normal mean and variance in the mercury contamination problem. Our prior parameter values were $\kappa = 3$, $\mu_0 = -2.45$, $n_0 = 27$, and $\sigma_0^2 = 0.402$. The summary statistics from the dataset were $\bar{y} = -2.563$, $s^2 = 0.385$, and $n = 21$.

We can use a two-step process to draw independent samples from this joint posterior density. First we'll draw random samples from the inverse gamma posterior density $p(\sigma^2|y)$. Then we'll use those sampled values of σ^2 in drawing from the conditional posterior density $p(\mu|\sigma^2, y)$. Here are the code and results:

```

> kappa <- 3
> mu0 <- -2.75
> n0 <- 27
> sigsq0 <- 0.402
> ybar <- -2.563
> ssq <- 0.385

```

```

> n <- 21
> sigsq <- 1/ rgamma( 1000, (n0+n)/2, (n0* sigsq0
+ (n-1) * ssq)/2 )
# In the next line of code, these 1000 values of
#   sigsq appear in the
#   standard deviation for drawing random normals.
# This means that each of the values of mu will be
#   drawn from a normal
#   density with a different standard deviation.
> mu <- rnorm( 1000, (kappa * mu0 + n * ybar)
+ ((kappa+n), sqrt( sigsq / (kappa+n) ) )

> mean(sigsq)
[1] 0.4033241
> quantile(sigsq, c(0.025, 0.975))
      2.5%      97.5%
0.2719035 0.6036209
> mean(mu)
[1] -2.584575
> quantile(mu, c(0.025, 0.975) )
      2.5%      97.5%
-2.852751 -2.327061

```

These results agree closely with the posterior means and credible sets obtained in Sect. 7.3.1. Note that drawing the μ s from normal densities, all with the same mean but each conditioning on a *different* value of σ^2 , was equivalent to drawing from the marginal t distribution for μ .

8.2.1.3 More on Independent Sampling

These simple examples, in which we can compare Bayesian estimation based on independent sampling from the posterior density to results obtained analytically, suggest that sampling-based methods may work well. Unfortunately, in most realistically complex Bayesian models, independent sampling from the posterior density is not straightforward at all. Much statistical research has focused on development of methods for drawing independent samples from nonstandard densities.

Unfortunately, when the joint posterior is very high dimensional, independent-sampling methods generally become computationally infeasible, just as happened with numeric and Monte Carlo integration.

8.3 Introduction to Markov Chain Monte Carlo Methods

Because of the shortcomings of the other approaches described above, Markov chain Monte Carlo (MCMC) methods have become the predominant computational strategy for fitting Bayesian models. MCMC makes it possible to draw samples from very high-dimensional joint posterior densities. The downside is that the samples are not independent, which requires extra care on the part of the statistician who wishes to use them for inference.

For more details on MCMC and other Monte Carlo methods, see [Robert and Casella \(2010\)](#).

8.3.1 Markov Chains

In this book, we are concerned only with the class of Markov chains called homogeneous discrete-time Markov chains with continuous state spaces. When I refer to “Markov chains,” I will mean this particular class.

Markov chains are random variables that are generated sequentially over time. A Markov chain is said to start at “time 0” at some initial value. At time 1, the chain moves to a random value generated from a probability distribution whose parameters depend on the initial value from time 0. At each successive time point, the chain again moves to a new random value generated from the same form of probability distribution, but with parameters depending on the value from the immediately preceding time point.

We think of Markov chains as potentially running until arbitrarily large times. Common notation for a Markov chain is $\{X_t\}_{t=0}^{\infty}$, where X_t represents the random variable at time t , and once time t has been reached, x_t denotes the realized value. The value x_t is called the state of the chain at time t . The time points t at which a Markov chain generates new values are often referred to as *iterations* and the generated values x_t as *iterates*. The values of a Markov chain may be either scalars or vectors, but for a given Markov chain, all the iterates will be of the same dimension.

The support from which all the random variables X_t are drawn is called the *state space* of the Markov chain. The probability distribution from which the state at each time t is drawn, conditional on the state from the previous time, is called the *transition kernel* of the chain and may be denoted $p(X_t|X_{t-1} = x_{t-1})$.

The defining characteristic of Markov chains is the Markov property which is that

$$p(X_t|X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots, X_0 = x_0) = p(X_t|X_{t-1} = x_{t-1})$$

In words, the Markov property says that, *conditional on the value at the time point immediately before it*, the value observed at any time t is independent of all the earlier values. Another way to say this is that, if we know the value of x_{t-1} , then the values x_0 through x_{t-2} contain no further information about X_t .

Under certain regularity conditions that are beyond the mathematical scope of this course, the draws generated by a Markov chain will *converge in distribution* to draws from a *target* probability distribution. This means that if a Markov chain is allowed to run for long enough (and it meets the regularity conditions), then this “convergence” occurs and all the subsequent iterates are draws from this target distribution. Another way to say the same thing is that the Markov chain eventually “forgets” its initial value and that Markov chains started from different initial values but having the same transition kernel will end up drawing from the same target distribution.

Even after a Markov chain has converged, the subsequent iterates still are dependent. We can see that from the fact that the transition kernel doesn’t change—each new value is generated from a probability distribution that depends on the values from the preceding iteration. Statistical methods appropriate for correlated samples are required for using MCMC output.

8.3.2 *Markov Chains for Bayesian Inference*

Markov chains are important in Bayesian statistics because it generally is possible to construct a Markov chain (i.e., to define its transition kernel) in such a way that the target distribution is the joint posterior distribution of all the unknown parameters in the Bayesian model of interest! Even for very high-dimensional models in which it is infeasible to draw samples directly from the joint posterior, it often is straightforward to define a transition kernel that draws conditionally, given an existing draw from the joint posterior. Thus, Markov chain Monte Carlo methods provide a way of drawing samples from the joint posterior distribution in realistic, high-dimensional Bayesian models.

MCMC had been known for decades before its implications for Bayesian statistical modeling were fully recognized. One way of defining transition kernels for model fitting, now called the Metropolis algorithm, was first published in 1953 in [Metropolis et al. \(1953\)](#). The method was generalized in [Hastings \(1970\)](#). Image processing is a field in which the potential of MCMC was recognized early ([Geman and Geman 1984](#)).

The seminal reference describing the application of MCMC to fitting Bayesian models is [Gelfand and Smith \(1990\)](#). Since its publication, MCMC has become the primary computational approach for Bayesian inference and has enabled the fitting of realistic, complex, high-dimensional Bayesian models to address research questions in diverse fields such as weather forecasting, climatology, psychometrics, and econometrics.

The Gibbs sampling algorithm is one particular way of constructing a transition kernel to produce a Markov chain with the desired target distribution. It is the method described in [Gelfand and Smith \(1990\)](#), and the algorithms used in WinBUGS ([Lunn et al. 2000](#)) and OpenBUGS ([Lunn et al. 2009](#)) are based on it. We will discuss the Gibbs sampling algorithm in detail in Chap. 9.

Using MCMC output for Bayesian inference places extra responsibility on the statistician. In particular, before basing posterior inference on MCMC samples, the statistician must attempt to assess three aspects of convergence:

1. At what time (or iteration) does convergence in distribution to the target distribution occur? There is no way to gauge this exactly, so we are seeking a point at which the iterates have become draws from a distribution close enough to the true target that our inference will be valid. The values from iterations preceding this point, commonly referred to as *burn-in*, need to be discarded. Then characteristics (sample means, medians, quantiles, etc.) of the remaining iterates for each model unknown are used to estimate the corresponding characteristics of the posterior marginal distributions of the unknown model quantities.
2. Do the post-burn-in samples that we have retained represent the entire support of the joint posterior distribution? Because MCMC samples exhibit serial dependence, it is possible for an MCMC chain to “get stuck” in one region of the parameter space and never visit some other parts. This risk is greatest when the target distribution is multimodal, with regions of very low posterior probability separating the modes.
3. After discarding the burn-in iterations, are there enough samples remaining to enable our estimation regarding the posterior to be as precise as we require?

In point of fact, only mathematical analysis can guarantee that a Markov chain has converged to its target distribution, and such analysis is generally prohibitively difficult for realistically complex models. This book presents practical procedures that are commonly used to attempt to assess MCMC convergence by running more than one chain for each model and examining the samples produced. While none of these methods offer guarantees, they can help to identify problems with MCMC samplers and provide some protection against invalid inference.

8.4 Introduction to OpenBUGS and WinBUGS

WinBUGS (Lunn et al. 2000) and OpenBUGS (Lunn et al. 2009) are the most widely used software packages for fitting Bayesian models using MCMC. Both grew out of the BUGS (Bayesian inference using Gibbs sampling) project that began in the early 1980s under David Spiegelhalter at the Medical Research Council Biostatistics Unit in Cambridge, UK.

WinBUGS and OpenBUGS enable the user to specify a Bayesian model (likelihood and prior(s)) in a simple language that resembles R. The software then uses artificial intelligence to devise the transition kernel for a Markov chain whose target distribution is the posterior that results from the user’s model specification. The user must input the number of Markov chains to run for the model and a set of initial values for each chain, and must request for how many iterations the chains are to be run. The software then generates the Markov chains. In addition, WinBUGS and OpenBUGS provide graphical and numeric functions for assessing convergence of the chains and summarizing posterior inference.

Both WinBUGS and OpenBUGS are, and always have been, freely available for download. However, WinBUGS is not open source. Development of OpenBUGS, an open-source version of WinBUGS, began in 2004. Today all development efforts are going into OpenBUGS, while WinBUGS remains available in its well-known and stable form. Downloads, documentation, examples, and discussion of WinBUGS are available from: <http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml>, while the same items for OpenBUGS are at: <http://www.openbugs.info/w/>. If you choose use WinBUGS, be sure to install the patch and the key from the download web page in order to obtain its full functionality.

WinBUGS and OpenBUGS are designed to run under Windows. However, they also run, complete with graphical user interface, under `wine` in Linux. A command-driven version (no graphics) runs natively in Linux and probably also on Macintosh.

This textbook shows examples run in OpenBUGS. The code, data, and initial values will work equally well in WinBUGS. The graphical user interfaces in WinBUGS and OpenBUGS are almost identical, with some differences of placement of items in pull-down menus. The differences between WinBUGS and OpenBUGS are likely to grow in the future, as more and more new capabilities are added to OpenBUGS.

The next sections provide an elementary introduction to OpenBUGS, by using it for simple models that we have already encountered. More detailed discussion of the convergence assessment and output analysis capabilities of OpenBUGS follows in the next chapter in the context of hierarchical models, for which these features are needed. Other introductory textbooks with many worked examples in WinBUGS are Congdon (2001, 2003) and Gill (2002).

8.4.1 Using OpenBUGS for the Problem of Estimating a Binomial Success Parameter

Figure 8.3 shows the graphical user interface in OpenBUGS, with the necessary code, data, and initial values to fit the survey-data problem from Chap. 3.

OpenBUGS has very extensive documentation available online as part of the software itself. Note the tabs for “Manuals” and “Help.” Under the “Examples” tab are several volumes of worked examples, complete with code, data, and detailed explanations.

To type a new model into OpenBUGS, click the “File” pull-down menu and choose “New.”

OpenBUGS requires three kinds of input from the user in order to fit a Bayesian model: the code specifying the model, the data, and initial values for the unknown random variables in the model (as many sets of initial values as there are Markov chains to be run).

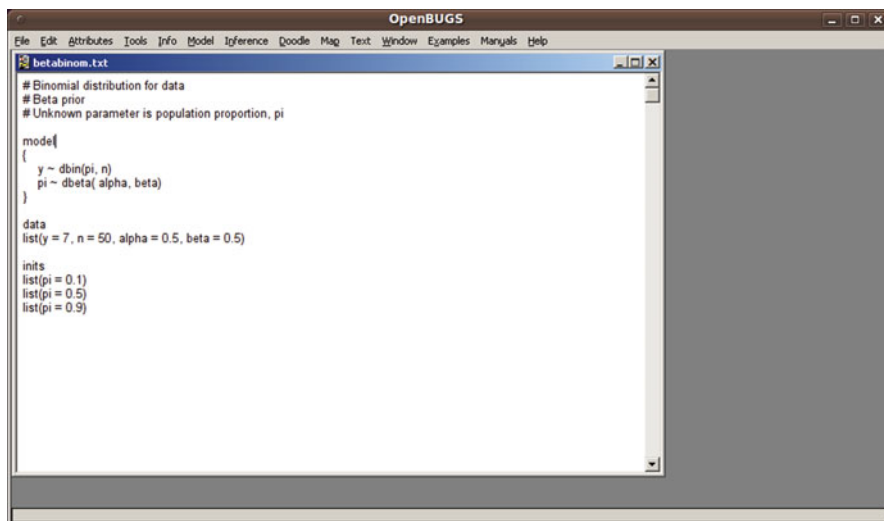


Fig. 8.3 OpenBUGS screenshot

8.4.2 Model Specification

The code for model specification must begin with the keyword `model`. The entire body of the code must be enclosed in curly braces. At minimum, the code must specify the likelihood (in the form of the distribution of the data values given the model) and the priors for all model parameters. The OpenBUGS symbol for “is distributed as” is the tilde, \sim . A list of all the probability mass functions and densities built into OpenBUGS is provided under the “Help” tab (the similar list in WinBUGS is inside the user manual).

For the survey problem, the complete model specification is:

```
model
{
  y ~ dbin(pi, n)
  pi ~ dbeta(alpha, beta)
}
```

8.4.3 Data and Initial Values Files

For OpenBUGS, not only the observed data values but all other constants must be provided as data. Data may be input to OpenBUGS either in list format or tabular format. The following is the data for our survey problem in list format. The keyword `list` is required, followed by a listing of the name and value of each quantity, enclosed in parentheses.

```
list(y = 7, n = 50, alpha = 0.5, beta = 0.5)
```

Which prior did I choose to use for the binomial success parameter π ? (Note that the parameters of the beta prior are given in the data list as 0.5 and 0.5.)

In addition to the data, the user must give OpenBUGS initial values with which to start off each Markov chain. For reasons detailed in the next chapter, I recommend always running several Markov chains for each model and starting them from very different initial values. For the example here, I want OpenBUGS to run three Markov chains. The only unknown quantity in the model is π , so the initial values will be provided in three separate lists, each containing a different value for π .

```
list(pi = 0.1)
list(pi = 0.5)
list(pi = 0.9)
```

8.4.3.1 MCMC Initial Values Are Not Like Priors!

Priors are part of the model specification. Priors must *not* be derived from the current dataset.

Initial values are part of the computing process. They do not contribute information about the model parameters. Basing the choice of initial values on a preliminary analysis of the current dataset is perfectly acceptable. For example, one possible way of choosing overdispersed initial values for MCMC chains for a Bayesian model is to first fit a frequentist model with the same likelihood as the Bayesian model. Then use the maximum likelihood estimates from the frequentist analysis as the initial values for one chain, the m.l.e.s minus three or four standard errors as the initial values for a second chain, and the m.l.e.s plus three or four standard errors as the initial values for a third chain.

8.4.4 Running the Model

The following steps are required to run a model in OpenBUGS or WinBUGS:

1. Use the mouse to highlight the word “model” at the beginning of the model section of your code. Then select the “Model” menu and from it select “Specification” and then “check model.” See Fig. 8.4. Watch for a message at the bottom of the OpenBUGS window either confirming the validity of the model or reporting errors as in Fig. 8.5.
2. Highlight the word “list” at the beginning of your data listing as in Fig. 8.6. From the “Specification tool” box select “load data.” Again check for a message confirming data loading or errors.
3. The number of parallel chains to be run must be set before compiling the model. In the “Specification tool” box, change the number of chains from 1 to 3 as in Fig. 8.7.

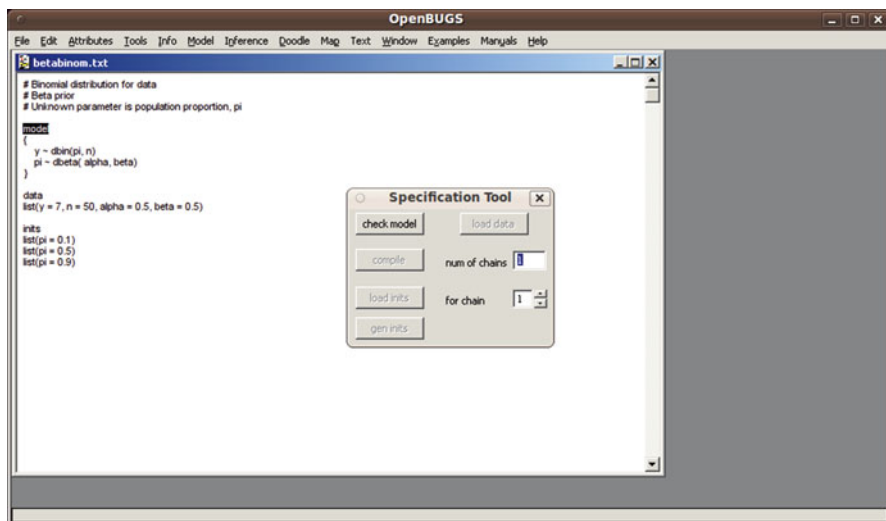


Fig. 8.4 Preparing to check the model

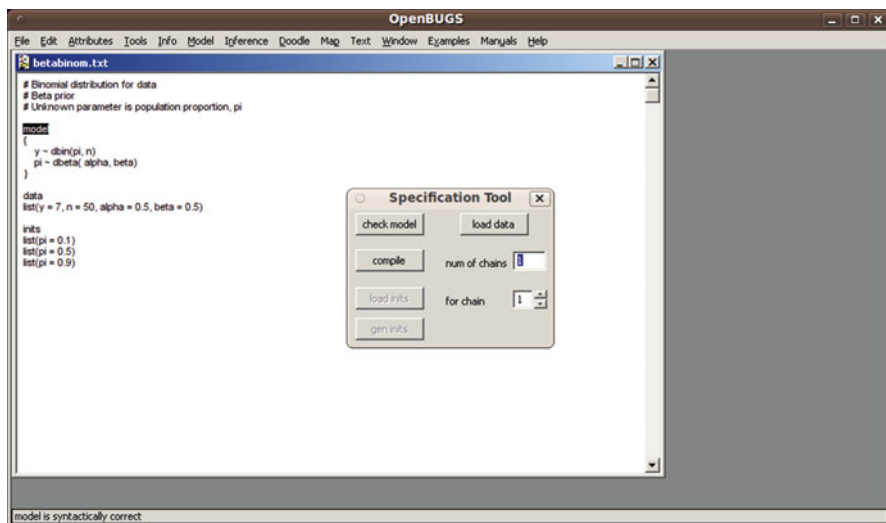


Fig. 8.5 Model is syntactically correct

4. In the “Specification tool” box, click “Compile.” This is the point at which OpenBUGS evaluates whether your model is logically sound and whether it can find a way to construct a Markov chain with the joint posterior distribution of the model unknowns as its target distribution. In Fig. 8.8, the model has compiled successfully. Check for the message at the bottom left of the OpenBUGS window.

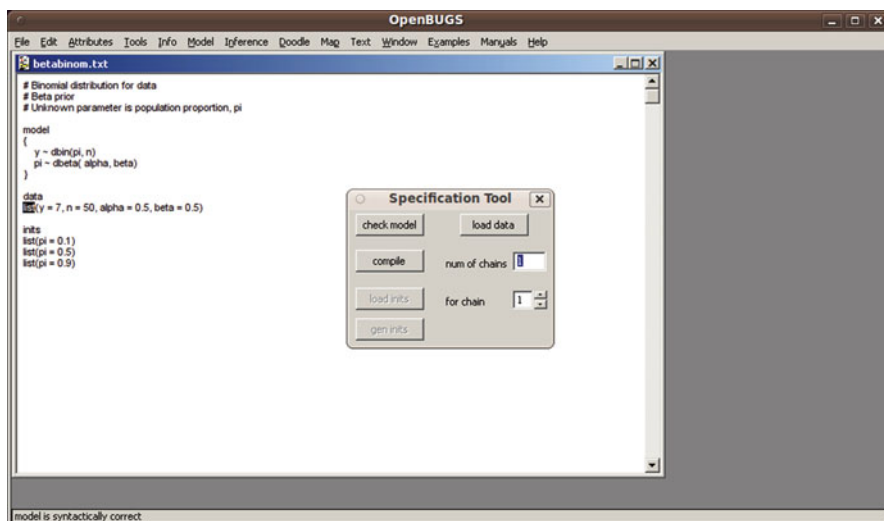
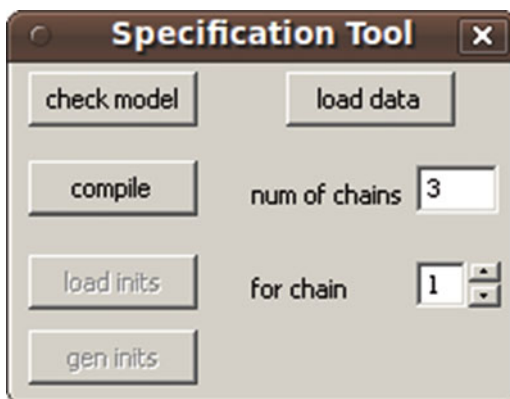


Fig. 8.6 Preparing to load the data

Fig. 8.7 Setting the number of chains



- Highlight the word “list” at the beginning of your initial values section. Select the “Model” menu and from it select “load inits.”

Again check for a message. You will get a message that some nodes are uninitialized, as in Fig. 8.9. Continue to load initial values for each of the other two chains until all are initialized (Fig. 8.10).

In this model, there is only one unknown parameter. In more complex models with many parameters, you may not wish to specify initial values for all of them. OpenBUGS and WinBUGS are capable of automatically generating initial values for some parameters. The user requests this by clicking the “Gen inits” button if the message about uninitialized nodes remains after the user-provided initial values have been loaded for all chains. Since OpenBUGS and WinBUGS generate initial values by drawing from the prior densities specified for the

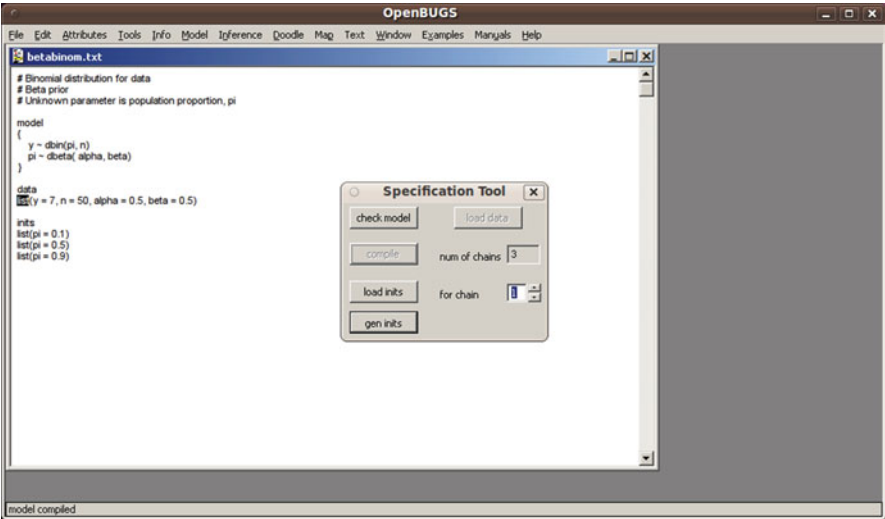


Fig. 8.8 Compiling

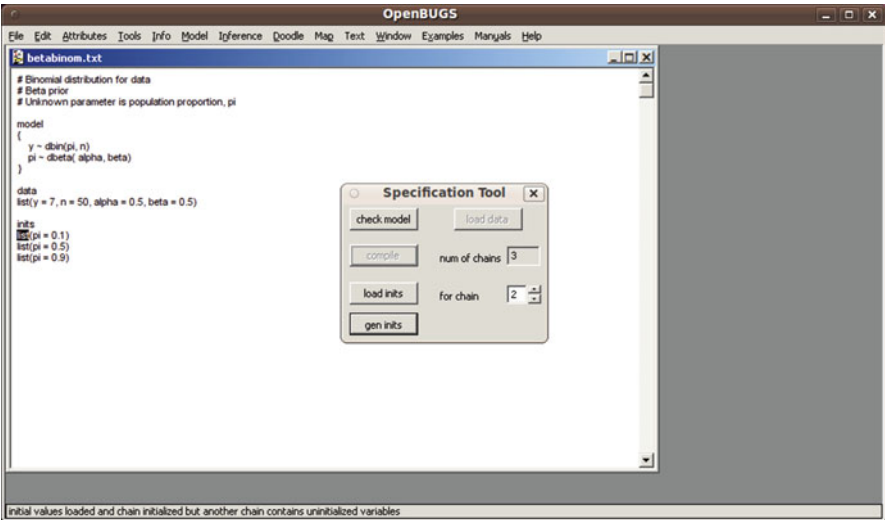


Fig. 8.9 Initial values for chain 1 loaded

- parameters, this cannot be done if the priors are improper or extremely vague. Furthermore, OpenBUGS and WinBUGS are *not* capable of auto-generating initial values for precision parameters.
- By default, OpenBUGS/WinBUGS does not save the values of the MCMC iterates that it generates. The user must specify for which model unknowns the sampled values should be stored and reported. To do this, select the “Inference”

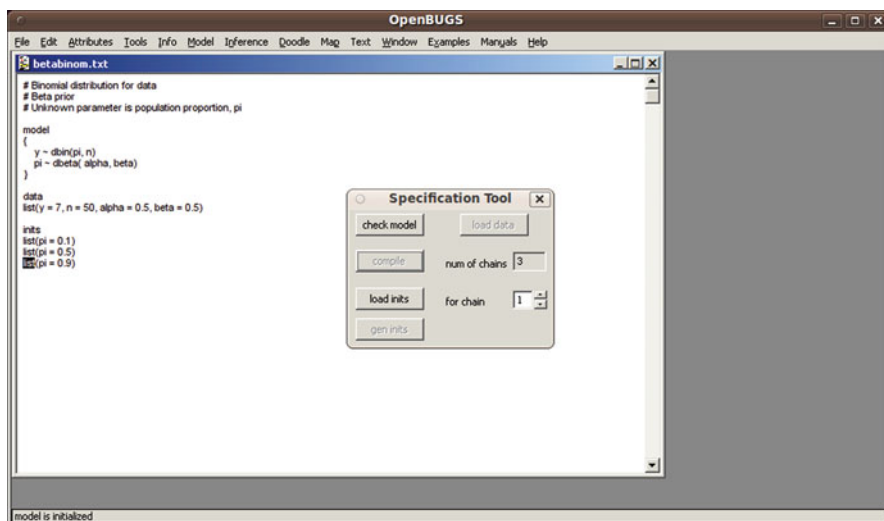


Fig. 8.10 All initial values loaded

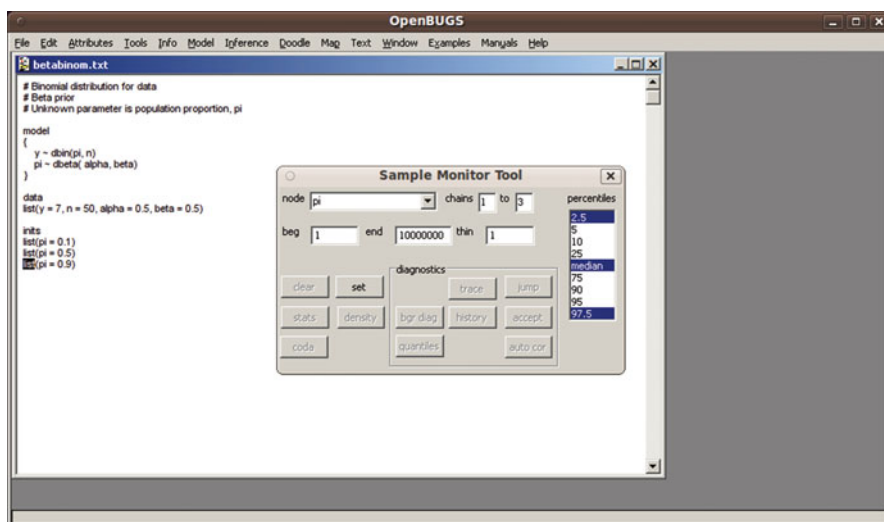


Fig. 8.11 Turning on sample monitors

menu and “Samples” from it. In the window in the prompt box, type the name of each parameter whose posterior distribution you want to study (this will be just π_i in this simple example), and click on “set” after each one. See Fig. 8.11.

- At last we can actually run the chains! Select the “Model” menu and “Update.” You will be prompted for how many iterations you want to run the sampler. For now, accept the default of 1,000, and click “Update.” OpenBUGS will report that it has completed the updates and the amount of time they took (Fig. 8.12).

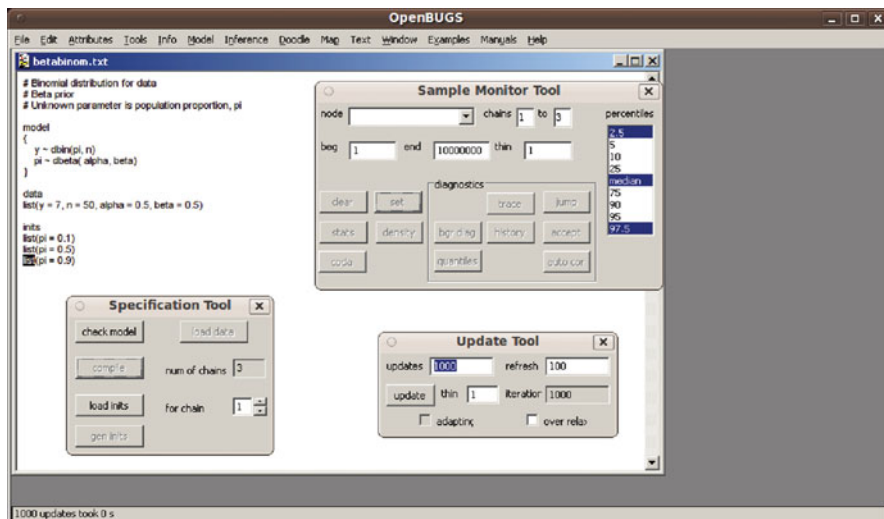


Fig. 8.12 Updating

8. Before inspecting the OpenBUGS output, go to the Model menu, and under the “Input/Output options” submenu, change the setting from “window” to “log.” This will cause all the output we are about to request to go into a single window instead of creating a bazillion small windows cluttering up the screen. (In WinBUGS, the choice of output to window or log appears in a separate pull-down menu called “Options.”)
9. Now we will use OpenBUGS graphical and numeric facilities for assessing MCMC sampler convergence. Go back to the “Sample monitor” box and select the desired parameter in the node box. Entering an asterisk requests all monitored nodes. Then, one at a time, click “history,” “autocorr,” and “GRdiag.” The output is described and interpreted in Sect. 8.4.5.

8.4.5 Assessing Convergence in OpenBUGS

The simple one-parameter model that we have fit results in an MCMC sampler with no convergence problems whatsoever. Therefore, in this section, you get to see the ideal in each convergence assessment. You rarely will see MCMC output plots as perfect as these. These may be used as a benchmark with which to compare the MCMC output from more complex models.

8.4.5.1 History or Trace Plots

History plots are one of the oldest methods of qualitatively assessing MCMC sampler performance. The MCMC iteration number is on the x-axis and the value of

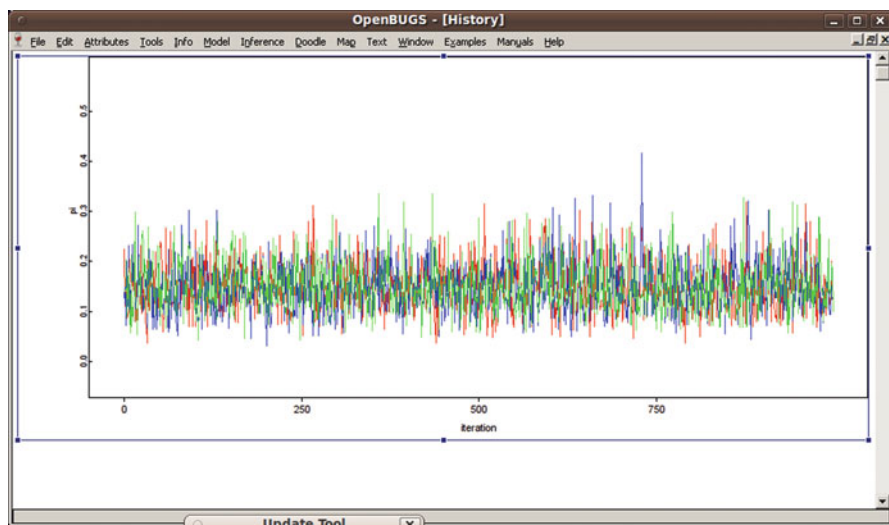


Fig. 8.13 OpenBUGS history plot

the parameter drawn at each iteration is on the y-axis. Successive values are joined by a line. OpenBUGS produces a separate history plot for each parameter. When more than one chain has been run, the lines from all chains are plotted in different colors in the same panel.

Figure 8.13 shows the history plot for the parameter π_i from three chains run for 1,000 iterations. Despite the fact that we started the three chains from widely spread initial values, they immediately settle into drawing from the same range of values. Furthermore, they look like white noise—just random noodling around in that range, without any consistent pattern. These are hallmarks of rapid MCMC convergence.

We can see the white-noise phenomenon more clearly by restricting the history plot to output from a single chain. To obtain this plot, with output only from chain 1, return to the Sample monitor tool. In the section that says “chain 1 to 3,” change the 3 to a 1. Then click “History.” The resulting plot is in Fig. 8.14.

8.4.5.2 Autocorrelation Plots

We have mentioned that there usually is dependence in the samples produced by a Markov chain. *Autocorrelation* is a quantitative measure of this dependence. You should be familiar with correlation as a measure of association between two quantitative variables measured on the same subjects. For example, we might calculate the correlation between weights and heights of a sample of people. As you know, the prefix “auto” means “self,” so autocorrelation literally is correlation with self. Like any correlation, autocorrelation values must fall between -1 (signifying perfect negative correlation) and $+1$.

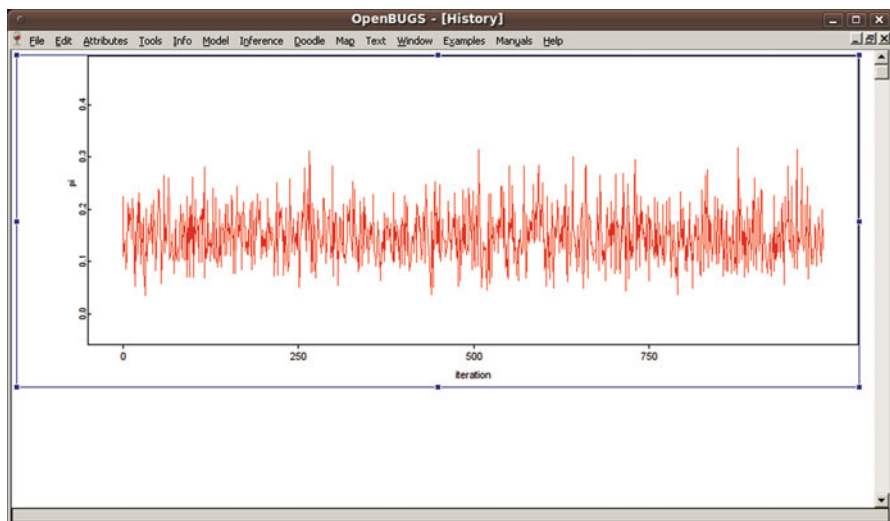


Fig. 8.14 History plot from first chain

Lag 1 autocorrelation in MCMC output is the correlation between samples from the same chain drawn 1 iteration apart. In our example, in which samples are drawn from the posterior density of π , we might call the values drawn from successive iterations π_1 , π_2 , π_3 , and so on. Then to calculate the lag 1 autocorrelation, we would pair up π_1 with π_2 , π_2 with π_3 , π_3 with π_4 , etc. Similarly, the lag k autocorrelation is the correlation between samples drawn k iterations apart.

In an autocorrelation plot, lags are on the x-axis and the height of each bar represents the magnitude of the autocorrelation at that lag. The first bar—for lag 0—always has a height of 1 (think about why this has to be true); it provides a visual scale with which to compare the heights of the remaining bars.

Usually in MCMC output, the lag 1 autocorrelation is positive, and the autocorrelation decreases as the lag increases until it reaches a threshold lag beyond which it is essentially 0. A Markov chain in which autocorrelation is large at lag 1 and decays slowly as the lag increases is said to be mix slowly, and it will converge slowly in all three senses: It will take a long time to find its stationary distribution; once in the stationary distribution, it will take many iterations to explore the entire support of the distribution; and a very large number of iterates will be needed in order to obtain usefully precise estimation of the characteristics of the posterior distribution.

Our simple one-parameter model causes none of these issues for MCMC. Like the history plots, the OpenBUGS plot of autocorrelation (Fig. 8.15 for chain 1) for this model shows the ideal—that is, from lag 1 on, the autocorrelation is arbitrarily close to zero. This is the pattern that we always wish to see in MCMC output but rarely do. It is consistent with the white noise that we saw in the history plots.

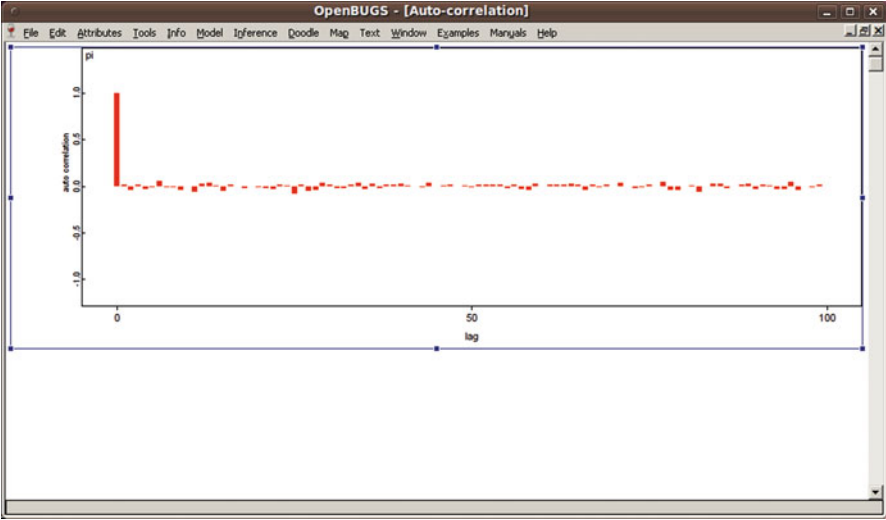


Fig. 8.15 OpenBUGS autocorrelation plot from the first chain

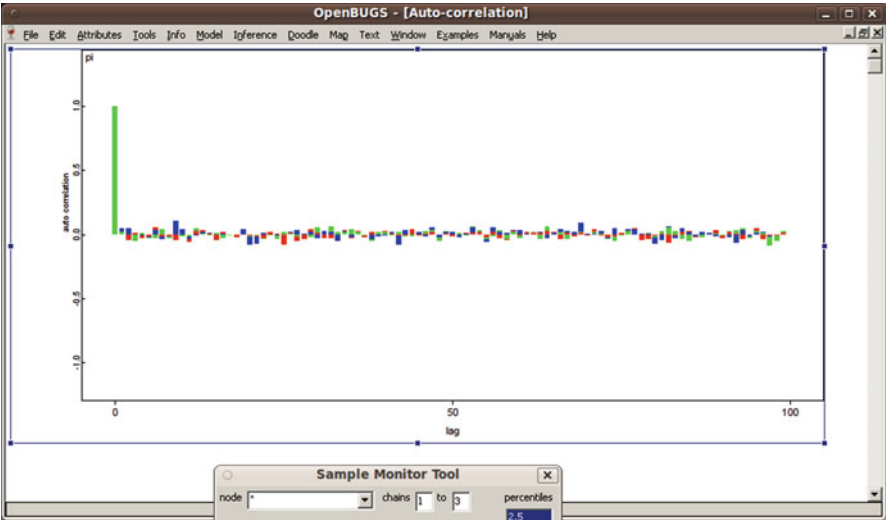


Fig. 8.16 OpenBUGS autocorrelation plot from all three chains

Figure 8.16 shows the autocorrelation plots for all three chains superimposed. It was obtained by changing the setting in the Sample monitor tool back to “chains 1 to 3” and then clicking the “Autocorr” button.

8.4.5.3 The Brooks, Gelman, and Rubin Diagnostic

In the early 1990s, when MCMC was becoming known as a computational method for fitting Bayesian models, a number of statistical methodologists sought to develop

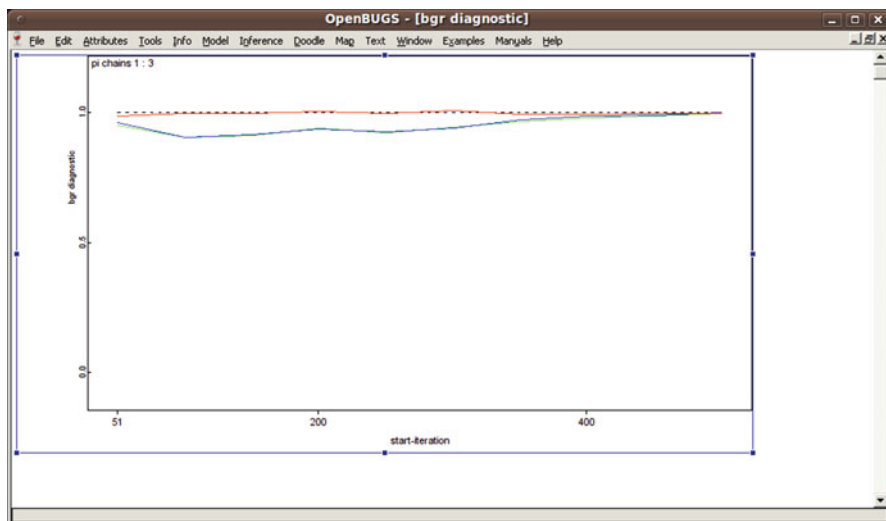


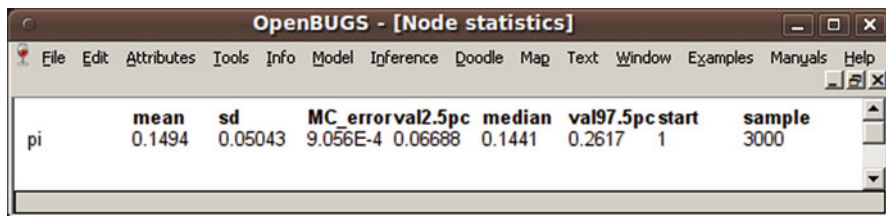
Fig. 8.17 OpenBUGS Brooks, Gelman, and Rubin diagnostic plot

quantitative methods that could be applied to the output of MCMC samplers to try to determine how many initial iterations needed to be discarded as burn-in. [Gelman and Rubin \(1992\)](#) proposed a diagnostic that required that two or more parallel chains be run from very different initial values. [Brooks and Gelman \(1998\)](#) corrected and generalized the Gelman and Rubin diagnostic. WinBUGS and OpenBUGS incorporate one of Brooks and Gelman’s diagnostics, which may be obtained with the “bgrdiag” button on the Samples monitor tool.

The BGR diagnostic is not helpful for models like our one-parameter beta-binomial, for which MCMC samplers converge instantly. It will be discussed in much more detail in Chap. 9, in the context of hierarchical models for which it can provide very useful insight. For completeness, the plot of the BGR diagnostic for the beta-binomial example is shown in Fig. 8.17.

8.4.5.4 The Monte Carlo Error (MC error)

The table of summary statistics of the MCMC output is produced with the “Stats” button on the Sample Monitor tool. Before using this table for inference, we need to use it for one more convergence check. The column called `MC_error` refers to the *Monte Carlo error*, which is a measure of the performance of the Markov chain (*not* an estimate of a characteristic of the posterior distribution!). Note that the first column in the table, labeled `mean`, is the estimated mean of the posterior distribution of the parameter, with the estimation based on the MCMC samples. The MC error measures the amount of uncertainty in this sampling-based estimate of the posterior mean. The MC error is much like the standard error of the mean,



	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
pi	0.1494	0.05043	9.056E-4	0.06688	0.1441	0.2617	1	3000

Fig. 8.18 OpenBUGS node statistics table

which you should be familiar with from frequentist statistics, except the MC error is adjusted for the autocorrelation in the MCMC sampler output. The interpretation of MC error is that the estimate of the posterior mean probably is accurate to within about \pm twice the MC error. If a more precise estimate of the posterior mean is needed, then the MCMC chains need to be run for more iterations.

A rule of thumb (stated, e.g., in the user manual included with OpenBUGS) is that an MCMC sampler should be run long enough after convergence that the MC error is less than $\frac{1}{20}$ as large as the estimated posterior standard deviation of the parameter. The estimate of the posterior standard deviation is in the *sd* column. You may need greater precision for a particular inferential purpose.

In our example (Fig. 8.18), the MC error is given as 9.056E–4, which is 0.0009056. This is only about 1/50th as large as the estimated standard deviation (0.0504). Thus, the rule of thumb is satisfied.

8.4.6 Posterior Inference Using OpenBUGS

We saw no evidence that our OpenBUGS chains failed to converge, so we can proceed to use their output for Bayesian inference.

As already mentioned, the table of node statistics exemplified in Fig. 8.18 provides estimates of the posterior mean and standard deviation. It also shows estimates of the posterior median and the endpoints of the 95% credible set, based on the 0.5, 0.025, and 0.975 empirical quantiles of the MCMC samples.

Let's check the performance of an OpenBUGS sampler by comparing the results in Fig. 8.18 to the exact analytic results for this model. The prior density on π was $Beta(0.5, 0.5)$, and the data was 7 successes in 50 trials. Thus, the exact posterior density is $Beta(7.5, 43.5)$. Table 8.1 compares the OpenBUGS sampling-based estimates with the corresponding characteristics of the exact beta posterior density.

The sampling-based approximations are accurate to two digits to the right of the decimal point. The estimate of the posterior mean is off by about twice the MC error (0.002), so that seems to be working correctly.

Table 8.1 Comparison of exact and OpenBUGS estimates based on 1,000 iterations

	Exact	OpenBUGS
Mean	0.147	0.149
sd	0.049	0.050
Median	0.142	0.144
95% credible set	(0.065, 0.255)	(0.067, 0.262)

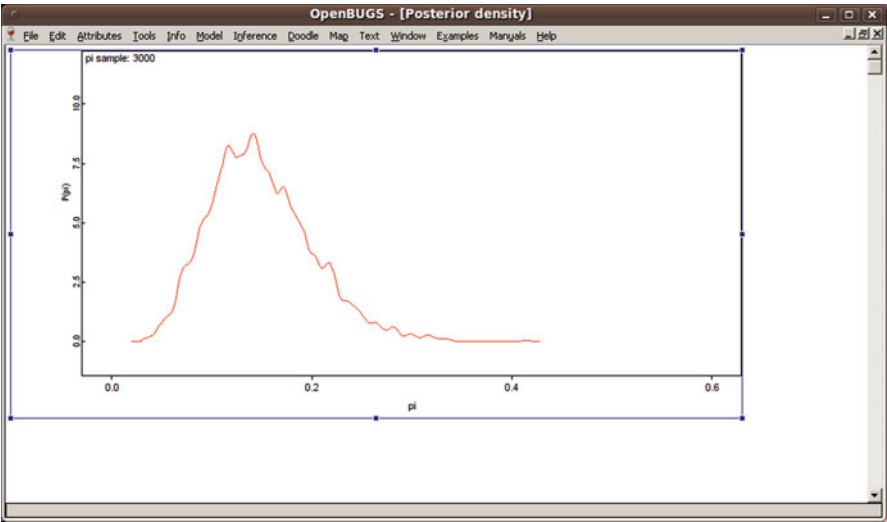
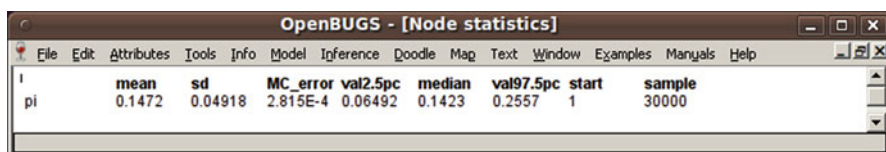


Fig. 8.19 Plot of estimated posterior density of pi

OpenBUGS and WinBUGS can also produce plots of the approximate marginal posterior density of each monitored parameter. These plots—essentially smoothed histograms of the MCMC samples—are obtained by clicking the “density” button on the Sample monitor tool. With 3,000 total samples (1,000 from each chain), the density plot in Fig. 8.19 is rough, but the right-skewed shape is clear.

8.4.6.1 More Precise Estimation from MCMC

To get more precision in posterior estimation (and a smoother density plot), we may return to the “Update” tool and request additional samples. The table of node statistics in Fig. 8.20 and the density plot in Fig. 8.21 were produced after 9,000 more iterations were run. Before examining these results, let’s think about what to expect. The posterior mean, standard deviation, and quantiles are characteristics of the posterior distribution. We determined that they had been estimated reasonably well based on the pooled first 1,000 samples from the three chains. Thus, the estimates based on pooling 10,000 samples from each chain probably will not be substantially different, although they are likely to be slightly closer to the true values. On the other hand, the MC error is a measure of the uncertainty in the



	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
pi	0.1472	0.04918	2.815E-4	0.06492	0.1423	0.2557	1	30000

Fig. 8.20 Node statistics from 10,000 iterations in three chains

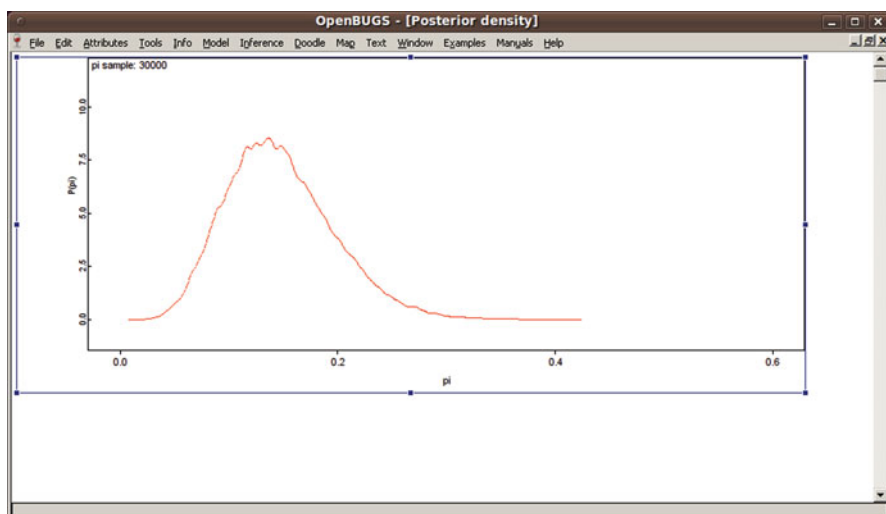


Fig. 8.21 Plot of estimated posterior density of π based on 10,000 MCMC samples from three chains

MCMC-based estimate of the posterior mean, and it should decrease systematically with increasing numbers of iterations—it should be roughly proportional to one over the square root of the number of MCMC iterations. Thus, we should expect the MC error to be only about 1/3 as large in the table in Fig. 8.20 as in Fig. 8.18. And the plot of the estimated posterior density should be smoother when it is based on more samples.

To print the content of any window, click on that window and then select the “File” menu and “Print.” If you wish, you may copy and paste graphical and tabular output from the “Sample monitor” windows into a single window for compact printing.

8.4.7 *OpenBUGS for Normal Models*

We will conclude this first introduction to OpenBUGS with code for performing posterior inference and prediction in models with a normal likelihood.

First, we will use OpenBUGS to work the example from Sect. 6.2.8, in which we sought to estimate the population mean of the log of mercury contamination in fish from the Des Moines River. Assuming that the population distribution of log mercury concentration was normal and that the precision was known to be 2.5 log units, we specified a normal prior on μ and performed Bayesian inference. Model 1 below expresses this model. Note that WinBUGS and OpenBUGS parameterize the normal density in terms of the mean and the *precision*.

There are 21 observations in the dataset. In order to estimate the posterior predictive distribution for a future observation from the same population, I have added an “NA” to the end of the data list. Like R, OpenBUGS and WinBUGS use “NA” to denote a missing value. They will treat this missing data value as another unknown quantity in the model and will draw a value for it at each iteration of the sampler. These draws are draws from the posterior predictive density. The OpenBUGS/WinBUGS node statistics for the missing data node (in this example `y[22]`) estimate the characteristics of the posterior predictive distribution.

Notice the *for loop* in Model 1 below. Since N is assigned as the constant value of 22 in the data list, the line `y[i] ~ dnorm(mu, tausq)` inside the curly braces says that each of the 22 y s (including the missing value) is a draw from the normal density with mean μ and precision tausq . Since tausq is given a constant value in the data list, it is being treated as a known parameter.

```
# Model 1
# Assuming data are draws from a normal population
# with known precision
# Population mean mu is unknown parameter
# We can also estimate the posterior predictive
# distribution
# by monitoring y[22]

model
{
  # likelihood
  for (i in 1:N) {
    y[i] ~ dnorm( mu, tausq )
  }
  # priors
  mu ~ dnorm(-2.75, 7.5)
}

#data
list(y=c(-2.526, -1.715, -1.427, -2.12, -2.659,
          -2.408, -3.219, -1.966,
          -2.526, -1.833, -2.813, -1.772, -2.813, -2.526,
          -3.219, -2.526,
```

```
-2.813, -2.526, -3.507, -2.996, -3.912, NA), N=22,
tausq= 2.5)
```

```
#inits for model 1
list(mu = -5)
list(mu = -2.5)
list(mu = 0)
```

In Model 2, below, both `mu` and `tausq` are treated as unknown parameters, and a close approximation to the improper noninformative joint prior from Sect. 7.1 is specified. The `dflat()` prior is the improper prior that is proportional to a constant over the whole real line. Because OpenBUGS and WinBUGS do not allow the improper Gamma prior with both parameters approaching 0, a common practice is to approximate it with a Gamma prior with very small values of both parameters, as was done her.

Note the line `sigmasq <- 1\tausq`. The symbol `<-` in OpenBUGS and WinBUGS indicates deterministic calculation. When running the MCMC sampler for this model, at each iteration, OpenBUGS will draw a new value of the precision `tausq` and then will calculate its inverse. By monitoring the node `sigmasq`, the user can obtain an estimate of the posterior marginal distribution of the population variance. This is an illustration of a very powerful feature of the use of MCMC for Bayesian model fitting—the ability to obtain estimates of the posterior marginal distribution of any function of model parameters, simply by calculating such functions at each iteration of the sampler.

Model 2 also illustrates an alternate way of providing data to WinBUGS and OpenBUGS. Vector and matrix data may be entered in a columnar format, the first row of which provides the variable names followed by square brackets to indicate that they are vectors. The last row in the table of data must be the keyword `END` in all capitals, followed by a carriage return (the invisible character obtained by pressing the Enter key.)

The data for Model 2 must be loaded in two steps: The list component is loaded as usual by highlighting the keyword `list` and then clicking “Load data.” Then the row of column headings for the tabular data (in this case, just `y[]`) is highlighted, and “Load data” is clicked again.

```
# Normal sampling density for data
# Both mu and tausq unknown

# model 2
model
{
  # likelihood
  for (i in 1:N) {
    y[i] ~ dnorm( mu, tausq )
  }
}
```

```

# priors
mu ~ dflat()
tausq ~ dgamma( 0.0001, 0.0001)
sigmasq <- 1/tausq
}

```

Here is a different way to give data to WinBUGS.

```

data
list(N = 22 )

```

```

additional data

```

```

y[]

```

```

-2.526

```

```

-1.715

```

```

-1.427

```

```

-2.12

```

```

-2.659

```

```

-2.408

```

```

-3.219

```

```

-1.966

```

```

-2.526

```

```

-1.833

```

```

-2.813

```

```

-1.772

```

```

-2.813

```

```

-2.526

```

```

-3.219

```

```

-2.526

```

```

-2.813

```

```

-2.526

```

```

-3.507

```

```

-2.996

```

```

-3.912

```

```

NA

```

```

END

```

```

inits for model 2

```

```

list(mu = 0, tausq = 1)

```

```

list(mu = 20, tausq = 100)

```

```

list(mu = 40, tausq = 1000)

```

Model 3 below shows how to code the conjugate prior from Sect. 7.3 in OpenBUGS. WinBUGS and OpenBUGS syntax does not permit doing calculations as parameters in density functions. Thus we cannot write:

```
mu ~ dnorm( -2.75, 3 * tausq)
```

Instead we have to do the calculation first and put the named result into the density function, like this:

```
tausq0 <- 3 * tausq
mu ~ dnorm( -2.75, tausq0)

# Normal sampling density for data
# Both mu and tausq unknown
# Conjugate joint prior

# model 3
model
{
  # likelihood
  for (i in 1:N) {
    y[i] ~ dnorm( mu, tausq )
  }
  # priors
  tausq0 <- 3 * tausq
  mu ~ dnorm( -2.75, tausq0)
  tausq ~ dgamma( 13.3, 5.35)
  sigmasq <- 1/tausq
}
```

8.5 Exercises

8.1. Section 8.1.3 includes a function for using Monte Carlo integration to approximate the integral of the unnormalized posterior in the example problem with a binomial likelihood and a histogram prior. Write an R function to approximate the posterior mean of π in this example using Monte Carlo integration. Compare your results with those obtained by numeric integration.

8.2. Go through the steps to use OpenBUGS for Model 1 for the fish mercury data. Note that, since $y[22]$ is an unknown quantity in the model, it needs an initial value. The easiest approach is to leave the initial values lists as they are, and then, after loading the initial values for the third chain, to click “Gen inits” to have OpenBUGS generate its own initial values for $y[22]$.

Compare your results to those obtained in Sect. 6.2.8.3.

8.3. Go through the steps to use OpenBUGS for Model 3 for the fish mercury data. Compare your results to those obtained in Sect. 7.1.1.

8.4. This problem is a continuation of Problem 6.2. Now use OpenBUGS or WinBUGS to carry out the analysis. You will have to specify the model in terms of the precision.

1. What is the conjugate family of prior distributions for a normal precision when the mean is known? (You will then use the same parameters in this prior as you used for the prior on the variance in Problem 6.3.)
2. Include the computation of the variance in your OpenBUGS/WinBUGS program.
3. Compare the posterior mean and variance obtained by OpenBUGS/WinBUGS for the variance with what you obtained analytically.

Chapter 9

Hierarchical Models and More on Convergence Assessment

So far, all of the Bayesian models that we have encountered have had only two components—the likelihood, which describes the data as draws from a probability distribution, and the prior, which specifies a probability distribution on the unknown parameters in the likelihood. Such a simple model is inadequate for many (probably most) real-world applications. As a result, more complex Bayesian models with additional levels are very commonly used. Such models are called *hierarchical* models.

Sometimes hierarchical models are needed because the structure of the data itself is a hierarchy. For example, consider children’s scores on a standardized test taken by some third graders across the USA. The individual children are in classrooms; the classrooms are in schools; the schools are in school districts; the school districts are in states, etc. In this case, a hierarchical model would enable us to estimate parameters at each level of the hierarchy so as to address questions such as: How variable are average test scores in different schools in the same school district? How variable are average test scores in different school districts?

More generally, hierarchical models are appropriate when there are natural groupings of observations in the data or of parameters in the model.

Early references on Bayesian hierarchical models are [Box and Tiao \(1973\)](#); [Good \(1965\)](#). Chapter 5 of [Gelman et al. \(2004\)](#) offers an up-to-date discussion, including computational aspects.

9.1 Specifying Bayesian Hierarchical Models Example: A Better Model for the College Softball Player’s Batting Average

In problems at the end of Chap. 3, you estimated a college softball player’s college-career batting average from her number of hits in 30 at bats that occurred during eight games. In answering Problem 3.3, you probably noted that considering each at

bat an independent Bernoulli trial was not justified, because the outcomes of at bats within a single game were likely to be correlated (since within the same game, the player was facing the same opposing team, playing in the same stadium, etc.)

Now we will use a hierarchical model to estimate the player's overall batting average while appropriately accounting for the correlation structure of the data. Specifically, we will assume that the at bats *within* each game are *exchangeable*—that is, that their outcomes are conditionally independent given the player's success probability *for that game*. Furthermore, we will assume that the player's probability π_i of getting a hit could be different in different games, $i = 1, \dots, 8$, but that all the π_i 's are random draws from the same probability distribution, with mean the true overall batting average that is of primary interest. We also have secondary interest in estimating the π_i 's themselves.

To fit a model reflecting these assumptions, we need the numbers of hits and at bats from the individual games, which are:

Game (i)	at bats (n_i)	hits (y_i)
1	1	5
2	0	4
3	1	3
4	1	5
5	1	3
6	0	4
7	1	4
8	0	2

9.1.1 The First Stage: The Likelihood

The levels of a Bayesian model are called *stages*. Regardless of the model's complexity, the first stage always is the likelihood—the stage that is closest to the data.

With the assumptions stated above for the softball player example, the likelihood will be the product of eight binomials:

$$p(y_1, y_2, \dots, y_8 | \pi_1, \pi_2, \dots, \pi_8) = \prod_{i=1}^8 \left[\binom{n_i}{y_i} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i} \right]$$

We can write the likelihood as a product because, *given the 8 parameters π_i* , the y_i 's are conditionally independent.

The y_i 's are not exchangeable however. Remember that the formal definition of exchangeability is “invariance to permutations of the indices.” In this example, that would mean that the numeric evaluation of the likelihood would not be changed if we decided to swap the subscripts on some of the data values—for example, if

we decided to rename the pair (n_6, y_6) as (n_3, y_3) and vice versa without making the same swap of subscripts of π_6 and π_2 . This permuted likelihood would include the following to two terms (using the original subscript numbering):

$$\binom{n_6}{y_6} \pi_3^{y_6} (1 - \pi_3)^{n_6 - y_6}$$

and

$$\binom{n_3}{y_3} \pi_6^{y_3} (1 - \pi_6)^{n_3 - y_3}$$

Since data pairs here are matched up with the wrong π , the likelihood evaluation would be different from in the unpermuted version.

9.1.2 *The Second Stage: Priors on the Parameters That Appeared in the Likelihood*

The second stage of a hierarchical model consists of probability distributions for some or all of the parameters that appeared in the likelihood. The important difference between the second stage of a hierarchical model and the second (and final) stage of the two-stage models we have studied previously is that the parameters of the priors at the second stage of a hierarchical model are more unknown parameters.

The softball player example illustrates why we do this. Recall that we want to express the notion that the π_i 's from different games are not all equal but that they probably aren't radically different (after all, they all pertain to the same player) by modeling them as random draws from the same probability distribution. The beta family is a natural choice of that probability distribution, since it would be the conjugate family in a two-stage model. With this choice, the second-stage prior is:

$$\pi_i \sim \text{Beta}(\alpha, \beta), \quad i = 1, \dots, 8$$

Note that the mean of this beta distribution (let's call it μ) is the mean of the π_i 's from all possible college games that this player will ever play—in other words, it's the unknown quantity we are trying to estimate using the data at hand. Now if we filled in fixed, known numbers for the parameters α and β , we would be claiming that we already knew the mean of the beta density—it would just be $\frac{\alpha}{\alpha + \beta}$. So we must treat α and β as unknown model parameters—which means that they in turn must have prior distributions. Since μ is a deterministic function of α and β , the priors on α and β will induce a prior on μ as well.

9.1.3 *The Third Stage: Priors on Any Parameters That Do Not Already Have Them*

In a three-stage hierarchical model, the third (and last) stage specifies prior distributions, with fixed numeric values of their parameters, for the remaining model unknowns.

For the softball player problem, the third stage consists of priors for α and β . There is no semi-conjugate family for the parameters of beta densities. Therefore, we will just choose a familiar family that has the correct support. As the parameters of a beta density, both α and β must be strictly positive. Continuous densities with support on the positive real line include gamma, inverse gamma, exponential, and chi-square. The last two are just special cases of gamma densities. Let's place gamma priors on both α and β . They must be proper densities in order for the whole joint posterior to be proper, but we don't want them to be very informative, as we would like the data to dominate our inference. Specifying:

$$\begin{aligned}\alpha &\sim \text{Exp}(1) \\ \beta &\sim \text{Exp}(0.33)\end{aligned}$$

gives α a prior mean of 1 and β a prior mean of 3 and therefore suggests that μ may be in the vicinity of 0.25.

9.1.4 *The Joint Posterior Distribution in Hierarchical Models*

Bayes' rule is the same for hierarchical models as for simpler Bayesian models: the posterior is proportional to the prior times the likelihood. To write an expression that is proportional to the joint posterior distribution in a hierarchical model, we must take the product of the likelihood and all the priors from all stages of the model.

In our hierarchical softball example, the procedure looks like this. The likelihood is:

$$\prod_{i=1}^8 \left[\binom{n_i}{y_i} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i} \right]$$

The second-stage prior is (note that we *cannot* drop the normalizing constant for the beta density here because α and β are not constants—they are unknown model parameters):

$$\prod_{i=1}^8 \left[\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \pi_i^{\alpha-1} (1 - \pi_i)^{\beta-1} \right]$$

and the third-stage prior is proportional to (we can drop the normalizing constants here because they are indeed fixed, known numbers):

$$\exp(-\alpha) \exp(-0.33\beta)$$

Multiplying everything together (and being very careful about which terms are inside the product), we get the following for the joint posterior density:

$$\begin{aligned} p(\pi, \alpha, \beta | \mathbf{y}) &\propto \prod_{i=1}^8 \left[\binom{n_i}{y_i} \pi_i^{y_i} (1 - \pi_i)^{n_i - y_i} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \pi_i^{\alpha-1} (1 - \pi_i)^{\beta-1} \right] \\ &\quad \exp(-\alpha) \exp(-0.33\beta) \\ &= \left[\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right]^8 \prod_{i=1}^8 \left[\pi_i^{y_i + \alpha - 1} (1 - \pi_i)^{n_i - y_i + \beta - 1} \right] \exp(-\alpha) \\ &\quad \exp(-0.33\beta) \end{aligned} \tag{9.1}$$

9.1.5 Higher-Order Hierarchical Models

For some applications, hierarchical models with more than three stages are needed. In these cases, the third stage will specify probability distributions as priors for the parameters of the second stage, but again will leave the parameters of these third-stage priors as unknowns to be estimated. It is possible to specify higher and higher stages of Bayesian models. For example, the test-scores data described in the introduction to this chapter could well require at least a five-stage model. Regardless of the number of stages, all unknowns in the model must be given priors, and the parameters of the priors at the final stage of the model must be fixed numeric values.

9.2 Fitting Bayesian Hierarchical Models

At the time of this writing, Markov chain Monte Carlo methods are almost universally used for fitting Bayesian hierarchical models. It rarely is feasible to determine analytically the marginal posterior distributions of the parameters of inferential interest in complex hierarchical models. Consequently, we use MCMC to draw samples from the joint posterior distribution.

WinBUGS/OpenBUGS code, data, and initial values for the hierarchical version of the softball player example are as follows:

```

model
{
  for (i in 1:N) {
    y[i] ~ dbin( pi[i], n[i] )    # likelihood
    pi[i] ~ dbeta( alpha, beta ) # second-stage prior
  }

  alpha ~ dgamma(1.0, 1.0)        # third-stage prior
  beta  ~ dgamma(1.0, 0.33 )      # third-stage prior

  mu <- alpha /( alpha + beta)    # calculate quantity
                                  # of interest
                                  # (function of
                                  # parameters)

}

#data
list( y = c( 1,0,1,1,1,0,1,0), n = c(5,4,3,5,3,4,4,2),
      N = 8)

# overdispersed initial values for 3 chains
list( alpha = 1, beta = 1, pi = c(.5, .5, .5, .5, .5, .
    .5, .5, .5) )

list(alpha = 100, beta = 10, pi = c(.9, .9, .9, .9,
    .9, .9, .9, .9) )

list(alpha = 10, beta = 100, pi = c(.1, .1, .1, .1,
    .1, .1, .1, .1))

```

Note that since there are eight games, each with its own data value y_i and success probability π_i , both the likelihood specification for the y 's and the second stage prior on the π s are stated inside a loop. On the other hand, since there is only one each of α and β , their third-stage prior specifications must be outside the loop.

Recall that the `<-` symbol in WinBUGS/OpenBUGS represents deterministic calculation. At every iteration of the Gibbs samplers, WinBUGS/OpenBUGS will draw new values for all of the π s, α and β . It will then use the current values of α and β to compute μ for that iteration. The resulting values of μ , from all iterations after sampler convergence, are draws from the posterior marginal distribution of μ and can be used for Bayesian inference. This ability to easily obtain samples from the posterior marginal distribution of functions of model parameters is a major advantage of MCMC.

9.3 Estimation Based on Hierarchical Models

Hierarchical models enable us to draw inference about parameters associated with individual observations or groups of observations (i.e., the parameters that appear in the first stage) as well as about the parameters of the distributions from which the first-stage parameters are drawn.

Output from our OpenBUGS sampler for the softball example is below. (Don't worry—I didn't forget to assess MCMC convergence before making this table! That process is examined in Sect. 9.4. We will note here that in the table below, all of the MC_errors are less than one twentieth of the corresponding estimated posterior standard deviations, so that aspect of convergence is satisfied.)

OpenBUGS>	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha	1.207	0.6828	0.01375	0.31	1.071	2.94	1001	15000
beta	5.343	3.019	0.05326	1.281	4.743	12.81	1001	15000
mu	0.1971	0.07808	9.751E-4	0.07403	0.187	0.3766	1001	15000
pi[1]	0.194	0.1223	0.001112	0.02389	0.1716	0.4911	1001	15000
pi[2]	0.1109	0.1038	0.001152	4.54E-4	0.08195	0.3787	1001	15000
pi[3]	0.2424	0.1509	0.001463	0.02978	0.216	0.5971	1001	15000
pi[4]	0.1948	0.1232	0.001205	0.02321	0.1731	0.493	1001	15000
pi[5]	0.2407	0.1486	0.001441	0.03032	0.2146	0.5952	1001	15000
pi[6]	0.1121	0.1044	0.001138	5.045E-4	0.08299	0.3818	1001	15000
pi[7]	0.2156	0.1353	0.001253	0.02514	0.191	0.5355	1001	15000
pi[8]	0.1401	0.1282	0.001433	8.319E-4	0.1057	0.4682	1001	15000

We are particularly interested in marginal inference on μ —the mean of the population distribution of the player's game-specific success probabilities in all of her college games. The posterior mean is 0.197, but the marginal posterior is quite spread out, producing a 95% posterior credible set of (0.074, 0.377).

The terms *shrinkage* and *borrowing strength* are used to describe the effect of hierarchical models on estimation of group-specific or observation-specific parameters (such as the π s in this model). Recall that in Sect. 4.2.1, we saw that the posterior mean of a parameter in a two-stage model was the result of “shrinking” an estimate based on the data alone toward the prior mean of the parameter. In a hierarchical model such as the softball model, in which the second stage is a prior distribution on many parameters of the same kind that appear in the likelihood, the shrinkage principle is expanded. In our example, the parameter π_i for each game could be estimated from the data from that game alone as the maximum likelihood estimator $\hat{\pi}_i = \frac{y_i}{n_i}$. The posterior mean $E(\pi_i|\mathbf{y})$ is the result of shrinking the individual-data-based value toward a common value determined by (a) *all* of the data, (b) the form of the second-stage, and (c) the third-stage priors. The degree of shrinkage (how far away each Bayesian posterior mean is from the corresponding data-only estimate) is driven by two factors: How much information there is in the data for estimating that parameter, and how far away the data-only estimate is from the common value.

Table 9.1 shows how all this plays out in the softball example. The data for game 1 is $y_1 = 1$ hit in $n_1 = 5$ tries, for an m.l.e. of $\hat{\pi}_1 = 0.2$. OpenBUGS estimates the posterior mean of π_1 as 0.194, with an M.C. error estimate of 0.0011, suggesting that the exact posterior mean is likely to be between 0.192 and 0.196. This is a

Table 9.1 Maximum likelihood estimates and posterior means of π s

	y	n	$\hat{\pi}_i$	$E(\pi_i \mathbf{y})$
1	1	5	0.200	0.194
2	0	4	0.000	0.111
3	1	3	0.333	0.242
4	1	5	0.200	0.195
5	1	3	0.333	0.241
6	0	4	0.000	0.112
7	1	4	0.250	0.215
8	0	2	0.000	0.140

little smaller than the frequentist estimate based on the data from game 1 alone: The posterior mean has been shrunk away from the m.l.e toward a common value smaller than 0.20.

Now look at the results for game 2. Based on no hits in 4 tries, the m.l.e. is 0. (Does this seem reasonable—should the fact that no successes were observed in that game mean that the best estimate of the underlying success probability for that game is exactly 0?) From OpenBUGS, $E(\pi_2|\mathbf{y})$ is estimated at 0.111, with an M.C. error of 0.0012. In this case, the posterior mean is “shrunk” upward from the m.l.e. to become more similar to the other π estimates. The difference between $\hat{\pi}_2$ and $E(\pi_2|\mathbf{y})$ is larger than the difference between $\hat{\pi}_1$ and $E(\pi_1|\mathbf{y})$ because $\hat{\pi}_1$ is farther away from the middle range of all the $\hat{\pi}$ s than $\hat{\pi}_2$ and because there is less data (4 at bats rather than 5) on which to estimate π_2 than π_1 .

Compare the OpenBUGS results for games 2 and 6. Since the data values are identical and the second-stage prior doesn’t introduce any *a priori* differences among the π s, the exact posterior means have to be equal. The small difference in the third decimal digit of the OpenBUGS posterior means estimates is due solely to random variability in MCMC sampling, and is smaller than twice the M.C. error estimate for either of the posterior means.

Now compare the estimation results for game 8, in which there were $y_8 = 0$ hits in $n_8 = 2$ at bats, with those for games 2 and 6. The frequentist m.l.e $\hat{\pi}_8$ again is 0, but the estimated value of $E(\pi_8|\mathbf{y})$ is 0.140, larger than those of games 2 and 6. Since there was less data (2 trials for game 8 versus 4 for games 2 and 6), the data offered less information to estimate π_8 , so the Bayesian model borrowed more strength from the data for the other games.

Games 3 and 5 produce the two largest m.l.e.s—0.333. The Bayesian-estimated posterior means are about 0.24. Again, we see that extreme values in either direction are shrunk in toward the middle.

9.3.1 Prediction from Hierarchical Models

Hierarchical models enable prediction of both unobserved data values and their associated parameters. We can illustrate this by further expanding our softball

example. Suppose we want to predict the player's success probability π_{new} at a typical future game and to predict her number of hits y_{new} in three at bats at such a game. Below is the WinBUGS/OpenBUGS code and data list with the necessary additions:

```

model
{
  for (i in 1:N) {
    y[i] ~ dbin( pi[i], n[i] )           # likelihood
    pi[i] ~ dbeta( alpha, beta )        # second-stage prior
  }

  alpha ~ dgamma(1.0, 1.0)              # third-stage prior
  beta ~ dgamma(1.0, 0.33 )             # third-stage prior

  mu <- alpha / ( alpha + beta )         # calculate function of parameters
                                          # that we wish to monitor

  pinew ~ dbeta( alpha, beta )           # draw from posterior predictive
                                          # distribution of pis
  ynew ~ dbin( pinew, nnew )            # draw from posterior predictive
                                          # distribution of new data
}

# data
list( y = c( 1,0,1,1,1,0,1,0 ), n = c(5,4,3,5,3,4,4,2), N = 8, nnew=3)

```

Note that the constant value of `nnew` is provided in the data list. The initial values list from the original model can be used with this expanded version; the `GenInits` facility can be used to generate initial values for `pinew` and `ynew`. The OpenBUGS posterior predictive summaries are below.

OpenBUGS>	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
pinew	0.1969	0.1773	0.001532	9.62E-4	0.1507	0.6527	1001	15000
ynew	0.5945	0.8153	0.006539	0.0	0.0	3.0	1001	15000

Now π_{new} is a parameter value drawn from a distribution with mean μ . Therefore, the posterior mean of π_{new} should be identical to that of μ , and it is (except for the small differences introduced by random MCMC sampling). However, the spread of the posterior distribution of π_{new} should be larger than that of μ , because the uncertainty about π_{new} includes all the uncertainty about μ plus the variability between possible π s for different games. Sure enough, both the estimated posterior standard deviation and the width of the 95% credible set for π_{new} are greater than those for μ .

Finally, the posterior summaries for y_{new} describe the posterior predictive distribution of the number of hits in a future game in which the player has three at bats. This predictive distribution should be centered at approximately three times the posterior mean of π_{new} ($3 \times 0.197 = 0.591$), and the value estimated by OpenBUGS is very close to this. Besides the uncertainty already present in the posterior distribution of π_{new} , the posterior predictive distribution of y_{new} also includes the binomial sampling variability that would exist even if π_{new} were known exactly. Thus, it is not surprising that the 95% posterior predictive interval for y_{new} spans (0,3)—all possible values of a binomial random variable when there are three trials.

9.4 More on Convergence Assessment in WinBUGS/OpenBUGS

Hierarchical models provide more challenge to MCMC convergence than the simple models for which we used OpenBUGS in Chap. 8. Although the OpenBUGS samplers for the hierarchical softball example are well behaved, the plots and diagnostics do not look quite as ideal as those from the previous chapter.

Figure 9.1 presents the history plots for the first 1,000 iterations of all three chains for the hierarchical softball example. (Although I monitored all eight of the π s, I am including only the first one in the plots and diagnostics shown here since the other seven look very similar.) Here we can see the advantage of using very different sets of initial values for the multiple chains, and then, for each model parameter, plotting the trajectories of all chains on the same axes. Figure 9.1 shows that within the first few iterations, all three chains for each parameter came together and started drawing from the same range of values. Although this wasn't instantaneous as in the simple models in the previous chapter, it was very quick. Can you figure out why the history plot for y_{new} looks so odd? It is just because y_{new} can take on only integer values (0, 1, 2, or 3).

Figures 9.2 and 9.3 focus on a single chain (chain 2), so that we can see what is happening in more detail. The plots for β are informative. Although the values change freely, the history plot doesn't look quite like white noise. There is some tendency for values from close-together iterations to be similar. This is borne out by the autocorrelation plot for β , which shows that the autocorrelation doesn't drop near zero until about 10 lags. We can get the numbers underlying an autocorrelation

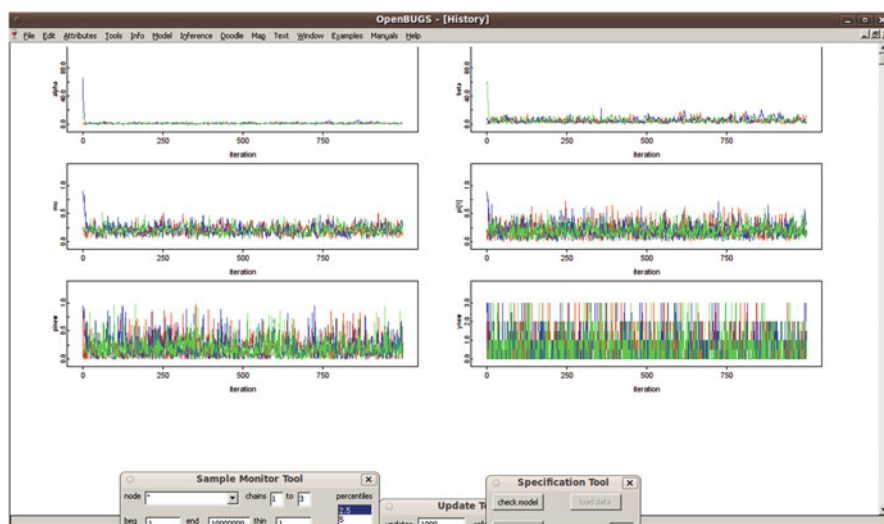


Fig. 9.1 History plots from first 1,000 iterations of all chains for hierarchical softball example

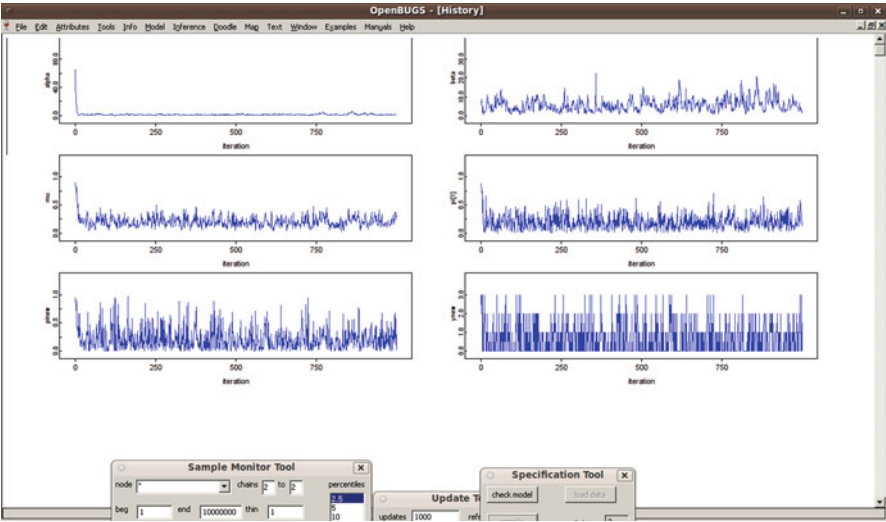


Fig. 9.2 History plots from first 1,000 iterations of chain 2 for hierarchical softball example

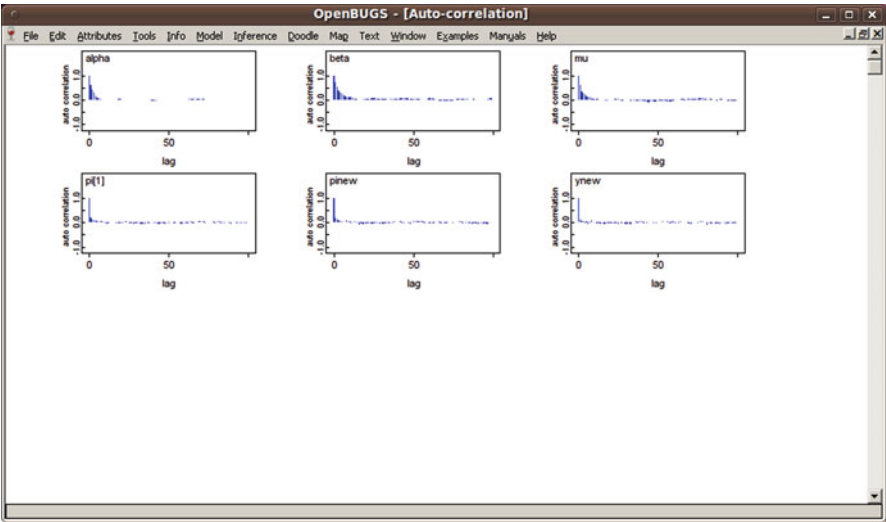


Fig. 9.3 Autocorrelation plots from first 1,000 iterations of chain 2 for hierarchical softball example

plot in OpenBUGS by right-clicking on the plot, then clicking “Properties,” and then “Data.” Here are the first lines from the resulting output for the β plot:

```
area under autocorrelation function up to lag 100
6.503
lag
```

1	0.7247
2	0.53
3	0.3804
4	0.2997
5	0.2573
6	0.208
7	0.1637
8	0.1368
9	0.1352
10	0.1311
11	0.1133
12	0.07763
13	0.0548
14	0.02805
15	0.01069

This amount of autocorrelation in MCMC sampler output is no cause for worry. It does mean that there will be less information in the sampler output than there would be in independent samples and that we should keep a careful eye on the MC error and make sure to run plenty of iterations.

9.4.1 The Brooks Gelman and Rubin Diagnostic

OpenBUGS and WinBUGS include one of [Brooks and Gelman \(1998\)](#) variants of the popular convergence diagnostic first proposed by [Gelman and Rubin \(1992\)](#). It can be used to decide how many early iterations to discard as burn-in.

The intuitive idea behind the diagnostic is that, if two or more MCMC chains have run from overdispersed initial values, we can assess whether the chains have escaped from their initial values and found the target distribution of the Markov chain by comparing the variability within each chain's output to the variability of the pooled samples from all chains. Once all the chains have converged (at least approximately) to the target distribution, then they are all drawing from the same distribution and, for any given parameter, the variability within chains should be approximately equal to the variability between chains. Prior to convergence, while each chain is drawing from a different part of the parameter space, the within-chain variability is likely to be smaller than the pooled-sample variability. [Brooks and Gelman \(1998\)](#) suggested several different measures of variability that could be used. The one implemented in OpenBUGS and WinBUGS is the widths of the 80% credible sets estimated from the samples. The numeric diagnostic, called R for ratio, is the width of the pooled-sample credible set divided by the mean width of the within-chain credible sets.

The OpenBUGS manual explains exactly what OpenBUGS calculates and plots when the user clicks the “bgr diag” button on the “Samples” tool. The fol-

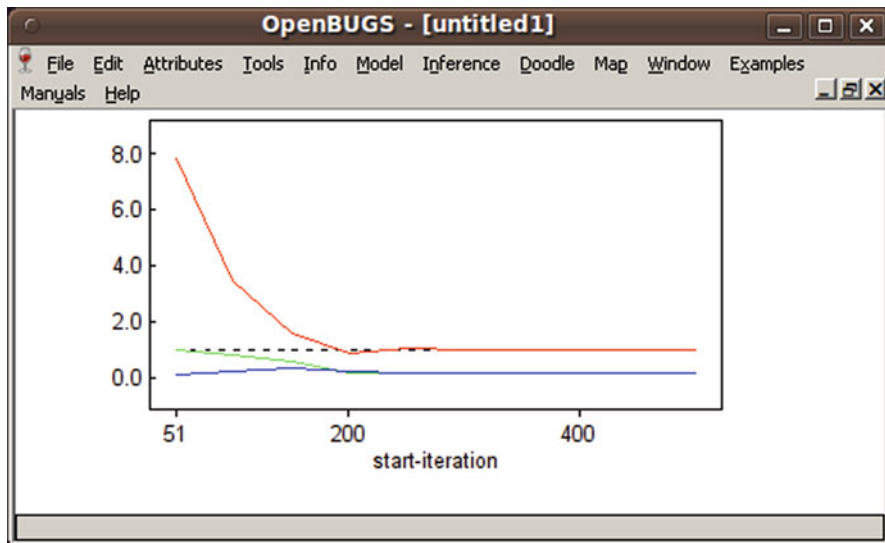


Fig. 9.4 Example of BGR diagnostic plot from OpenBUGS manual

lowing excerpt, including the idealized example plot, (Fig. 9.4) is taken from the Inference/Samples section of the user manual under the “Manuals” tab for OpenBUGS version 3.2.1.

bgr diag: calculates the Gelman-Rubin statistic, as modified by Brooks and Gelman (1998). The basic idea is to generate multiple chains starting at overdispersed initial values and assess convergence by comparing within- and between-chain variability over the second half of those chains. We denote the number of chains generated by M and the length of each chain by $2T$. We take as a measure of posterior variability the width of the $100(1-\alpha)\%$ credible interval for the parameter of interest (in OpenBUGS, $\alpha = 0.2$). From the final T iterations, we calculate the empirical credible interval for each chain. We then calculate the average width of the intervals across the M chains and denote this by W . Finally, we calculate the width B of the empirical credible interval based on all MT samples pooled together. The ratio $R = B / W$ of pooled to average interval widths should be greater than 1 if the starting values are suitably overdispersed; it will also tend to 1 as convergence is approached, and so we might assume convergence for practical purposes if $R < 1.05$, say.

Rather than calculating a single value of R , we can examine the behavior of R over iteration time by performing the above procedure repeatedly for an increasingly large fraction of the total iteration range, ending with all of the final T iterations contributing to the calculation as described above. Suppose, for example, that we have run 1,000 iterations ($T = 500$) and we wish to use the resulting sample to calculate 10 values of R over iteration time, ending with the calculation involving iterations 501–1,000. Calculating R over the final halves of iterations 1–100, 1–200, 1–300, ..., 1–1,000, say, will give a clear picture of the convergence of R to 1 (assuming the total number of iterations is sufficiently large). If we plot against the starting iteration of each range (51, 101, 151, ..., 501), then we can immediately read off the approximate point of convergence, for example,

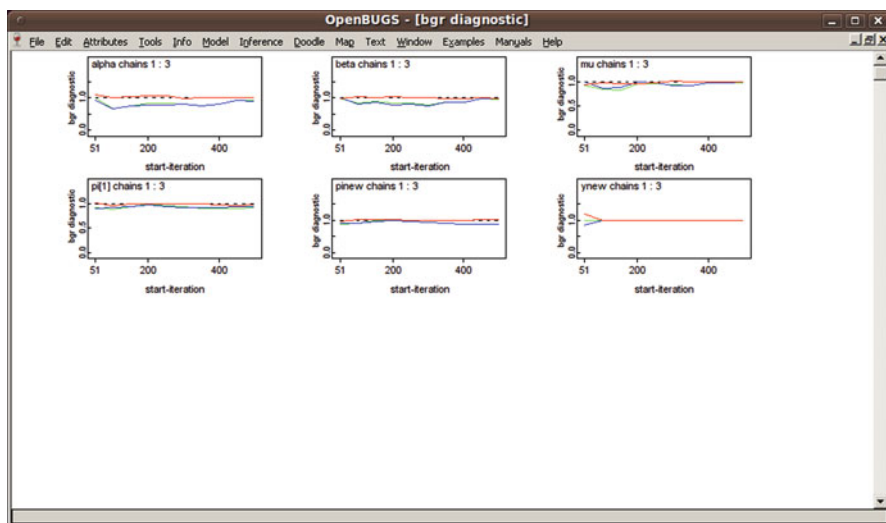


Fig. 9.5 BGR diagnostic plots for first 1,000 iterations for hierarchical softball example

OpenBUGS automatically chooses the number of iterations between the ends of successive ranges: $\max(100, 2T / 100)$. It then plots R in red, B (pooled) in green, and W (average) in blue. Note that B and W are normalized so that the maximum estimated interval width is one—this is simply so that they can be seen clearly on the same scale as R . Brooks and Gelman (1998) stress the importance of ensuring not only that R has converged to 1 but also that B and W have converged to stability. This strategy works because both the length of the chains used in the calculation and the start iteration are always increasing. Hence, we are guaranteed to eventually (with an increasing sample size) discard any burn-in iterations and include a sufficient number of stationary samples to conclude convergence.

In the above plot, convergence can be seen to occur at around iteration 250.

To use the BGR diagnostic, examine the BGR plots for each model parameter. Identify the plot that requires the greatest number of iterations for the value of the BGR diagnostic (red line) to remain below 1.2 and the blue and green lines to come together and become horizontal. The approximate convergence point in that plot is the number of iterations to discard as burn-in. To tell WinBUGS or OpenBUGS to ignore the burn-in iterations in calculating posterior summaries, simply enter the iteration number at which you want the calculations to begin in the `beg` box in the Sample monitor tool.

Figure 9.5 shows the BGR diagnostic plots for the hierarchical softball example. For all parameters, the red line representing the ratio R begins near 1 and stays there. However, for α and β , it is not clear that the blue and green lines have settled down even for the last interval.

We can obtain a table of the plotted results for the β parameter by right-clicking on the plot, then clicking “Properties,” and “Data.” The resulting table, shown in Fig. 9.6, confirms that the widths of the within chain and pooled 80%

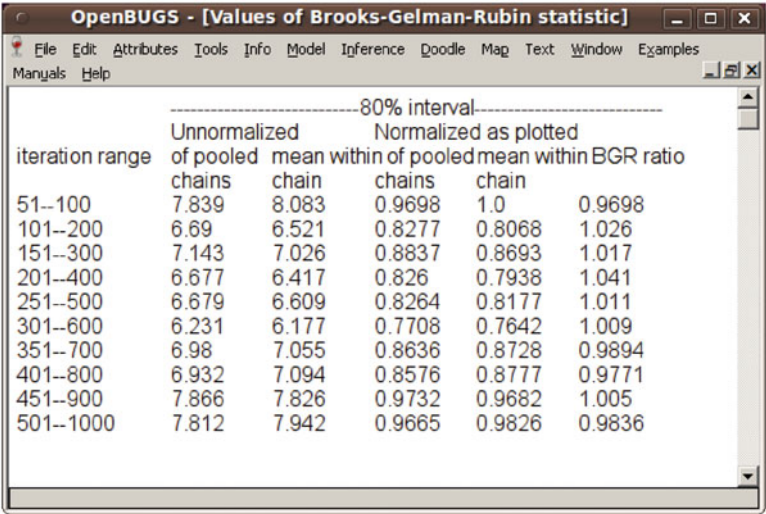


Fig. 9.6 Table of values underlying BGR plot for first 1,000 iterations

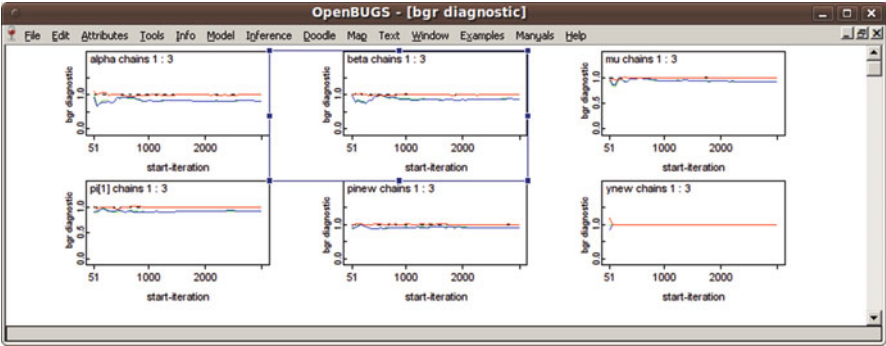


Fig. 9.7 BGR diagnostic plot for 6,000 iterations

credible sets were still bouncing around as of iterations 451–900, but might have settled down by iterations 501–1,000. More iterations are needed, so I ran an additional 5,000.

The BGR plots for the extended chains (now 6,000 total iterations in each) are in Fig. 9.7. For all parameters, everything looks great starting at approximately iteration 1,000. Therefore, before producing the output to be used for inference (the table of statistics and the posterior density plots in Sect. 9.3), I set the beginning iteration in the sample monitor tool to 1,001 so that the first 1,000 burn-in iterations would not be included.

9.4.2 *Convergence in the Hierarchical Softball Example with a Vague Prior*

We might want to do a sensitivity analysis to assess the effect of our third-stage prior specification on the inference in the hierarchical softball example. Below is OpenBUGS code for the same model as in the previous section, except that I have commented out the original priors on α and β and replaced them with very vague gamma priors. In order to emphasize the point of this section, I also made the initial values for α and β more extreme.

```
model
{
  for (i in 1:N) {
    y[i] ~ dbin( pi[i], n[i] )      # likelihood
    pi[i] ~ dbeta( alpha, beta )    # second-stage prior
  }

  alpha ~ dgamma(0.001,0.001)      # third-stage prior
  beta ~ dgamma(0.001, 0.001 )    # third-stage prior

  #alpha ~ dgamma(1.0, 1.0)        # third-stage prior
  #beta ~ dgamma(1.0,0.33 )        # third-stage prior

  mu <- alpha / ( alpha + beta )    # calculate function of parameters
                                   # that we wish to monitor

  pinew ~ dbeta( alpha, beta )
  ynew ~ dbin( pinew, nnew)

}

# data
list( y = c( 1,0,1,1,1,0,1,0 ), n = c(5,4,3,5,3,4,4,2), N = 8, nnew=3)

# overdispersed initial values for 3 chains (used for first example in ch 9)
list( alpha = .2, beta = 1, pi = c(.5, .5, .5, .5, .5, .5, .5, .5) )
list(alpha = 100, beta = 500, pi = c(.9, .9, .9, .9, .9, .9, .9, .9) )
list(alpha = 200, beta = 1000, pi = c(.1, .1, .1, .1, .1, .1, .1, .1))
```

Running this version of the model illustrates a feature of OpenBUGS and WinBUGS that we have not seen previously. During the first 500 iterations of the update, there is a check mark by the word “Adapting” in the Update tool, as shown in Fig. 9.8. This indicates that, at the beginning of the sampler run, OpenBUGS is using a method of sampling that learns from the samples already generated how to sample more efficiently as it goes along. The slice sampling algorithm [Neal \(2003\)](#) as implemented in OpenBUGS and WinBUGS adapts in this way for the first 500 iterations, then settles on a single updating scheme that is used for the remaining iterations. Because the sampling method during the adaptation phase is not strictly a Markov chain (see [Atchade and Rosenthal \(2005\)](#)), samples generated during that phase should not be used for inference. OpenBUGS and WinBUGS protect the user by refusing to include adaptation iterations in inference-related results, such as the statistics table, posterior density plots, and the BGR diagnostic. However, all samples can be included in history plots. The OpenBUGS/WinBUGS

Fig. 9.8 Update Tool showing adaptation phase

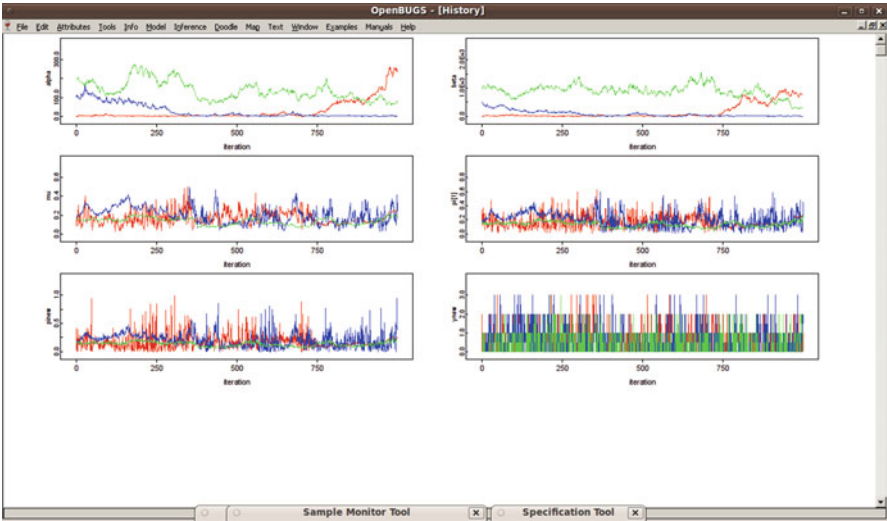
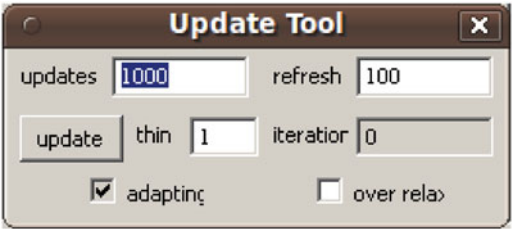


Fig. 9.9 History plots for first 1,000 iterations of model with vague priors

implementation of a different sampling algorithm, Metropolis–Hastings (Hastings 1970; Metropolis et al. 1953), by default adapts for the first 4,000 iterations.

Figure 9.9 presents the first 1,000 iterations of the hierarchical softball model with vague priors. The last 600 iterations in these plots are post-adaptation and reflect the behavior of the Markov chains. Figure 9.10 presents the corresponding autocorrelation plots. Both sets of plots indicate extremely slow mixing and high autocorrelation. The BGR diagnostic plots (representing iterations 501–1,000 only) shown in Fig. 9.11 confirm that convergence has not occurred for at least some parameters within the first 1,000 iterations.

The source of the convergence problems in this example is that, although the data (the eight pairs n_i and y_i) contain some information about the average success probability across games, they provide little information about the variability among the π_i s for different games. In other words, the data inform about the ratio $\frac{\alpha}{\alpha+\beta}$ but not very much about α and β individually. In our first version of this model, the prior densities on α and β provided enough additional information about these parameters that they were not prohibitively difficult to estimate (although the 95% posterior credible set for β shown in the table in Sect. 9.3 is quite wide). The vague

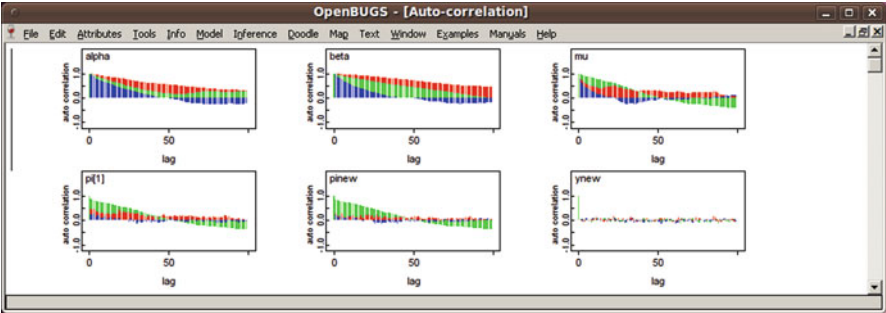


Fig. 9.10 Autocorrelation plots for first 1,000 iterations of model with vague priors

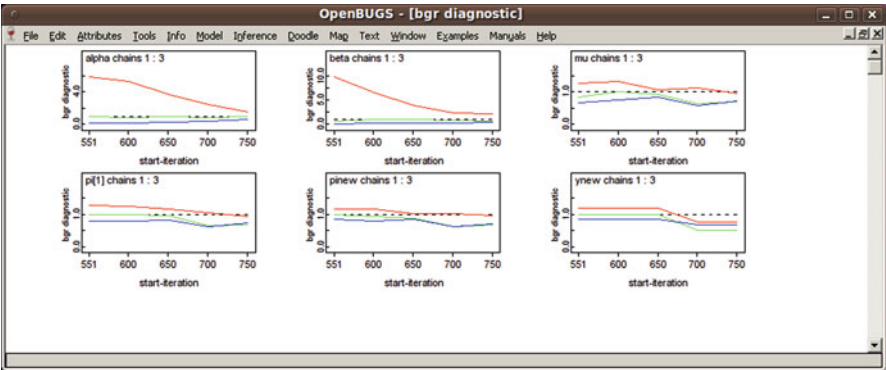


Fig. 9.11 BGR diagnostic plot for first 1,000 iterations of model with vague priors

priors in the second version are not strong enough to compensate for the lack of information in the data. Using MCMC makes estimation even more difficult, because a chain can “get stuck” in a low-posterior-probability range of extreme values of α and β , as long as the ratio they produce is reasonable.

I ran an additional 29,000 iterations, then 20,000 more, bringing the total to 50,000 per chain. Figures 9.12 and 9.13 are the history and BGR diagnostic plots.

It appears that these chains are not becoming better behaved. For α and β , the sample paths continue to go on occasional extreme excursions, but the BGR plots suggest that from approximately iteration 22,000 on, all three chains are traversing the same parameter space.

I discarded the first 22,000 iterations as burn-in. Below are the autocorrelation plots (Fig. 9.14) and table of statistics (Fig. 9.15) from iterations 22,001 to 50,000 for all chains. Although the autocorrelation has not dropped substantially compared to what we saw in the first 1,000 iterations, the simple tactic of running many, many iterations has decreased the MC errors. In a simple, low-dimensional model like this (only ten actual parameters), despite poor mixing of the MCMC samplers, we could easily run enough iterations to have reasonable confidence that the chains

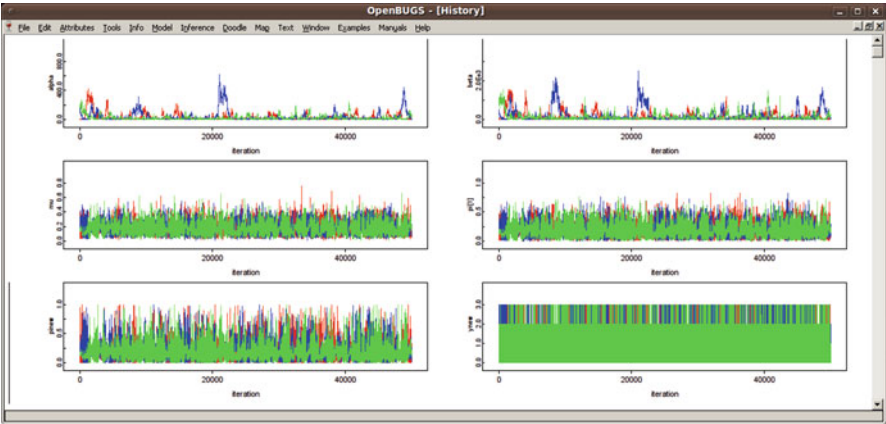


Fig. 9.12 History plots for 50,000 iterations of model with vague priors

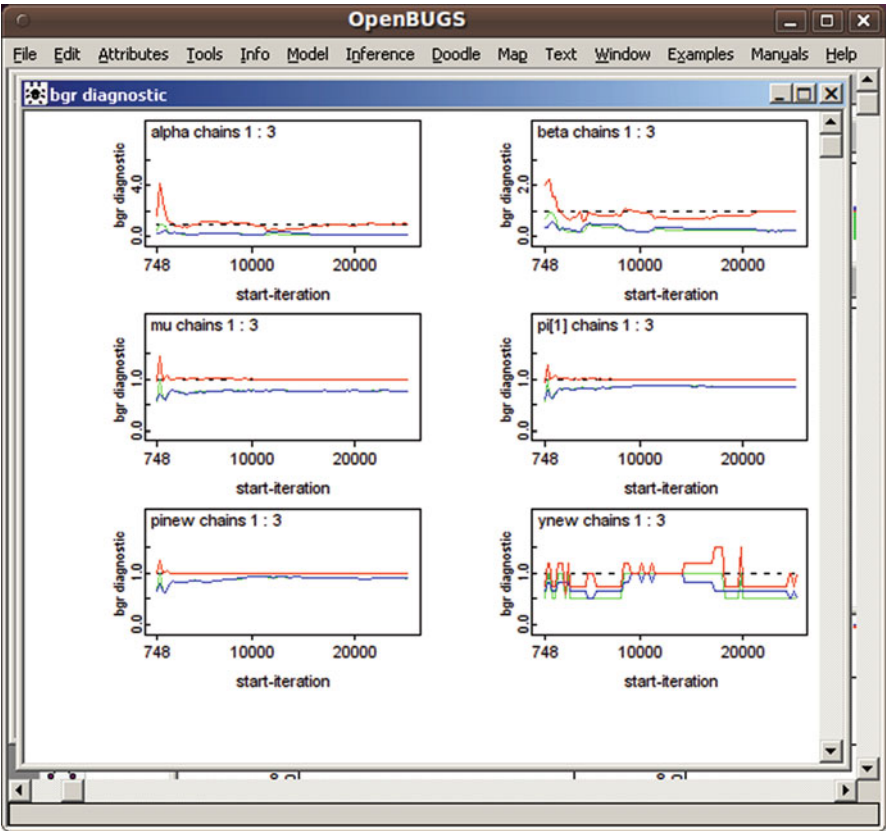


Fig. 9.13 BGR diagnostic plot for 50,000 iterations of model with vague priors

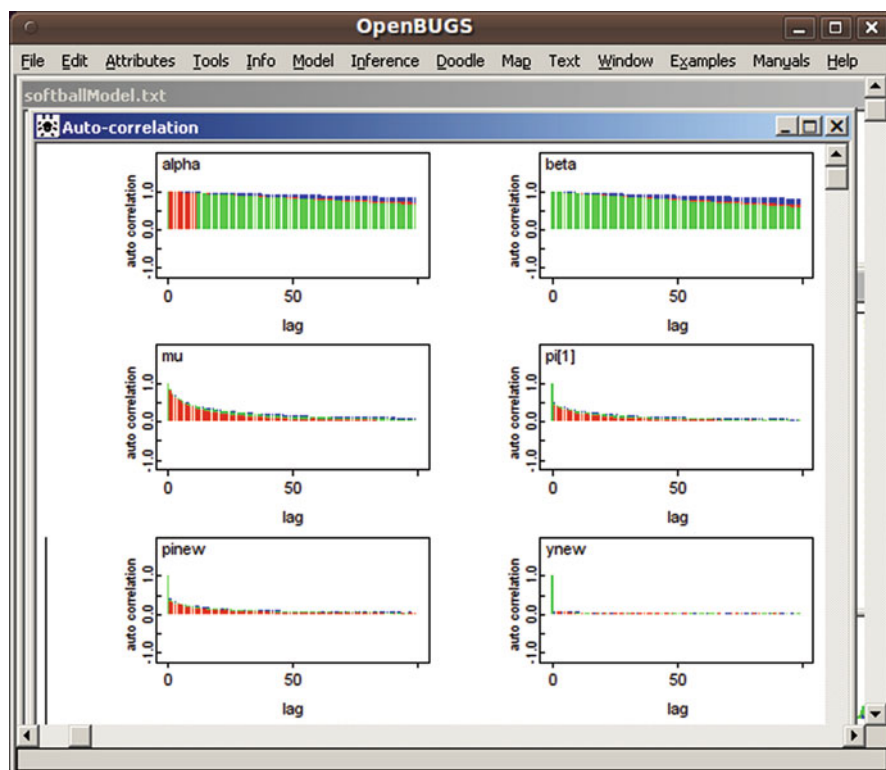


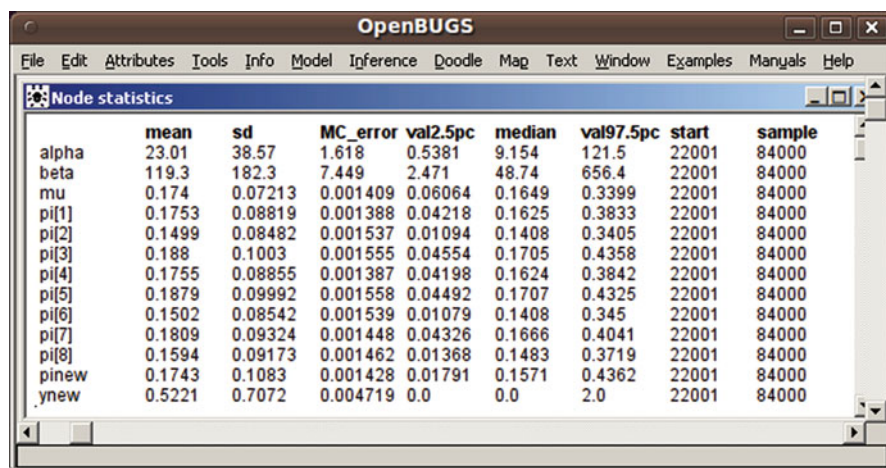
Fig. 9.14 Autocorrelation plots for iterations 22,001–50,000 of model with vague priors

had found their target distribution and had traversed the entire parameter space and that we have run enough iterations to have a satisfactory degree of precision in our estimation.

The inference resulting from this model is somewhat different from that obtained with the more informative priors. The posterior marginal 95% credible sets for α and β are very wide, indicating how little information is contained in the data and priors about these parameters. However, the other parameters and functions of interest are estimated more precisely. The posterior means of the eight π_i s are more similar in this model than in the other version, and the posterior mean of μ is smaller.

Some important general principles are illustrated by this example:

1. Very weak priors at the final stage of a hierarchical model may lead to very imprecise inference. In many cases, improper priors at the final stage produce an improper joint posterior distribution, so that inference is not possible.
2. When an MCMC sampler runs very fast (usually the case with a small dataset and a low-dimensional model), it may be possible to run enough iterations to



	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
alpha	23.01	38.57	1.618	0.5381	9.154	121.5	22001	84000
beta	119.3	182.3	7.449	2.471	48.74	656.4	22001	84000
mu	0.174	0.07213	0.001409	0.06064	0.1649	0.3399	22001	84000
pi[1]	0.1753	0.08819	0.001388	0.04218	0.1625	0.3833	22001	84000
pi[2]	0.1499	0.08482	0.001537	0.01094	0.1408	0.3405	22001	84000
pi[3]	0.188	0.1003	0.001555	0.04554	0.1705	0.4358	22001	84000
pi[4]	0.1755	0.08855	0.001387	0.04198	0.1624	0.3842	22001	84000
pi[5]	0.1879	0.09992	0.001558	0.04492	0.1707	0.4325	22001	84000
pi[6]	0.1502	0.08542	0.001539	0.01079	0.1408	0.345	22001	84000
pi[7]	0.1809	0.09324	0.001448	0.04326	0.1666	0.4041	22001	84000
pi[8]	0.1594	0.09173	0.001462	0.01368	0.1483	0.3719	22001	84000
pinew	0.1743	0.1083	0.001428	0.01791	0.1571	0.4362	22001	84000
ynew	0.5221	0.7072	0.004719	0.0	0.0	2.0	22001	84000

Fig. 9.15 Node statistics for iterations 22,001–50,000 of model with vague priors

obtain reasonable inference even when convergence is slow. In other situations, more challenging remedies for slow convergence may be needed.

3. Running a few MCMC samplers from carefully chosen overdispersed initial values helps in detecting slow convergence.

9.5 Other Hierarchical Models

There are many possible hierarchical models with a three-stage structure similar to the one in our example, but with different distributions at the three stages. Volume 1 of the examples that come with WinBUGS and OpenBUGS includes two such models, and I encourage you to study both of them carefully. “Pump” is a hierarchical Poisson/gamma model. We will discuss it in the context of graphical representation of Bayesian models in Sect. 9.6. The “Dyes” example is a hierarchical normal model, which raises some interesting issues that are explored in the next subsection.

9.5.1 Hierarchical Normal Means

A hierarchical normal model is considered when the data consist of two or more groups of samples drawn from different normal subpopulations of a larger population. The different subpopulations may have different means and/or variances, but these means (or variances) are similar to each other, and that relationship may be expressed by saying that they are draws from a common probability distribution.

The observations in such data require two subscripts: y_{ij} denotes the j th observation in group i .

The “Dyes” example in WinBUGS/OpenBUGS is a problem taken from the classic Bayesian textbook (Box and Tiao 1973). Here the subpopulations are batches of product produced by a factory, and 5 items were randomly drawn from each of 6 batches. The model is constructed under the assumption that different batches have different means μ_i , $i = 1, \dots, 6$ but that the between-item variability is the same within all batches. If we denote the within-batch precision as τ_{with} , the likelihood looks like this:

$$y_{ij} | \mu_i, \tau_{with} \sim N\left(\mu_i, \frac{1}{\tau_{with}}\right), \quad i = 1, \dots, 6, \quad j = 1, \dots, 5$$

or

$$p(\mathbf{y} | \mu_1, \dots, \mu_6, \tau_{with}) = \prod_{i=1}^6 \prod_{j=1}^5 \frac{\sqrt{\tau_{with}}}{\sqrt{2\pi}} \exp\left[-\frac{\tau_{with}(y_{ij} - \mu_i)^2}{2}\right] \quad (9.2)$$

9.5.1.1 Prior Specification at the Second and Third Stages

Standard practice in constructing hierarchical models is to have the second stage consist of a prior or priors on set(s) of parameters of the same kind and to leave the prior(s) on any one-of-a-kind parameters from the likelihood to the third stage.

In the Dyes example, the unknown parameters in the likelihood are the six μ s and the single τ_{with} . The prior for the set of μ s will comprise the second stage. Although other choices certainly are possible, the most common choice of parametric family for the second-stage prior on a set of normal means is the semi-conjugate normal prior, and that was what Box and Tiao, and the WinBUGS/OpenBUGS example, used:

$$\mu_i | \theta, \tau_{btw} \sim N\left(\theta, \frac{1}{\tau_{btw}}\right), \quad i = 1, \dots, 6$$

This prior says that the μ s are like draws from the population of all possible μ s (the means of all possible batches of product from the factory). The precision parameter τ_{btw} reflects how different the means of different batches are. The parameter θ represents the overall mean of all possible batch means. Both of these are unknown parameters to be estimated.

The third stage provides the priors on the unknown parameters from the second stage, as well as τ_{btw} from the likelihood. In the original version of the WinBUGS/OpenBUGS example, vague semi-conjugate priors were specified on all three of these parameters—gammas for the precisions and normal for θ . The code is below:

```
model
```



```

{
  for(i in 1 : batches) {
    mu[i] ~ dnorm(theta, tau.btw)
    for(j in 1 : samples) {
      y[i , j] ~ dnorm(mu[i], tau.with)
    }
  }
  sigma2.with <- 1 / tau.with
  sigma2.btw <- 1 / tau.btw
  tau.with ~ dgamma(0.001, 0.001)
  tau.btw ~ dgamma(0.001, 0.001)
  theta ~ dnorm(0.0, 1.0E-10)
}

```

9.5.1.2 Third-Stage Priors and Proper and Improper Posterior Distributions

As you will see in Problem 9.5, the vague gamma prior on τ_{btw} leads to very poor MCMC sampler convergence for this model and to poor estimation of this parameter even if tens of thousands of iterations are run. There is no problem with a vague prior for τ_{with} , because it is the precision of observable data values, so the actual data provide a lot of information about it. On the other hand, τ_{btw} is the precision of unknown and unknowable parameters, so it is far harder to estimate.

It is actually the case that an improper prior on τ_{btw} will result in an improper joint posterior density, so that no valid posterior inference is possible. The gamma(0.001, 0.001) prior used in the OpenBUGS example is proper, so the resulting posterior actually is proper. However, it is just barely proper, and the additional uncertainty introduced by the use of MCMC to fit the model causes the difficulty that you will observe in Problem 9.5.

Typically the statistician will have little prior information about the variability between the parameters modeled at the second stage of a hierarchical normal model, so specifying a prior on τ_{btw} presents a real challenge. Continuing Bayesian methodological research seeks ways of specifying priors on the between-means precision that contain little information but lead to better-behaved posterior densities and MCMC samplers than do vague gammas. For this purpose, Gelman (2006) recommends using a non-conjugate vague uniform prior on the between-means standard deviation (i.e., on $\frac{1}{\sqrt{\tau_{btw}}}$). At the time of this writing, this prior is included as an alternative in the WinBUGS version of the Dyes example, but not in the OpenBUGS version.

9.6 Directed Graphs for Hierarchical Models

Graph theory is an area of mathematics and computer science that can offer insight into statistical models. In this setting, a *graph* is a collection of vertices or nodes representing some type of objects. Pairwise relationships among the nodes are represented by *edges* connecting them. A graph may be either *directed* or *undirected*.

The kinds of Bayesian models that OpenBUGS and WinBUGS can fit may be expressed by a particular kind of graph called a *directed acyclic graph* or DAG (Lauritzen and Spiegelhalter 1998). Each quantity in the model is represented by a node in the graph. The edges connecting the nodes are shown as arrows pointing into nodes from the nodes on which they depend directly.

The developers of WinBUGS and OpenBUGS recommend drawing DAGs while developing Bayesian models and have included the “DoodleBUGS” feature to facilitate this process. Not only does DoodleBUGS assist the user in drawing the graph, but also, for models that are not too complicated, DoodleBUGS can translate the graph into WinBUGS/OpenBUGS code which then can be run in the usual way.

9.6.1 Parts of a DAG

Figure 9.16—the DAG for the WinBUGS/OpenBUGS hierarchical Poisson/gamma model called “Pump”—provides examples of the symbols used in directed acyclic graphs.

The object that looks like a pile of papers is a set of *plates*. Plates show that there are more than one of each of the objects drawn inside them and include a notation of how many repetitions there are (the `for i IN 1 : N` in this example).

The different kinds of quantities in a Bayesian model are represented in a DAG by the following kinds of nodes:

Constants (values fixed by the design of the study) are represented by single- or double-edged rectangles. Since they are fixed, constants do not depend on any other quantities in the model. Thus, they have no arrows running into them, but do have arrows running out from them and into the nodes that depend on them. Another way of saying this using the language of DAGs is that constants have no *parents* but are the parents of other nodes. In the pump example, the `t[i]`s are constants.

Stochastic nodes are variables that are given a distribution. They may be observed data, whose probability distribution is given in the likelihood or unobservable parameters whose distributions are given in later stages of the model. They generally are *children* in the graph, with their parents characterizing their distribution, but they may be parents as well. Stochastic nodes generally are represented by circles or ovals, although observed data often are denoted by single-edged rectangles. In the pump example, `alpha`, `beta`, the `theta[i]`s, and the `x[i]`s are stochastic nodes.

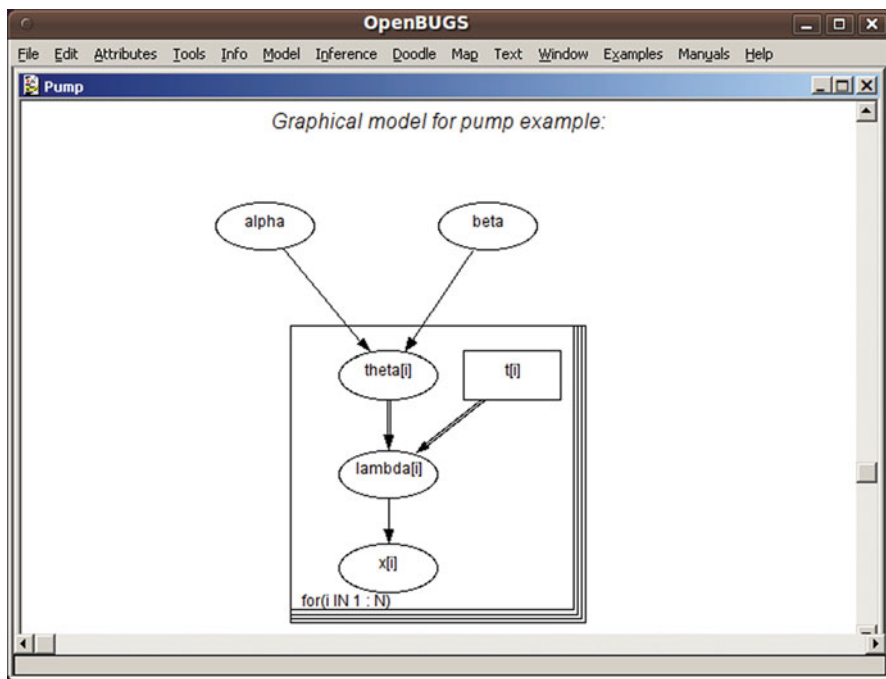


Fig. 9.16 The directed acyclic graph for the hierarchical Poisson/gamma Pump example

Deterministic nodes are calculated functions of other nodes. WinBUGS/OpenBUGS refers to the functions as *logical functions* and to the resulting nodes as *logical nodes*. Since one or more stochastic nodes usually are needed in the calculations for a logical node, logical nodes are also represented by circles or ovals.

Two kinds of dependence can be represented by arrows in a DAG:

Stochastic dependence is represented by a solid arrow. For example, in the graph for the pump model, the solid arrows from α and β into the $\theta[i]$ s indicate that the θ s are random draws from a distribution with parameters α and β .

Deterministic dependence is represented by either dashed arrows or hollow arrows. In the DAG for the pump model, the hollow arrows from $\theta[i]$ and $t[i]$ into $\lambda[i]$ indicate that the $\lambda[i]$ s are calculated deterministically given the $\theta[i]$ s and the $t[i]$ s.

9.7 *Gibbs Sampling for Hierarchical Models

Recall that a Markov chain is generated one iteration at a time and that the *transition kernel* is the distribution from which the values for each iteration are drawn,

conditional on the existing values from the previous iteration. The Gibbs sampling algorithm is one way to construct a transition kernel to produce a Markov chain whose stationary distribution is the joint posterior distribution in a Bayesian model. The seminal references on Gibbs sampling are [Gelfand and Smith \(1990\)](#); [Geman and Geman \(1984\)](#); [Hastings \(1970\)](#); [Metropolis et al. \(1953\)](#).

The key to Gibbs sampling is *full conditional distributions*. The full conditional distribution for a model quantity is the distribution of that quantity conditional on assumed known values of all of the other quantities in the model. For example, in a simple normal model in which the two unknown parameters were the mean μ and the variance σ^2 and the observed data was \mathbf{y} , there would be two full conditionals:

$$p(\mu|\sigma^2, \mathbf{y})$$

$$p(\sigma^2|\mu, \mathbf{y})$$

The mathematical foundation of the Gibbs sampling algorithm is the fact that, subject to regularity conditions (which unfortunately are outside the scope of this course), a joint distribution is uniquely determined by the corresponding full conditional distributions.

The Gibbs sampling algorithm can be described generically as follows. Suppose that we have a Bayesian model with p unknown parameters, which we will call $\theta_1, \theta_2, \dots, \theta_p$, and that the observed data are denoted \mathbf{y} . Thus, the joint posterior distribution is $p(\theta_1, \theta_2, \dots, \theta_p|\mathbf{y})$. As with any Markov chain, the user must provide initial values. I'll indicate iteration numbers with superscripts in parentheses, so the initial values are $\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_p^{(0)}$. Then the Gibbs sampler draws the values for each iteration in p steps by drawing a new value for each parameter from its full conditional given the most recently drawn values of all other parameters. In symbols, the steps for any iteration, say iteration k , are as follows:

- Draw $\theta_1^{(k)}$ from $p(\theta_1|\theta_2^{(k-1)}, \theta_3^{(k-1)}, \dots, \theta_p^{(k-1)})$
- Draw $\theta_2^{(k)}$ from $p(\theta_2|\theta_1^{(k)}, \theta_3^{(k-1)}, \dots, \theta_p^{(k-1)})$
- \vdots
- Draw $\theta_p^{(k)}$ from $p(\theta_p|\theta_1^{(k)}, \theta_2^{(k)}, \dots, \theta_{(p-1)}^{(k)})$

Thus, the difficult task of sampling from the possibly complicated, high-dimensional joint posterior distribution is broken down into a large number of simpler, feasible sampling tasks.

9.7.1 Deriving Full Conditional Distributions

But how can we figure out what probability distribution each full conditional is? Here are the steps for the analytical procedure for deriving the full conditional distribution of any individual unknown model quantity:

1. Write the mathematical form of the (unnormalized) joint posterior distribution.
2. Pull out every term in the joint posterior that contains the quantity of interest.
3. Write the product of all the terms from step 2; this product is proportional to the needed full conditional distribution.
4. If possible, identify the parametric family of which the full conditional is a member.

Let's find the full conditionals for each parameter in the hierarchical beta-binomial softball example. The unknown parameters are $\pi_1, \pi_2, \dots, \pi_8, \alpha$, and β . We already did step 1, and the result is in (9.1.4), which is reproduced here.

$$p(\pi, \alpha, \beta | \mathbf{y}) \propto \left[\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right]^8 \prod_{i=1}^8 \left[\pi_i^{y_i + \alpha - 1} (1 - \pi_i)^{n_i - y_i + \beta - 1} \right] \exp(-\alpha) \exp(-0.33\beta)$$

To find the full conditional for any individual π_i , $i = 1, \dots, 8$, we write the product of all the terms in (9.1.4) that contain π_i and then simplify. I will use the notation $\pi_{(-i)}$ to represent all the π s except π_i .

$$p(\pi_i | \pi_{(-i)}, \alpha, \beta, \mathbf{y}) \propto \pi_i^{y_i + \alpha - 1} (1 - \pi_i)^{n_i - y_i + \beta - 1}$$

We recognize this as the kernel of a beta density with parameters $y_i + \alpha$ and $n_i - y_i + \beta$. OpenBUGS/WinBUGS uses one of the well-known algorithms for sampling from beta densities. Note that the other seven π s do not appear in this density.

How about α ?

$$\begin{aligned} p(\alpha | \pi_i, i = 1, \dots, 8, \beta, \mathbf{y}) &\propto \left[\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \right]^8 \prod_{i=1}^8 \left[\pi_i^{y_i + \alpha - 1} \right] \exp(-\alpha) \\ &\propto \left[\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)} \right]^8 \prod_{i=1}^8 [\pi_i^\alpha] \exp(-\alpha) \end{aligned}$$

Hmm, have you ever seen a density that looked like that? (Remember, α is the random variable here.) No, this definitely is not any familiar density, and OpenBUGS/WinBUGS will have to use one of their general-purpose algorithms for sampling from nonstandard densities.

Now for β :

$$\begin{aligned} p(\beta | \pi_i, i = 1, \dots, 8, \alpha, \mathbf{y}) &\propto \left[\frac{\Gamma(\alpha + \beta)}{\Gamma(\beta)} \right]^8 \prod_{i=1}^8 \left[(1 - \pi_i)^{n_i - y_i + \beta - 1} \right] \exp(-0.33\beta) \\ &\propto \left[\frac{\Gamma(\alpha + \beta)}{\Gamma(\beta)} \right]^8 \prod_{i=1}^8 \left[(1 - \pi_i)^\beta \right] \exp(-0.33\beta) \end{aligned}$$

This is another nonstandard density. Note that the data (\mathbf{y} and \mathbf{n}) do not appear in the full conditionals for α or β .

There is another method of deriving full conditional distributions, which is based on directed acyclic graphs. Here are the steps:

1. Draw a directed graph of your complete model.
2. Identify the “parents” and “children” of the parameter of interest.
3. Write the product of the conditional distributions of
 - the parameter given its parents
 - the children given their parents.

You will have a chance to try this method in some of the exercises for this chapter.

No Bayesian inference can be directly based on full conditional distributions. They are useful only as part of the computing methods for fitting Bayesian models. Fifteen or more years ago, Bayesian statisticians often had to code their own Gibbs samplers to fit models that they developed. Today BUGS, WinBUGS, OpenBUGS, and R packages for Bayesian model fitting are available and can handle most Bayesian data analysis needs, so the skills of deriving full conditionals and coding MCMC samplers are rarely needed.

9.8 Recommendations for Using MCMC to Fit Bayesian Models

Here are some general comments and recommendations for using MCMC to fit Bayesian models.

9.8.1 *How Many Chains*

There is controversy over whether it is better to run a single long MCMC chain, or multiple shorter ones. Obviously it is wasteful to run more than one chain, because burn-in iterations have to be discarded from all of them. On the other hand, if you run only one chain, you have no way of knowing whether there are parts of the parameter space that it never visited. My recommendation is to run a small number (three to five) of chains with initial values selected to give you information about sampler performance.

9.8.2 *Initial Values*

Students often confuse initial values with prior parameters, but they are totally different. Priors are part of the model specification. Priors must *not* be derived from

the current dataset. Initial values are part of the computing process. Initial values can be derived from the current dataset.

Here are some comments on specifying initial values:

- Initial values may be randomly generated from priors.
- If it is possible to fit a frequentist model that is similar to your Bayesian model, then initial values may be based on frequentist estimates. For example, you might initialize one chain with the maximum likelihood estimates for all parameters, one chain with the m.l.e.s minus 4 standard errors, and one with the m.l.e.s plus 4 standard errors.
- Initial values may be chosen systematically to represent extreme regions of the parameter space to aid in assessing convergence.
- Initial values must be specified for variance components. WinBUGS/OpenBUGS usually can automatically generate initial values for other parameters, but it's often advantageous to specify even those that can be auto-generated.

9.8.3 General Advice

Learn as much as possible about your model before ever running an MCMC sampler. Fit an analogous frequentist model. Fit a simple Bayesian model first; then gradually add complexity. These steps will enable you to recognize whether your MCMC sampler output is giving reasonable inference.

To assess MCMC sampler convergence, monitor *all* model parameters, not only the parameters of substantive interest. If your model has too many parameters to permit individual monitoring of all of them, then least monitor examples of all *types* of parameters. For example, if there are 6,000 random effects, monitor a half dozen or so of them as well as all precision parameters in a model. Examine history plots, autocorrelation plots, BGR plots, and MC errors. Be aware that there is no guarantee that any or all of these methods will detect convergence failure.

For more detail on MCMC convergence diagnostics and their use, see [Cowles and Carlin \(1996\)](#).

9.9 Exercises

9.1. Draw a directed acyclic graph of the hierarchical model for the softball problem. You may either use DoodleBUGS or draw it freehand.

9.2. For the “pump” example in WinBUGS/OpenBUGS, write an expression to which the joint posterior distribution of all the model parameters is proportional.

9.3. Derive the posterior full conditional distributions that would be used in a Gibbs sampler for the pump model. First, start from your joint posterior in the previous

problem to extract each full conditional. Then check whether you get the same results if you start from the directed acyclic graph.

9.4. Modify the WinBUGS/OpenBUGS code for the pump model to create nodes representing (a) the mean of the distribution from which the individual pumps' failure rates are drawn, (b) the failure rate of a new pump that is not part of the current dataset, and (c) the number of failures of the new pump in 1,000 h. Run the expanded model, monitoring your new nodes, and comment on the results.

9.5. Consider the "Dyes" example in the first set of WinBUGS/OpenBUGS examples. At the time of this writing, the versions of this example are slightly different in the two software packages. The WinBUGS version includes three different choices of prior on the between-groups precision parameter, while the OpenBUGS version has only one choice (the choice that is not recommended by WinBUGS!).

1. Under this model, are all the y_{ij} 's (the yields for all the different samples in different batches) considered exchangeable? Why or why not?
2. Under this model, are all the y_{2j} 's (the yields for all the samples in batch 2) considered exchangeable?
3. Under this model, are the μ_i 's considered exchangeable?
4. Run three parallel chains for 15,000 iterations to fit the model as given in the example. If you are using WinBUGS, use the *third prior* (the one that is *not* recommended). If you are using OpenBUGS and there is only one prior, use it.
 - For one chain, use the initial values provided in the example.
 - For another chain, use


```
list(mu = c(1,525, 1,525, 1,525, 1,525, 1,525, 1,525), theta = 1,525, tau.btw = 10,000, tau.within = 0.001)
```
 - For another chain, set the μ 's equal to values that are very different from each other and from θ , and reverse the values of τ_{between} and τ_{within} given previously.
 - Monitor θ , μ , τ_{between} , τ_{within} , $\sigma_{\text{between}}^2$, and σ_{within}^2 , beginning with the first iteration.
 - Obtain autocorrelation plots, history plots, Gelman and Rubin diagnostics, and output statistics. Look at plots based on each chain individually as well as on the combined chains. You do not have to print and turn in all the plots! Choose a few typical ones to print and comment on.
5. Run another 15,000 iterations and obtain the same output. Do the autocorrelation plots change much with additional iterations? Does the Monte Carlo error decrease with additional iterations?
6. The use of very vague priors on both of the variance components, as shown in the example, is a bad idea. If an improper prior is placed on τ_{between} , the posterior will be improper as well. The priors on τ_{within} and τ_{between} are so vague that this is *almost* the case here. That is the reason why so many iterations were required in the sampler run.

Replace the vague prior on $\tau_{between}$ with an informative prior. Choose its parameters this way. You want an *inverse gamma* prior on $\sigma_{between}^2$ (the *variance* of batch means) that has a mean of 2,000 and a variance of 250,000. Find the correct values of the parameters α and β . Now in WinBUGS, put a *gamma* prior with the same values of α and β on $\tau_{between}$ (the *precision*).

7. Repeat steps (4) and (5) above with your modified model, answering the same questions. In addition, compare the autocorrelation plots and Monte Carlo errors between the original model and the model with the informative prior. (Again, you don't have to print out many of the plots. Summarize the comparison in a few sentences.)
8. If you are working from code with three choices of prior, repeat step (4):
 - using Prior 1 in the example
 - using Prior 2 in the example

Again, summarize what you see in history plots, autocorrelation plots, and BGR diag plots.

Chapter 10

Regression and Hierarchical Regression Models

Linear regression is one of the most commonly used methods in both classical and Bayesian statistics.

10.1 Review of Linear Regression

Recall that in regression analysis, we have two or more variables that can be measured on the same subjects. We wish to use one or more of them—the *predictor variables* (also called *independent variables* or *covariates*)—to explain or predict a *response variable* (also called an *outcome variable* or a *dependent variable*). How we define which variable is the response and which are predictors depends on our research question. In *linear regression*, the response variable is quantitative. In *simple linear regression*, there is only one predictor variable, and the relationship between the response variable and the predictor is roughly linear.

Typically, the notation Y is used for the response variable and X for a predictor, so that y_i and x_i denote the observed values of the response and the predictor for the i th subject in a dataset.

The population *regression equation* with one covariate is

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

where β_0 is the intercept (usually defined as the expected value of Y when $X = 0$) and β_1 is the slope (the expected difference between two Y values whose corresponding X values differ by one unit).

A crucial assumption underlying linear regression is that the expected values of the Y variable, when plotted against the values of the X variable, lie on a straight line. The ε s represent the random differences between individual observed Y values and their expected values on the regression line. The term for these random differences is *errors*, but with no implication that they are mistakes or wrong in any way.

A second regression assumption, which is needed in frequentist analysis to calculate p-values and confidence intervals, is that the errors follow a normal distribution with zero mean. Thus, the three unknown parameters in simple linear regression are β_0 , β_1 , and the variance σ^2 of the normal distribution of the errors. The slope β_1 is usually of greatest interest, since it captures the relationship between the two variables.

10.1.1 Centering the Covariate

When all of the possible values of the covariate are of the same sign and lie far away from zero, the mathematical definition of the intercept may not make sense substantively. For example, suppose the population of interest is adult males, the covariate is height in inches, and the response variable is weight in pounds. Then, although the intercept is a perfectly valid mathematical construct and it is needed to make the line lie in the right place, the notion of an adult with height 0 inches is nonsensical.

In such cases, a common practice is to center the covariate before using sample data to estimate the regression coefficients and variance. Centering simply means calculating the sample mean of the covariate, and subtracting this mean from each covariate value. In symbols, before centering, the model being fit is

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

but with centering, it becomes

$$Y_i = \beta_0^* + \beta_1 (X_i - \bar{X}) + \varepsilon_i \quad (10.1)$$

The interpretation and value of the slope β_1 are unchanged by centering. However, centering gives the intercept a different meaning and usually a very different value. We can see this by rearranging the right side of (10.1).

$$Y_i = \beta_0^* - \beta_1 \bar{X} + \beta_1 X_i + \varepsilon_i$$

The intercept in the centered model is the expected value of the response variable when the centered covariate is 0—that is, when the covariate on the original scale has the typical value \bar{X} rather than the unreasonable value of 0. Thus, it has a meaningful interpretation.

An additional advantage of centering covariates for Bayesians is that centering can improve convergence of MCMC samplers for fitting complicated Bayesian regression models.

10.1.2 Frequentist Estimation in Regression

In frequentist linear regression, the maximum likelihood estimates of the intercept and slope are calculated by a method called *ordinary least squares (OLS)*, which

chooses the best-fitting line by minimizing the sum of the squared differences between each data point and the fitted line. Using the notation y_i $i = 1, \dots, n$ for the observed values of the response variable and x_i $i = 1, \dots, n$ for the observed values of the covariate, the formulas for the m.l.e.s of the coefficients in simple linear regression with covariate centering are as follows:

$$\hat{\beta}_0 = \bar{y}$$

$$\hat{\beta}_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

$\hat{\beta}_0$ and $\hat{\beta}_1$ are the sufficient statistics for β_0 and β_1 .

To provide more notations that we will use throughout this chapter, the fitted values \hat{y}_i are

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

and the *residuals* r_i are the differences between the observed values and the fitted values:

$$r_i = y_i - \hat{y}_i$$

The sum of the squared residuals

$$SSR = \sum_i \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 (x_i - \bar{x}) \right)^2$$

is the sufficient statistic for σ^2 . The unbiased estimator of σ^2 when there is only one covariate is

$$\frac{\sum_i \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 (x_i - \bar{x}) \right)^2}{n - 2} = \frac{\sum_i r_i^2}{n - 2}$$

10.1.3 Example: Mercury Deposited by Precipitation Near the Brule River in Wisconsin

As we mentioned in Sect. 6.2.1, eating fish contaminated with mercury is a health hazard, particularly to infants and children. Mercury gets into fish through the following cycle. Mercury is emitted into the atmosphere from both human and natural sources. Then rain washes mercury out of the air and deposits it on land and in surface water. Such deposition is the major way that mercury gets into lakes and rivers. Fish ingest mercury from their food sources in the water in which they live.

The Mercury Deposition Network (MDN) (<http://nadp.sws.uiuc.edu/mdn/>) has monitored mercury deposition in precipitation at locations in the United States and

Canada since 1996. MDN is a part of the National Atmospheric Deposition Program (NADP), a cooperative monitoring effort involving governmental, academic, tribal, and private organizations. Currently MDN operates over 100 sites from which precipitation is collected and analyzed weekly following rigorous quality assurance procedures. The resulting data is publicly available at <http://nadp.sws.uiuc.edu/MDN/mdndata.aspx>. These data can be used to assess trends in mercury deposition over time and space, and to predict deposition levels at unsampled locations or future times.

I was interested in assessing whether there was a systematic change in mercury concentration levels in the Midwest during the decade and a half of MDN's existence. I decided to begin by looking at the time trend at a single site, and to expand the analysis to include multiple sites as a later step. There are no MDN sites in Iowa. I chose the Brule River site in nearby Wisconsin as the first location to study. The weekly Brule station data spans the last half of 1996 through the beginning of 2011. Both mercury emissions and precipitation are known to have seasonal patterns that were not the focus of my interest, so I decided to aggregate that data into annual averages. Furthermore, the distribution of the raw weekly concentrations was highly right skewed. For these reasons, I first log transformed the weekly mercury concentrations, then averaged all valid measurements within each year. I discarded the 1996 and 2011 values, because they did not include the full years. This process produced annual values for 14 years, 1997–2010, which are plotted in Fig. 10.1. The figure suggests a somewhat linear relationship, with a slightly negative slope. We need to use statistics to determine whether this apparent relationship is more than would be likely to occur by chance.

Specifically, we will use frequentist methods to estimate the slope—the annual rate of change of mean log mercury concentration, and to test the null hypothesis that the slope is greater than or equal to 0.

$$H_0 : \beta_1 \geq 0$$

$$H_1 : \beta_1 < 0$$

Yet another assumption in linear regression is that the errors are uncorrelated. We need to check that assumption carefully in data like this, in which the measurements were taken sequentially over time. Autocorrelation would be visible in the scatterplot if adjacent values of the Y variable tended to be very similar. The raw weekly data exhibited high autocorrelation, but we see much less evidence in the aggregated annual data. We will proceed with our analysis, but will check again after we have fit the regression model and have estimates of the errors in the form of the residuals.

The covariate is the year, with all strictly positive values far away from 0. If we don't center the covariate, the interpretation of the intercept would be the expected value of mean log mercury concentration in the year 0, which has little meaning. So we will center the covariate. This will not change the estimated slope, but now the intercept will be the expected value of Y in the middle of the year 2003.

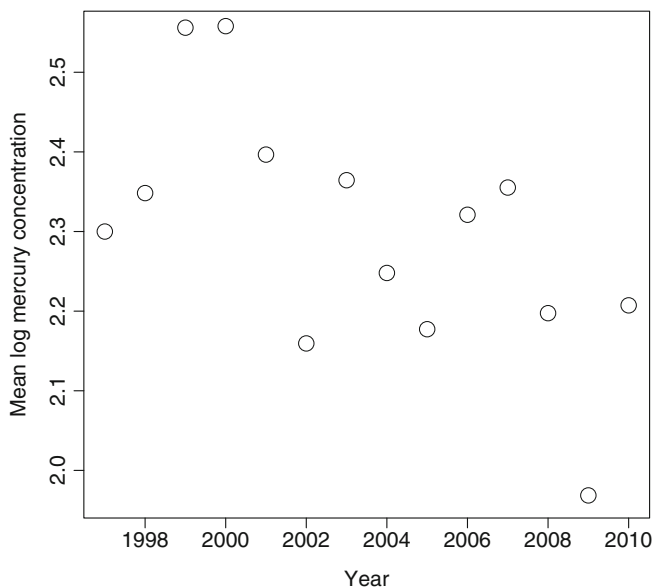


Fig. 10.1 Scatterplot of mean log mercury concentration at the Brule River MDN site versus year

Below is R code for fitting the regressions, first without centering and then with centering. The values of the response variable are in a vector called `brulemeanLogHgConcYr`. The `lm` function is used to store all the regression calculations from each model into an object (`brulelmout` for the analysis with uncentered covariate and `brulelmout2` for the analysis with centering). We can then summarize or plot aspects of these objects.

```
# uncentered covariate
> x <- 1997:2010
> brulelmout <- lm(brulemeanLogHgConcYr ~ x)

# centered covariate
> xcent <- x - mean(x)
> brulelmout2 <- lm(brulemeanLogHgConcYr ~ xcent)
```

Before interpreting the results, we should perform some diagnostic checks. The residual plot (residuals r_i on the y-axis and the fitted values \hat{y}_i on the x-axis) in Fig. 10.2 is reassuring: it looks like a random scatter of points with no curvature, no clustering of points, and no systematic change of spread from left to right. Thus, it provides no evidence of violation of the assumptions of linearity, uncorrelated errors, and equality of variance.

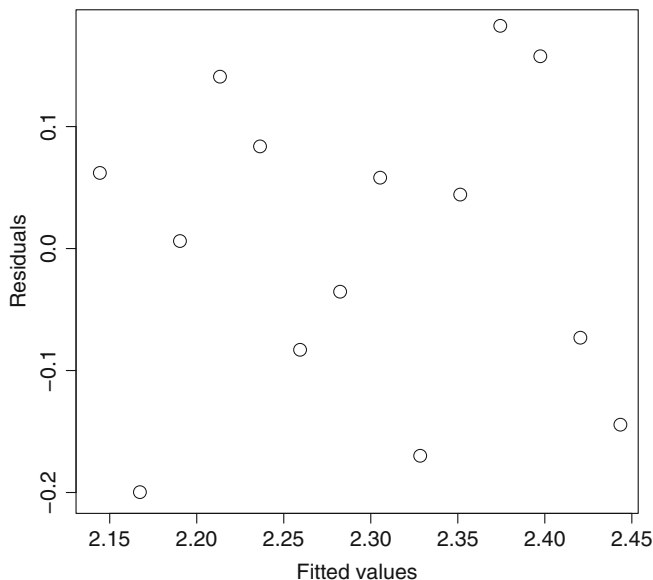


Fig. 10.2 Residual plot for regression of mean log mercury concentration at the Brule River MDN site versus year

```
plot( brulelmout2$fitted.values,
      brulelmout2$residuals)
```

The normal QQ plot of the residuals (Fig. 10.3) also shows no clear violation of the normality assumption.

Finally, we use the Durbin–Watson test in the R package `lmtest` to check numerically that the residuals do not exhibit autocorrelation:

```
> library(lmtest)
> dwout <- dwtest( brulelmout2,
  alternative='two.sided')
> dwout
```

Durbin--Watson test

```
data: brulelmout2
DW = 1.7589, p-value = 0.4232
alternative hypothesis: true autocorrelation is not 0
```

This output says that we would have had better than a 40% chance of getting sample data that produced residuals with as much serial autocorrelation as those in our dataset even if the true population lag 1 autocorrelation is 0. While this certainly does not prove that the population autocorrelation really is 0, it suggest that we don't

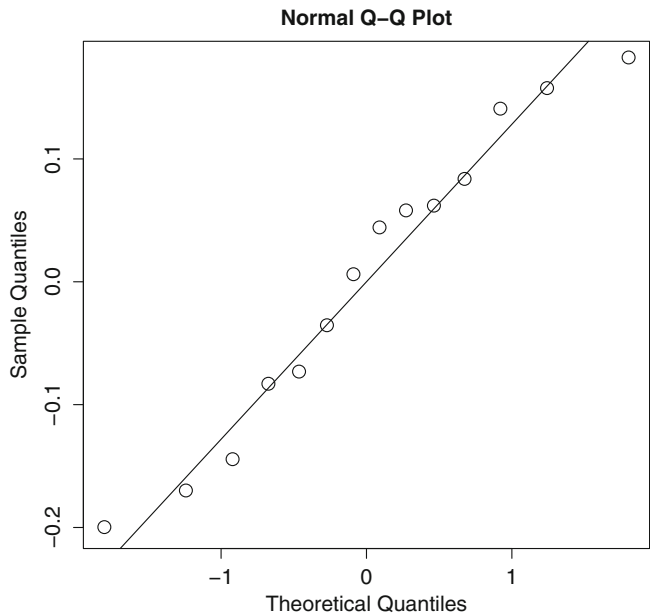


Fig. 10.3 Normal QQ plot of residuals from regression of mean log mercury concentration at the Brule River MDN site versus year

need to specifically account for temporal correlation in our analysis—the simple linear regression model probably is adequate.

Let’s examine the regression output.

```
# uncentered

> summary(brulelmout)

Call:
lm(formula = brulemeanLogHgConcYr ~ x)

Residuals:
    Min       1Q   Median       3Q      Max
-0.20183 -0.08267  0.02307  0.07616  0.18046

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  48.385393   17.031873    2.841   0.0149 *
x           -0.023006    0.008501   -2.706   0.0191 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

```

Residual standard error: 0.1282 on 12 degrees of
freedom
Multiple R-squared: 0.379, Adjusted R-squared: 0.3273
F-statistic: 7.324 on 1 and 12 DF, p-value: 0.01909

```

```

> confint(brulelmout)
                2.5 %          97.5 %
(Intercept) 11.27613049 85.494656439
x            -0.04152854 -0.004484178

```

```
# centered
```

```
> summary(brulelmout2)
```

```
Call:
```

```
lm(formula = brulemeanLogHgConcYr ~ xcent)
```

```
Residuals:
```

```

      Min       1Q   Median       3Q      Max
-0.20183 -0.08267  0.02307  0.07616  0.18046

```

```
Coefficients:
```

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.292155    0.034269  66.888  <2e-16 ***
xcent        -0.023006    0.008501  -2.706   0.0191 *
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

Residual standard error: 0.1282 on 12 degrees of
freedom

```

```

Multiple R-squared: 0.379, Adjusted R-squared: 0.3273
F-statistic: 7.324 on 1 and 12 DF, p-value: 0.01909

```

```

> confint(brulelmout2)
                2.5 %          97.5 %
(Intercept)  2.21749013  2.366820717
xcent        -0.04152854 -0.004484178

```

As expected, the estimates of the slope are the same in both models: $\hat{\beta}_1 = -0.0230$, implying that the annual mean of log mercury concentration drops by about -0.023 log units per year on average. The 95% confidence interval for the slope is $(-0.0415, -0.0045)$.

The sign of the estimated slope is negative—consistent with the alternative hypothesis in our one-sided test. The `lm` function in R automatically reports the p-value for a two-sided test of the null hypothesis that the slope is zero. To get the frequentist p-value for our one-sided hypothesis test, we can halve the reported p-value: $0.0191/2 = 0.0096$. Thus, for the frequentist, the result is statistically significant at any reasonable significance level. The data provide strong evidence of a negative slope at this site.

Also as expected, the estimates of the intercept differ between the two models. For the uncentered model, the point estimate is 48.39 log units, and the 95% confidence interval is extremely wide—(11.28, 85.49). The data contains little information about what would be going on in the year 0, so there is huge uncertainty in estimating the intercept in the uncentered model. In the model with covariate centering, the point estimate of the intercept is 2.292, and the 95% confidence interval is narrow—(2.217, 2.367).

The residual standard error, 0.1282 in both models, is the point estimate of σ —the standard deviation of points around the regression line. The unbiased estimate of σ^2 is $0.1282^2 = 0.0164$.

10.2 Introduction to Bayesian Simple Linear Regression

Bayesian simple linear regression is closely related to the normal models that we encountered in Chap. 7. The difference is that instead of a single population mean, each observation i has its own mean that depends on the regression coefficients and the i th covariate value. Detailed descriptions of Bayesian linear regression are presented in [Box and Tiao \(1973\)](#); [Gelman et al. \(2004\)](#); [Lee \(2004\)](#); [Zellner \(1971\)](#).

In what follows, we will assume that the covariate has been centered, as that simplifies some calculations.

The likelihood can be written distributionally as follows:

$$y_i | x_i, \beta_0, \beta_1, \sigma^2 \sim N(\beta_0 + \beta_1(x_i - \bar{x}), \sigma^2), \quad i = 1, \dots, n$$

The parameter of greatest interest in Bayesian simple linear regression usually is the slope, β_1 . Thus, we need to find the joint posterior density of all three regression parameters and then integrate out β_0 and σ^2 to get the posterior marginal density of β_1 .

10.2.1 Standard Noninformative Prior

We will consider first the standard noninformative prior that yields Bayesian inference analogous to the frequentist results. As in the models with normal likelihood and both mean and variance unknown, the standard noninformative prior

in Bayesian regression is the product of independent improper priors on the means-related parameters and the variance parameter. The parameters related to the means are the regression coefficients β_0 and β_1 . Multiplying flat priors (proportional to a constant over the whole real line) on both coefficients times an inverse gamma prior with both parameters going to 0 for σ^2 yields

$$p(\beta_0, \beta_1, \sigma^2) \propto \frac{1}{\sigma^2}, \quad -\infty < \beta_0, \beta_1 < \infty, \quad 0 < \sigma^2 < \infty$$

We approximate this prior in WinBUGS with either vague normal (or “dflat()”) priors on β_0 and β_1 and a very vague gamma prior on the precision parameter.

The three sufficient statistics simplify deriving the joint and marginal posterior distributions of the regression parameters.

$$\begin{aligned}\hat{\beta}_0 &= \bar{y} \\ \hat{\beta}_1 &= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} \\ SSR &= \sum_i \left(y_i - \hat{\beta}_0 - \hat{\beta}_1(x_i - \bar{x}) \right)^2\end{aligned}$$

Also recall that the sample variance in regression is $s^2 = \frac{SSR}{n-2}$.

With the standard noninformative prior and a centered covariate, the joint posterior density is

$$\begin{aligned}p(\beta_0, \beta_1, \sigma^2 | \mathbf{y}) &\propto \frac{1}{\sigma^2} \frac{1}{(\sigma^2)^{\left(\frac{n}{2}\right)}} \exp \left[\frac{-\sum_i (y_i - \beta_0 - \beta_1(x_i - \bar{x}))^2}{2\sigma^2} \right] \\ &= \frac{1}{(\sigma^2)^{\left(\frac{n+2}{2}\right)}} \exp \left[-\frac{SSR + n(\beta_0 - \hat{\beta}_0)^2 + \sum_i (x_i - \bar{x})^2 (\beta_1 - \hat{\beta}_1)^2}{2\sigma^2} \right]\end{aligned}$$

If we integrate β_1 out of the joint posterior, we get

$$p(\beta_0, \sigma^2 | \mathbf{y}) \propto \frac{1}{(\sigma^2)^{\left(\frac{n+1}{2}\right)}} \exp \left[-\frac{SSR + n(\beta_0 - \hat{\beta}_0)^2}{2\sigma^2} \right]$$

To obtain the posterior marginal density of β_0 , we must integrate σ^2 out of the preceding expression which, after some algebra, produces

$$\beta_0 | \mathbf{y} \sim t \left(\hat{\beta}_0, \frac{s^2}{n}, n-2 \right)$$

at distribution with mean $\hat{\beta}_0$, scale parameter $\frac{s^2}{n}$, and degrees of freedom $n-2$. A similar pair of integrations lead to the posterior marginal density that is our

primary concern:

$$\beta_1 | \mathbf{y} \sim t \left(\hat{\beta}_1, \frac{s^2}{\sum (x_i - \bar{x})^2}, n-2 \right)$$

Based on these t distributions, the Bayesian credible sets for β_0 and β_1 will have exactly the same endpoints as the frequentist t confidence intervals for the same parameters.

And finally,

$$\sigma^2 | \mathbf{y} \sim IG \left(\frac{n-2}{2}, \frac{SSR}{2} \right)$$

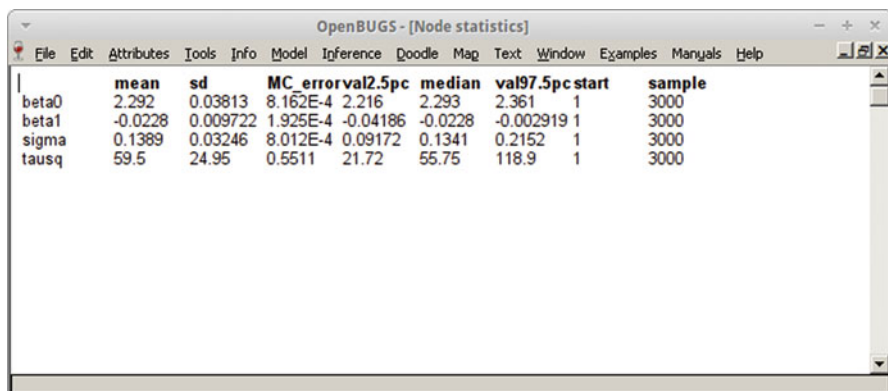
10.2.1.1 Verifying that the Posterior Density Is Proper

Recall that whenever a statistician uses an improper prior, he or she must verify that the data provides sufficient information to make the posterior density proper. In the case of simple linear regression and the standard improper joint prior, the data requirements are that the sample size n must be strictly greater than two, and that not all the covariate values are equal.

10.2.2 *Bayesian Analysis of the Brule River Mercury Concentration Data*

Here is OpenBUGS code for a simple Bayesian linear regression analysis of the Brule River mercury concentration data using a close approximation to the standard noninformative prior for linear regression. Note that the covariate can be centered by the WinBUGS/OpenBUGS code itself, as in the `for` loop at the beginning of the model program.

```
model
{
  for( i in 1:N) {
    xcent[i] <- x[i] - mean(x[])
  }
  for (i in 1:N) {
    mu[i] <- beta0 + beta1 * xcent[i]
    y[i] ~ dnorm( mu[i], tausq )
  }
  beta0 ~ dflat()
  beta1 ~ dflat()
  tausq ~ dgamma( 0.001, 0.001)
  sigma <- 1 / sqrt(tausq)          # regression standard
                                   deviation
}
```



The screenshot shows the 'OpenBUGS - [Node statistics]' window. It contains a table with the following data:

	mean	sd	MC error	val2.5pc	median	val97.5pc	start	sample
beta0	2.292	0.03813	8.162E-4	2.216	2.293	2.361	1	3000
beta1	-0.0228	0.009722	1.925E-4	-0.04186	-0.0228	-0.002919	1	3000
sigma	0.1389	0.03246	8.012E-4	0.09172	0.1341	0.2152	1	3000
tausq	59.5	24.95	0.5511	21.72	55.75	118.9	1	3000

Fig. 10.4 OpenBUGS results from regression of mean log mercury concentration at the Brule River MDN site versus year

```
#data
list( x = c(1997, 1998, 1999, 2000, 2001, 2002, 2003,
            2004, 2005, 2006, 2007, 2008, 2009, 2010),
      y = c(2.2952, 2.3435, 2.5512, 2.5531, 2.3918,
            2.1546, 2.3596,
            2.2431, 2.1725, 2.3162, 2.3504, 2.1926,
            1.9638, 2.2025),
      N = 14)

# inits
list( beta0 = 0, beta1 = 0, tausq = 1)
```

Convergence of OpenBUGS samplers for simple linear regression models is almost instantaneous. You will verify that for this example in an exercise. For now, we will move on to examine the output.

The OpenBUGS posterior summaries from 3,000 iterations from a single chain are shown in Fig. 10.4. The posterior means for the regression coefficients are close to the frequentist m.l.e.s, as we would expect for parameters with symmetric posterior marginal densities since we are using the noninformative prior. The posterior densities of `tausq` and `sigma` are right skewed, so their posterior means are larger than their posterior modes; it is the latter that would correspond to the frequentist m.l.e.s for these parameters.

We can expand our OpenBUGS program to do further inference and prediction. In the model code below, I have inserted the line

```
postprob <- step(beta1)
```

to carry out the Bayesian test of the null hypothesis that $\beta_1 \geq 0$. The WinBUGS/OpenBUGS `step` function returns 0 if its argument is negative and 1 if

its argument is 0 or greater. Thus, the posterior mean of the node `postprob` will estimate the posterior probability of the null hypothesis.

In order to predict the value of mean log mercury concentration for the year 2011, I added 2011 to the end of the vector of covariate values and an NA to the end of the vector of response variable values. Accordingly, I changed `N` from 14 to 15. I still wanted to center the covariate around the mean of the covariate values corresponding to the observed response variables (i.e., the first 14 observations in the dataset); notice the change in the first `for` loop that makes OpenBUGS ignore the fake 15th observation when computing the mean.

```

model
{
  for( i in 1:N) {
    xcent[i] <- x[i] - mean(x[1:14])
  }
  for (i in 1:N) {
    mu[i] <- beta0 + beta1 * xcent[i]
    y[i] ~ dnorm( mu[i], tausq )
  }
  beta0 ~ dflat()
  beta1 ~ dflat()
  tausq ~ dgamma( 0.001, 0.001)
  sigma <- 1 / sqrt(tausq)
  postprob <- step(beta1)          # count the
                                   iterations in which
                                   beta1

}

#data
list( x = c( 1997, 1998, 1999, 2000, 2001, 2002, 2003,
             2004, 2005, 2006, 2007, 2008, 2009,
             2010, 2011),
      y = c(2.2952, 2.3435, 2.5512, 2.5531, 2.3918,
             2.1546, 2.3596, 2.2431, 2.1725, 2.3162,
             2.3504, 2.1926, 1.9638, 2.2025, NA),
      N = 15)

# inits
list( beta0 = 0, beta1 = 0, tausq = 1)

```

The OpenBUGS posterior summaries from 3,000 iterations from a single chain are shown in Fig. 10.5.

The statistics for `postpred` indicate that $Pr(\beta_1 \geq 0|y)$ is about 0.0089. Furthermore, the results for `y[15]` show that the prediction interval for the mean log mercury concentration in the year 2011 is (1.803, 2.432).

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
beta0	2.292	0.03726	6.824E-4	2.216	2.293	2.367	1	3000
beta1	-0.02321	0.009488	1.415E-4	-0.0422	-0.02329	-0.004855	1	3000
mu[15]	2.118	0.08024	0.001181	1.962	2.117	2.278	1	3000
postprob	0.008333	0.09091	0.001634	0.0	0.0	0.0	1	3000
sigma	0.1379	0.03189	7.617E-4	0.09174	0.1329	0.2142	1	3000
y[15]	2.117	0.1588	0.002831	1.803	2.119	2.432	1	3000

Fig. 10.5 OpenBUGS results from regression of mean log mercury concentration at the Brule River MDN site versus year

10.2.3 Informative Prior Densities for Regression Coefficients and Variance

What if we have prior information that we would like to include in our Bayesian model? The simplest (and probably most commonly used) procedure is to assume *a priori* independence among β_0 , β_1 , and σ^2 and to place independent proper normal priors on β_0 and β_1 and a proper inverse gamma prior on the variance σ^2 (or equivalently, a gamma prior on the precision). The product of these three prior densities is not a *conjugate* prior, because the resulting posterior density will not factor into three independent densities from the same families.

10.3 Generalized Linear Models

There are many kinds of response variables for which linear regression based on a normal likelihood won't work. The most common example is binary response variables.

As a concrete application, let's revisit the telephone survey regarding a sales tax to support flood prevention measures in Iowa City and Coralville, first introduced in Problem 4.2. The main question asked whether the person supported the sales tax; responses could be coded 1 for "yes" and 0 for "no." The survey also included demographic questions, such as how long the person had lived in the area.

Suppose we want to model the relationship between years lived in the area (predictor variable X) and support/nonsupport of the sales tax (response variable Y). Two problems with trying to do this with normal linear regression immediately become obvious. Our model would look like

$$E(Y_i) = \beta_0 + \beta_1 X_i$$

Since all Y_i s are zeroes and ones, $E(Y_i)$ actually is the probability π_i that Y_i equals 1, given the value of X_i . Now, if we fit a frequentist linear regression model, the fitted values would be

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

We would hope that the fitted values would be between 0 and 1. However, suppose that $\hat{\beta}_1$ is negative (implying that longer residence in the area is associated with lower probability of supporting the sales tax). It is almost certain that for some very large values of x_i , the corresponding \hat{y}_i s would be negative. On the other hand, if $\hat{\beta}_1$ turned out to be positive, very large x_i s would produce \hat{y}_i s that were greater than 1.

Furthermore, an assumption of normal linear regression is that the errors follow a normal distribution. Recall that the errors ε_i are the differences between the actual values of the Y variable and the values predicted by the true population regression equation.

$$\varepsilon_i = Y_i - \beta_0 - \beta_1 X_i$$

If all the Y_i s are zeroes and ones, subtracting a linear function of the predictor variable from each is not likely to produce draws from a normal density.

Binary data are not the only kind of response variables for which normal linear regression is a poor choice. Count data, which might arise from a Poisson distribution, are another common example.

A class of models that works well in these cases is called *generalized linear models* or GLMs [McCulloch and Nelder \(1989\)](#); [Nelder and Wedderburn \(1972\)](#). The idea is that, instead of modeling the expected value of the response variable directly, we model a transformation of it.

For binary response variables, several transformations work well for this purpose, but by far, the most commonly used is the logit transformation, which we encountered in Sect. 5.3.3.1.

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right)$$

While a probability π_i must lie in the interval $[0, 1]$, a logit may be anywhere on the whole real line. (Verify this for yourself by attempting to evaluate $\text{logit}(0)$, $\text{logit}(0.5)$, and $\text{logit}(1)$.) Thus, the first problem with regression with a binary response variable is eliminated. Binary regression using the logit transformation is called *logistic regression*. An excellent textbook on frequentist logistic regression is [Hosmer and Lemeshow \(2000\)](#).

WinBUGS and OpenBUGS provide a simple but surprising syntax for fitting Bayesian logistic regression models, as illustrated in the “Beetles” program in Volume II of WinBUGS/OpenBUGS examples.

The data (first published in [Bliss 1935](#)) involves eight groups of approximately 60 beetles each that were exposed to different doses of an insecticide. The variables for each group i are as follows: r_i is the number of beetles killed, x_i is the log of the

insecticide dose, and n_i is the number of beetles in the group. Below is the code, to which I have added comments on the most important lines.

```
model
{
  for( i in 1 : N ) {
    # likelihood: each r[i] is a draw from a binomial
    # distribution with success probability p[i] and
    # number of trials n[i]
    r[i] ~ dbin(p[i],n[i])
    # the logit of p[i] is a linear function of the
    # centered log dose x[i]
    logit(p[i]) <- alpha.star + beta * (x[i] - mean(x[]))
    #probit(p[i]) <- alpha.star + beta * (x[i] - mean(x[]))
    #cloglog(p[i]) <- alpha.star + beta * (x[i] - mean(x[]))
    rhat[i] <- n[i] * p[i]
  }
  alpha <- alpha.star - beta * mean(x[])
  beta ~ dnorm(0.0,0.001)
  alpha.star ~ dnorm(0.0,0.001)
}
```

The parameter of greatest interest is the slope β . A positive slope is expected, indicating that higher doses of insecticide are associated with higher probabilities of beetles being killed. The interpretation of the slope β in logistic regression with a single covariate is that, for each one-unit increase in the value of the predictor variable, we expect a β -unit change in the log odds of the response variable Y being equal to 1. An equivalent statement that is a little bit more understandable is the for each one-unit increase in the value of the predictor variable, the odds in favor of $Y = 1$ are multiplied by $\exp(\beta)$.

The results reported in the Beetles example in WinBUGS/OpenBUGS is that the posterior mean of the slope on log of insecticide dose is 34.29. This means that each increase of one log unit in insecticide dose multiplies the log of the odds in favor of beetles being killed by 34.29.

The two lines that are commented out code other choices of transformations that can be used in regression with binary response variables, the probit transformation and the complementary log-log transformation. Like the logit transformation, both of these map the interval $[0, 1]$ to the whole real line. You will work with this example in Problem 11.4 in Chap. 11.

For an in-depth discussion of Bayesian generalized linear models, see Chap. 16 of [Gelman et al. \(2004\)](#).

10.4 Hierarchical Normal Linear Models

Hierarchical normal linear models combine the principles of hierarchical models and Bayesian regression. They are needed when we want to use regression to relate covariates and response variables, but the problem we are considering is inherently hierarchical in structure and/or the observations in our dataset are grouped. Early presentations of Bayesian hierarchical normal linear models are ([Lindley and Smith 1972](#); [Novick et al. 1972](#)).

10.4.1 *Example: Estimating the Slope of Mean Log Mercury Concentration Throughout North America Using Data from Multiple MDN Sites*

I would like to use the MDN data to estimate the overall mean slope on year of log mercury concentration for all of North America, and I would like to assess the variability among slopes in different locations. For this purpose, I will randomly select one site from each of six regions in North America: northeastern, southeastern, north central, south central, northwestern, and southwestern.

What would be wrong with combining the annual mean log mercury concentration data from all six sites together and doing a simple linear regression (Bayesian or frequentist) to estimate the intercept and slope on year? The problem is that, due to differing conditions of climate, emissions, etc. at each site, the observations within each site are likely to be correlated—to be more similar to each other than to the observations at other sites. For example, perhaps a site in Pennsylvania always has higher mercury concentrations in rainfall than the Brule River site. Such clustering violates the independence assumption of linear regression. If we analyze data as if they are independent when they are not, we tend to underestimate the uncertainty in our inference.

One possible way of modeling the correlation among repeated measurements at the same site is to let each site have its own pair of regression coefficients—its own intercept and its own slope on year. We might be able to assume that, *given the site-specific regression coefficients*, the observations within each site are *conditionally independent* of each other and of the observations at other sites. We might consider the intercept–slope pairs from the six sites in our dataset as random draws from the population of the intercept–slope pairs from all possible locations in North America where MDN rain gauges could be placed. The mean of the population distribution of slopes is our parameter of greatest interest.

10.4.2 *Stages of a Hierarchical Normal Linear Model*

As always, the first stage of the model provides the distribution of the data given certain model parameters. Let y_{ij} represent the observed value of mean log mercury concentration at site i in year t_{ij} , $i = 1, \dots, N = 6$ and $j = 1, \dots, 14$. Then the likelihood may be expressed as

$$y_{ij} | \alpha_{0i}, \alpha_{1i}, \tau_y^2 \sim N \left(\alpha_{0i} + \alpha_{1i} t_{ij}, \frac{1}{\tau_y^2} \right)$$

where τ_y^2 is the precision of the points around the site-specific regression line, and α_{0i} and α_{1i} are the site-specific intercept and slope. This model assumes that all sites share the same precision parameter; other assumptions could be made.

There are at least two different ways of formulating the second stage of the model. Both assume that the site-specific intercepts $\alpha_{0,i}$ and $\alpha_{1,i}$ are draws from normal densities centered at the overall mean intercept and slope for North America, represented by β_0 and β_1 .

10.4.3 Univariate Formulation of the Second Stage

In the univariate formulation, the second-stage prior consists of independent normal densities for the site-specific intercepts and slopes:

$$\alpha_{0i}|\beta_0, \tau_{\alpha_0}^2 \sim N\left(\beta_0, \frac{1}{\tau_{\alpha_0}^2}\right)$$

$$\alpha_{1i}|\beta_1, \tau_{\alpha_1}^2 \sim N\left(\beta_1, \frac{1}{\tau_{\alpha_1}^2}\right)$$

Here the precision parameter $\tau_{\alpha_0}^2$ captures the variability among intercepts at different sites. If this precision is small (so its inverse, the between-intercept variance, is large), then the intercepts at different sites tend to be very different from one another. Similarly, the precision parameter $\tau_{\alpha_1}^2$ captures how different the slopes on year are at different sites. As expected in a hierarchical model, the parameters in the second-stage priors (β_0 , β_1 , $\tau_{\alpha_0}^2$, and $\tau_{\alpha_1}^2$) are all unknown quantities that we wish to estimate.

10.4.4 Bivariate Formulation of the Second Stage

The univariate formulation described in the previous section generally works fairly well. However, it does not include any parameter that captures correlation between intercepts and slopes. A bivariate normal joint prior on the intercept–slope pairs is needed to do that:

$$\begin{bmatrix} \alpha_{0i} \\ \alpha_{1i} \end{bmatrix} | \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \Sigma_\alpha \sim N_2\left(\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \Sigma_\alpha\right)$$

Here Σ_α is the variance–covariance matrix of the bivariate normal prior density. The upper left hand entry is the variance of the site-specific intercepts; the lower right-hand entry is the variance of the site-specific slopes, and the off-diagonal entries are the covariance between the site-specific intercepts and slopes. If sites with larger intercepts tend also to have larger slopes, then the covariance will be positive; if the opposite is the case, it will be negative. The inverse of the variance–covariance matrix is the precision matrix. As you might guess, WinBUGS and OpenBUGS parameterize multivariate normal densities in terms of their means and precision matrices.

This bivariate formulation has one more parameter (the covariance) than the univariate version. Thus, it is a bit more complicated, but it also gives more information about the relationship between site-specific intercepts and slopes.

10.4.5 Third Stage: Univariate Formulation

The third stage consists of prior distributions for all the remaining unknown parameters. The conventional, semi-conjugate priors are as follows (with the symbols on the right-hand side of each representing fixed, known numbers):

$$\begin{aligned}\beta_0 &\sim N\left(\mu_0, \frac{1}{\tau_0^2}\right) \\ \beta_1 &\sim N\left(\mu_1, \frac{1}{\tau_1^2}\right) \\ \tau_y^2 &\sim G(a_y, b_y) \\ \tau_{\alpha_0}^2 &\sim G(a_{\alpha_0}, b_{\alpha_0}) \\ \tau_{\alpha_1}^2 &\sim G(a_{\alpha_1}, b_{\alpha_1})\end{aligned}$$

The prior on τ_y^2 can be very vague, because the observed data values make estimation of this precision parameter quite easy. However, $\tau_{\alpha_0}^2$ and $\tau_{\alpha_1}^2$ are the precisions of unknown and unobservable parameters. If we put improper gamma priors on them, the joint posterior density of all the model unknowns would be improper. Even vague proper priors on $\tau_{\alpha_0}^2$ and $\tau_{\alpha_1}^2$ can lead to model instability and slow convergence of an MCMC sampler for model fitting.

10.4.6 Third Stage: Bivariate Formulation

If the bivariate formulation is used at the second stage of the hierarchical normal linear model, the corresponding semi-conjugate third-stage priors are as follows:

$$\begin{aligned}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \mid \begin{bmatrix} \mu_0 \\ \mu_1 \end{bmatrix}, \Sigma_0 &\sim N_2\left(\begin{bmatrix} \mu_0 \\ \mu_1 \end{bmatrix}, \Sigma_0\right) \\ \tau_y^2 &\sim G(a_y, b_y) \\ \Sigma_\alpha^{-1} &\sim \text{Wishart}(R[2, 2], \rho)\end{aligned}$$

The *Wishart* density is new to us, and we will devote an entire subsection to it and its implementation in WinBUGS/OpenBUGS.

10.4.7 The Wishart Density

The Wishart distribution is the conjugate prior for the precision matrix of a multivariate normal distribution with known mean. It is the standard choice of prior for precision matrices in realistic multivariate normal-based models with means (and possibly many other parameters) unknown because it leads to a Wishart full conditional distribution for the precision matrix that simplifies MCMC-based model fitting. The two parameters of the Wishart distribution are a mean matrix and a scalar parameter called the degrees of freedom.

We have mentioned elsewhere in this textbook that many common distributions are parameterized differently by different authors and software packages. This is certainly (and confusingly) the case with the Wishart density. If X denotes a $p \times p$ symmetric, positive definite random matrix, R is a fixed $p \times p$ symmetric, positive definite matrix, ν is a strictly positive scalar, and the p.d.f. of X is

$$p(X|R, \nu) \propto |R|^{\frac{\nu}{2}} |X|^{\frac{\nu-p-1}{2}} \exp \left[-\frac{1}{2} \text{tr}(RX) \right] \quad (10.2)$$

then the references below define the two parameters as follows:

Reference	Parameterization
Spiegelhalter et al. (1995)	$X \sim \text{dwish}(R, \nu)$
Anderson (1984)	
Carlin and Louis (2000)	$X \sim \text{dwish}(R^{-1}, \nu)$
Gelman et al. (2004)	
Robert (2007)	
Box and Tiao (1973)	$X \sim \text{dwish}(R^{-1}, \nu - p + 1)$

In this textbook, we will use the WinBUGS/OpenBUGS parameterization. The Wishart distribution is proper if ν (the degrees of freedom) is $\geq p$ (the dimension of the matrix on which we are placing the Wishart prior). If $X \sim \text{dwish}(R, \nu)$, and X_{ij} is the entry in the i th column and j row of X , then the moments are as follows:

$$\begin{aligned} E(X_{ij}) &= \nu(R^{-1})_{ij} \\ \text{Var}(X_{ij}) &= \nu \left[(R^{-1})_{ij}^2 + (R^{-1})_{ii}(R^{-1})_{jj} \right] \\ \text{Cov}(X_{ij}, X_{kl}) &= \nu \left[(R^{-1})_{ik}(R^{-1})_{jl} + (R^{-1})_{il}(R^{-1})_{jk} \right] \end{aligned}$$

Note that the gamma distribution is a special (one-dimensional) case of the Wishart. If X and R are scalars, and the p.d.f. of X is proportional to $x^{\frac{\nu}{2}-1} \exp\left(-\frac{Rx}{2}\right)$ then

$$W(R, \nu) = G\left(\frac{\nu}{2}, \frac{R}{2}\right)$$

WinBUGS and OpenBUGS do not allow the use of their Wishart distribution with one-dimensional matrices, however.

If $X \sim dwish(R, \nu)$, then X^{-1} has an *inverse Wishart* distribution: $X^{-1} \sim IW(R, \nu)$, where

$$E(X_{ij}^{-1}) = \frac{R_{ij}}{\nu - p - 1}$$

The inverse Wishart distribution is always proper; however, it has a degenerate form if $\nu < p$, and obviously the first moment is negative or infinite unless $\nu > p + 1$.

Since statisticians and subject-matter experts tend to be better able to think in terms of variances and correlations rather than elements of precision matrices, the following way of specifying a prior on a covariance matrix, say Σ , in WinBUGS/OpenBUGS is attractive:

1. Let S equal your prior guess for the mean of the $p \times p$ variance/covariance matrix Σ .
2. Choose a degrees-of-freedom parameter ν ($> p + 1$) that roughly represents an “equivalent prior sample size”—your belief in S as the value of Σ is as strong as if you had seen ν previous vectors with sample covariance matrix S .
3. Define a matrix $\Omega = (\nu - p - 1)S$.
4. In WinBUGS, put a $Wishart(\Omega, \nu)$ prior on the corresponding precision matrix Σ^{-1} :

```
Sigmainv[1:p,1:p] ~ dwish( Omega[,], nu )
```

Then

- $E(\Sigma_{i,j}) = S_{i,j}$.
- The larger ν is, the smaller the prior variance is.
- $E(\Sigma_{i,j}^{-1}) = \frac{\nu}{\nu - p - 1}(S^{-1})_{i,j}$.

If you have used the above procedure and you wish to monitor the elements of the variance/covariance matrix Σ , then you must add a line to your program that uses the *inverse* function to invert the precision matrix:

```
Sigma[1:p,1:p] <- inverse(Sigmainv[,])
```

10.5 WinBUGS Examples for Hierarchical Normal Linear Models

The volumes of examples provided in the drop-down menus in WinBUGS and OpenBUGS include examples of hierarchical normal linear models with both the univariate and the bivariate formulations of the second and third stages. Their example dataset, used in [Gelfand and Smith \(1990\)](#), has the weights in grams of 30

baby rats who were weighed every 7 days from age 8 days to 36 days. The covariate is age in days at time of each weighing. The model gives each baby rat his or her own regression line (called a growth curve in this context) defined by an intercept and a slope. The parameter of greatest interest is the population mean growth rate (in grams per day) of baby rats of the species of interest being fed the diet used in the study. In my notation in Sects. 10.4.3 and 10.4.4, this parameter is β_1 .

10.5.1 Example with Univariate Formulation at Second and Third Stages

Below are the code, data, and initial values given in the WinBUGS Volume 1 example called “Rats.” I am quoting it in full here because as of this writing (January 2012), the version included in OpenBUGS does not include the alternative priors at the third stage.

```
model
{
  for( i in 1 : N ) {
    for( j in 1 : T ) {
      Y[i , j] ~ dnorm(mu[i , j],tau.c)
      mu[i , j] <- alpha[i] + beta[i] * (x[j] - xbar)
    }
    alpha[i] ~ dnorm(alpha.c,tau.alpha)
    beta[i] ~ dnorm(beta.c,tau.beta)
  }
  tau.c ~ dgamma(0.001,0.001)
  sigma <- 1 / sqrt(tau.c)
  alpha.c ~ dnorm(0.0,1.0E-6)

  # Choice of prior of random effects variances
  # Prior 1: uniform on SD
  sigma.alpha~ dunif(0,100)
  sigma.beta~ dunif(0,100)
  tau.alpha<-1/(sigma.alpha*sigma.alpha)
  tau.beta<-1/(sigma.beta*sigma.beta)

  #Prior 2: (not recommended)
  #tau.alpha ~ dgamma(0.001,0.001)

  beta.c ~ dnorm(0.0,1.0E-6)

  alpha0 <- alpha.c - xbar * beta.c
}

Data
list(x = c(8.0, 15.0, 22.0, 29.0, 36.0), xbar = 22, N = 30, T = 5,
     Y = structure(
       .Data = c(151, 199, 246, 283, 320,
                  145, 199, 249, 293, 354,
                  147, 214, 263, 312, 328,
                  155, 200, 237, 272, 297,
                  135, 188, 230, 280, 323,
                  159, 210, 252, 298, 331,
                  141, 189, 231, 275, 305,
                  159, 201, 248, 297, 338,
                  177, 236, 285, 350, 376,
                  134, 182, 220, 260, 296,
                  160, 208, 261, 313, 352,
```

```

143, 188, 220, 273, 314,
154, 200, 244, 289, 325,
171, 221, 270, 326, 358,
163, 216, 242, 281, 312,
160, 207, 248, 288, 324,
142, 187, 234, 280, 316,
156, 203, 243, 283, 317,
157, 212, 259, 307, 336,
152, 203, 246, 286, 321,
154, 205, 253, 298, 334,
139, 190, 225, 267, 302,
146, 191, 229, 272, 302,
157, 211, 250, 285, 323,
132, 185, 237, 286, 331,
160, 207, 257, 303, 345,
169, 216, 261, 295, 333,
157, 205, 248, 289, 316,
137, 180, 219, 258, 291,
153, 200, 244, 286, 324),
.Dim = c(30,5)))

#inits
# for model with prior 1
list(alpha = c(250, 250, 250, 250, 250, 250, 250, 250, 250,
               250, 250, 250, 250, 250, 250,
               250, 250, 250, 250, 250, 250, 250, 250, 250,
               250, 250, 250, 250, 250, 250),
      beta = c(6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
               6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6),
      alpha.c = 150, beta.c = 10,
      tau.c = 1, sigma.alpha = 1, sigma.beta = 1)

#for model with prior 2
list(alpha = c(250, 250, 250, 250, 250, 250, 250, 250, 250,
               250, 250, 250, 250, 250, 250,
               250, 250, 250, 250, 250, 250, 250, 250, 250,
               250, 250, 250, 250, 250, 250),
      beta = c(6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
               6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6),
      alpha.c = 150, beta.c = 10,
      tau.c = 1, tau.alpha = 1, tau.beta = 1)

```

In the WinBUGS symbols for the model, each baby rat has his or her own intercept and slope, α_i and β_i . Since the baby rats in the study are considered a random sample from the population of this type of rat fed the diet used in the study, the individual intercepts α_i are modeled as random draws from the population of intercepts of all such rats. Similarly, the individual slopes β_i are modeled as random draws from the population of slopes (growth rates) of all such rats. The mean of the population distribution of slopes, called β_c in the WinBUGS code, is the parameter of greatest interest.

Note that the covariate, age, is centered in this model. Thus, the intercept parameters (the α s and α_c) relate to the weights at the mean of the covariate values, which is 22 days.

In the WinBUGS notation, τ_α and τ_β are the precision parameters in the normal priors on the α 's and β 's, respectively. As mentioned in Sect. 10.4.5, placing very vague gamma priors on the precision parameters of these unobserved regression parameters leads to an almost improper posterior density and poor

MCMC convergence. Therefore, in the WinBUGS example, an alternative prior—specified on the standard deviations rather than on the precisions themselves—is recommended.

Note that using different parameterizations in the third-stage priors requires different sets of initial values. Initial values must be provided for parameters that have priors—not for deterministic transformations of those parameters. Thus, the first set of initial values given above, which goes with “Prior 1,” has initial values for the standard deviations σ_α and σ_β , whereas the second set of initial values (for “Prior 2”) has initial values for τ_α and τ_β .

10.5.2 Example with Bivariate Formulation at Second and Third Stages

The “Birats” example from Volume 2 of WinBUGS/OpenBUGS examples uses the same dataset on baby rats but illustrates the bivariate formulation of the priors in the second and third stages. Below are code, initial values, and data from this example, which is the same in both WinBUGS and OpenBUGS:

```
model
  for( i in 1 : N ) {
    beta[i , 1:2] ~ dnmnorm(mu.beta[ , ], R[ , ])
    for( j in 1 : T ) {
      Y[i , j] ~ dnorm(mu[i , j], tauC)
      mu[i , j] <- beta[i , 1] + beta[i , 2] * x[j]
    }
  }

  mu.beta[1:2] ~ dnmnorm(mean[ , ], prec[ , ])
  R[1:2 , 1:2] ~ dwish(Omega[ , ], 2)
  tauC ~ dgamma(0.001, 0.001)
  sigma <- 1 / sqrt(tauC)
}

# Inits
list(mu.beta = c(0,0), tauC = 1,
      beta = structure(
        .Data = c(100,6,100,6,100,6,100,6,100,6,
                  100,6,100,6,100,6,100,6,100,6,
                  100,6,100,6,100,6,100,6,100,6,
                  100,6,100,6,100,6,100,6,100,6,
                  100,6,100,6,100,6,100,6,
                  100,6,100,6,100,6,100,6,100,6),
        .Dim = c(30, 2)),
      R = structure(.Data = c(1,0,0,1), .Dim = c(2, 2)))

#Data

list(x = c(8.0, 15.0, 22.0, 29.0, 36.0), N = 30, T = 5,
      Omega = structure(.Data = c(200, 0, 0, 0.2), .Dim = c(2, 2)),
      mean = c(0,0),
      prec = structure(.Data = c(1.0E-6, 0, 0, 1.0E-6),
        .Dim = c(2, 2)),
      Y = structure(
        .Data = c(151, 199, 246, 283, 320,
                  145, 199, 249, 293, 354,
                  147, 214, 263, 312, 328,
```



```

155, 200, 237, 272, 297,
135, 188, 230, 280, 323,
159, 210, 252, 298, 331,
141, 189, 231, 275, 305,
159, 201, 248, 297, 338,
177, 236, 285, 350, 376,
134, 182, 220, 260, 296,
160, 208, 261, 313, 352,
143, 188, 220, 273, 314,
154, 200, 244, 289, 325,
171, 221, 270, 326, 358,
163, 216, 242, 281, 312,
160, 207, 248, 288, 324,
142, 187, 234, 280, 316,
156, 203, 243, 283, 317,
157, 212, 259, 307, 336,
152, 203, 246, 286, 321,
154, 205, 253, 298, 334,
139, 190, 225, 267, 302,
146, 191, 229, 272, 302,
157, 211, 250, 285, 323,
132, 185, 237, 286, 331,
160, 207, 257, 303, 345,
169, 216, 261, 295, 333,
157, 205, 248, 289, 316,
137, 180, 219, 258, 291,
153, 200, 244, 286, 324),
.Dim = c(30,5))

```

In this code, the i th baby rat's intercept-slope pair is denoted `beta[i, 1:2]`. This means that WinBUGS considers all these pairs as a matrix with 30 rows (the number of rats) and 2 columns. The likelihood defines the probability density function of each data point `Y[i, j]` given the individual rat's intercept and slope and the common precision `tauC`.

```

for( i in 1:N ) {
  for( j in 1 : T ) {
    Y[i , j] ~ dnorm(mu[i , j], tauC)
    mu[i , j] <- beta[i , 1] + beta[i , 2] * x[j]
  }
}

```

The second-stage prior on the β_i s is expressed in the lines

```

for (i in 1:N) {
  beta[i , 1:2] ~ dmnorm(mu.beta[], R[ , ])
}

```

`dmnorm` is the WinBUGS/OpenBUGS name for the multivariate normal density. `mu.beta` is the vector containing the population mean intercept and slope, so `mu.beta[2]` is the population slope of greatest substantive interest. `R[,]` is the precision matrix of the multivariate normal prior. Note that the dimension(s) of vectors or matrices must be specified for the quantity on the left side of the \sim character and are not needed for quantities on the right side. In this line of code, the "1:2" in `beta[i, 1:2]` tells WinBUGS that each `beta[i,]` is a vector of length 2, so the mean of its multivariate normal density (`mu.beta[]`) must also be of length 2, and the precision matrix `R[,]` must be 2 by 2.

The third-stage prior densities are expressed in the following lines (which are outside of any `for` loop):

```
mu.beta[1:2] ~ dmnorm(mean[], prec[ , ])
R[1:2 , 1:2] ~ dwish(Omega[ , ], 2)
tauC ~ dgamma(0.001, 0.001)
```

Since this is the last stage of the model, all of the parameters in these priors must be known constants. The easiest way to assign constant values to vectors and matrices in OpenBUGS/WinBUGS is to put those numbers in the data list rather than the model code. That is what was done in this example, in the following lines from the data list:

```
Omega = structure(.Data = c(200, 0, 0, 0.2), .Dim = c(2, 2)),
mean = c(0,0),
prec = structure(.Data = c(1.0E-6, 0, 0, 1.0E-6), .Dim = c(2, 2)),
```

Note how a matrix is defined within a `structure` function. The `.Data` component contains the numeric values in order row by row. The `.Dim` component provides the number of rows followed by the number of columns.

In choosing the numeric values for the Ω matrix, the authors of the example did not follow the procedure that I recommended in Sect. 10.4.7. Instead they opted to make their prior as noninformative as possible while still keeping it proper; therefore, they set the degrees-of-freedom parameter equal to 2—the same as the dimension of the matrix. The combination of this minimally informative prior with the lack of covariate centering leads to slow convergence of the MCMC sampler for this problem.

Problems

10.1. Rerun the WinBUGS/OpenBUGS Bayesian regression from Sect. 10.2.2. Run 3 parallel chains (make up your own additional sets of initial values), and run them for at least 5,000 iterations.

1. Verify that convergence is immediate using the convergence-assessment methods that you know.
2. Compare your Bayesian estimates of the regression coefficients to the frequentist results given in Sect. 10.1.3.
3. Compare your estimate posterior probability that $\beta_1 > 0$ to the frequentist p-value. In a one-sided hypothesis test like this one, when the standard noninformative prior is used in the Bayesian analysis, these two probabilities should be equal. Explain how their interpretations differ.

10.2. Carry out a new Bayesian regression analysis of the Brule River MDN data, this time with informative priors on the regression coefficients. Assume that you had previous information that the population intercept is likely to be in the interval (1.5,

2.5) and the population slope in the interval $(-0.5, 0.5)$. Convert this information into normal prior densities on the appropriate parameters. Use a vague prior on the precision. Run the new model, and comment on how results differ from those in the previous problem.

10.3. The data for this problem, in WinBUGS/OpenBUGS format, may be downloaded from the textbook web page. The file is called `MDN6sites.txt`. It presents annual means of log mercury concentration for 6 sites monitored by the Mercury Deposition Network for each year in 1997 through 2010. Because the sites are hundreds of miles apart (one in each of 6 regions of the U.S.), we do not need to worry about spatial correlation among the values.

Use WinBUGS or OpenBUGS to fit a hierarchical normal linear model to these data, giving each site its own intercept and slope on year. You may use the WinBUGS examples discussed in Sect. 10.5 as templates. Your instructor will tell you whether to use the univariate formulation (the “Rats” example) or the bivariate formulation (the “Birats” example).

You will need to choose different initial values and possibly different priors for your analysis from those that worked for the baby rats data.

1. Fit a hierarchical normal linear model to the data. Turn in your code, data list, and initial values list.
2. Assess convergence in the ways that you have learned. You do not have to print any plots; just write a few sentences telling what you did.
3. Produce estimated posterior means and 95% credible sets for the following quantities:
 - a. The estimated population intercept of mean log mercury concentration for all sites at the beginning of the observation period—1997. (If you center the covariate, think about how to do this.)
 - b. The estimated population slope of mean log mercury concentration on year for the entire continent.
 - c. The standard deviation of mean log mercury concentration around the site-specific regression lines.
 - d. The standard deviation that captures between-site variability in slopes.
 - e. The individual intercept and slope for site 2.

Chapter 11

Model Comparison, Model Checking, and Hypothesis Testing

“All models are wrong. Some models are useful.” This well-known quotation by the statistician George E. P. Box captures one challenge facing the applied statistician: how to determine whether the statistical model(s) entertained to address a particular research question is not wrong in ways that lead to useless or incorrect conclusions regarding important aspects of the question at hand.

In most real data analysis situations, researchers consider several statistical models that might be appropriate for the application. They establish criteria for determining which of the candidate models is best, and whether even that model is good enough to use as the basis for inference. This chapter explores Bayesian methods of comparing models, testing hypotheses, and assessing model adequacy. Specifically, we look at two Bayesian tools for model comparison—Bayes factors (Kass and Raftery 1995) and the more recently proposed Deviance Information Criterion (Spiegelhalter et al. 2002). We then see how to apply posterior predictive model checking (Gelman et al. 1996) to determine whether a chosen model is adequate for the research purpose.

11.1 Bayes Factors for Model Comparison and Hypothesis Testing

We first will investigate *Bayes factors*, which have a long history in Bayesian model comparison and hypothesis testing.

11.1.1 Bayes Factors in the Simple/Simple Case

Let’s first consider the most straightforward application of Bayes factors—that of making a decision between two models for the state of the world, or equivalently,

Table 11.1 Prior likelihood, and posterior probabilities

Model	Prior probabilities	Likelihood for M+	Prior × likelihood	Posterior probabilities
M_0 No breast cancer	0.9955	0.0274	0.0273	0.893
M_1 Breast cancer	0.0045	0.724	0.0033	0.107

between two simple hypotheses about an unknown parameter. Recall that a *simple hypothesis* is a statement that a parameter takes on a specific value.

As an example, let’s revisit the breast cancer problem from Chap. 1. Recall that the two possible states of the world (or *models*) were that my friend had breast cancer and that she did not have breast cancer (with prior probabilities 0.0045 and 0.9955, respectively). Furthermore, the probability of a positive screening mammogram was 0.724 for a woman who has breast cancer and 0.0274 for a woman who does not. Table 11.1 is a rearrangement of Table 1.3, in which we used Bayes’ theorem to move from the prior probabilities of the two models, through the likelihood, to the posterior probabilities.

Since the model (breast cancer or not) determines the probability of a positive mammogram, this problem can be cast equivalently as a hypothesis test about the parameter representing this probability. If we call the parameter π , then the hypotheses would be written as

$$H_0 : \pi = 0.0274$$

$$H_1 : \pi = 0.7240$$

The prior probabilities on the two models can also be interpreted as the prior probabilities on the two possible values of π and hence on the null and alternative hypotheses. Both H_0 and H_1 are *simple* hypotheses, because each states that the unknown parameter is equal to a single numeric value.

11.1.1.1 Prior Odds and Posterior Odds

The notion of odds is needed as preparation for Bayes factors. In probability as in horse racing, the *odds* in favor of any event or statement is a ratio—the probability that the event occurs (or the statement is true) over the probability that the event does not occur (or the statement is false).

In a Bayesian analysis in which there are only two possible states of the world, M_0 and M_1 (or equivalently, two simple hypotheses, H_0 and H_1) to compare, the prior probability $Pr(M_0) = 1 - Pr(M_1)$. Thus, the *prior odds* in favor of M_1 (or H_1) are

$$\frac{Pr(M_1)}{Pr(M_0)} = \frac{Pr(H_1)}{Pr(H_0)}$$

In the mammogram example, the prior odds in favor of my friend having breast cancer (or of the probability of her mammogram coming out positive being 0.7240) are

$$\frac{Pr(M_1)}{Pr(M_0)} = \frac{Pr(H_1)}{Pr(H_0)} = \frac{0.0045}{0.9955} = 0.00452$$

The *posterior odds* in favor of a model or a hypothesis are the analogous ratio of posterior probabilities:

$$\frac{Pr(M_1|y)}{Pr(M_0|y)} = \frac{Pr(H_1|y)}{Pr(H_0|y)}$$

where y represents the observed data.

In the mammogram example, the posterior odds in favor of my friend having breast cancer (or of $\pi = 0.7240$) are

$$\frac{Pr(M_1|+)}{Pr(M_0|+)} = \frac{Pr(H_1|+)}{Pr(H_0|+)} = \frac{0.107}{0.893} = 0.120$$

where “+” indicates that the data value was a positive test.

11.1.1.2 The Bayes Factor

The Bayes factor in favor of a model or hypothesis is the ratio of the posterior odds to the prior odds. Thus, the Bayes factor in favor of Model 1 versus Model 0 is

$$BF_{1,0} = \frac{\frac{Pr(M_1|y)}{Pr(M_0|y)}}{\frac{Pr(M_1)}{Pr(M_0)}} \quad (11.1)$$

In the simple/simple case, (11.1) simplifies:

$$\begin{aligned} BF_{1,0} &= \frac{\frac{Pr(M_1|y)}{Pr(M_0|y)}}{\frac{Pr(M_1)}{Pr(M_0)}} \\ &= \frac{\frac{Pr(M_1)Pr(y|M_1)}{Pr(M_0)Pr(y|M_0)}}{\frac{Pr(M_1)}{Pr(M_0)}} \\ &= \frac{Pr(y|M_1)}{Pr(y|M_0)} \end{aligned} \quad (11.2)$$

That is, in the simple/simple case, the Bayes factor is the ratio of the likelihoods under the two competing models or hypotheses. In other words, it is the evidence contained in the data alone (uninfluenced by the prior) in favor of one model or the other.

In the mammogram example, (11.1) yields the following value of the Bayes factor in favor of Model 1:

$$BF_{1,0} = \frac{0.120}{0.00452} = 26.5$$

while (11.2) gives

$$BF_{1,0} = \frac{0.0724}{0.0274} = 26.4$$

The small difference is due to rounding error.

11.1.2 Interpreting a Bayes Factor

Recall that $BF_{1,0}$ in the simple/simple case is the weight of evidence contained *in the data alone* in favor of M_1 versus M_0 . Thus, it ignores any evidence provided by the prior. (In the mammogram example, the prior information concerns the extremely small proportion of women who actually have breast cancer when they have their first screening mammogram). The Bayes factor usually is reported on the \log_{10} scale. A review paper by Kass and Raftery (1995) recommends the interpretations of intervals of values of the Bayes factor shown in Table 11.2:

For example, the Bayes factor in the mammogram problem would provide strong evidence against the null hypothesis of no breast cancer.

11.1.3 The Bayes Factor in More General Models

In general Bayesian models, calculation of the Bayes factor is usually conceptually, but not computationally, straightforward.

First, remember that, in general models, Bayes rule, stated in terms of equality rather than proportionality, says

$$p(\theta|\mathbf{y}) = \frac{p(\theta)p(\mathbf{y}|\theta)}{\int p(\theta)p(\mathbf{y}|\theta)d\theta}$$

where θ denotes the vector of unknown parameters in the model.

The denominator—the normalizing constant required to make the joint posterior density integrate to 1—is

$$\begin{aligned}
& \int p(\theta)p(\mathbf{y}|\theta)d\theta \\
&= \int p(\theta, \mathbf{y})d\theta \\
&= p(\mathbf{y})
\end{aligned}$$

This denominator is called the *marginal likelihood*. It does not depend on any parameter values, since the parameters have been integrated out. The numeric value of the marginal likelihood is determined by the data and the entire Bayesian model (the form of the likelihood and all levels of priors). It is intuitively clear that if two different Bayesian models are fit to the same data, the model with the larger marginal likelihood is more consistent with the data.

Suppose we wish to compare models M_1 and M_0 for the same data. The two models may have different likelihoods, different numbers of unknown parameters, etc. For example, we might want to compare a model with a normal likelihood and priors on the unknown mean and variance to a model with a t likelihood and priors on the unknown mean, scale parameter, and degrees of freedom parameter, or we might want to compare two non-nested regression models.

The Bayes factor in the general case is the ratio of the marginal likelihoods under the two candidate models. That is, if θ_1 and θ_0 denote parameters under models M_1 and M_0 respectively, then

$$\begin{aligned}
p(\mathbf{y}|M_1) &= \int p(\theta_1)p(\mathbf{y}|\theta_1)d\theta_1 \\
p(\mathbf{y}|M_0) &= \int p(\theta_0)p(\mathbf{y}|\theta_0)d\theta_0 \\
BF_{10} &= \frac{p(\mathbf{y}|M_1)}{p(\mathbf{y}|M_0)}
\end{aligned}$$

In this case, the Bayes factor cannot be interpreted as the evidence in the data alone, because clearly the priors affect each marginal likelihood and therefore the Bayes factor itself.

In realistically complex models, the integrations required to calculate marginal likelihoods (and therefore, Bayes factors for model comparison) analytically are infeasible. Some approaches to using MCMC output to approximate marginal likelihoods and Bayes factors are discussed in [Carlin and Chib \(1995\)](#); [Chib \(1995\)](#); [Chib and Jeliazkov \(2001\)](#), and [Cowles \(2003\)](#). Carlin and Chib's method, the only one of these that can be implemented straightforwardly in WinBUGS/OpenBUGS, is illustrated in the "Pines" example from Volume III of the examples included in OpenBUGS and on the webpage <http://www.mrc-bsu.cam.ac.uk/bugs/documentation/exampVol2/node20.html>.

11.2 Bayes Factors and Bayesian Hypothesis Testing

The more general form of the Bayes factor also can be used in Bayesian hypothesis testing. Here, only one model is being considered, and the goal is to compare two different sets in which the parameter or parameters could lie. Generically, if θ represents the parameter of interest in a model and Θ_0 and Θ_1 are two nonoverlapping regions that cover the entire possible range of values of θ , then

$$H_0 : \theta \in \Theta_0$$

$$H_1 : \theta \in \Theta_1$$

Carrying out this Bayesian hypothesis test involves calculating posterior probabilities:

$$P(\theta \in \Theta_0 | \mathbf{y})$$

$$P(\theta \in \Theta_1 | \mathbf{y})$$

To illustrate, we revisit the simplified example of mercury in fish tissue from Sect. 6.2.8, for which the Bayes factor can be computed easily. Recall that we had data on log-transformed concentrations of mercury in 21 samples of tissue from fish caught in the Des Moines River. Furthermore, the Natural Resources Defense Council defines low mercury concentration in fish as less than 0.09 parts per million, or less than -2.41 on the log scale. We wish to use our data to test the null hypothesis that the population mean μ of mercury concentration in fish in the Des Moines River falls into that “low” category. Thus, our hypotheses are

$$H_0 : \mu \leq -2.41$$

$$H_1 : \mu > 2.41$$

In Sect. 6.2.8, we assumed that our sample data y_i , $i = 1, \dots, 21$ were draws from a normal distribution with unknown mean μ and known variance $\sigma^2 = \frac{1}{2.5}$. Thus, the likelihood could be expressed as

$$\bar{y} | \mu \sim N\left(\mu, \frac{1}{21 \times 2.5}\right)$$

We specified a conjugate normal prior on μ :

$$\mu \sim N\left(-2.75, \frac{1}{7.5}\right)$$

This prior density induces prior probabilities on the two hypotheses:

Table 11.2 Kass and Raftery’s table for interpreting Bayes factors

$\log_{10}(B_{10})$	$B_{1,0}$	Evidence against H_0 (or M_0)
0 to 1/2	1 to 3.2	Not worth more than bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
> 2	> 100	Decisive

$$Pr(H_0) = Pr(\mu < -2.41) = 0.8241$$

$$Pr(H_1) = Pr(\mu \geq -2.41) = 0.1759$$

These probabilities can be obtained with R:

```
> pnorm( -2.41, -2.75, sqrt(1/7.5) )  
[1] 0.8241064  
> 1- pnorm( -2.41, -2.75, sqrt(1/7.5) )  
[1] 0.1758936
```

Combining the observed data $\bar{y} = -2.563$ with the prior produces the following posterior density:

$$\mu|y \sim N\left(2.586, \frac{1}{60}\right)$$

The posterior probabilities on the two hypotheses also can be calculated in R:

```
pnorm( -2.41, -2.586, sqrt(1/60) )  
[1] 0.9136045  
> 1-pnorm( -2.41, -2.586, sqrt(1/60) )  
[1] 0.08639554
```

giving

$$Pr(H_0|y) = Pr(\mu \leq -2.41|y) = 0.9136$$

$$Pr(H_1|y) = Pr(\mu > -2.41|y) = 0.0864$$

The conclusion is that, given the prior information and the current data, the probability that $\mu < -2.41$ log units is 0.914.

The Bayes factor is calculated by dividing the posterior odds by the prior odds:

$$BF_{0,1} = \frac{\frac{0.9136}{0.0864}}{\frac{0.8241}{0.1759}} = 2.257$$

According to Table 11.2, this result is worth only a bare mention: Incorporating the data did little to strengthen the evidence already present in the prior regarding the hypotheses.

Note that if an improper prior is used, it is impossible to calculate the prior odds needed in computing the Bayes factor.

11.2.1 *Obtaining Posterior Probabilities from WinBUGS/OpenBUGS*

The `step` function in WinBUGS and OpenBUGS makes it easy to estimate the posterior probability that a model parameter is above or below a value of interest. This function returns 1 if its argument is greater than or equal to 0 and returns 0 otherwise.

The code below fits the model from Sect. 11.2 and estimates the posterior probability $Pr(H_0|\mathbf{y}) = Pr(\mu < -2.41|\mathbf{y})$. This is the OpenBUGS program called “Model 1” in Sect. 8.4.7 with one additional line:

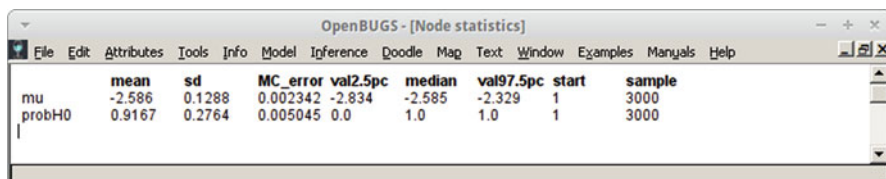
```
probH0 <- step( -2.41 - mu )    # test H0: mu <= -2.41
```

At each iteration of the MCMC sampler, the current draw for μ is subtracted from -2.41 , and the value 1 is assigned to `probH0` if the result is 0 or greater; 0 is assigned otherwise. Thus, the mean of `probH0` will be the proportion of the iterations in which μ was less than or equal to -2.41 .

```
# Model 1
# Assuming data are draws from a normal
# population with known precision
# Population mean mu is unknown parameter

model
{
  # likelihood
  for (i in 1:N) {
    y[i] ~ dnorm( mu, tausq )
  }
  # priors
  mu ~ dnorm(-2.75, 7.5)
  probH0 <- step( -2.41 - mu )    # test H0: mu <=
                                -2.41
}

#data
list(y=c(-2.526, -1.715, -1.427, -2.12, -2.659,
        -2.408, -3.219,
        -1.966, -2.526, -1.833, -2.813, -1.772, -2.813,
        -2.526, -3.219,
        -2.526, -2.813, -2.526, -3.507, -2.996, -3.912, NA),
     N=22,
     tausq= 2.5)
```



	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
mu	-2.586	0.1288	0.002342	-2.834	-2.585	-2.329	1	3000
probH0	0.9167	0.2764	0.005045	0.0	1.0	1.0	1	3000

Fig. 11.1 OpenBUGS statistics showing posterior probability of null hypothesis

```
#inits for model 1
list(mu = -5)
list(mu = -2.5)
list(mu = 0)
```

The OpenBUGS statistics show the posterior probability of H_0 as 0.917 (Fig. 11.1). The small difference between this value and the value obtained from R (0.914) is due to MCMC random sampling variability.

11.2.2 Bayesian Viewpoint on Point Null Hypotheses

Two-sided hypothesis testing, typically with a *point null hypothesis*, is very common in frequentist statistical practice. Here, the null hypothesis is that the unknown parameter has some particular numeric value of interest, and the alternative hypothesis is that it does not. Generically

$$H_0 : \theta = \theta_0$$

$$H_1 : \theta \neq \theta_0$$

where θ is the unknown parameter and θ_0 is a number. For example, I was taught in elementary school health class that the normal body temperature for healthy adults is 98.6 °F. To test whether this was true, I might let μ represent the population mean body temperature in healthy adults. I might measure the temperatures of a random sample of healthy people and use the data to test the hypotheses:

$$H_0 : \mu = 98.6$$

$$H_1 : \mu \neq 98.6$$

Bayesians observe that the point null hypothesis actually is unlikely to be of substantive interest. Almost certainly there is an interval of values around θ_0 —say, $[\theta_0 - \varepsilon, \theta_0 + \varepsilon]$ for some small, positive value of ε —such that, if the parameter θ lies in that interval, it “might as well be equal” to θ_0 . Exactly how the relevant interval is

defined depends entirely on the research question of interest. For example, regarding the body temperature question, if the true values of μ were between 98.55 and 98.65, I certainly wouldn't feel that my health class teacher had been lying!

From the Bayesian perspective, there also is a mathematical problem with testing point null hypotheses. If the parameter of interest is being treated as if it were a continuous random variable, then the probability that it is exactly equal to any specific numeric value is 0. In the body temperature example, if a normal prior was placed on μ , then the prior probability that $\mu = 98.6$ would be 0. We wouldn't even have to collect any data—we would already know that the null hypothesis was false!

To demonstrate how different frequentist p-values are from Bayesian posterior probabilities, in (Berger, 1985, Sect. 4.10), Berger describes a procedure that enables Bayesian two-sided tests of point null hypotheses. The idea is to divide the prior probability mass into two parts that sum to one: a discrete part that is a lump of probability mass on the value specified in H_0 , and continuous part for the rest of the support of the unknown parameter. This procedure is not often used in Bayesian practice and is not further explored here.

11.3 The Deviance Information Criterion

Since MCMC is the most frequently used computational approach to fitting Bayesian models, Bayesian statisticians need a model-comparison method that is based on MCMC output. The Deviance Information Criterion Spiegelhalter et al. (2002) meets that requirement, and in fact, can be calculated within WinBUGS/OpenBUGS for most models. The following section of the WinBUGS webpage gives details about the DIC and its implementation in WinBUGS and OpenBUGS:

<http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/dicpage.shtml>.

The goal of the DIC is to compare candidate models with respect to their ability to predict new data of the same kind. The DIC protects against overfitting (excessive model complexity driven by peculiarities of the particular dataset being used) by penalizing models with larger numbers of *effective parameters*. The number of effective parameters is essentially a count of the number of unknown parameters in the model, except that highly correlated parameters and parameters that are strongly influenced by their priors count for less than one each. When comparing different candidate models for the same data, smaller values of DIC suggest better predictive ability. It is not possible to interpret whether a single DIC value for one model in isolation is “good” or “bad.”

The DIC can be used for many kinds of comparisons including regression models with completely different sets of predictor variables, generalized linear models with different link functions, and Bayesian models with different numbers of stages. One restriction is that the data (in the case of regression models, the response variable) must have the same form in all models being compared. For example, the DIC could not be used to compare a regression model in which the response variable was

log transformed with a regression model for the same data in which the response variable was not transformed.

Computation of the DIC requires two quantities, both of which are easily approximated using MCMC sampler output. Let $D(\mathbf{y}, \theta)$ denote -2 times the log likelihood conditioned on a particular value of the model parameters—that is, $-2\log p(\mathbf{y}|\theta)$, where θ is the vector of all unknown parameters in the model. Define

- $\hat{D}_{avg}(\mathbf{y})$: D averaged over the posterior distribution of θ
- $D_{\hat{\theta}}(\mathbf{y})$: D evaluated at the posterior mean of θ

Then the effective number of parameters is estimated as

$$p_D = \hat{D}_{avg}(\mathbf{y}) - D_{\hat{\theta}}(\mathbf{y})$$

and the DIC is

$$\begin{aligned} DIC &= \hat{D}_{avg}(\mathbf{y}) + p_D \\ &= 2\hat{D}_{avg}(\mathbf{y}) - D_{\hat{\theta}}(\mathbf{y}) \end{aligned}$$

The “Dyes” problem that you encountered in Problem 9.5 provides a good example of how the DIC works. Recall that the data consisted of measurements on five samples from each of six batches of dyestuff. The question of interest was how the variability within batches compared to the variability between batches. We will use the DIC to compare the hierarchical model given in the WinBUGS/OpenBUGS example with a simple two-stage normal model that ignores the grouping of the observations.

Here are the code, data, and initial values as they appear in the OpenBUGS Dyes example:

```
model
{
  for(i in 1 : batches) {
    mu[i] ~ dnorm(theta, tau.btw)
    for(j in 1 : samples) {
      y[i , j] ~ dnorm(mu[i], tau.with)
    }
  }
  sigma2.with <- 1 / tau.with
  sigma2.btw <- 1 / tau.btw
  tau.with ~ dgamma(0.001, 0.001)
  tau.btw ~ dgamma(0.001, 0.001)
  theta ~ dnorm(0.0, 1.0E-10)
}

# data
```

Table 11.3 DIC for the hierarchical model for the Dyes data

	Dbar	Dhat	DIC	pD
y	323.4	316.9	329.9	6.463
Total	323.4	316.9	329.9	6.463

```
list(batches = 6, samples = 5,
y = structure(
.Data = c(1545, 1440, 1440, 1520, 1580,
          1540, 1555, 1490, 1560, 1495,
          1595, 1550, 1605, 1510, 1560,
          1445, 1440, 1595, 1465, 1545,
          1595, 1630, 1515, 1635, 1625,
          1520, 1455, 1450, 1480, 1445), .Dim = c(6, 5)))

# initial values
list(theta=1500, tau.with=1, tau.btw=1)
list(theta=3000, tau.with=0.1, tau.btw=0.1)
```

Let's count how many parameters there are in this model. This means how many quantities have prior distributions at any level of the model. It does *not* include quantities that are deterministically computed. The answer is one overall mean parameter (`theta`), six batch means parameters (the `mus`), and two precisions (`tau.with` and `tau.btw`), for a grand total of nine parameters.

To obtain the Deviance Information Criterion in WinBUGS or OpenBUGS, one must first run the chains for the model until it appears that they have converged. Then one selects "DIC" from the "Inference" pull-down menu, and clicks the "Set" button to turn it on. Then one runs additional iterations, after which one clicks "Stats" on the DIC Tool window to get the values.

I ran two chains for 25,000 iterations, then started the DIC and ran 25,000 additional iterations. The results are in Table 11.3.

Note that the DIC is calculated for every quantity that WinBUGS/OpenBUGS interprets as data. Then the columns are totaled. Since `y` is our only data vector, the total is the same as the entry for `y`. The estimated effective number of parameters, `pD`, is 6.463—smaller than the total number that we counted. This is because each of the `mus` is counted as less than one parameter because of the influence of their second-stage prior. The value that will be used for model comparison is the DIC itself—329.9.

Below is a simple, two-stage model for the same data:

```
model
{
  for(i in 1 : batches) {
    for(j in 1 : samples) {
      y[i , j] ~ dnorm(theta, tau.with)
    }
  }
}
```

Table 11.4 DIC for the simplified model for the Dyes data

	Dbar	Dhat	DIC	pD
y	334.8	332.7	336.8	2.027
Total	334.8	332.7	336.8	2.027

```

sigma2.with <- 1 / tau.with
tau.with ~ dgamma(0.001, 0.001)
theta ~ dnorm(0.0, 1.0E-10)
}

# inits
list(theta=1500, tau.with = 0.1)
list(theta = 0, tau.with = 10)

```

The DIC results obtained from iterations 2,001–4,000 are in Table 11.4:

In a simple model like this, in which there are no groups of parameters all drawn from the same prior, the number of effective parameters pD is almost exactly equal to the actual number of parameters in the model (2 in this case). The DIC for this simple model, 336.8, is larger than the DIC obtained for the hierarchical model. That means that even though the hierarchical model is penalized more heavily for complexity (larger pD than the simpler model), its predictive ability is deemed to be so much greater that it is still preferred. In a hierarchical model like this, the DIC is evaluating how well the model would predict new data in the same groupings, so the result is not surprising.

11.4 Posterior Predictive Checking

As we mentioned at the beginning of this chapter, almost no model is ever truly “correct.” There always are predictor variables that we did not know about, characteristics of the population distribution that cannot be captured by any standard parametric family, etc. Thus, even after we have identified the “best” model among a set of candidate models, we still must attempt to determine whether the selected model has flaws that will threaten the validity of the inference that we wish to make.

Posterior predictive model checking [Gelman et al. \(2004, 1996\)](#) is a method of assessing whether a model is inconsistent with the data in ways that are likely to affect the conclusions drawn. It is easily carried out in WinBUGS or OpenBUGS. The idea is to determine whether datasets generated by the model resemble the real dataset in crucial ways. The specific steps are as follows:

1. Identify a *discrepancy measure* (also called a *test quantity*). This is a quantity that (a) can be calculated from data values (or possibly from data and parameter values) and (b) would be expected to have systematically different values in data that met important model assumptions compared to data that did not.

2. Calculate the value of the discrepancy measure from the real dataset that is being analyzed. Denote this value by $T(\mathbf{y})$.
3. Fit the Bayesian model of interest to the real data, and simulate a large number of *replicate datasets* from the posterior predictive distribution. A replicate dataset has the same number of observations as the real dataset. If the model of interest involves a response variable and predictor variables, then each replicate dataset has the same values of the predictor variables that appeared in the real dataset, and new values are simulated only for the response variable. Denote the i th replicate dataset by \mathbf{y}_i^{rep} .
4. Calculate the value of the discrepancy measure $T(\mathbf{y}_i^{rep})$ (for each replicate dataset).
5. Determine the proportion of the $T(\mathbf{y}_i^{rep})$ values that are smaller than $T(\mathbf{y})$. This proportion is an estimate of the *Bayesian posterior predictive p-value*.

Intuitively, if the model is consistent with the data, then the discrepancy measure value from the real data, $T(\mathbf{y})$, will be typical of discrepancy measure values from datasets drawn from the posterior predictive density—that is, in a histogram of the $T(\mathbf{y}_i^{rep})$ s, $T(\mathbf{y})$ will not fall near the extreme low end nor the extreme high end. In other words, a posterior predictive p-value close to either 0 or 1 is evidence of serious disagreement between the data and the model: It indicates that a better model is needed for trustworthy inference.

Different discrepancy measures will be appropriate for assessing data and model characteristics in different problems. For example, to check whether outliers in a dataset are too extreme to justify the use of a normal likelihood, the *range* of the data (maximum minus minimum) could be an effective discrepancy measure. Below, we devise a discrepancy measure to assess whether the assumption of independent errors in linear regression is violated.

In Sect. 10.1.3, we performed linear regression using data on mean log mercury concentration measured in each of 14 consecutive months. The fact that the data were collected sequentially in time raised the question of whether there would be autocorrelation in the errors, thus violating the assumption of independent errors. In Sect. 10.1.3, we used the Durbin–Watson test to do a frequentist check for autocorrelation. Here we use posterior predictive checking to perform a similar check from a Bayesian perspective. The lag 1 autocorrelation of the residuals is an ideal discrepancy measure for this purpose, and it can be calculated in WinBUGS/OpenBUGS code with a little effort. Recall that a sample Pearson correlation coefficient can be calculated from the formula

$$r = \frac{1}{(N-1)s_x s_y} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

where N is the sample size, \bar{x} and \bar{y} are the sample means, and s_x and s_y are the sample standard deviations of the two variables of interest. For a lag 1 autocorrelation, the X variable values are observations 1 to $N-1$, and the Y variable values are observations 2 to N of the same vector. WinBUGS and

OpenBUGS have built-in functions (mean and sd) for calculating sample means and standard deviations. Here is an expanded version of the OpenBUGS program from Sect. 10.2.2, which provides a Bayesian posterior predictive p-value for checking whether the autocorrelation in the residuals from the real data “looks like” what we would find in data generated by the simple linear regression model:

```

model
{
  # center covariate
  for( i in 1:N) {
    xcent[i] <- x[i] - mean(x[])
  }

  # likelihood
  for (i in 1:N) {
    mu[i] <- beta0 + beta1 * xcent[i]
    y[i] ~ dnorm( mu[i], tausq )
    resid[i] <- y[i] - mu[i]

    # draw a replicate dataset
    yrep[i] ~ dnorm(mu[i], tausq)
    residrep[i] <- yrep[i] - mu[i]
  }

  # calculate lag 1 autocorrelation in residuals from
  # real data
  mean1 <- mean( resid[1:(N-1)] )
  mean2 <- mean( resid[2:N] )
  for(i in 1:(N-1)) {
    summand[i] <- (resid[i] - mean1) *
      (resid[i+1] - mean2 )
  }
  lag1auto <- sum( summand[] ) / ( (N-1) *
    sd( resid[1:(N-1)]) * sd( resid[2:N] ) )

  # calculate lag 1 autocorrelation in replicate dataset

  mean1rep <- mean( residrep[1:(N-1)] )
  mean2rep <- mean( residrep[2:N] )
  for(i in 1:(N-1)) {
    summandrep[i] <- (residrep[i] - mean1rep) *
      residrep[i+1] - mean2rep )
  }
}

```

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
lag1auto	0.1194	0.09205	0.00414	0.04634	0.08021	0.3742	1001	1000
lag1autorep	-0.06266	0.2405	0.006584	-0.5214	-0.05266	0.3947	1001	1000
pppval	0.762	0.4259	0.01068	0.0	1.0	1.0	1001	1000

Fig. 11.2 Posterior predictive p-value for the regression model for mercury concentrations in precipitation

```

lag1auto <- sum( summandrep[] ) / ( (N-1) *
  sd( residrep[1:(N-1)] ) * sd( residrep[2:N] ) )

# is lag1auto > lag1autorep?
pppval <- step(lag1auto - lag1autorep)

beta0 ~ dflat()
beta1 ~ dflat()
tausq ~ dgamma( 0.001, 0.001)
sigma <- 1 / sqrt(tausq)          # regression standard
                                deviation
}

#data
list( x = c( 1997, 1998, 1999, 2000, 2001, 2002, 2003,
             2004, 2005, 2006,
             2007, 2008, 2009, 2010),
      y = c(2.2952, 2.3435, 2.5512, 2.5531, 2.3918,
             2.1546, 2.3596,
             2.2431, 2.1725, 2.3162, 2.3504, 2.1926,
             1.9638, 2.2025),
      N = 14)

# inits
list( beta0 = 0, beta1 = 0, tausq = 1)

```

Figure 11.2 shows the results for the quantities needed for the posterior predictive check (from the last 1,000 iterations of a 2,000-iteration chain). The autocorrelation in the residuals from the real data is generally positive but small, while the autocorrelation in the residuals from replicate datasets centers around 0 (as expected). The mean of `pppval` is 0.762, indicating that about 76% of the replicate datasets produced residuals with smaller autocorrelation than the corresponding residuals from the real dataset. Although a value closer to 0.5 would be preferred, 0.76 is not

extreme—the real data is not atypical of data generated from this model. A value greater than 0.90 or 0.95 would have indicated a need to expand the model to account for autocorrelation in the data.

11.5 Exercises

11.1. This question is a continuation of your first homework for this class.

Refer to question 2.3 in this textbook. The question of interest is whether or not the woman is a carrier of the hemophilia gene.

Suppose the woman has 1 son, whose hemophilia status is $y = 0$. (This is different from the outcomes discussed in this book.)

Compute:

1. Prior odds in favor of $\theta = 0$ versus $\theta = 1$
2. Bayes factor in favor of $\theta = 0$ versus $\theta = 1$
3. Posterior odds in favor of $\theta = 0$ versus $\theta = 1$

11.2. Attached is WinBUGS code and output for the Dyes example, which you know from a previous homework. I have added 4 lines to the program. You will need to look up the `ranked` function in the WinBUGS or OpenBUGS manual to understand what it does. Refer to the code and output to answer the following questions (just a sentence or two for each):

1. Explain the meaning of the following new nodes:
 - a. `ypred`
 - b. `resid`
 - c. `presid`
 - d. `pppv`
2. What could you learn by monitoring “pppv”?
3. Does the output suggest any problems with model fit?

```
model
{
  for( i in 1 : batches ) {
    m[i] ~ dnorm(theta, tau.btw)
    for( j in 1 : samples ) {
      y[i , j] ~ dnorm(m[i], tau.with)
      ypred[i,j] ~ dnorm(m[i], tau.with)
      resid[i,j] <- abs(y[i,j] - m[i])
      presid[i,j] <- abs(ypred[i,j] - m[i])
    }
    large[i] <- ranked(resid[i,], samples)
    largepred[i] <- ranked(presid[i,], samples)
    pppv[i] <- step(large[i] - largepred[i])
  }
  sigma2.with <- 1 / tau.with
  sigma2.btw <- 1 / tau.btw
  tau.with ~ dgamma(0.001, 0.001)
```

```

tau.btw ~ dgamma(0.001, 0.001)
theta ~ dnorm(0.0, 1.0E-10)
}

```

node	mean	sd	MC	2.5	97.5%	start	sample
			error	% median			
m[1]	1515.0	20.31	0.4391	1472.0	1517.0	1550.0	10001 40000
m[2]	1528.0	18.62	0.2291	490.0	1527.0	1566.0	10001 40000
m[3]	1548.0	22.12	0.5962	1512.0	1547.0	1594.0	10001 40000
m[4]	1511.0	21.42	0.5458	1466.0	1513.0	1547.0	10001 40000
m[5]	1569.0	30.38	1.143	1517.0	1572.0	1624.0	10001 40000
m[6]	1495.0	27.22	0.953	1443.0	1494.0	1544.0	10001 40000
pppv[1]	0.561	0.4963	0.005088	0.0	1.0	1.0	10001 40000
pppv[2]	0.1466	0.3537	0.003352	0.0	0.0	1.0	10001 40000
pppv[3]	0.3195	0.4663	0.003411	0.0	0.0	1.0	10001 40000
pppv[4]	0.6383	0.4805	0.006751	0.0	1.0	1.0	10001 40000
pppv[5]	0.5282	0.4992	0.003993	0.0	1.0	1.0	10001 40000
pppv[6]	0.2517	0.434	0.004228	0.0	0.0	1.0	10001 40000
sigma2.btw	2125.0	3716.0	65.35	0.004705	1237.0	9942.0	10001 40000
sigma2.with	3083.0	1125.0	31.27	1571.0	2855.0	5842.0	10001 40000

11.3. Use the DIC to compare the fits of two different models for the data involving growth of baby rats. First use the model, data, and initial values exactly as given in the “Rats” example in Volume 1 of WinBUGS or OpenBUGS examples. Then use the model, data, and initial values as given in the “Birats” example in Volume 2. In both cases, run at least 1,000 burn-in iterations before you set the DIC. Then use output for the DIC based on at least 10,000 additional iterations. Turn in the tables of DIC results for both models, and answer the following questions:

1. What is the estimated number of free parameters in the “Rats” model? In the “Birats” model? What could explain the difference between the two estimates?
2. Is one model strongly preferred over the other after the penalty for model complexity is taken into account? Justify your answer.

11.4. Refer to the Beetles example in Volume II of the OpenBUGS/WinBUGS examples. Use the DIC to compare the fits of the three models with three transformations (logit, probit, and complementary log–log). This is a comparison that a frequentist deviance analysis could not make since the models are not nested.

Tables of Probability Distributions

Table A.1 Discrete distributions

Distribution	Probability mass function	Mean	Mode	Variance
Binomial $Y \sim \text{Bin}(n, \pi)$ $0 < \pi < 1$	$p(y \pi) = \binom{n}{y} \pi^y (1 - \pi)^{n-y}$ $y = 0, 1, \dots, n$	$n\pi$	$\lfloor (n+1)\pi \rfloor$	$n\pi(1 - \pi)$
Poisson $Y \sim \text{Pois}(\lambda)$ $\lambda > 0$	$p(y \lambda) = \frac{\exp(-\lambda)\lambda^y}{y!}$ $y = 0, 1, \dots$	λ	$\lfloor \lambda \rfloor$	λ

Table A.2 Univariate continuous distributions

Distribution	Density	Mean Mode	Variance
Beta $Y \sim \text{Beta}(\alpha, \beta)$ $\alpha, \beta > 0$	$p(y \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} y^{\alpha-1} (1-y)^{\beta-1}$ $0 < y < 1$	$\frac{\alpha}{\alpha+\beta}$ $\frac{\alpha-1}{\alpha+\beta-2}$	$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$
Gamma $Y \sim \text{Gamma}(\alpha, \beta)$ $\alpha, \beta > 0$	$p(y \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\beta y)$ $0 < y < \infty$	$\frac{\alpha}{\beta}$ $\frac{\alpha-1}{\beta}, \alpha \geq 1$	$\frac{\alpha}{\beta^2}$
Inverse gamma $Y \sim \text{IG}(\alpha, \beta)$ $\alpha, \beta > 0$	$p(y \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{1}{y^{\alpha+1}} \exp(-\frac{\beta}{y})$ $0 < y < \infty$	$\frac{\beta}{\alpha-1}, \alpha > 1$ $\frac{\beta}{\alpha+1}$	$\frac{\beta^2}{(\alpha-1)^2(\alpha-2)}, \alpha > 2$
Normal $Y \sim N(\mu, \sigma^2)$ $\sigma^2 > 0$	$p(y \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(y-\mu)^2}{2\sigma^2})$ $-\infty < y < \infty$	μ μ	σ^2
Student's t $Y \sim t(\mu, \sigma^2, \nu)$ $\sigma^2 > 0, \nu \geq 1$	$p(y \mu, \sigma^2, \nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi\sigma^2}} \frac{1}{(1+\frac{1}{\nu}(\frac{y-\mu}{\sigma})^2)^{\frac{\nu+1}{2}}}$ $-\infty < y < \infty$	$\mu, \nu > 1$ μ	$\frac{\nu}{\nu-2} \sigma^2, \nu > 2$
Uniform $Y \sim U(a, b)$	$p(y a, b) = \frac{1}{b-a}$ $a \leq y \leq b$	$\frac{b-a}{2}$ none	$\frac{(b-a)^2}{12}$

Table A.3 Multivariate continuous distributions

Distribution	Density	Mean Mode	Variance
Normal $\mathbf{Y} \sim N_d(\mu, \Sigma)$ Σ pos def symm matrix	$p(\mathbf{y} \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} \Sigma ^{\frac{1}{2}}} \exp(-\frac{(\mathbf{y}-\mu)^T \Sigma^{-1} (\mathbf{y}-\mu)}{2})$ \mathbf{y} a vector of length d	μ μ	Σ
Wishart $T \sim W_d(S, \nu)$ S pos def symm matrix	$p(T S, \nu) \propto T ^{\frac{\nu}{2}} \frac{\nu-d-1}{2} \exp(-\frac{\text{tr}(ST)}{2})$ T pos def symm matrix	$E(T_{ij}) = \nu S_{ij}^{-1}$	

References

- Albert, J. (1997). *Bayesian computation using minitab*. Belmont: Wadsworth.
- Albert, J. (2007). *Bayesian computation with R*. New York: Springer.
- Albert, J. (2008). *LearnBayes: Functions for Learning Bayesian Inference*. R package version 2.0
- Anderson, T.W. (1984). *An introduction to multivariate statistical analysis*, 2nd edn. New York: Wiley.
- Atchade, Y.F., & Rosenthal, J.S. (2005). On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, 11, 815–828.
- Bauer, R.L., Sung, J., Eckhert, K.H.E. Jr., Koul, A., Castillo, N.B., Nemoto, T. (1997). Comparison of histologic diagnosis between stereotactic core needle biopsy and open surgical biopsy. *Annals of Surgical Oncology*, 4(4), 316–320.
- Berger, J.O. (1985). *Statistical decision theory and Bayesian analysis*. Berlin: Springer
- Berry, D.A. (1996). *Statistics: A Bayesian perspective*. Belmont: Duxbury.
- Bliss, C.I. (1935). The calculation of the dosage-mortality curve. *Annals of Applied Biology*, 22, 134–167.
- Brooks, S.P., & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4), 434–455.
- Box, G.E., & Tiao, G.C. (1973). *Bayesian inference in statistical analysis*. Reading: Addison-Wesley.
- Burden, R.L., & Faires, D. (2011). *Numerical analysis*, 9th edn. Boston: Brooks/Cole.
- Carlin, B.P., & Chib, S. (1995). Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, Series B*, 57(3), 473–484.
- Carlin, B.P., & Louis, T.A. (2000). *Bayes and empirical Bayes methods for data analysis*, 2nd edn. London/Boca Raton: Chapman & Hall/CRC.
- Carlin, B.P., & Sargent, D.J. (1996). Robust Bayesian approaches for clinical trial monitoring. *Statistics in Medicine*, 15(11), 1093–1106.
- Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, 90(432), 1313–1321.
- Chib, S., & Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association*, 96(453), 270–281.
- Congdon, P. (2001). *Bayesian statistical modelling*. New York: Wiley.
- Congdon, P. (2003). *Applied Bayesian modelling*. New York: Wiley.
- Cowles, M.K. (2003). Efficient model-fitting and model-comparison for high dimensional Bayesian geostatistical models. *Journal of Statistical Planning and Inference*, 112(1–2), 221–239.
- Cowles, M.K., & Carlin, B.P. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91, 883–904.

- Gelfand, A.E., & Smith, A.F.M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85, 398–409.
- Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3), 515–533.
- Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B. (2004). *Bayesian data analysis*, 2nd edn. London: Chapman & Hall.
- Gelman, A., Meng, X.L., Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 6, 733–807.
- Gelman, A., & Rubin, D. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*, 7, 457–511.
- Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6, 721–741.
- Gill, J. (2002). *Bayesian methods for the social and behavioral sciences*. London: Chapman and Hall.
- Good, I.J. (1965). *The estimation of probabilities: an essay on modern Bayesian methods*. Cambridge: M.I.T. Press.
- Greenhouse, J., & Wasserman, L. (1995). Robust Bayesian methods for monitoring clinical trials. *Statistics in Medicine*, 14(12), 1379–1391.
- Guidance for the use of bayesian statistics in medical device clinical trials. <http://www.fda.gov/medicaldevices/deviceregulationandguidance/guidancedocuments/ucm071072.htm>. Accessed on July 2012.
- Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their application. *Biometrika*, 57, 97–109.
- Hosmer, D.W., & Lemeshow, S. (2000). *Applied logistic regression (Wiley Series in probability and statistics)*, 2nd edn. New York: Wiley-Interscience.
- Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London, Series A, Mathematical and Physical Sciences*, 186(1007), 453–461.
- Jensen, J. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1), 175–193.
- Johnson, V., & Albert, J. (1999). *Ordinal data modeling*. Berlin: Springer.
- Kass, R.E., & Raftery, A.E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430), 773–795.
- Lauritzen, S.L., & Spiegelhalter, D.J. (1988). Local computation with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society, Series B*, 50, 205–247. Reprinted in Shafer and Pearl (1990).
- Lee, P.M. (2004). *Bayesian statistics: an introduction*, 3rd edn. London: Hodder Arnold.
- Lindley, D., & Smith, A. (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society Series B*, 34, 1–41.
- Lunn, D., Spiegelhalter, D., Thomas, A., Best, N. (2009). The BUGS project: evolution, critique and future directions (with discussion). *Statistics in Medicine*, 28, 3049–3082.
- Lunn, D., Thomas, A., Best, N., Spiegelhalter, D. (2000). WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing*, 10, 325–337.
- McCulloch, P., & Nelder, J.A. (1989). *Generalized linear models*, 2nd edn. Boca Raton: CRC.
- McGrayne, S.B. (2011). Book review of *Why Bayes Rules: The History of a Formula That Drives Modern Life*. *Scientific American*, 15.
- McGrayne, S.B. (2011). *The theory that would not die: how Bayes rule cracked the enigma code, hunted down Russian submarines and emerged triumphant from two centuries of controversy*. New Haven: Yale University Press.
- Metropolis, N., Rosenbluth, A., Rosenbluth, A., Teller, A., Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6), 1087–1092.
- Moore, D.S. (2007). *The basic practice of statistics*, 4th edn. New York: W.H. Freeman and Company.

- Neal, R.M. (2003). "Slice sampling" (with discussion). *Annals of Statistics*, 31, 705–767.
- Nelder, J.A., Wedderburn, R.W.M. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A, General*, 135, 370–384.
- Novick, M., Jackson, P., Thayer, D., Cole, N. (1972). Estimating multiple regressions in m -groups: a cross validation study. *British Journal of Mathematical and Statistical Psychology*, 25, 33–50.
- Poplack, S.P., Tosteson, A.N., Grove, M.R., Wells, W.A., Carney, P.A. (2000). Mammography in 53,803 women from the New Hampshire mammography network. *Radiology*, 217, 832–840.
- R Development Core Team (2008). *R: a language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing, <http://www.R-project.org>. ISBN 3-900051-07-011/27/12.
- Robert, C.P. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*. New York: Springer.
- Robert, C.P., & Casella, G. (2010). *Introducing Monte Carlo methods with R*. Berlin: Springer.
- Spiegelhalter, D.J., Best, N.G., Carlin, B.P., Van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B*, 64(4), 583–616.
- Spiegelhalter, D.J., Freedman, L.S., Parmar, M. (1993). Applying Bayesian ideas in drug development and clinical trials. *Statistics in Medicine*, 11, 1501–1511.
- Spiegelhalter, D., Thomas, A., Best, N., Gilks, W. (1995). *BUGS examples, version 0.5*. Cambridge: MRC Biostatistics Unit.
- Woodworth, G.G. (2004). *Biostatistics: A Bayesian introduction*. Hoboken: Wiley.
- Zellner, A. (1971). *An introduction to Bayesian inference in econometrics*. New York: Wiley.

Index

A

autocorrelation, 134

B

Bayes factors, 207

Bernoulli, 26

binary, 26

binomial random variable, 26

borrowing strength, 153

burn-in, 125

C

complement, 14

composite midpoint rule, 113

conditional probability, 15

conjugate prior, 34

continuous random variable, 20

credible set, 57

D

discrepancy measure, 219

discrete random variables, 20

disjoint, 14

E

equal-tail credible set, 57

equally likely, 13

equivalent prior sample size, 35

estimating, 61

event, 13

exchangeability, 84

exchangeable, 36, 148

exhaustive, 14

F

full conditional distribution, 172

G

Gaussian quadrature, 113

generalized linear models, 193

graph, 170

H

hierarchical models, 147

highest posterior density region, 57

history plots, 133

hypotheses, 5

I

improper densities, 73

independent, 16

intersection, 14

J

Jeffreys prior, 74

K

kernel, 27, 34

L

likelihood, 7

likelihood function, 27

logistic regression, 193

logit, 73

M

marginal likelihood, 211
marginal probability, 15
Markov chain, 123
models, 5
Monte Carlo error, 137
multiplicative rule, 15
mutually exclusive, 14

N

noninformative priors, 72
normal-inverse gamma density, 106
normalizing constant, 27
null event, 14
numeric integration, 113

O

odds, 208

P

p-value, 53
point estimate, 50
point null hypothesis, 215
posterior marginal, 101
posterior predictive interval, 65
posterior odds, 209
posterior predictive p-value, 220
posterior predictive probability, 62
posterior probability, 7
precision, 86
predicting, 61
prior odds, 208
prior probabilities, 6
probability density function, 21

probability distributions, 20
probability mass function, 21

R

random variable, 20
reparameterization, 74
robust, 67

S

sample space, 14
sampling distribution, 51
semi-conjugate, 106
sensitivity, 6, 8
sensitivity analysis, 67, 68
shrinkage, 54, 153
simple hypothesis, 208
space, 20
specificity, 8
stages, 148
statistical inference, 49
sufficient statistic, 85, 95

T

test quantity, 219
transformation of variables, 74
transition kernel, 123, 171

U

uniform, 28
union, 14

W

Wishart density, 198