Bayesian Modeling, Inference and Prediction

David Draper

Department of
Applied Mathematics and Statistics
University of California, Santa Cruz

draper@ams.ucsc.edu
www.ams.ucsc.edu/~draper/

Draft 6 (December 2005): Many sections still quite rough. Comments welcome.

This book was typeset by the author using a PostScript-based phototypesetter (© Adobe Systems, Inc.). The figures were generated in PostScript using the R data analysis language (R Project, 2005), and were directly incorporated into the typeset document. The text was formatted using the LaTeX language (Lamport, 1994), a version of TeX (Knuth, 1984).

To Andrea,
from whom I've learned so much

Contents

Preface

Santa Cruz, California David Draper
December 2005

Chapter 1

Background and basics

1.1 Quantification of uncertainty

Case study: Diagnostic screening for HIV. Widespread **screening for HIV** has been proposed by some people in some countries (e.g., the U.S.). Two blood tests that screen for HIV are available: *ELISA*, which is relatively inexpensive (roughly \$20) and fairly accurate; and *Western Blot (WB)*, which is considerably more accurate but costs quite a bit more (about \$100). Let's say[1] I'm a physician, and a new patient comes to me with symptoms that suggest the patient (male, say) may be HIV positive. **Questions:**

- Is it appropriate to use the language of probability to quantify my uncertainty about the proposition $A =$\{this patient is HIV positive\}?

- If so, what kinds of probability are appropriate, and how would I assess $P(A)$ in each case?

- What strategy (e.g., *ELISA*, *WB*, both?) should I employ to decrease my uncertainty about A? If I decide to run a screening test, how should my uncertainty be updated in light of the test results?

[1] As will become clear, the Bayesian approach to probability and statistics is explicit about the role of personal judgment in uncertainty assessment. To make this clear I'll write in the first person in this book, but as you read I encourage you to constantly imagine yourself in the position of the person referred to as "I" and to think along with that person in quantifying uncertainty and making choices in the face of it.

Statistics might be defined as **the study of uncertainty: how to measure it, and what to do about it**, and probability as the part of mathematics (and philosophy) devoted to the quantification of uncertainty. The systematic study of probability is fairly recent in the history of ideas, dating back to about 1650 (e.g., Hacking, 1975). In the last 350 years three main ways to define probability have arisen (e.g., Oakes, 1990):

- **Classical**: Enumerate the **elemental outcomes** (EOs) (the fundamental possibilities in the process under study) in a way that makes them **equipossible** on, e.g., symmetry grounds, and compute $P_C(A) =$ ratio of $n_A =$(number of EOs favorable to A) to $n =$(total number of EOs).

- **Frequentist**: Restrict attention to attributes A of **events**: phenomena that are inherently (and independently) repeatable under "identical" conditions; define $P_F(A) =$ the limiting value of the relative frequency with which A occurs in the repetitions.

- **Personal**, or **"Subjective"**, or **Bayesian**: I imagine betting with someone about the truth of **proposition** A, and ask myself what **odds** O (in favor of A) I would need to give or receive in order that I judge the bet fair; then (for me) $P_B(A) = \frac{O}{(1+O)}$.

Other approaches not covered here include **logical** (e.g., Jeffreys, 1961) and **fiducial** (Fisher, 1935) probability.

Each of these probability definitions has general advantages and disadvantages:

- **Classical**

  - $\boxed{\text{Plus:}}$ Simple, when applicable (e.g., idealized coin-tossing, drawing colored balls from urns, choosing cards from a well-shuffled deck, and so on).

  - $\boxed{\text{Minus:}}$ The only way to define "equipossible" without a circular appeal to probability is through the **principle of insufficient reason**—I judge EOs equipossible if I have no grounds (empirical, logical, or symmetrical) for favoring one over another—but this can lead to paradoxes (e.g., assertion of equal uncertainty is not invariant to the choice of scale on which it's asserted).

- **Frequentist**
  - $\boxed{\text{Plus:}}$ Mathematics relatively tractable.
  - $\boxed{\text{Minus:}}$ Only applies to inherently repeatable events, e.g., from the vantage point of (say) 2005, P_F(the Republicans will win the White House again in 2008) is (strictly speaking) undefined.
- **Bayesian**
  - $\boxed{\text{Plus:}}$ All forms of uncertainty are in principle quantifiable with this approach.
  - $\boxed{\text{Minus:}}$ There's no guarantee that the answer I get by asking myself about betting odds will retrospectively be seen by me or others as "good" (but how should the quality of an uncertainty assessment itself be assessed?).

Returning to $P(A) = P$(this patient is HIV-positive), data are available from medical journals on the prevalence of HIV-positivity in various subsets of $\mathcal{P} = \{$all humans$\}$ (e.g., it's higher in gay people and lower in women). All three probabilistic approaches require me to use my **judgment** to identify the **recognizable subpopulation** $\mathcal{P}_{\text{this patient}}$ (Fisher, 1956; Draper et al., 1993): this is

> *the smallest subset to which this patient belongs for which the HIV prevalence differs from that in the rest of $\mathcal{P}$ by an amount I judge as* ***large enough to matter in a practical sense****.*

The idea is that within $\mathcal{P}_{\text{this patient}}$ I regard HIV prevalence as close enough to constant that the differences aren't worth bothering over, but the differences between HIV prevalence in $\mathcal{P}_{\text{this patient}}$ and its complement matter to me. Here $\mathcal{P}_{\text{this patient}}$ might consist (for me) of everybody who matches this patient on gender, age (category, e.g., 25–29), and sexual orientation. **NB** This is a modeling choice based on judgment; different reasonable people might make different choices.

As a classicist I would then (a) use this definition to establish equipossibility within $\mathcal{P}_{\text{this patient}}$, (b) count $n_A =$(number of HIV-positive people in $\mathcal{P}_{\text{this patient}}$) and $n =$(total number of people in $\mathcal{P}_{\text{this patient}}$), and (c) compute $P_C(A) = \frac{n_A}{n}$.

As a frequentist I would (a) equate $P(A)$ to P(a person chosen at random with replacement (**independent identically distributed (IID)** sampling) from $\mathcal{P}_{\text{this patient}}$ is HIV-positive), (b) imagine repeating this random sampling indefinitely, and (c) conclude that the limiting value of the relative frequency of HIV-positivity in these repetitions would also be $P_F(A) = \frac{n_A}{n}$. **NB** Strictly speaking I'm not allowed in the frequentist approach to talk about P(this patient is HIV-positive)—he either is or he isn't. In the frequentist paradigm I can only talk about the *process* of sampling people like him from $\mathcal{P}_{\text{this patient}}$.

As a Bayesian, with the information given here I would regard this patient as **exchangeable** (de Finetti, e.g., 1964, 1974/5) with all other patients in $\mathcal{P}_{\text{this patient}}$—meaning informally that I judge myself equally uncertain about HIV-positivity for all the patients in this set—and this judgment, together with the axioms of **coherence**, would also yield $P_{B:\text{You}}(A) = \frac{n_A}{n}$ (although I've not yet said why this is so). Exchangeability and coherence will be defined and explored in more detail in what follows.

Note that with the same information base the three approaches in this case have led to the same answer, although the meaning of that answer depends on the approach, e.g., frequentist probability describes the *process* of observing a repeatable event whereas Bayesian probability is an attempt to quantify my uncertainty about something, repeatable or not.

The classical and frequentist approaches have sometimes been called **objective**, whereas the Bayesian approach is clearly **subjective**, and—since objectivity sounds like a good goal in science—this has sometimes been used as a claim that the classical and frequentist approaches are superior. I'd argue, however, that in interesting applied problems of realistic complexity, the judgment of **equivalence** or **similarity** (equipossibility, IID, exchangeability) that's central to all three theories makes them all subjective in practice.

Imagine, for example, that I'm given data on HIV prevalence in a large group of people, along with many variables (possible predictors) that might or might not be relevant to identifying the recognizable subpopulations. I might well differ (with other reasonable people working independently) in my judgments on which of these predictors are relevant (and how they should be used in making the prediction), and the result could easily be noticeable variation in the estimates of P(HIV positive) obtained by the other analysts and me, even if I and the other people all attempt to use "objective" methods to arrive at these judgments (there are many such methods, and they don't always lead to the same conclusions). Thus the assessment of complicated probabilities is inherently subjective—there are "judgment calls" built into probabilistic and statistical analysis.

With this in mind attention in all three approaches should evidently shift away from trying to achieve "objectivity" toward two things: (1) the explicit statement of the assumptions and judgments made on the way to my probability assessments, so that other people may consider their plausibility, and (2) **sensitivity analyses** exploring the mapping from assumptions to conclusions. (To a Bayesian saying that $P_B(A)$ is objective just means that lots of people more or less agree on its value.)

1.2 Sequential learning; Bayes' Theorem

Let's say that, with this patient's values of relevant demographic variables, the prevalence of HIV estimated from the medical literature, $P(A) = P(\text{he's HIV-positive})$, in his recognizable subpopulation is about $\frac{1}{100} = 0.01$. To improve this estimate by gathering data specific to this patient, I decide to take some blood and get a result from *ELISA*. Suppose the test comes back positive—what should the updated $P(A)$ be?

Bayesian probability has that name because of the simple updating rule that has been attributed to Thomas Bayes (1763), who was one of the first people to define conditional probability and make calculations with it:

Bayes' Theorem
for propositions

$$P(A|D) = \frac{P(A)\,P(D|A)}{P(D)}$$

(Actually—Stigler, 1986; Bernardo and Smith, 1994—Bayes only stated and worked with a special case of this; the general form was first used by Laplace, 1774.)

In the usual application of this A is an unknown quantity (such as the truth value of some proposition) and D stands for some **data** relevant to my uncertainty about A:

$$\begin{aligned} P(\text{unknown}|\text{data}) &= \frac{P(\text{unknown})\,P(\text{data}|\text{unknown})}{\text{normalizing constant}} \\ \text{posterior} &= c \cdot \text{prior} \cdot \text{likelihood} \end{aligned}$$

The terms **prior** and **posterior** emphasize the sequential nature of the learning process: $P(\text{unknown})$ was my uncertainty assessment before the data arrived; this is updated multiplicatively on the probability scale by the **likelihood** $P(\text{data}|\text{unknown})$, and renormalized so that total probability remains 1.

Writing the Theorem both for A and (not A) and combining gives a (perhaps even more) useful version:

$$\boxed{\textbf{Bayes' Theorem in odds form}}$$

$$\begin{array}{ccccc} \dfrac{P(A|\text{data})}{P(\text{not }A|\text{data})} & = & \dfrac{P(A)}{P(\text{not }A)} & \cdot & \dfrac{P(\text{data}|A)}{P(\text{data}|\text{not }A)} \\ \\ \begin{array}{c}\textbf{posterior}\\ \textbf{odds}\end{array} & = & \begin{array}{c}\textbf{prior}\\ \textbf{odds}\end{array} & \cdot & \begin{array}{c}\textbf{Bayes}\\ \textbf{factor}\end{array} \end{array}$$

Another name for the Bayes factor is the **likelihood ratio**, since this factor measures the relative plausibility of the data given A and (not A).

Applying this to the HIV example requires additional information about *ELISA* obtained by screening the blood of people with known HIV status:

$$\begin{array}{rcl} \textbf{sensitivity} & = & P(\textit{ELISA}\text{ positive}|\text{HIV positive}) \quad \text{and} \\ \textbf{specificity} & = & P(\textit{ELISA}\text{ negative}|\text{HIV negative}) \end{array}$$

In practice *ELISA*'s operating characteristics are (or at least seem) rather good—sensitivity about 0.95, specificity about 0.98—so you might well expect that

$$P(\text{this patient HIV positive}|\textit{ELISA}\text{ positive})$$

would be close to 1.

Here the updating produces a surprising result: the Bayes factor comes out

$$B = \frac{\text{sensitivity}}{1 - \text{specificity}} = \frac{0.95}{0.02} = 47.5,$$

which sounds like strong evidence that this patient is HIV positive, but the prior odds are quite a bit stronger the other way ($\frac{P(A)}{1-P(A)} = 99$ to 1 *against* HIV) leading to posterior odds of $\frac{99}{47.5} \doteq 2.08$ *against* HIV, i.e., $P(\text{HIV positive}|\text{data}) = \frac{1}{1+\text{odds}} = \frac{95}{293} \doteq 0.32$ (!).

The reason for this is that *ELISA* was designed to have a vastly better **false negative** rate—$P(\text{HIV positive}|\ \textit{ELISA}\text{ negative}) = \frac{5}{9707} \doteq 0.00052 \doteq$

1 in 1941—than **false positive** rate—P(HIV negative| *ELISA* positive) = $\frac{198}{293} \doteq 0.68 \doteq 2$ in 3. This in turn is because *ELISA*'s developers judged that it's far worse to tell somebody who's HIV positive that they're not than the other way around (this is a reasonable position when using *ELISA* for, e.g., blood bank screening, which is one of the main uses for the test). This false positive rate would make widespread screening for HIV based only on *ELISA* a truly bad idea.

Formalizing the consequences of the two types of error in diagnostic screening would require quantifying the misclassification costs, which shifts the focus from (scientific) **inference** (the acquisition of knowledge for its own sake: Is this patient really HIV-positive?) to **decision-making** (putting that knowledge to work to answer a public policy or business question: What use of *ELISA* and *Western Blot* would yield the optimal screening strategy?).

1.3 Bayesian decision theory

Axiomatic approaches to rational decision-making date back to Ramsay (1926), with von Neumann and Morgenstern (1944) and Savage (1954) also making major contributions. The ingredients of a general decision problem (e.g., Bernardo and Smith, 1994) include

- A set $\{a_i, i \in I\}$ of available **actions**, one of which I will choose;
- For each action a_i, a set $\{E_j, j \in J\}$ of **uncertain outcomes** describing what will happen if I choose action a_i;
- A set $\{c_j, j \in J\}$ of **consequences** corresponding to the outcomes $\{E_j, j \in J\}$; and
- A **preference relation** $\leq$, expressing my preferences between pairs of available actions ($a_1 \leq a_2$ means "a_1 is not preferred by me to a_2").

  Define $a_1 \sim a_2$ ("a_1 and a_2 are **equivalent**" to me) iff $a_1 \leq a_2$ and $a_2 \leq a_1$.

This preference relation induces a qualitative ordering of the uncertain outcomes ($E \leq F$ means "E is not more likely than F"), because if I compare two dichotomized possible actions, involving the same consequences and differing only in their uncertain outcomes, the fact that I prefer one action to another means that I must judge it more likely that if I take that action the preferred consequence will result.

Within this framework I have to make further assumptions—the **coherence** axioms—to ensure that my actions are internally consistent. Informally (see Bernardo and Smith, 1994, for the formalism) these are:

- An axiom insisting that I be willing to express preferences between simple dichotomized possible actions ($\{a, \text{not } a\}$);
- A **transitivity** axiom in which (for all actions a, a_1, a_2, a_3) $a \leq a$, and if $a_1 \leq a_2$ and $a_2 \leq a_3$ then $a_1 \leq a_3$; and
- An axiom based on the **sure-thing principle**: if, in two situations, no matter how the first comes out the corresponding outcome in the second is preferable, then I should prefer the second situation overall.

This puts $\leq$ on a sound footing for qualitative uncertainty assessment, but does not yet imply how to quantify—it's like being able to say that one thing weighs less than another but not to say by how much. To go further requires a fourth assumption, analogous to the existence of a set of reference standards (e.g., an official kg weight, half-kg, and so on) and the ability to make arbitrarily precise comparisons with these standards:

- An axiom guaranteeing that for each outcome E there exists a **standard outcome** S (e.g., "idealized coin lands heads") such that $E \sim S$.

This framework implies the existence and uniqueness of a (personal) probability P_B (abbreviated P), mapping from outcomes E to [0,1] and corresponding to the judgments in my definition of $\leq$, and a **utility function** U (large values preferred, say), mapping from consequences c to R and quantifying my preferences.

This has all been rather abstract. Two concrete results arising from this framework may make its implications clearer:

- Bayes' original definition of personal probability is helpful in thinking about how to quantify uncertainty. Pretending that consequences are monetary (e.g., US$), for me to say that $P_B(E) = p$ for some uncertain outcome E whose truth value will be known in the future is to say that I'm indifferent between (a) receiving $\$p \cdot m$ for sure (for some (small) hypothetical amount of money \$$m$) and (b) betting with someone in such a way that I'll get \$$m$ if E turns out to be true and nothing if not (this can be used this to estimate $P_B(E)$).

- It turns out that any coherent set of probability judgments must satisfy the standard axioms and theorems of a finitely additive probability measure:

  - $0 \leq P(E) \leq 1$ and $P(E^c) = 1 - P(E)$;
  - $P(E_1 \text{ or } \ldots \text{ or } E_J) = \sum_{j \in J} P(E_j)$ for any finite collection $\{E_j, j \in J\}$ of disjoint outcomes;
  - $P(E \text{ and } F) = P(E) \cdot P(F)$ for any two independent outcomes (informally, E and F are **independent** if my uncertainty judgments involving one of them are unaffected by information about the other); and
  - Conditional probability has a natural definition in this setup, corresponding to the updating of my uncertainty about E in light of F, and with this definition $P(E|F) = \frac{P(E \text{ and } F)}{P(F)}$.

  Otherwise (de Finetti, 1964) someone betting with me on the basis of my probability judgments can make **Dutch book** against me, i.e., get me to agree to a series of bets that are guaranteed to lose me money.

  Thus coherent Bayesian probability obeys the same laws as with the classical and frequentist approaches (apart from a technical issue about finite versus countable additivity).

Nothing so far has said clearly what choice to make in a decision problem if I wish to avoid incoherence. If the outcomes were certain I'd evidently choose the action that maximizes my utility function, but since they're not the best action must involve weighing both my probabilities for the uncertain outcomes and the utilities I place on their consequences. It's a direct implication of the framework here that the form this weighing should take is simple and clear:

> **Maximization of Expected Utility (MEU)** Given my probability and utility judgments, my decision-making is coherent iff for each action a_i, with associated uncertain outcomes

Table 1.1: *Probabilities and utilities for action a_1.*

Probability	True HIV Status	*ELISA* Status	Utility
.0095	$+$	$+$	$-c_1$
.0005	$+$	$-$	$-c_1 - L_{\text{I}}$
.0198	$-$	$+$	$-c_1 - L_{\text{II}}$
.9702	$-$	$-$	$-c_1$

$\{E_j, j \in J\}$ and consequences $\{c_j, j \in J\}$, I compute the **expected utility** $\text{EU}_i = \sum_{j \in J} U(c_j) P(E_j)$ and choose the action that **maximizes** $\{\text{EU}_i, i \in I\}$.

Example: HIV screening. As a simplified version of this problem consider choosing between two actions:

- a_1: Obtain *ELISA* results at a cost of $c_1 = \$20$; if positive conclude this patient is HIV+, if negative conclude HIV–.
- a_2: Same as a_1 except if *ELISA* comes out positive, obtain *Western Blot* (WB) results at an additional cost of $c_2 = \$100$; if *WB* is positive conclude HIV+, if negative conclude HIV–.

With action a_1 the probabilities, uncertain outcomes, and utilities are as in Table 1.1. Here L_{I} and L_{II} are the false negative (false positive) monetary losses suffered by this patient if he really is HIV+ (HIV–) but *ELISA* says he's HIV– (HIV+). The expected utility with action a_1 is thus

$$\begin{aligned} \text{EU}_1 &= .0095(-c_1) + .0005(-c_1 - L_{\text{I}}) + \ldots + .9702(-c_1) \\ &= -(c_1 + .0005 L_{\text{I}} + .0198 L_{\text{II}}) \, . \end{aligned}$$

The corresponding information for action a_2 is given in Table 1.2. These probabilities arise from *WB*'s design (the goal was to have about the same false negative rate as *ELISA* and a much lower false positive rate (about 0.1), leading to a slightly worse sensitivity (0.949) but much improved specificity (0.999)).

Table 1.2: *Probabilities and utilities for action a_2.*

Probability	True HIV Status	*ELISA* Status	*WB* Status	Utility
.00945	$+$	$+$	$+$	$-c_1 - c_2$
.00005	$+$	$+$	$-$	$-c_1 - c_2 - L_{\text{I}}$
.00004	$+$	$-$	$+$	$-c_1 - L_{\text{I}}$
.00046	$+$	$-$	$-$	$-c_1 - L_{\text{I}}$
.0001	$-$	$+$	$+$	$-c_1 - c_2 - L_{\text{II}}$
.0197	$-$	$+$	$-$	$-c_1 - c_2$
.00095	$-$	$-$	$+$	$-c_1$
.96925	$-$	$-$	$-$	$-c_1$

The expected utility with action a_2 comes out

$$\begin{aligned} \text{EU}_2 &= .00945(-c_1 - c_2) + \ldots + .9604(-c_1) \\ &= -(c_1 + .0293 c_2 + .00055 L_{\text{I}} + .0001 L_{\text{II}}) \ . \end{aligned}$$

By MEU I should prefer a_2 to a_1 iff $\text{EU}_2 > \text{EU}_1$, i.e., iff

$$.0197 L_{\text{II}} - .00005 L_{\text{I}} - .0293 c_2 > 0 \ .$$

Thus a_2 becomes more desirable as the loss suffered with a false positive (negative) increases (decreases), and less desirable as *WB*'s cost increases, all of which makes good sense.

It's interesting to note that with a modest value for L_{II} (e.g., \$1,000), the monetary advantage from taking action a_2 is quite small even with a realistically huge value for L_{I} (e.g., \$100,000, which leads to an edge for a_2 of only about \$12). This is due to the extremely low false negative rate for both tests—L_{I} would have to be over \$335,000 for a_1 to dominate!

1.4 Problems

1. (Conditional probability; elaboration of the HIV case study in this chapter) Consider the HIV screening example, in which A = {the patient in question is HIV positive} and D = {ELISA says he's HIV

positive}. Let p stand for the prevalence of HIV among people similar to this patient (recall that in the case study $p = 0.01$), and let ϵ and π stand for the sensitivity and specificity of the ELISA screening test, respectively (in the case study $\epsilon = 0.95$ and $\pi = 0.98$).

(a) By using either Bayes' Theorem (in probability or odds form) or 2×2 contingency tables, write down explicit formulas in terms of p, ϵ, and π for the **positive predictive value** (PPV), $P(A|D)$, and **negative predictive value** (NPV), $P(\text{not } A|\text{not } D)$, of screening tests like ELISA (recall that ELISA's PPV and NPV with patients like the one in our case study were 0.32 and 0.99948, respectively). These formulas permit analytic study of the tradeoff between PPV and NPV.

(b) Interest focused in this chapter on why ELISA's PPV is so bad for people (like the man considered in the case study) for whom HIV is relatively rare ($p = 0.01$).

(i) Holding ϵ and π constant at ELISA's values of 0.95 and 0.98, respectively, obtain expressions (from those in (a)) for the PPV and NPV as a function of p, and plot these functions as p goes from 0 to 0.1.

(ii) Show (e.g., by means of Taylor series) that in this range the NPV is closely approximated by the simple linear function $(1 - 0.056\,p)$.

(iii) How large would p have to be for ELISA's PPV to exceed 0.5? 0.75?

(iv) What would ELISA's NPV be for those values of p?

(v) Looking at both PPV and NPV, would you regard ELISA as a good screening test for subpopulations with (say) $p = 0.1$? Explain briefly.

(b) Suppose now that p is held constant at 0.01 and you're trying to improve ELISA for use on people with that background prevalence of HIV, where "improve" for the sake of this part of the problem means raising the PPV while not suffering too much of a decrease (if any) of the NPV. ELISA is based on the level L of a particular antibody in the blood, and uses a rule of the form {if $L \geq c$ announce that the person is HIV positive}. This means that if you change c the sensitivity and specificity change in a tug-of-war fashion: altering c to make ϵ go up makes π go down, and vice versa.

(i) By using the formulas in (a) or 2×2 contingency tables, show that as ϵ approaches 1 with π no larger than 0.98, the NPV will approach 1 but the biggest you can make the PPV is about 0.336. Thus if you want to raise the PPV you would be better off trying to increase π than ϵ.

(ii) Suppose there were a way to change c which would cause π to go up while holding ϵ arbitrarily close to 0.95. Show that π would have to climb to about 0.997 to get the PPV up to 0.75. Is the NPV still at acceptable levels under these conditions? Explain briefly.

2. (Coherence and Dutch book) On 2 Apr 2001 a senior writer for the web page *Sportsline.com*, Mark Soltau, posted an article about the Masters golf tournament which was about to be held on 5–8 Apr 2001 (copy attached). Among other things he identified the 24 players (among the 93 golfers in the field) who were, in his view, most likely to win the tournament, and he posted odds *against* each of them winning (for example, his quoting of 10–1 odds on Phil Mickelson meant that his personal probability that Mickelson would win was $\frac{1}{1+10} \doteq 0.091$), which are summarized in Tables 1.3–1.4 below.

(a) If the 24 odds quoted by Mr. Soltau were taken literally, show that the personal probability specification implied by his posted odds was incoherent. (In fact Mr. Soltau may well have been quoting un-normalized odds, which is a fairly common practice in sports, but let's take him literally in this part of the problem.)

(b) It would be nice to demonstrate Mr. Soltau's incoherence by explicitly providing a set of bets which would be guaranteed to lose him money, but that's actually fairly complicated (hint for the previous part of this question: that's *not* what I had in mind for you to do in (a)). To take a simpler example that has the same flavor as Mr. Soltau's mistake (if his odds are taken literally), pretend that he's handicapping (setting odds for) a tournament in which only Tiger Woods, Phil Mickelson, and some other unnamed golfers are playing, and he announces 3 to 1 odds in *favor* of Woods winning and 1 to 1 odds in favor of Mickelson (again without specifying any odds for the other golfers). (To be clear on the relationship between odds and money, here's how it works in horse-racing (and Mr. Soltau would have to play by the same rules): suppose that a bookie at the horse track offers odds of 4 to 1 against horse A, and I bet (say) \$1 on that horse to win; if horse A wins I enjoy a net gain of \$4, otherwise I suffer a net loss of \$1.) Work out an explicit set of bets to offer Mr. Soltau which would constitute a Dutch book against him. If Mr. Soltau were willing to accept arbitrarily large bets, is there any theoretical limit to the amount of money you would be guaranteed to win from him? Explain briefly.

(c) (c) In practice sports bookies only allow people to make bets *for* individual golfers, so that in reality you're not allowed to construct a wager like $\{\$x$ on Woods to win and $\$y$ on Mickelson to lose$\}$. Can you make Dutch book against Mr. Soltau under these conditions? Explain briefly.

3. (Bayes' Theorem; based on problem 7 in chapter 1 of Gelman et al., 2003) In the old television game show *Let's Make a Deal*, there are three doors; behind one of the doors is a car, and behind the other two are goats, with the assignment of prizes to doors made at random. You—the contestant, who prefers cars to goats—are asked to pick a door. After you choose (of course you can do no better than picking at random), the Master of Ceremonies, Monte Hall, who knows where the car is, opens one of the other doors to reveal a goat, and he offers you the choice of staying with the door you originally picked or switching to the other unopened door. Suppose that Monte Hall uses the following algorithm to decide which door to reveal to you after you've chosen (without loss of generality) door 1: if the car is behind door 2 he shows you door 3; if it's behind door 3 he shows you door 2; and if it's behind door 1 he randomizes between showing you doors 2 and 3 with equal probability. Should you switch or stay with your original choice?

   (a) Explicitly use Bayes' Theorem to work out the chance of winning the car under each strategy.

   (b) How would you explain intuitively to someone who favors the inferior strategy why the other one is better?

4. (Conditional probability, and review of the normal distribution; based on problem 4 in chapter 1 of Gelman et al., 2003) (American) football (not soccer) experts provide a *point spread* (PS) for every football game as a measure of the difference in ability between the two teams. For example, team A might be a 3.5–point favorite over team B. This means that the proposition that A (the favorite) defeats B (the underdog) by 4 or more points is considered a fair bet, i.e., $P(A$ wins by more than 3.5 points$) = \frac{1}{2}$. If the PS is an integer, the implication is that A is as likely to win by more points than the PS as it is to win by fewer points than the PS (or to lose); there is a positive probability that A will win by exactly the PS, in which case neither side is paid off. In Chapter 1 Gelman et al. (2003) present data on the PS and actual game outcome for 672 professional football games played during the 1981 and 1983–84 seasons, and they show that the histogram of the quantity (actual outcome – PS) is well approximated by a normal distribution with mean 0.07 and standard deviation (SD) 13.86, suggesting that a good predictive distribution for the actual result of an NFL football game would be normal with mean equal to the PS and SD 14 points (two touchdowns). (If you're in the habit of betting on NFL games this should give you pause, e.g., if a team is favored by a touchdown the chance it will win, according to this uncertainty assessment, is only about 69%.) It turns out that there were 12 games in this data base with PS values of 8 points, and the actual outcomes in those games were –7, –5, –3, –3, 1, 6, 7, 13, 15, 16, 20, and 21, with positive (negative) values indicating wins by the favorite (underdog) by the number of points indicated. Consider the following conditional probabilities:

$$P(\text{favorite wins}|\text{PS} = 8)$$
$$P(\text{favorite wins by at least } 8|\text{PS} = 8)$$
$$P(\text{favorite wins by at least } 8|\text{PS} = 8 \text{ and favorite wins})$$

   (a) Estimate each of these using the relative frequencies of the games with an 8–point PS.

   (b) Estimate each using the normal approximation to the distribution of (actual outcome – PS). (You can use a normal table from any statistics book, or the error function `erf` in `Maple`.)

(c) Which of these approaches to uncertainty assessment seems to have produced better answers here? How should "better" be defined? Explain briefly.

5. (**Cromwell's Rule** and its implications for Bayesian learning) Prove the following two facts: for any D such that $P(D) > 0$,

   (a) If $P(A) = 0$ then $P(A|D) = 0$.

   (b) If $P(A) = 1$ then $P(A|D) = 1$.

   In the usual application of these facts (as in the HIV case study in this chapter), A is a proposition whose truth value is unknown to me (such as the HIV status of the patient) and D represents some data relevant to A (such as the result of a screening test like *ELISA*); in this setting (a) and (b) together are referred to as *Cromwell's Rule* (history). What are the implications of Cromwell's Rule for the use of Bayes' Theorem as a formal model for learning from data? Explain briefly.

Table 1.3: *Odds posted by sports writer Mark Soltau against each of the top 24 golfers competing in the Masters golf tournament, April 2001 (part 1 of table).*

Player	Best Finish	Odds	Comment
Tiger Woods	1st in 1997	3–1	His tournament to lose
Phil Mickelson	3rd in 1996	10–1	Overdue for major breakthrough
Vijay Singh	1st in 2000	10–1	Faldo successfully defended in 1990
Davis Love III	2nd in 1999	15–1	Has come oh-so-close before
Colin Montgomerie	Tied for 8th in 1998	15–1	Sooner or later he'll get it right
José Maria Olazabal	1st in 1994, 1999	20–1	Fearless competitor who never quits
Tom Lehman	2nd in 1994	25–1	Has all the tools to contend again
Nick Price	5th in 1986	25–1	If putter holds up, could be a factor
Ernie Els	2nd in 2000	25–1	Play lacking lately, but ready to rise up
David Duval	Tied for 2nd in 1998	25–1	Wrist, back only question marks
Jesper Parnevik	Tied for 21st in 1997	30–1	A major is next for gritty Swede
Mark Calcavecchia	2nd in 1998	30–1	Streaky player, never backs off

Table 1.4: *Odds posted by sports writer Mark Soltau against each of the top 24 golfers competing in the Masters golf tournament, April 2001 (part 2 of table).*

Player	Best Finish	Odds	Comment
Sergio Garcia	Tied for 38th in 1999	35–1	Doesn't lack game or confidence
Justin Leonard	Tied for 7th in 1997	35–1	Good grinder who won't beat himself
Jim Furyk	4th in 1998	35–1	Will long putter bag a major?
Greg Norman	2nd in 1996	35–1	Everybody's sentimental favorite
Paul Azinger	5th in 1998	40–1	Playing well and knows the layout
Darren Clarke	Tied for 8th in 1998	50–1	Cigar will come in handy at Amen Corner
Loren Roberts	Tied for 3rd in 2000	50–1	Splendid short game comes in handy
Brad Faxon	Tied for 9th in 1993	50–1	Has he ever hit a poor putt?
Fred Couples	Tied for 2nd in 1998	60–1	Never count him out
John Huston	Tied for 3rd in 1990	60–1	The man is a birdie machine
Mike Weir	Tied for 28th in 2000	60–1	Canadian continues to impress
Bernhard Langer	1st in 1993	65–1	Tough, determined and unflappable

Chapter 2

Exchangeability and conjugate modeling

2.1 Quantification of uncertainty about observables. Binary outcomes

Case Study: *Hospital-specific prediction of mortality rates.* Let's say I'm interested in measuring the quality of care (e.g., Kahn et al., 1990) offered by one particular hospital. I'm thinking of the Dominican Hospital (DH) in Santa Cruz, CA (you would of course almost certainly have a different hospital in mind if you were thinking along these lines for yourself). As part of this I decide to examine the medical records of all patients treated at the DH in one particular time window, say January 2002 to December 2005 (inclusive), for one particular medical condition for which there is a strong *process-outcome link*, say acute myocardial infarction (AMI; heart attack). (*Process* is what health care providers do on behalf of patients; *outcomes* are what happens as a result of that care.) I can tell from past data that in the time window I'm interested in there will be about $n = 400$ AMI patients at the DH.

To keep things simple let's ignore process for the moment and focus here on one particular binary outcome: death status (mortality) as of 30 days from hospital admission, coded 1 for dead and 0 for alive. (In addition to process this will also depend on the sickness at admission of the AMI patients, but for simplicity let's ignore that initially too.) From the vantage point of

December 2001, say, what may be said about the roughly 400 1s and 0s I will observe in 2002–05?

The meaning of probability. I'm definitely uncertain about the 0–1 death outcomes $Y_1, \ldots, Y_n$ before I observe any of them. Probability is supposed to be the part of mathematics concerned with quantifying uncertainty; can probability be used here? In Chapter 1 I argued that the answer was yes, and that three types of probability—classical, frequentist, and Bayesian—are available (in principle) to quantify uncertainty like that encountered here. I'll focus on the approaches with the most widespread usage—frequentist and Bayesian—in what follows (the classical approach is too hard to apply in any but the simplest of problems).

2.2 Review of frequentist modeling

How can the frequentist definition of probability be applied to the hospital mortality problem? By definition the frequentist approach is based on the idea of hypothetical or actual repetitions of the process being studied, under conditions that are as close to *independent identically distributed* (IID) sampling as possible. When faced with a data set like the 400 1s and 0s $(Y_1, \ldots, Y_n)$ here, the usual way to do this is to think of it as a *random sample*, or *like* a random sample, from some *population* that is of direct interest to me. Then the randomness in my probability statements refers to the process of what I might get if I were to repeat the sampling over and over—the Y_i become *random variables* whose probability distribution is determined by this hypothetical repeated sampling.

Figure 2.1 gives a sketch which illustrates the basis of the frequentist approach to inference. In this figure SD stands for standard deviation, the most common measure of the extent to which the observations y_i in a data set vary, or are spread out, around the center of the data. The center is often measured by the mean $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$, and the SD of a sample of size n is then given by

$$SD = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2}. \tag{2.1}$$

The population size is denoted by N; this is often much larger than the sample size n.

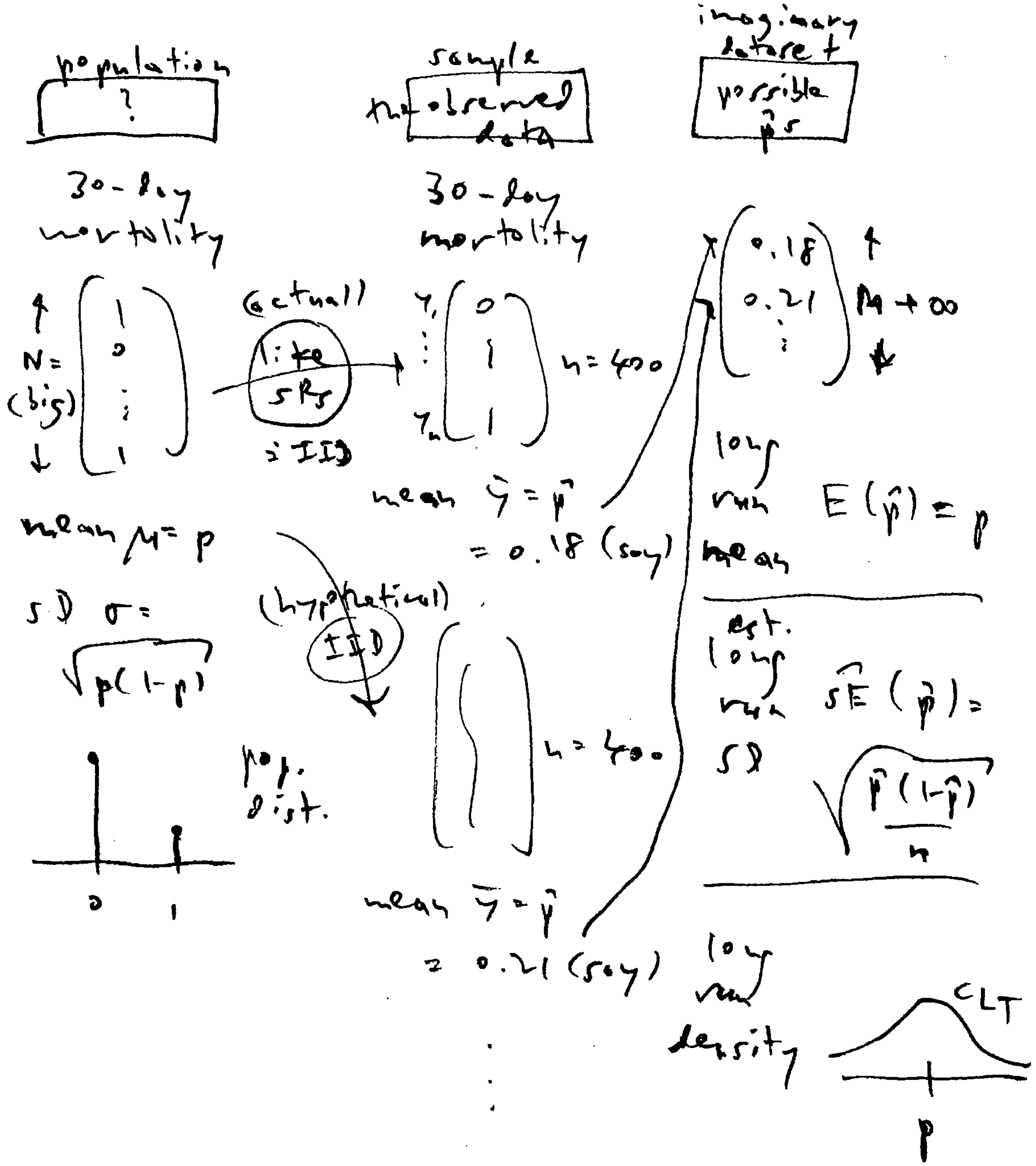


Figure 2.1: *The basis of the frequentist approach to inference.*

With 0/1 (dichotomous) data, like the mortality outcomes in this case study, the population mean μ simply records the proportion p of 1s in the population (check this), and similarly the sample mean $\bar{y}$ keeps track automatically of the observed death rate $\hat{p}$ in the sample. As $N \rightarrow \infty$ the population SD σ with 0/1 data takes on a simple form (check this):

$$\sigma = \sqrt{p(1-p)}. \tag{2.2}$$

It's common in frequentist modeling to make a notational distinction between the random variables Y_i (the placeholders for the process of making IID draws from the population over and over) and the values y_i that the Y_i might take on (although I'll abuse this notation with $\hat{p}$ below). In Figure 2.1 the relationship between the population and the sample data sets can be usefully considered in each of two directions:

- If the population is known you can think about how the sample is likely to come out under IID sampling—this is a probability question.

  Here in this case p would be known and you're trying to figure out the random behavior of the sample mean $\bar{Y} = \hat{p}$.

- If instead only the sample is known your job is to infer the likely composition of the population that could have led to this IID sample—this is a question of statistical inference.

In this problem the sample mean $\bar{y} = \hat{p}$ would be known and your job would be to estimate the population mean p.

Suppose that $N \gg n$, i.e., that even if SRS was used you are effectively dealing with IID sampling. Intuitively both SRS and IID should be "good" (representative) sampling methods (representative here means that you would judge (your uncertainty about) the sampled and unsampled units in the population as exchangeable), so that $\hat{p}$ should be a "good" estimate of p, but what exactly does the word "good" mean in this sentence?

Evidently a good estimator $\hat{p}$ would be likely to be close to the truth p, especially with a lot of data (i.e., if n is large). In the frequentist approach to inference quantifying this idea involves imagining how $\hat{p}$ would have come out if the process by which the observed $\hat{p} = 0.18$ came to you were repeated under IID conditions. This gives rise to the *imaginary data set*, the third part of the diagram in Figure 2.1: we consider all possible $\hat{p}$ values based on an IID sample of size n from a population with $100p\%$ 1s and $100(1-p)\%$ 0s.

Let M be the number of hypothetical repetitions in the imaginary data set. The long-run mean (as $M \to \infty$) of these imaginary $\hat{p}$ values is called the *expected value* of the random variable $\hat{p}$, written $E(\hat{p})$ or $E_{\text{IID}}(\hat{p})$ to emphasize the mechanism of drawing the sample from the population. The long-run SD of these imaginary $\hat{p}$ values is called the *standard error* of the random variable $\hat{p}$, written $SE(\hat{p})$ or $SE_{\text{IID}}(\hat{p})$.

It's natural in studying how the hypothetical $\hat{p}$ values vary around the center of the imaginary data set to make a *histogram* of these values: this is a plot with the possible values of $\hat{p}$ along the horizontal scale and the frequency with which $\hat{p}$ takes on those values on the vertical scale. It's helpful to draw this plot on the *density scale*, which just means that the vertical scale is chosen so that the total area under the histogram is 1. The long-run histogram of the imaginary $\hat{p}$ values on the density scale is called the *(probability) density* of the random variable $\hat{p}$.

The values of $E(\hat{p})$ and $SE(\hat{p})$, and the basic shape of the density of $\hat{p}$, can be determined mathematically (under IID sampling) and verified by simulation. It turns out that

$$E_{\text{IID}}(\hat{p}) = p \quad \text{and} \quad SE_{\text{IID}}(\hat{p}) = \frac{\sigma}{\sqrt{n}} = \sqrt{\frac{p(1-p)}{n}}, \tag{2.3}$$

and the density of $\hat{p}$ for large n is well approximated by the *normal curve* or *Gaussian distribution* (this result is the famous *Central Limit Theorem (CLT)*).

Suppose the sample of size $n = 400$ had 72 1s and 328 0s, so that $\hat{p} = \frac{72}{400} = 0.18$. Thus you would estimate that the population mortality rate p is around 18%, but how much uncertainty should be attached to this estimate?

The above standard error formula is not directly usable because it involves the unknown p, but we can estimate the standard error by plugging in $\hat{p}$:

$$\widehat{SE}(\hat{p}) = \sqrt{\frac{\hat{p}\,(1-\hat{p})}{n}} = \sqrt{\frac{(0.18)(0.82)}{400}} \doteq 0.019. \tag{2.4}$$

In other words, I think p is around 18%, give or take about 1.9%.

A probabilistic uncertainty band can be obtained with the frequentist approach by appeal to the CLT, which says that (for large n) in repeated sampling $\hat{p}$ would fluctuate around p like draws from a normal curve with mean p and SD (SE) 0.019, i.e.,

$$\begin{aligned} 0.95 &\doteq P_F\Big[p - 1.96\,\widehat{SE}(\hat{p}) \le \hat{p} \le p + 1.96\,\widehat{SE}(\hat{p})\Big] \\ &= P_F\Big[\hat{p} - 1.96\,\widehat{SE}(\hat{p}) \le p \le \hat{p} + 1.96\,\widehat{SE}(\hat{p})\Big]\,. \qquad (2.5)\end{aligned}$$

Thus (Neyman 1923) a 95% (frequentist) *confidence interval* for p runs from $\hat{p} - 1.96\,\widehat{SE}(\hat{p})$ to $\hat{p} + 1.96\,\widehat{SE}(\hat{p})$, which in this case is from $0.180 - (1.96)(0.019) = 0.142$ to $0.180 + (1.96)(0.019) = 0.218$, i.e., I am "95% confident that p is between about 14% and 22%". But what does this mean?

Everybody *wants* the confidence interval (CI) to mean

$$P_F(0.142 \le p \le 0.218) \doteq 0.95, \qquad (2.6)$$

but it *can't* mean that in the frequentist approach to probability: in that approach p is treated as a fixed unknown constant, which either is or is not between 0.142 and 0.218. So what does it mean? The answer involves a kind of *calibration* of the CI process: about 95% of the nominal 95% CIs would include the true value, if you were to generate a lot of them via independent IID samples from the population.

The diagram in Figure 2.1 takes up a lot of space; it would be nice to have a more succinct summary of it. A random variable Y is said to follow the *Bernoulli* distribution with *parameter* $0 < p < 1$—this is summarized by saying $Y \sim \text{Bernoulli}(p)$—if Y takes on only the values 1 and 0 and

$$P(Y = y) = \left\{ \begin{array}{cc} p & \text{if } y = 1 \\ 1 - p & \text{if } y = 0 \end{array} \right\} = p^y\,(1-p)^{1-y}. \qquad (2.7)$$

A parameter is just a fixed unknown constant. Another popular name for the parameter p in this model is θ.

Evidently what the population and sample parts of the diagram on page 6 are trying to say, in this notation, is that $(Y_1, \dots, Y_n)$ are drawn in an IID fashion from the Bernoulli distribution with parameter θ.

In the usual shorthand, which I'll use from now on, this is expressed as

$$Y_i \stackrel{\text{IID}}{\sim} \text{Bernoulli}(\theta), \quad i = 1, \dots, n \quad \text{for some } 0 < \theta < 1. \qquad (2.8)$$

This is the frequentist statistical model for the AMI mortality data, except that I have forgotten so far to specify an important ingredient: what is the population of patients whose mean (underlying death rate) is θ? As a frequentist (as noted above), to use probability to quantify my uncertainty about the 1s and 0s, I have to think of them as either literally a random sample or *like* a random sample from some population, either hypothetical or actual. What are some possibilities for this population?

- All AMI patients who might have come to the DH in 2002–05 if the world had turned out differently; or

- Assuming sufficient *time-homogeneity* in all relevant factors, you could try to argue that the collection of all 400 AMI patients at the DH from 2002–05 is like a random sample of size 400 from the population of all AMI patients at the DH from (say) 1997–2006; or

- *Cluster sampling* is a way to choose, e.g., patients by taking a random sample of hospitals and then a random sample of patients *nested* within those hospitals. What we actually have here is a kind of cluster sample of all 400 AMI patients from the DH in 2002–2005 (and no patients from any other hospitals). Cluster samples tend to be less informative than SRS samples of the same size because of (positive) intracluster correlation (patients in a given hospital tend to be more similar in their outcomes than would an SRS of the same size from the population of all the patients in all the hospitals). Assuming the DH to be representative of some broader collection of hospitals in California and (unwisely) ignoring intracluster correlation, you could try to argue that these 400 1s and 0s were like a simple random sample of 400 AMI patients from this larger collection of hospitals.

I think you would agree with me that none of these options is entirely compelling.

If you're willing to ignore this difficulty and pretend the data are like a sample from some population, interest would then focus on inference about the parameter θ, the "underlying death rate" in this larger collection of patients to which you feel comfortable generalizing the 400 1s and 0s: if θ were unusually high, that would be prima facie evidence of a possible quality of care problem at the DH.

2.2.1 The likelihood function

Suppose (as above) that

$$Y_i \overset{\text{IID}}{\sim} \text{B}(\theta), \quad i = 1, \ldots, n \quad \text{for some } 0 < \theta < 1. \tag{2.9}$$

Since the Y_i are independent, the joint sampling distribution of all of them, $P(Y_1 = y_1, \ldots, Y_n = y_n)$, is the product of the separate, or *marginal*, sampling distributions $P(Y_1 = y_1), \ldots, P(Y_n = y_n)$:

$$\begin{aligned} P(Y_1 = y_1, \ldots, Y_n = y_n) &= P(Y_1 = y_1) \cdots P(Y_n = y_n) \\ &= \prod_{i=1}^{n} P(Y_i = y_i)\,. \end{aligned} \tag{2.10}$$

But since the Y_i are also identically distributed, and each one is Bernoulli with parameter θ, i.e., $P(Y_i = y_i) = \theta^{y_i} (1-\theta)^{1-y_i}$, the joint sampling distribution can be written

$$P(Y_1 = y_1, \ldots, Y_n = y_n) = \prod_{i=1}^{n} \theta^{y_i} (1-\theta)^{1-y_i}. \tag{2.11}$$

Let's use the symbol y to stand for the vector of observed data values $(y_1, \ldots, y_n)$.

Before any data have arrived, this joint sampling distribution is a function of y for fixed θ—it tells you how the data would be likely to behave in the future if you were to take an IID sample from the Bernoulli(θ) distribution. In 1921 Fisher had the following idea: after the data have arrived it makes more sense to interpret (2.11) as a function of θ for fixed y—he called this the *likelihood function* for θ in the Bernoulli(θ) model:

$$\begin{aligned} l(\theta|y) &= l(\theta|y_1, \ldots, y_n) = \prod_{i=1}^{n} \theta^{y_i} (1-\theta)^{1-y_i} \qquad (2.12) \\ &= P(Y_1 = y_1, \ldots, Y_n = y_n) \text{ but interpreted} \\ & \text{as a function of } \theta \text{ for fixed } y. \end{aligned}$$

Fisher tried to create a theory of inference about θ based only on this function—we will see below that this is an important ingredient, but not the only important ingredient, in inference from the Bayesian viewpoint.

The Bernoulli(θ) likelihood function can be simplified as follows:

$$l(\theta|y) = \theta^s(1-\theta)^{n-s}, \tag{2.13}$$

where $s = \sum_{i=1}^n y_i$ is the number of 1s in the sample and $(n-s)$ is the number of 0s. What does this function look like?

With $n = 400$ and $s = 72$ it's easy to get `Maple` to plot it:

```
rosalind 329> maple

    |\^/|     Maple 9 (SUN SPARC SOLARIS)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo
 \  MAPLE  /  Maple Inc. 2003. ll rights reserved. Maple is a
 <____ ____>  trademark of Waterloo Maple Inc.
      |       Type ? for help.

> l := ( theta, s, n ) -> theta^s * ( 1 - theta )^( n - s );

                                         s            (n - s)
           l := (theta, s, n) -> theta  (1 - theta)

> plotsetup( x11 );

> plot( l( theta, 72, 400 ), theta = 0 .. 1 );

> plot( l( theta, 72, 400 ), theta = 0.12 .. 0.25 );
```

The results are graphed in Figure 2.2 for $\theta \in (0,1)$ and again in Figure 2.3 in a kind of closeup in the interesting region, $\theta \in (0.12, 0.25)$. Does this function remind you of anything? It reminds me a lot of a Gaussian distribution for θ, although Fisher fought hard to resist this interpretation (thinking of the likelihood function for large n as approximately a Gaussian density for θ is, as will be seen below, completely natural from the Bayesian viewpoint, and Fisher—with his likelihood theory—was trying hard to do something non-Bayesian, for reasons that I'll mention along the way as this story unfolds).

Note that the likelihood function $l(\theta|y) = \theta^s(1-\theta)^{n-s}$ in this problem depends on the data vector y only through $s = \sum_{i=1}^n y_i$—Fisher referred to any such data summary as a *sufficient statistic* (with respect to the given likelihood function).

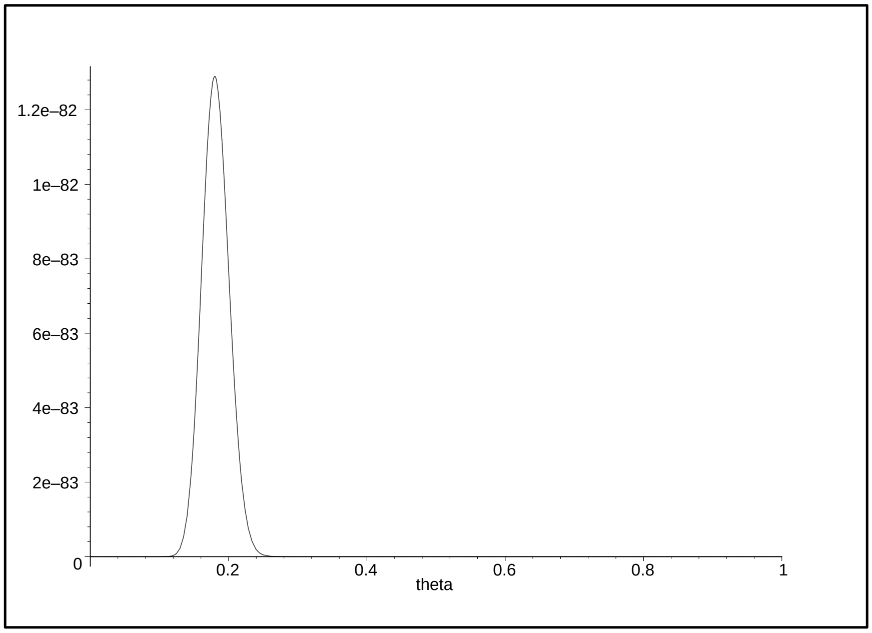


Figure 2.2: *Plot of the Bernoulli likelihood function* $\theta^s(1-\theta)^{n-s}$ *for* $\theta \in (0,1)$ *with* $n = 400$ *and* $s = 72$*.*

There's an interesting thing about Figures 2.2 and 2.3: the vertical scale shows that `Maple` has cheerfully worked without complaining with likelihood values on the order of 10^{-82}. In fact, in floating point calculations, at least in principle, `Maple` is supposed to be able to work smoothly with numbers all the way from 10^{-400} to 10^{+400} or so (for comparison, there are only about 10^{80} elementary particles in the universe), but the possibility of numerical instabilities with likelihood values so small makes me a bit nervous, no matter what the software manufacturer says. Let's see; the likelihood values got that small by multiplying 400 numbers between 0 and 1 together. This is a situation that's crying out for taking logarithms: sums are more numerically stable than products. Fisher came to the same conclusion (for somewhat different reasons): it's often at least as useful to look at the logarithm of the likelihood function as the likelihood function itself.

```
  > ll := ( theta, s, n ) -> log( l( theta, s, n ) );
```

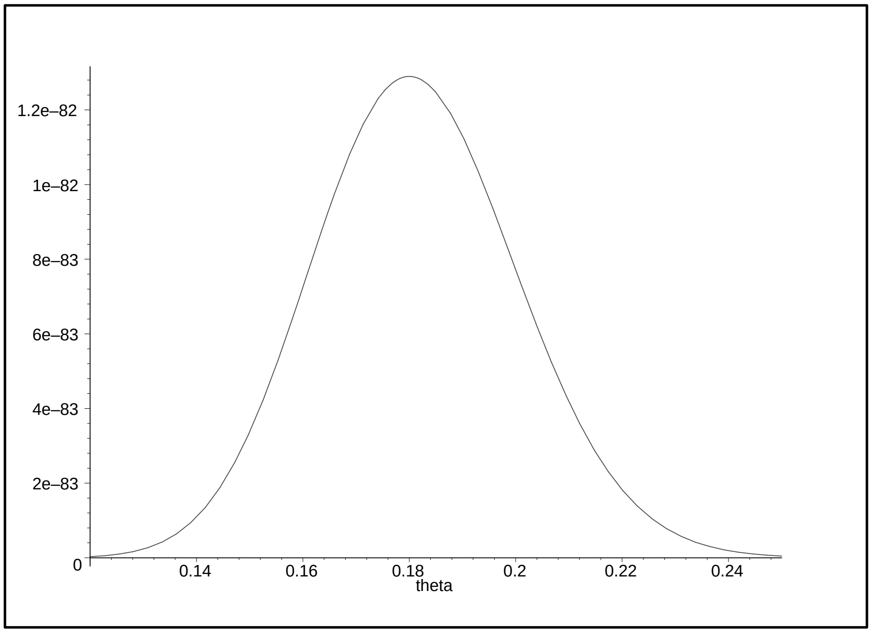


Figure 2.3: *Closeup of* $\theta^{72}(1-\theta)^{328}$ *for* $\theta \in (0.12, 0.25)$.

```
> plot( ll( theta, 72, 400 ), theta = 0.12 .. 0.25 );
```

In this case, as is often true for large n, the log likelihood function looks locally quadratic around its maximum.

Fisher had the further idea that the *maximum* of the likelihood function would be a good estimate of θ (we'll look later at conditions under which this makes sense from the Bayesian viewpoint). Since the logarithm function is monotone increasing, it's equivalent in maximizing the likelihood to maximize the log likelihood, and for a function as well behaved as this you can do that by setting its first partial derivative with respect to θ (the so-called *score function*) to 0 and solving:

```
> score := simplify( diff( ll( theta, s, n ), theta ) );

                              s - n theta
              score := - ------------------
                           theta (-1 + theta)
```

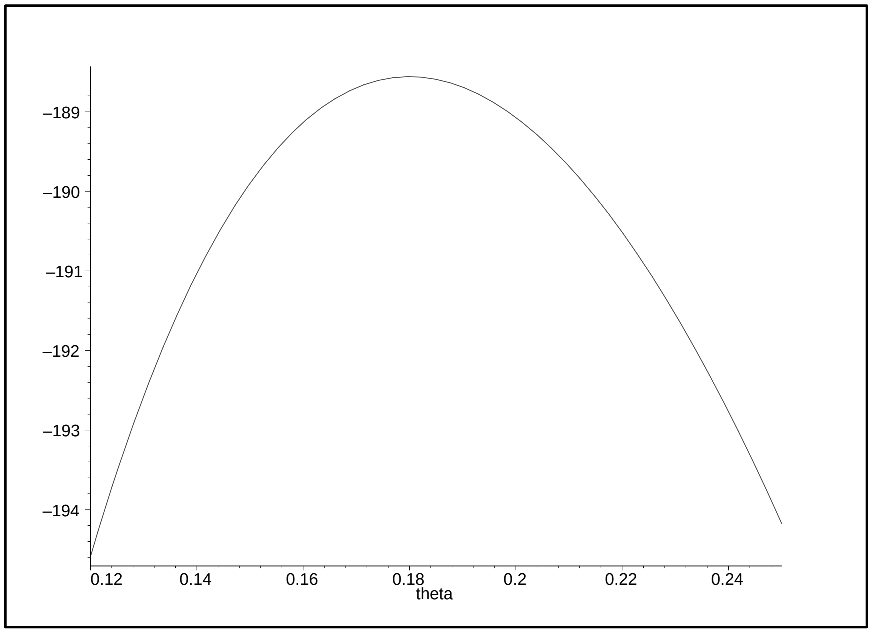


Figure 2.4: *Log likelihood function in the Bernoulli model with $n = 400$ and $s = 72$.*

```
  > solve( score = 0, theta );

                                    s/n
```

Fisher called the function of the data that maximizes the likelihood (or log likelihood) function is the *maximum likelihood estimate* (MLE) $\hat{\theta}_{\text{MLE}}$. You can see that in this case $\hat{\theta}_{\text{MLE}}$ is just the sample mean $\bar{y} = \frac{s}{n}$, which we've previously seen (via Neyman's approach to inference) is a sensible estimate of θ.

Note also that if you maximize $l(\theta|y)$ and I maximize $c\,l(\theta|y)$ for any constant $c > 0$, we'll get the same thing, i.e., the likelihood function is only defined up to a positive multiple; Fisher's actual definition was

$$l(\theta|y) = c\,P(Y_1 = y_1, \ldots, Y_n = y_n)$$

for any (normalizing constant) $c > 0$ (this will be put to Bayesian use below).

From now on c in expressions like the likelihood function above will be a generic (and often unspecified) positive constant.

2.2.2 Calibrating the MLE

Maximum likelihood provides a basic principle for estimation of a (population) parameter θ from the frequentist/likelihood point of view, but how should the accuracy of $\hat{\theta}_{\text{MLE}}$ be assessed? Evidently in the frequentist approach I want to compute the variance or standard error of $\hat{\theta}_{\text{MLE}}$ in repeated sampling, or at least estimates of these quantities—let's focus on the estimated variance $\hat{V}\left(\hat{\theta}_{\text{MLE}}\right)$. Fisher (1922) proposed an approximation to $\hat{V}\left(\hat{\theta}_{\text{MLE}}\right)$ that works well for large n and makes good intuitive sense.

In the AMI mortality case study, where $\hat{\theta}_{\text{MLE}} = \hat{\theta} = \frac{s}{n}$ (the sample mean), we already know that

$$V\left(\hat{\theta}_{\text{MLE}}\right) = \frac{\theta(1-\theta)}{n} \quad \text{and} \quad \hat{V}\left(\hat{\theta}_{\text{MLE}}\right) = \frac{\hat{\theta}(1-\hat{\theta})}{n}, \tag{2.14}$$

but Fisher wanted to derive results like this in a more basic and general way.

Imagine quadrupling the sample size in this case study from $n = 400$ to $n = 1600$ while keeping the observed death rate constant at 0.18—what would happen to the log likelihood function? To answer this question, recall that as far as maximizing the likelihood function is concerned it's equally good to work with any (positive) constant multiple of it, which is equivalent to saying that I can add any constant I want to the log likelihood function without harming anything.

In the `Maple` plot in Figure 2.5 I've added a different constant to each of the log likelihood functions with $(s, n) = (72, 400)$ and $(288, 1600)$ so that they both go through the point $(\hat{\theta}_{\text{MLE}}, 0)$:

```
> plot( { ll( theta, 72, 400 ) - evalf( ll( 72 / 400, 72,
    400 ) ), ll( theta, 288, 1600 ) - evalf( ll( 288 / 1600,
    288, 1600 ) ) }, theta = 0.12 .. 0.25, color = black );
```

Notice in Figure 2.5 that what's happened as n went from 400 to 1600 while holding the MLE constant at 18% mortality is that the curve has become substantially steeper around its maximum, which means that the second derivative of the log likelihood function at $\hat{\theta}_{\text{MLE}}$, a negative number,

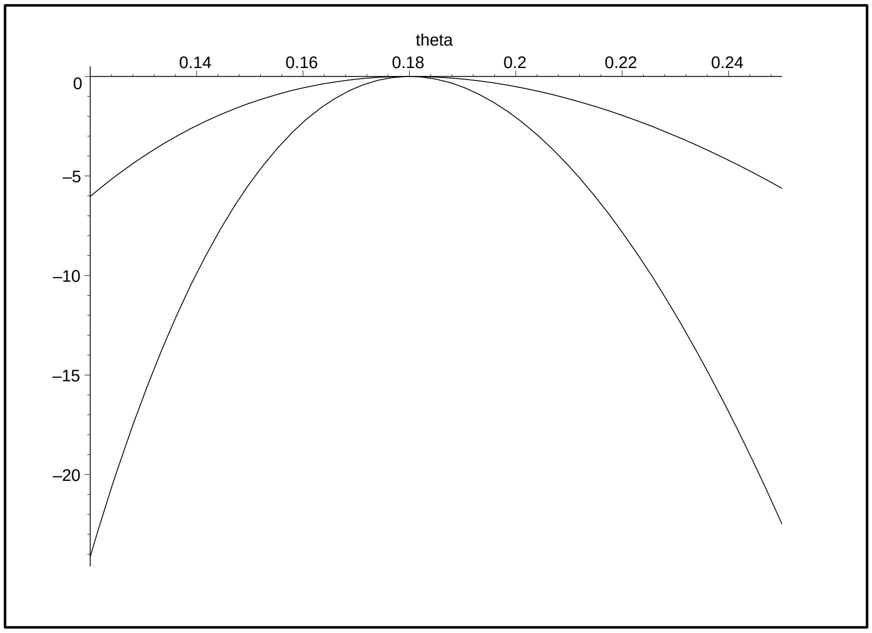


Figure 2.5: *Bernoulli log likelihood functions with* $(s, n) = (72, 400)$ *(flatter curve) and* $(288, 1600)$ *(steeper curve).*

has increased in magnitude. This led Fisher to define a quantity he called the *information* in the sample about θ—in his honor it's now called the (observed) *Fisher information*:

$$\hat{I}\left(\hat{\theta}_{\text{MLE}}\right) = \left[-\frac{\partial^2}{\partial \theta^2} \log l(\theta | y) \right]_{\theta = \hat{\theta}_{\text{MLE}}} . \tag{2.15}$$

This quantity increases as n goes up, whereas our uncertainty about θ based on the sample, as measured by $\hat{V}\left(\hat{\theta}_{\text{MLE}}\right)$, should go down with n. Fisher conjectured and proved that the information and the estimated variance of the MLE in repeated sampling have the following simple inverse relationship when n is large:

$$\hat{V}\left(\hat{\theta}_{\text{MLE}}\right) \doteq \hat{I}^{-1}\left(\hat{\theta}_{\text{MLE}}\right) . \tag{2.16}$$

He further proved that for large n (a) the MLE is approximately *unbiased*, meaning that in repeated sampling

$$E\left(\hat{\theta}_{\text{MLE}}\right) \doteq \theta, \tag{2.17}$$

and (b) the sampling distribution of the MLE is approximately Gaussian with mean θ and estimated variance given by (2.16):

$$\hat{\theta}_{\text{MLE}} \overset{\cdot}{\sim} N\left[\theta, \hat{I}^{-1}\left(\hat{\theta}_{\text{MLE}}\right)\right]. \tag{2.18}$$

Thus for large n an approximate 95% confidence interval for θ is given by $\hat{\theta}_{\text{MLE}} \pm 1.96\sqrt{\hat{I}^{-1}\left(\hat{\theta}_{\text{MLE}}\right)}$.

You can differentiate to compute the information yourself in the AMI mortality case study, or you can use `Maple` to do it for you:

```
> score := ( theta, s, n ) -> simplify( diff( ll( theta, s, n ),
    theta ) );

      score := (theta, s, n) -> simplify(diff(ll(theta, s, n),
        theta))

> score( theta, s, n );

                                 s - n theta
                            - ------------------
                              theta (-1 + theta)

> diff2 := ( theta, s, n ) -> simplify( diff( score( theta, s,
    n ), theta ) );

    diff2 := (theta, s, n) -> simplify(diff(score(theta, s, n),
      theta))

> diff2( theta, s, n );
                                 2
                         -n theta  - s + 2 s theta
                         -------------------------
                                2             2
                           theta  (-1 + theta)
```

```
> information := ( s, n ) -> simplify( eval( - diff2( theta, s,
    n ), theta = s / n ) );

> information( s, n );

                                        3
                                       n
                                - ----------
                                  s (-n + s)

> variance := ( s, n ) -> 1 / information( s, n );

                                                 1
                  variance := (s, n) -> -----------------
                                        information(s, n)

> variance( s, n );

                                  s (-n + s)
                                - ----------
                                        3
                                       n
```

This expression can be further simplified to yield

$$\hat{V}\left(\hat{\theta}_{\text{MLE}}\right) \doteq \frac{\frac{s}{n}\left(1-\frac{s}{n}\right)}{n} = \frac{\hat{\theta}(1-\hat{\theta})}{n}, \tag{2.19}$$

which coincides with (2.14).

From (2.19) another expression for the Fisher information in this problem is

$$\hat{I}\left(\hat{\theta}_{\text{MLE}}\right) = \frac{n}{\hat{\theta}(1-\hat{\theta})}. \tag{2.20}$$

As n increases, $\hat{\theta}(1-\hat{\theta})$ will tend to the constant $\theta(1-\theta)$ (this is well-defined because we've assumed that $0 < \theta < 1$, because $\theta = 0$ and 1 are probabilistically uninteresting), which means that information about θ on the basis of $(y_1, \ldots, y_n)$ in the IID Bernoulli model increases at a rate proportional to n as the sample size grows.

This is generally true of the MLE (i.e., in *regular* parametric problems):

$$\hat{I}\left(\hat{\theta}_{\text{MLE}}\right) = O(n) \quad \text{and} \quad \hat{V}\left(\hat{\theta}_{\text{MLE}}\right) = O\left(n^{-1}\right), \tag{2.21}$$

as $n \rightarrow \infty$, where the notation $a_n = O(b_n)$ means that the ratio $\left| \frac{a_n}{b_n} \right|$ is bounded as n grows. Thus uncertainty about θ on the basis of the MLE goes down like $\frac{c_{\text{MLE}}}{n}$ on the variance scale with more and more data (in fact Fisher showed that c_{MLE} achieves the lowest possible value: the MLE is *efficient*).

2.3 Bayesian modeling

As a Bayesian in this situation, my job is to quantify my uncertainty about the 400 binary observables I'll get to see starting in 2002, i.e., my initial modeling task is *predictive* rather than inferential. There is no samples-and-populations story in this approach, but probability and random variables arise in a different way: quantifying my uncertainty (for the purpose of betting with someone about some aspect of the 1s and 0s, say) requires *eliciting* from myself a joint predictive distribution that accurately captures my judgments about what I'll see: $P_{B:\text{me}}(Y_1 = y_1, \ldots, Y_n = y_n)$.

Notice that in the frequentist approach the random variables describe the process of observing a repeatable event (the "random sampling" appealed to earlier), whereas in the Bayesian approach I use random variables to quantify my uncertainty about observables I haven't seen yet. I'll argue later that the concept of probabilistic accuracy has two components: I want my uncertainty assessments to be both internally and externally consistent, which corresponds to the Bayesian and frequentist ideas of coherence and calibration, respectively.

2.3.1 Exchangeability

Eliciting a 400-dimensional distribution doesn't sound easy; major simplification is evidently needed. In this case, and many others, this is provided by *exchangeability* considerations.

If (as in the frequentist approach) I have no relevant information that distinguishes one AMI patient from another, my uncertainty about the 400 1s and 0s is symmetric, in the sense that a random permutation of the order in which the 1s and 0s were labeled from 1 to 400 would leave my uncertainty about them unchanged. de Finetti (1930, 1964) called random variables with this property exchangeable:

Definition. $\{Y_i, i = 1, \ldots, n\}$ are *exchangeable* if the distributions of $(Y_1, \ldots, Y_n)$ and $(Y_{\pi(1)}, \ldots, Y_{\pi(n)})$ are the same for all permutations $(\pi(1), \ldots, \pi(n))$.

NB Exchangeability and IID are not the same: IID implies exchangeability, and exchangeable Y_i do have identical marginal distributions, but they're not independent (if you were expecting a priori about 15% 1s, say (that's the 30-day death rate for AMI with average-quality care), the knowledge that in the first 50 outcomes at the DH 20 of them were deaths would certainly change your prediction of the 51st).

de Finetti also defined partial or conditional exchangeability (e.g., Draper et al., 1993): if, e.g., the gender X of the AMI patients were available, and if there were evidence from the medical literature that 1s tended to be noticeably more likely for men than women, then you would probably want to assume conditional exchangeability of the Y_i given X (meaning that the male and female 1s and 0s, viewed as separate collections of random variables, are each unconditionally exchangeable). This is related to Fisher's (1956) idea of recognizable subpopulations.

de Finetti's Theorem for 1s and 0s. The judgment of exchangeability still seems to leave the joint distribution of the Y_i quite imprecisely specified.

After defining the concept of exchangeability, however, de Finetti went on to prove a remarkable result: if you're willing to regard the $\{Y_i, i = 1, \ldots, n\}$ as part (for instance, the beginning) of an infinite exchangeable sequence of 1s and 0s (meaning that every finite subsequence is exchangeable), then there's a simple way to characterize your joint distribution, if it's to be coherent (e.g., de Finetti, 1975; Bernardo and Smith, 1994).

(Finite versions of the theorem have since been proven, which say that the longer the exchangeable sequence into which you're willing to embed $\{Y_i, i = 1, \ldots, n\}$, the harder it becomes to achieve coherence with any probability specification that's far removed from the one below.)

de Finetti's Representation Theorem. If $Y_1, Y_2, \ldots$ is an infinitely exchangeable sequence of 0–1 random quantities with probability measure P, there exists a distribution function $Q(\theta)$ such that the joint distribution $p(y_1, \ldots, y_n)$ for $Y_1, \ldots, Y_n$ is of the form

$$p(y_1, \ldots, y_n) = \int_0^1 \prod_{i=1}^n \theta^{y_i} (1 - \theta)^{1 - y_i} \, dQ(\theta) \ , \tag{2.22}$$

$$\text{where } Q(\theta) = \lim_{n \to \infty} P\left(\frac{1}{n} \sum_{i=1}^n Y_i \leq \theta\right) \text{ and } \theta \overset{P}{=} \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^n Y_i \ .$$

θ can also be interpreted as the marginal probability $P(Y_i = 1)$ that any of the Y_i is 1.

The Law of Total Probability. The distribution function Q will generally be well-behaved enough to have a density: $dQ(\theta) = p(\theta) d\theta$.

In this case de Finetti's Theorem says

$$p(y_1, \ldots, y_n) = \int_0^1 \prod_{i=1}^n \theta^{y_i} (1-\theta)^{1-y_i} \, p(\theta) d\theta. \tag{2.23}$$

Important digression. We saw in part 1 of the lecture notes that for the true-false propositions D and A,

$$\begin{aligned} P(D) &= P(D \text{ and } A) + P[D \text{ and } (\text{not } A)] \\ &= P(A)\, P(D|A) + P(\text{not } A)\, P(D|\text{not } A). \end{aligned} \tag{2.24}$$

This is a special case of the Law of Total Probability (LTP).

Notice that A and (not A) divide, or partition, the collection of all possible outcomes into two non-overlapping (mutually exclusive) and exhaustive possibilities.

Let $A_1, \ldots, A_k$ be any finite partition, i.e., $P(A_i \text{ and } A_j) = 0$ (mutually exclusive) and $\sum_{i=1}^k P(A_i) = 1$ (exhaustive).

Then a more general version of the LTP gives that

$$\begin{aligned} P(D) &= P(D \text{ and } A_1) + \ldots + P(D \text{ and } A_k) \\ &= P(A_1)\, P(D|A_1) + \ldots + P(A_k)\, P(D|A_k) \\ &= \sum_{i=1}^k P(A_i)\, P(D|A_i). \end{aligned} \tag{2.25}$$

2.3.2 Hierarchical (mixture) modeling

There is a continuous version of the LTP: by analogy with (25), if X and Y are real-valued random variables

$$p(y) = \int_{-\infty}^{\infty} p(x)\, p(y|x)\, dx. \tag{2.26}$$

$p(x)$ in this expression is called a mixing distribution.

Intuitively (26) says that the overall probability behavior $p(y)$ of Y is a mixture (weighted average) of the conditional behavior $p(y|x)$ of Y given X, weighted by the behavior $p(x)$ of X.

Another way to put this is to say that you have a choice: you can either model the random behavior of Y directly, through $p(y)$, or hierarchically, by first modeling the random behavior of X, through $p(x)$, and then modeling the conditional behavior of Y given X, through $p(y|x)$.

Notice that X and Y are completely general in this discussion—in other words, given any quantity Y that you want to model stochastically, you're free to choose any X upon which Y depends and model Y hierarchically given X instead, if that's easier.

Symbolically

$$Y \quad \leftrightarrow \quad \left\{ \begin{array}{c} X \\ Y|X \end{array} \right\}. \tag{2.27}$$

The reason for bringing all of this up now is that (23) can be interpreted as follows, with θ playing the role of x:

$$\begin{aligned} p(y_1, \ldots, y_n) &= \int_0^1 p(y_1, \ldots, y_n|\theta)\, p(\theta)\, d\theta \\ &= \int_0^1 \prod_{i=1}^n \theta^{y_i} (1-\theta)^{1-y_i}\, p(\theta)\, d\theta. \end{aligned} \tag{2.28}$$

2.3.3 The simplest mixture model

(28) implies that in any coherent expression of uncertainty about exchangeable binary quantities $Y_1, \ldots, Y_n$,

$$p(y_1, \ldots, y_n|\theta) = \prod_{i=1}^n \theta^{y_i} (1-\theta)^{1-y_i}. \tag{2.29}$$

But (a) the left side of (29), interpreted as a function of θ for fixed $y = (y_1, \ldots, y_n)$, is recognizable as the likelihood function for θ given y, (b) the right side of (29) is recognizable as the likelihood function for θ in IID Bernoulli sampling, and (c) (29) says that these must be the same.

Thus, to summarize de Finetti's Theorem intuitively, the assumption of exchangeability in your uncertainty about binary observables $Y_1, \ldots, Y_n$ amounts to behaving as if

- there's a quantity called θ, interpretable as either the long-run relative frequency of 1s or the marginal probability that any of the Y_i is 1,
- you need to treat θ as a random quantity with density $p(\theta)$, and
- conditional on this θ the Y_i are IID $\mathrm{B}(\theta)$.

In yet other words, for a Bayesian whose uncertainty about binary Y_i is exchangeable, the model may effectively be taken to have the simple mixture or hierarchical representation

$$\left\{ \begin{array}{ccc} \theta & \sim & p(\theta) \\ (Y_i|\theta) & \stackrel{\text{IID}}{\sim} & \mathrm{B}(\theta), \ i = 1, \dots, n \end{array} \right\}. \tag{2.30}$$

This is our first of many examples of a parametric Bayesian model (as contrasted with non-parametric and semi-parametric Bayesian models, which will come up in Section 2.8 below; parametric modeling is the main topic in this course).

2.3.4 Conditional independence

This is the link between frequentist and Bayesian modeling of binary outcomes: exchangeability implies that you should behave like a frequentist as far as the likelihood function is concerned (taking the Y_i to be IID Bernoulli with parameter θ), but a frequentist who treats θ as a random variable with a mixing distribution $p(\theta)$.

NB This is the first example of a general pattern:

$$Y_i \text{ exchangeable} \leftrightarrow \left\{ \begin{array}{c} Y_i \text{ conditionally IID} \\ \text{given one or more parameters} \end{array} \right\}. \tag{2.31}$$

So exchangeability is a special kind of conditional independence: binary exchangeable y_i are not independent, but they become conditionally independent given θ.

(30) is an example of the simplest kind of hierarchical model (HM): a model at the top level for the underlying death rate θ, and then a model below that for the binary mortality indicators Y_i conditional on θ (this is a basic instance of (27): it's not easy to model the predictive distribution for $(Y_1, \dots, Y_n)$ directly, but it becomes a lot easier when θ is introduced at the top level of a 2–level hierarchy).

To emphasize an important point mentioned above, to make sense of this in the Bayesian approach you have to treat θ as a random variable, even though logically it's a fixed unknown constant.

This is the main conceptual difference between the Bayesian and frequentist approaches: as a Bayesian you use the machinery of random variables to express your uncertainty about unknown quantities.

Approach	Fixed	Random
Frequentist	θ	Y
Bayesian	y	θ

2.3.5 Prior, posterior, and predictive distributions

What's the meaning of the mixing distribution $p(\theta)$?

$p(\theta)$ doesn't involve $y = (y_1, \ldots, y_n)$, so it must represent your information about θ before the data set y arrives—it makes sense to call it your prior distribution for θ.

I'll address how you might go about specifying this distribution below.

Q: If $p(\theta)$ represents your information about θ before the data arrive, what represents this information after y has been observed?

A: It has to be $p(\theta|y)$, the conditional distribution for θ given how y came out.

It's natural to call this the posterior distribution for θ given y.

Q: How do you get from $p(\theta)$ to $p(\theta|y)$, i.e., how do you update your information about θ in light of the data?

A: Bayes' Theorem for continuous quantities:

$$p(\theta|y) = \frac{p(\theta)\, p(y|\theta)}{p(y)}. \tag{2.32}$$

This requires some interpreting. As a Bayesian I'm conditioning on the data, i.e., I'm thinking of the left-hand side of (32) as a function of θ for fixed y, so that must also be true of the right-hand side. Thus (a) $p(y)$ is just a constant—in fact, you can think of it as the normalizing constant, put into the equation to make the product $p(\theta)\, p(y|\theta)$ integrate to 1; and (b) $p(y|\theta)$ may look like the usual frequentist sampling distribution for y given θ (Bernoulli, in this case), but I have to think of it as a function of θ for fixed y. We've already encountered this idea (p. 15): $l(\theta|y) = c\, p(y|\theta)$ is Fisher's likelihood function.

So Bayes' Theorem becomes

$$\begin{array}{rcccl} p(\theta|y) & = & c & \cdot\ p(\theta) \cdot & l(\theta|y)\ , \\ \text{posterior} & = & \left(\begin{array}{c} \text{normalizing} \\ \text{constant} \end{array} \right) & \cdot \text{ prior } \cdot & \text{likelihood}\ . \end{array} \tag{2.33}$$

You can also readily construct predictive distributions for the y_i before they're observed, or for future y_i once some of them are known.

For example, by the LTP, the posterior predictive distribution for $(y_{m+1}, \ldots, y_n)$ given $(y_1, \ldots, y_m)$ is

$$\begin{aligned} p(y_{m+1}, \ldots, y_n | y_1, \ldots, y_m) = \\ \int_0^1 p(y_{m+1}, \ldots, y_n | \theta, y_1, \ldots, y_m) \, p(\theta | y_1, \ldots, y_m) \, d\theta. \end{aligned} \tag{2.34}$$

Consider $p(y_{m+1}, \ldots, y_n | \theta, y_1, \ldots, y_m)$: if you knew θ, the information $y_1, \ldots, y_m$ about how the first m of the y_i came out would be irrelevant (imagine predicting the results of IID coin-tossing: if you somehow knew that the coin was perfectly fair, i.e., that $\theta = 0.5$, then getting (say) 6 heads in the first 10 tosses would be useless to you in quantifying the likely behavior of the next (say) 20 tosses—you'd just use the known true value of θ).

Thus $p(y_{m+1}, \ldots, y_n | \theta, y_1, \ldots, y_m)$ is just $p(y_{m+1}, \ldots, y_n | \theta)$, which in turn is just the sampling distribution under IID $\mathrm{B}(\theta)$ sampling for the binary observables $y_{m+1}, \ldots, y_n$, namely $\prod_{i=m+1}^n \theta^{y_i} (1 - \theta)^{1 - y_i}$.

And finally $p(\theta | y_1, \ldots, y_m)$ is recognizable as just the posterior distribution for θ given the first m of the binary outcomes.

Putting this all together gives

$$\begin{aligned} p(y_{m+1}, \ldots, y_n | y_1, \ldots, y_m) = \\ = \int_0^1 \prod_{i=m+1}^n \theta^{y_i} (1 - \theta)^{1 - y_i} \, p(\theta | y_1, \ldots, y_m) \, d\theta \end{aligned} \tag{2.35}$$

(we can't compute (35) yet because $p(\theta | y_1, \ldots, y_m)$ depends on $p(\theta)$, which we haven't specified so far).

This also brings up a key difference between a parameter like θ on the one hand and the Y_i, before you've observed any data, on the other: parameters are inherently unobservable.

This makes it harder to evaluate the quality of your uncertainty assessments about θ than to do so about the observable y_i: to see how well you're doing in predicting observables you can just compare your predictive distributions for them with how they actually turn out, but of course this isn't possible with things like θ which you'll never actually see.

Inference and prediction. Coherence and calibration The de Finetti approach to modeling emphasizes the prediction of observables as a valuable adjunct to inference about unobservable parameters, for at least two reasons:

• Key scientific questions are often predictive in nature: e.g., rather than asking “Is drug A better than B (on average across many patients) for lowering blood pressure?” (inference) the ultimate question is “How much more will drug A lower this patient’s blood pressure than drug B?” (prediction); and

• Good diagnostic checking is predictive: An inference about an unobservable parameter can never be directly verified, but often you can reasonably conclude that inferences about the parameters of a model which produces poor predictions of observables are also suspect.

With the predictive approach parameters diminish in importance, especially those that have no physical meaning—such parameters (unlike θ above) are just place-holders for a particular kind of uncertainty on your way to making good predictions.

It’s arguable (e.g., Draper, 1995) that the discipline of statistics, and particularly its applications in the social sciences, would be improved by a greater emphasis on predictive feedback.

This is not to say that parametric thinking should be abolished.

As the calculations on the previous pages emphasized, parameters play an important simplifying role in forming modeling judgments: the single strongest simplifier of a joint distribution is independence of its components, and whereas, e.g., in the mortality example the Y_i are not themselves independent, they become so conditional on θ.

Where Does the Prior Come From? de Finetti’s Theorem for 0–1 outcomes says informally that if you’re trying to make coherent (internally consistent) probability judgments about a series of 1s and 0s that you judge exchangeable, you may as well behave like a frequentist—IID B(θ)—with a prior distribution $p(\theta)$.

But where does this prior come from?

NB Coherence doesn’t help in answering this question—it turns out that any prior $p(\theta)$ could be part of somebody’s coherent probability judgments.

Some people regard the need to answer this question in the Bayesian approach as a drawback, but it seems to me (and to many other people) to be a positive feature, as follows.

From Bayes’ Theorem the prior is supposed to be a summary of what you know (and don’t know) about θ before the y_i start to arrive: from previous datasets of which you’re aware, from the relevant literature, from expert opinion, ... from all “good” source(s), if any exist.

Such information is almost always present, and should presumably be used when available—the issue is how to do so “well.”

> The goal is evidently to choose a prior that you’ll retrospectively be proud of, in the sense that your predictive distributions for the observables (a) are well-centered near the actual values and (b) have uncertainty bands that correspond well to the realized discrepancies between actual and predicted values. This is a form of calibration of your probability judgments.

There is no guaranteed way to do this, just as there is no guaranteed way to arrive at a “good” frequentist model (see “Where does the likelihood come from?” below).

Choosing a “good” prior. Some general comments on arriving at a “good” prior:

• There is a growing literature on methodology for elicitation of prior information (e.g., Kadane et al., 1980; Craig et al., 1997; Kadane and Wolfson, 1997; O’Hagan, 1997), which brings together ideas from statistics and perceptual psychology (e.g., people turn out to be better at estimating percentiles of a distribution than they are at estimating standard deviations (SDs)).

• Bayes’ Theorem on the log scale says (apart from the normalizing constant)

$$\log(\text{posterior}) = \log(\text{prior}) + \log(\text{likelihood}), \tag{2.36}$$

i.e., (posterior information) = (data information) + (prior information). This means that close attention should be paid to the information content of the prior by, e.g., density-normalizing the likelihood and plotting it on the same scale as the prior: it’s possible for small n for the prior to swamp the data, and in general you shouldn’t let this happen without a good reason for doing so.

Comfort can also be taken from the other side of this coin: with large n (in many situations, at least) the data will swamp the prior, and specification errors become less important.

• When you notice you’re quite uncertain about how to specify the prior, you can try sensitivity or (pre-posterior) analysis: exploring the mapping from prior to posterior, before the data are gathered, by (a) generating some possible values for the observables, (b) writing down several plausible forms for the prior, and (c) carrying these forward to posterior distributions.

If the resulting distributions are similar (i.e., if "all reasonable roads lead to Rome"), you've uncovered a useful form of stability in your results; if not you can try to capture the prior uncertainty hierarchically, by, e.g., adding another layer to a model like (30) above.

- Calibration can be estimated by a form of cross-validation: with a given prior you can (a) repeatedly divide the data at random into modeling and validation subsets, (b) update to posterior predictive distributions based on the modeling data, and (c) compare these distributions with the actual values in the validation data.

I'll illustrate some examples of this idea later.

Note that calibration is inherently frequentist in spirit (e.g., "What percentage of the time do your 95% central predictive intervals include the actual value?"). This leads to a useful synthesis of Bayesian and frequentist thinking:

> Coherence keeps you internally honest; calibration keeps you in good contact with the world.

2.3.6 Conjugate analysis. Comparison with frequentist modeling

Example: Prior specification in the mortality data. Let's say (a) you know (from the literature) that the 30-day AMI mortality rate given average care and average sickness at admission in the U.S. is about 15%, (b) You know little about care or patient sickness at the DH, but (c) You'd be somewhat surprised (e.g., on Central Limit Theorem grounds) if the "underlying rate" at the DH was much less than 5% or more than 30% (note the asymmetry). To quantify these judgments you seek a flexible family of densities on (0,1), one of whose members has mean .15 and (say) 95% central interval (.05,.30).

A convenient family for this purpose is the beta distributions,

$$\text{Beta}(\theta|\alpha,\beta) = c\,\theta^{\alpha-1}(1-\theta)^{\beta-1}, \tag{2.37}$$

defined for $(\alpha > 0, \beta > 0)$ and for $0 < \theta < 1$.

`Maple` can be used to evaluate the normalizing constant c. I define the un-normalized density, and ask `Maple` to symbolically integrate it:

```
> assume( alpha > 0, beta > 0, theta > 0, theta < 1 );

> p1 := ( theta, alpha, beta ) -> theta^( alpha - 1 ) *
    ( 1 - theta )^( beta - 1 );

     p1 := (theta, alpha, beta) ->
            (alpha - 1)            (beta - 1)
       theta            (1 - theta)

> integrate( p1( theta, alpha, beta ), theta = 0 .. 1 );

                             Beta(alpha~, beta~)
```

Well, that's interesting; what's the Beta function?

```
> help( Beta );

Beta - The Beta function

Calling Sequence:
     Beta( x, y )

Parameters:
     x - an expression
     y - an expression

Description:

- The Beta function is defined as follows:

      Beta( x, y ) = ( GAMMA( x ) * GAMMA( y ) ) / GAMMA( x + y )
```

Thank you very much, `Maple`; what's the GAMMA function?

```
> help( GAMMA );

GAMMA - The Gamma and Incomplete Gamma Functions

lnGAMMA - The log-Gamma function

Calling Sequence:
     GAMMA( z )
     GAMMA( a, z )
     lnGAMMA( z )

Parameters:

     z - an expression
     a - an expression

Description:

- The Gamma function is defined for Re( z ) > 0 by

      GAMMA(z) =  int( exp( -t ) * t^( z - 1 ), t = 0 ..
        infinity )

  and is extended to the rest of the complex plane,
  less the non-positive integers, by analytic continuation.
  GAMMA has a simple pole at each of the points
  z = 0, -1, -2, ... .

- For positive real arguments z, the lnGAMMA function is
    defined by:

      lnGAMMA( z ) = ln( GAMMA( z ) )

> plotsetup( x11 );

> plot( GAMMA( x ), x = 0 .. 5, color = black );
```

Notice that $\Gamma(1) = 1, \Gamma(2) = 1, \Gamma(3) = 2, \Gamma(4) = 6$, and $\Gamma(5) = 24$—the pattern here is that

$$\Gamma(n) = (n-1)! \quad \text{for integer } n. \tag{2.38}$$

Thus the Gamma function is a kind of continuous generalization of the factorial function.

What all of this has shown is that the normalizing constant in the beta

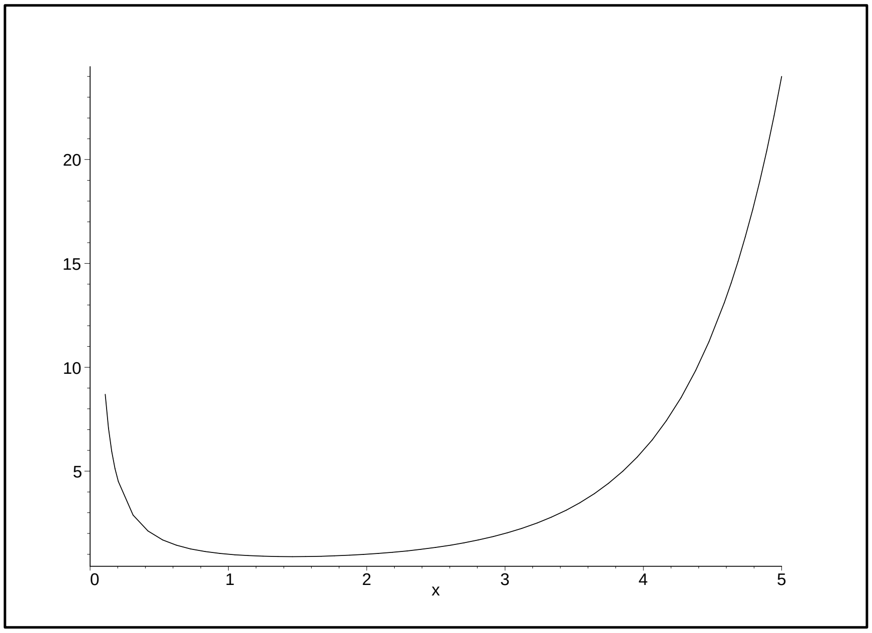


Figure 2.6: $\Gamma(x)$ *for* $x \in (0, 5)$.

distribution is

$$c = \left[\int_0^1 \theta^{\alpha - 1} \, (1 - \theta)^{\beta - 1} \, d\theta \right]^{-1} = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \, \Gamma(\beta)}, \tag{2.39}$$

so that the full definition of the beta distribution is

$$\text{Beta}(\theta | \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \, \Gamma(\beta)} \, \theta^{\alpha - 1} (1 - \theta)^{\beta - 1}, \tag{2.40}$$

for $(\alpha > 0, \beta > 0)$ and for $0 < \theta < 1$.

The beta family is convenient for two reasons:

(1) It exhibits a wide variety of distributional shapes (e.g., Johnson and Kotz, 1970):

```
> p := ( theta, alpha, beta ) -> ( GAMMA( alpha + beta ) /
    ( GAMMA( alpha ) * GAMMA( beta ) ) ) *
    theta^( alpha - 1 ) * ( 1 - theta )^( beta - 1 );

p := (theta, alpha, beta) ->

                           (alpha - 1)            (beta - 1)
  GAMMA(alpha + beta) theta            (1 - theta)
  ----------------------------------------------------------
                   GAMMA(alpha) GAMMA(beta)

> plot( { p( theta, 0.5, 0.5 ), p( theta, 1, 1 ),
    p( theta, 2, 3 ), p( theta, 30, 20 ) }, theta = 0 .. 1,
    color = black );
```

(2) As we saw above, the likelihood in this problem comes from the Bernoulli sampling distribution for the Y_i,

$$p(y_1, \ldots, y_n|\theta) = l(\theta|y) = \theta^s (1-\theta)^{n-s}, \tag{2.41}$$

where s is the sum of the y_i.

Now Bayes' Theorem says that to get the posterior distribution $p(\theta|y)$ you multiply the prior $p(\theta)$ and the likelihood—in this case $\theta^s (1-\theta)^{n-s}$—and renormalize so that the product integrates to 1.

Rev. Bayes himself noticed back in the 1740s that if you take the prior to be of the form $c\,\theta^u\,(1-\theta)^v$, the product of the prior and the likelihood will also be of this form, which makes the computations more straightforward.

The beta family is said to be conjugate to the Bernoulli/binomial likelihood.

Conjugacy of a family of prior distributions to a given likelihood is a bit hard to define precisely, but the basic idea—given a particular likelihood function—is to try to find a family of prior distributions so that the product of members of this family with the likelihood function will also be in the family.

Conjugate analysis—finding conjugate priors for standard likelihoods and restricting attention to them on tractability grounds—is one of only two fairly general methods for getting closed-form answers in the Bayesian approach (the other is asymptotic analysis; see Bernardo and Smith, 1994).

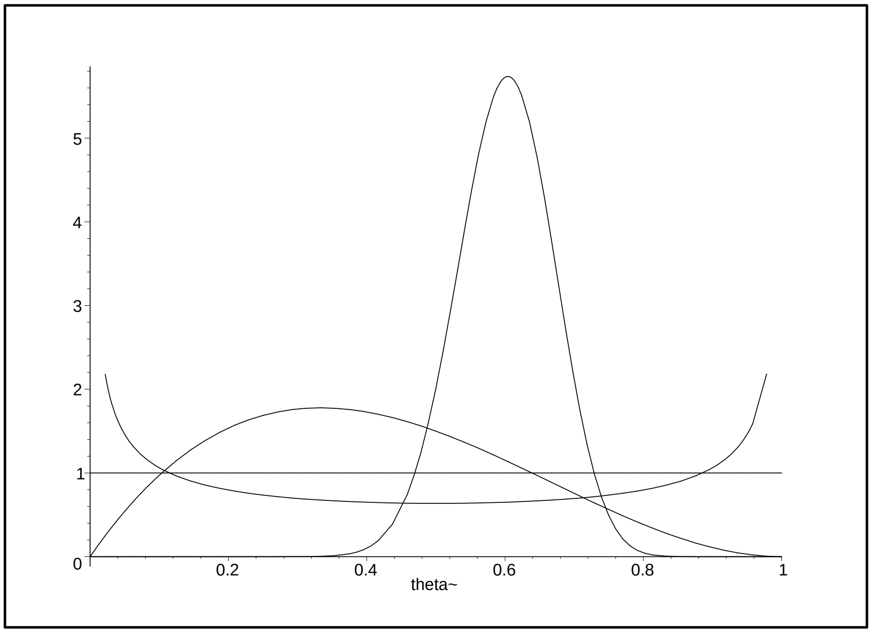


Figure 2.7: *The Beta*$(0.5, 0.5)$ *(U-shaped), Beta*$(1, 1)$ *(flat), Beta*$(2, 3)$ *(skewed right, with a mode at around 0.3), and Beta*$(30, 20)$ *(skewed left, with a mode at around 0.6) densities.*

Suppose we restrict attention (for now) to members of the beta family in trying to specify a prior distribution for θ in the AMI mortality example.

I want a member of this family which has mean 0.15 and 95% central interval (0.05, 0.30).

```
> mean := integrate( theta * p( theta, alpha, beta ),
    theta = 0 .. 1 );

                                        alpha~
                           mean := --------------
                                   alpha~ + beta~

> variance :=simplify( integrate( ( theta - alpha /
    ( alpha + beta ) )^2 * p( theta, alpha, beta ),
    theta = 0 .. 1 ) );
```

```
                                       alpha~ beta~
              variance := --------------------------------------
                                          2
                          (alpha~ + beta~)  (alpha~ + beta~ + 1)
```

As Maple has demonstrated,

$$\text{If } \theta \sim \text{Beta}(\alpha, \beta), \text{ then } E(\theta) = \frac{\alpha}{\alpha + \beta} \tag{2.42}$$
$$\text{and } V(\theta) = \frac{\alpha \beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}.$$

```
> solve( mean = 15 / 100, beta );

                                 17/3 alpha~

> solve( integrate( p( theta, alpha, 17 * alpha / 3 ),
    theta = 0.05 .. 0.30 ) = 0.95, alpha );

bytes used=3005456, alloc=1834672, time=0.82
bytes used=4006628, alloc=2293340, time=1.18
bytes used=5007408, alloc=2489912, time=1.58
```

Maple can't solve this equation symbolically (and neither could you), but it can do so numerically:

```
> fsolve( integrate( p( theta, alpha, 17 * alpha / 3 ),
    theta = 0.05 .. 0.30 ) = 0.95, alpha );

bytes used=7083468, alloc=2686484, time=2.50

  (output suppressed)

bytes used=27099104, alloc=3538296, time=11.99

                                  4.506062414

> 17 * 4.506062414 / 3;

                                 25.53435368
```

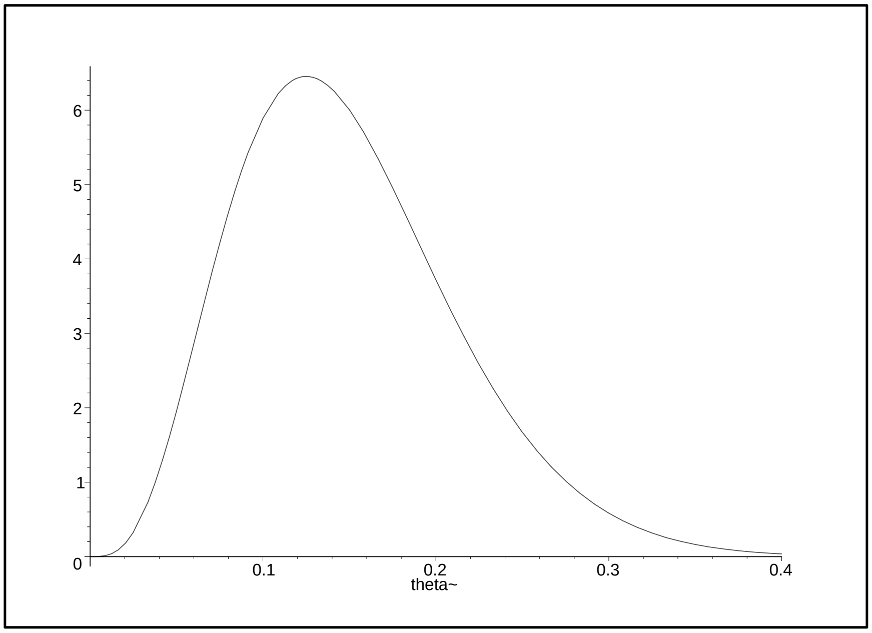


Figure 2.8: *The Beta* (4.5, 25.5) *density as a prior distribution in the hospital mortality case study.*

Thus the beta distribution with $(\alpha, \beta) = (4.5, 25.5)$ meets my two prior specifications.

```
  > plot( p( theta, 4.5, 25.5 ), theta = 0 .. 0.4 );
```

This prior distribution looks just like I want it to: it has a long right-hand tail and is quite spread out: the prior SD with this choice of (α, β) is $\sqrt{\frac{(4.5)(25.5)}{(4.5+25.5)^2(4.5+25.5+1)}} \doteq 0.064$, i.e., my prior says that I think the underlying AMI mortality rate at the DH is around 15%, give or take about 6 or 7%.

In the usual jargon α and β are called hyperparameters since they're parameters of the prior distribution.

Written hierarchically the model we've arrived at is

$$\begin{array}{rcll} (\alpha, \beta) & = & (4.5, 25.5) & \text{(hyperparameters)} \\ (\theta | \alpha, \beta) & \sim & \text{Beta}(\alpha, \beta) & \text{(prior)} \\ (Y_1, \dots, Y_n | \theta) & \stackrel{\text{IID}}{\sim} & \text{Bernoulli}(\theta) & \text{(likelihood)} \end{array} \qquad (2.43)$$

(43) suggests what to do if you're not sure about the specifications that led to $(\alpha, \beta) = (4.5, 25.5)$: hierarchically expand the model by placing a distribution on (α, β) centered at $(4.5, 25.5)$.

This is an important Bayesian modeling tool: if the model is inadequate in some way, expand it hierarchically in directions suggested by the nature of its inadequacy (I'll give more examples of this later).

Q: Doesn't this set up the possibility of an infinite regress, i.e., how do you know when to stop adding layers to the hierarchy?

A: (1) In practice people stop when they run out of (time, money), after having made sure that the final model passes diagnostic checks, and comfort may be taken from the empirical fact that (2) there tends to be a kind of diminishing returns principle: the farther a given layer in the hierarchy is from the likelihood (data) layer, the less it tends to affect the answer.

The conjugacy of the prior leads to a simple closed form for the posterior here: with y as the vector of observed $Y_i, i = 1, \ldots, n$ and s as the sum of the y_i (a sufficient statistic for θ with the Bernoulli likelihood),

$$\begin{aligned} p(\theta|y, \alpha, \beta) &= c\, l(\theta|y)\, p(\theta|\alpha, \beta) \\ &= c\, \theta^s\, (1-\theta)^{n-s}\, \theta^{\alpha-1} (1-\theta)^{\beta-1} \\ &= c\, \theta^{(s+\alpha)-1} (1-\theta)^{(n-s+\beta)-1}, \end{aligned} \tag{2.44}$$

i.e., the posterior for θ is $\text{Beta}(\alpha + s, \beta + n - s)$.

This gives the hyperparameters a nice interpretation in terms of effective information content of the prior: it's as if the data $(\text{Beta}(s+1, n-s+1))$ were worth $(s+1) + (n-s+1) \doteq n$ observations and the prior $(\text{Beta}(\alpha, \beta))$ were worth $(\alpha + \beta)$ observations.

This can be used to judge whether the prior is "too informative"—here it's equivalent to $(4.5 + 25.5) = 30$ binary observables with a mean of 0.15.

(44) can be summarized by saying

$$\left\{ \begin{array}{c} \theta \sim \text{Beta}(\alpha, \beta) \\ (Y_i|\theta) \overset{\text{IID}}{\sim} \text{Bernoulli}(\theta), \\ i = 1, \ldots, n \end{array} \right\} \rightarrow (\theta|y) \sim \text{Beta}(\alpha + s, \beta + n - s), \tag{2.45}$$

where $y = (y_1, \ldots, y_n)$ and $s = \sum_{i=1}^n y_i$.

Suppose the $n = 400$ observed mortality indicators consist of $s = 72$ 1s and $(n - s) = 328$ 0s.

Then the prior is Beta(4.5,25.5), the likelihood is Beta(73,329), the posterior for θ is Beta(76.5,353.5), and the three densities plotted on the same graph come out as follows:

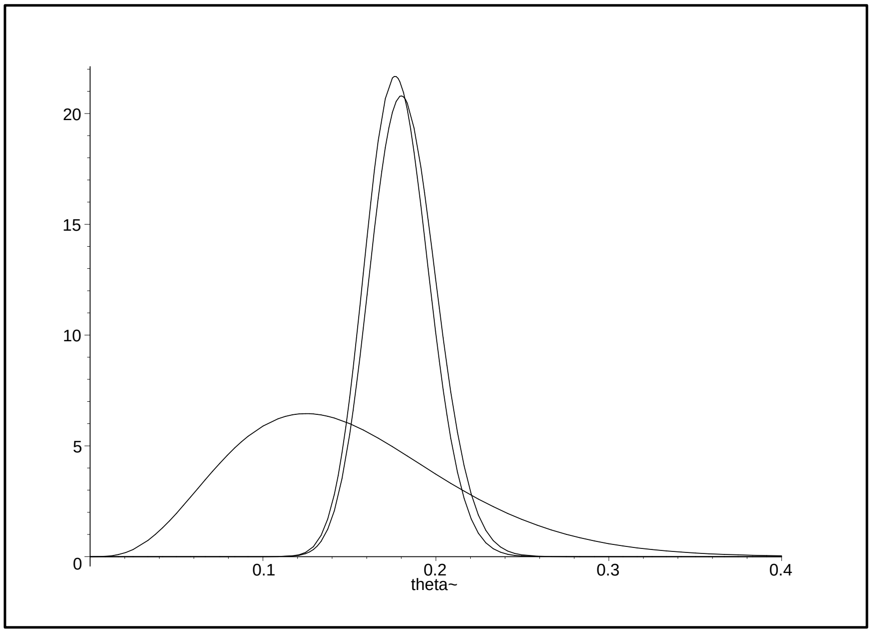


Figure 2.9: *Prior (Beta*$(4.5, 25.5)$*), likelihood (Beta*$(73.0, 329.0)$*), and posterior (Beta*$(76.5, 353.5)$*) densities.*

```
> plot( { p( theta, 4.5, 25.5 ), p( theta, 73.0, 329.0 ),
    p( theta, 76.5, 353.5 ) }, theta = 0 .. 0.4, color = black );
```

In this case the posterior and the likelihood nearly coincide, because the data information outweighs the prior information by $\frac{400}{30}$ = more than 13 to 1.

The mean of a Beta(α, β) distribution is $\frac{\alpha}{\alpha+\beta}$; with this in mind the posterior mean has a nice expression as a weighted average of the prior mean and data mean, with weights determined by the effective sample size of the prior, $(\alpha + \beta)$, and the data sample size n:

$$\begin{array}{ccc} \dfrac{\alpha+s}{\alpha+\beta+n} & = & \left(\dfrac{\alpha+\beta}{\alpha+\beta+n}\right)\left(\dfrac{\alpha}{\alpha+\beta}\right)+\left(\dfrac{n}{\alpha+\beta+n}\right)\left(\dfrac{s}{n}\right) \\ \begin{array}{c}\text{posterior}\\ \text{mean}\end{array} & = & \left(\begin{array}{c}\text{prior}\\ \text{weight}\end{array}\right)\left(\begin{array}{c}\text{prior}\\ \text{mean}\end{array}\right)+\left(\begin{array}{c}\text{data}\\ \text{weight}\end{array}\right)\left(\begin{array}{c}\text{data}\\ \text{mean}\end{array}\right) \end{array}$$

$$.178 \quad = \quad (.070) \quad (.15) \quad + \quad (.93) \quad (.18) \; .$$

Another way to put this is that the data mean, $\bar{y} = \frac{s}{n} = \frac{72}{400} = .18$, has been shrunk toward the prior mean .15 by (in this case) a modest amount: the posterior mean is about .178, and the shrinkage factor is $\frac{30}{30+400}$ = about .07.

Comparison with frequentist modeling. As we saw back on pp. 9–10, to analyze these data as a frequentist you would appeal to the Central Limit Theorem: $n = 400$ is big enough so that the sampling distribution of $\bar{Y}$ is approximately $\mathrm{N}\left[\theta, \frac{\theta(1-\theta)}{n}\right]$, so an approximate 95% confidence interval for θ would be centered at $\hat{\theta} = \bar{y} = 0.18$, with an estimated standard error of $\sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{n}} = 0.0192$, and would run roughly from 0.142 to 0.218.

By contrast the posterior for θ is also approximately Gaussian (see the graph on the next page), with a mean of 0.178 and an SD of

$$\sqrt{\frac{\alpha^* \beta^*}{(\alpha^* + \beta^*)^2 (\alpha^* + \beta^* + 1)}} = 0.0184,$$

where α^* and β^* are the parameters of the beta posterior distribution; a 95% central posterior interval for θ would then run from about $0.178 - (1.96)(0.0184) = 0.142$ to $0.178 + (1.96)(0.0184) = 0.215$.

```
> g := ( theta, mu, sigma ) -> exp( - ( theta - mu )^2 /
    ( 2 * sigma^2 ) ) / ( sigma * sqrt( 2 * Pi ) );
                                                               2
                                                   (theta - mu)
                                         exp(- 1/2 -------------)
                                                           2
                                                      sigma
              g := (theta, mu, sigma) -> ------------------------
                                             sigma sqrt(2 Pi)

> plot( { p( theta, 76.5, 353.5 ), g( theta, 0.178, 0.0184 ) },
    theta = 0.10 .. 0.26, color = black );
```

The Bayesian analysis here is equivalent to one in which a dataset consisting of $(0.15)(30) = 4.5$ 1s and $(1 - 0.15)(30) = 25.5$ 0s is appended to the observed data, and a frequentist analysis is carried out on this merged dataset.

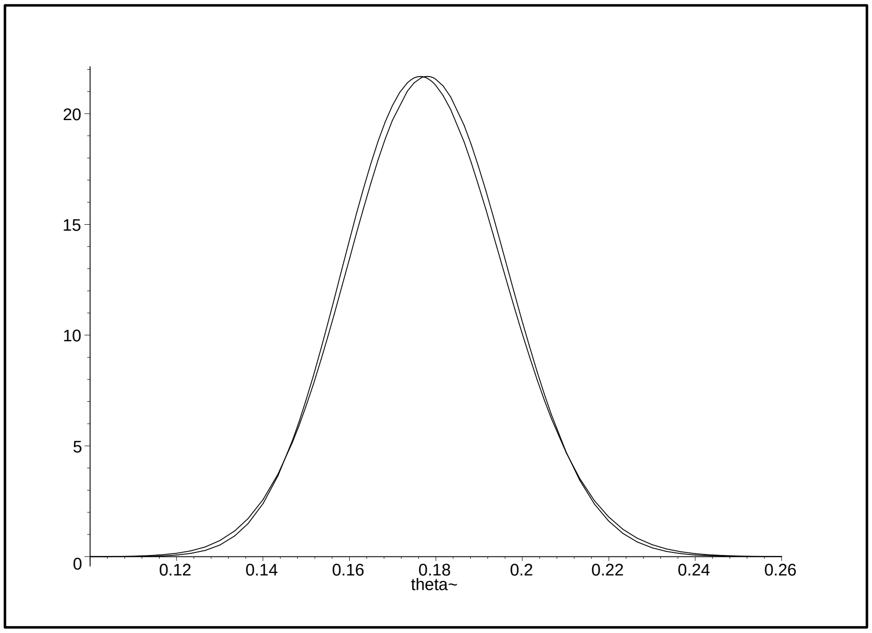


Figure 2.10: *The (Beta*$(76.5, 353.5)$*) posterior and the Gaussian distribution, with the same mean and variance, as an approximation to it.*

The two approaches (frequentist based only on the sample, Bayesian based on the sample and the prior I'm using) give almost the same answers in this case, a result that's typical of situations with fairly large n and relatively diffuse prior information.

Note, however, that the interpretation of the two analyses differs somewhat:

• In the frequentist approach θ is fixed but unknown and $\bar{Y}$ is random, with the analysis based on imagining what would happen if the hypothetical random sampling were repeated, and appealing to the fact that across these repetitions $(\bar{Y} - \theta) \stackrel{\cdot}{\sim} \text{N}(0, .019^2)$; whereas

• In the Bayesian approach $\bar{y}$ is fixed at its observed value and θ is treated as random, as a means of quantifying your posterior uncertainty about it: $(\theta - \bar{y} | \bar{y}) \stackrel{\cdot}{\sim} \text{N}(0, .018^2)$.

This means among other things that, while it's not legitimate with the frequentist approach to say that $P_f(.14 \leq \theta \leq .22) \doteq .95$, which is what many users of confidence intervals would like them to mean, the corresponding statement $P_B(.14 \leq \theta \leq .22 | y, \text{diffuse prior information}) \doteq .95$ is a natural consequence of the Bayesian approach.

In the case of diffuse prior information this justifies the fairly common practice of computing inferential summaries in a frequentist way and then interpreting them Bayesianly.

When nondiffuse prior information is available and you use it, your answer will differ from a frequentist analysis based on the same likelihood.

If your prior is retrospectively seen to have been well-calibrated you will get a better answer than with the frequentist approach; if poorly calibrated, a worse answer (Samaniego and Reneau, 1994):

“bad” Bayesian $\leq$ frequentist $\leq$ “good” Bayesian

What you make of this depends on your risk-aversion: Is it better to try to land on the right in this box, running some risk of landing on the left, or to steer a middle course?

(NB I'll give several examples later in which a Bayesian analysis is better even with diffuse prior information.)

Bernoulli Prediction. The predictive distribution for future Y_i in the Bernoulli model was shown back on p. 33 (equation (35)) to be

$$\begin{aligned} p(Y_{m+1} = y_{m+1}, \ldots, Y_n = y_n | y_1, \ldots, y_m) = \\ = \int_0^1 \prod_{i=m+1}^{n} \theta^{y_i} (1-\theta)^{1-y_i} \, p(\theta | y_1, \ldots, y_m) \, d\theta \end{aligned} \tag{2.46}$$

We've seen that if the prior is taken to be Beta(α, β) the posterior $p(\theta | y_1, \ldots, y_m)$ in this expression is Beta(α^*, β^*), where $\alpha^* = \alpha + s$ and $\beta^* = \beta + (n - s)$.

As an example of an explicit calculation with (46) in this case, suppose that we've observed n of the Y_i, obtaining data vector $y = (y_1, \ldots, y_n)$, and we want to predict Y_{n+1}.

Obviously $p(Y_{n+1} = y_{n+1} | y)$ has to be a Bernoulli(θ^*) distribution for some θ^*, and intuition says that θ^* should just be the mean $\frac{\alpha^*}{\alpha^* + \beta^*}$ of the posterior distribution for θ given y.

(46) and an application of (39) in this case give for $p(Y_{n+1} = y_{n+1} | y)$ the expressions

$$\int_0^1 \theta^{y_{n+1}} (1-\theta)^{1-y_{n+1}} \frac{\Gamma(\alpha^* + \beta^*)}{\Gamma(\alpha^*) \, \Gamma(\beta^*)} \theta^{\alpha^* - 1} (1-\theta)^{\beta^* - 1} \, d\theta \tag{2.47}$$

$$= \frac{\Gamma(\alpha^* + \beta^*)}{\Gamma(\alpha^*)\,\Gamma(\beta^*)} \int_0^1 \theta^{\alpha^* + y_{n+1} - 1} (1-\theta)^{(\beta^* - y_{n+1} + 1) - 1} \, d\theta$$

$$= \left[\frac{\Gamma(\alpha^* + y_{n+1})}{\Gamma(\alpha^*)} \right] \left[\frac{\Gamma(\beta^* - y_{n+1} + 1)}{\Gamma(\beta^*)} \right] \left[\frac{\Gamma(\alpha^* + \beta^*)}{\Gamma(\alpha^* + \beta^* + 1)} \right]$$

Now it's a fact about the Gamma function, which you can verify with `Maple`, that for any real number x, $\frac{\Gamma(x+1)}{\Gamma(x)} = x$:

```
> assume( x, real );

> simplify( GAMMA( x + 1 ) / GAMMA( x ) );

                                      x~
```

So (47), for example in the case $y_{n+1} = 1$, becomes

$$\begin{aligned} p(Y_{n+1} = 1 | y) &= \left[\frac{\Gamma(\alpha^* + 1)}{\Gamma(\alpha^*)} \right] \left[\frac{\Gamma(\alpha^* + \beta^*)}{\Gamma(\alpha^* + \beta^* + 1)} \right] \\ &= \frac{\alpha^*}{\alpha^* + \beta^*}, \qquad (2.48) \end{aligned}$$

confirming intuition.

For example, with $(\alpha, \beta) = (4.5, 25.5)$ and $n = 400$ with $s = 72$, we saw earlier that the posterior for θ was Beta$(76.5, 353.5)$, and this posterior distribution has mean $\frac{\alpha^*}{\alpha^* + \beta^*} = 0.178$.

In this situation you would expect the next AMI patient who comes along to die within 30 days of admission with probability 0.178, so the predictive distribution above makes good sense.

The Binomial Distribution. We've seen that a sufficient statistic for θ with a Bernoulli likelihood is the sum $s = \sum_{i=1}^n y_i$ of the 1s and 0s.

This means that if you buy into the model $(Y_i | \theta) \overset{\text{IID}}{\sim}$ Bernoulli(θ) you don't care whether you observe the entire data vector $Y = (Y_1, \ldots, Y_n)$ or its sum $S = \sum_{i=1}^n Y_i$.

The distribution of S in repeated sampling has a familiar form: it's just the binomial distribution Binomial(n, θ), which counts the number of successes in a series of IID success/failure trials.

Recall that if $S \sim$ Binomial(n, θ) then S has discrete density

$$p(S = s | \theta) = \left\{ \begin{array}{cc} \left(\begin{smallmatrix} n \\ s \end{smallmatrix} \right) \theta^s (1-\theta)^{n-s} & \text{if } s = 0, \ldots, n \\ 0 & \text{otherwise} \end{array} \right\}.$$

Thus we've learned another conjugate updating rule in simple Bayesian modeling, more or less for free: if the data set just consists of a single draw S from a binomial distribution, then the conjugate prior for the success probability θ is Beta(α, β), and the updating rule, which follows directly from (45), is

$$\left\{ \begin{array}{c} \theta \sim \text{Beta}(\alpha, \beta) \\ (S|\theta) \sim \text{Binomial}(n, \theta) \end{array} \right\} \rightarrow (\theta|s) \sim \text{Beta}(\alpha + s, \beta + n - s). \qquad (2.49)$$

Two important general points.

1 (the sequential nature of Bayesian learning) Suppose you and I are observing data $(y_1, \ldots, y_n)$ to learn about a parameter θ, and we have no reason throughout this observation process to change (the sampling distribution/likelihood part of) our model.

We both start with the same prior $p_1(\theta)$ before any of the data arrive, but we adopt what appear to be different analytic strategies:

- You wait until the whole data set $(y_1, \ldots, y_n)$ has been observed and update $p_1(\theta)$ directly to the posterior distribution $p(\theta|y_1, \ldots, y_n)$, whereas
- I stop after seeing $(y_1, \ldots, y_m)$ for some $m < n$, update $p_1(\theta)$ to an intermediate posterior distribution $p(\theta|y_1, \ldots, y_m)$, and then I want to go on from there, observing $(y_{m+1}, \ldots, y_n)$ and finally updating to a posterior on θ that takes account of the whole data set $(y_1, \ldots, y_n)$.

$\mathbf{Q}_1$ What should I use for my intermediate prior distribution $p_2(\theta)$?

$\mathbf{A}_1$ Naturally enough, the right thing to do is to set $p_2(\theta)$ to the current posterior $p(\theta|y_1, \ldots, y_m)$.

The informal way people refer to this is to say that yesterday's posterior distribution is today's prior distribution.

$\mathbf{Q}_2$ If I use the posterior in A_1, do you and I get the same answer for $p(\theta|y_1, \ldots, y_n)$ in the end?

$\mathbf{A}_1$ Yes (you can check this).

2 (the generality of conjugate analysis) Having seen conjugate priors used with binary outcomes, you can see that conjugate analysis has a variety of advantages:

- It's mathematically straightforward;
- The posterior mean turns out to be a weighted average of the prior and data means; and
- You get the nice interpretation of the prior as an information source that's equivalent to a data set, and it's easy to figure out the prior sample

size.

It's natural to wonder, though, what's lost in addition to what's gained by adopting a conjugate prior.

The main disadvantage of conjugate priors is that in their simplest form they're not flexible enough to express all possible forms of prior information.

For example, in the AMI mortality case study, what if you wanted to combine a bimodal prior distribution with the Bernoulli likelihood?

This isn't possible when using a single member of the Beta(α, β) family.

However, it's possible to prove the following:

Theorem (Diaconis and Ylvisaker 1985). Given a particular likelihood that's a member of the exponential family (this will be covered in Section 2.7 below), any prior distribution can be expressed as a mixture of priors that are conjugate to that likelihood.

For example, in the AMI case study the model could be

$$\begin{array}{rcl} J & \sim & p(J) \\ (\theta \vert J) & \sim & \text{Beta}(\alpha_J, \beta_J) \\ (Y_i \vert \theta) & \stackrel{\text{IID}}{\sim} & \text{B}(\theta), \ i = 1, \ldots, n, \end{array} \tag{2.50}$$

for some distribution $p(J)$ on the positive integers—this is completely general but loses some of the advantages of simple conjugate analysis (e.g., closed-form computations are no longer possible).

The exponential family. In our first (and only, so far) example of conjugate analysis, with the Bernoulli/binomial likelihood (41), we worked out the form of the conjugate prior just by looking at the likelihood function.

This works in simple problems, but it would be nice to have a general way of figuring out what the conjugate prior has to be (if it exists) once the likelihood is specified.

It was noticed a long time ago that many of the standard sampling distributions that you're likely to want to use in constructing likelihood functions in parametric Bayesian modeling have the same general form, which is referred to as the exponential family.

I bring this up here because there's a simple theorem which specifies the conjugate prior for likelihoods that belong to the exponential family.

With the Bernoulli likelihood (41) in the hospital mortality case study, the unknown quantity θ in the likelihood function was a scalar (1–dimensional; univariate), but this will not always be true: more generally and more usually θ is a vector of length (say) k.

We'll begin to look at problems with multivariate θ ($k > 1$) in Section 2.9, but—for continuity with the later material—here I'm going to give the definition of the exponential family for vector θ.

Definition (e.g., Bernardo and Smith, 1994): Given data y_1 (a sample of size 1) and a parameter vector $\theta = (\theta_1, \ldots, \theta_k)$, the (marginal) sampling distribution $p(y_1|\theta)$ belongs to the k-dimensional exponential family if it can be expressed in the form

$$p(y_1|\theta) = c\, f_1(y_1)\, g_1(\theta)\, \exp\left[\sum_{j=1}^k \phi_j(\theta)\, h_j(y_1)\right] \qquad (2.51)$$

for $y_1 \in \mathcal{Y}$ and 0 otherwise; if $\mathcal{Y}$ doesn't depend on θ the family is called regular.

The vector $[\phi_1(\theta), \ldots, \phi_k(\theta)]$ in (51) is called the natural parameterization of the exponential family.

In this case the joint distribution $p(y|\theta)$ of a sample $y = (y_1, \ldots, y_n)$ of size n which is conditionally IID from (51) (which also defines, as usual, the likelihood function $l(\theta|y)$) will be

$$\begin{aligned} p(y|\theta) &= l(\theta|y) = \prod_{i=1}^n p(y_i|\theta) && (52) \\ &= c\left[\prod_{i=1}^n f_1(y_i)\right] [g_1(\theta)]^n \exp\left[\sum_{j=1}^k \phi_j(\theta) \sum_{i=1}^n h_j(y_i)\right]. \end{aligned}$$

This leads to another way to define the exponential family: in (52) take $f(y) = \prod_{i=1}^n f_1(y_i)$ and $g(\theta) = [g_1(\theta)]^n$ to yield

Definition: Given data $y = (y_1, \ldots, y_n)$ (a conditionally IID sample of size n) and a parameter vector $\theta = (\theta_1, \ldots, \theta_k)$, the (joint) sampling distribution $p(y|\theta)$ belongs to the k-dimensional exponential family if it can be expressed in the form

$$p(y|\theta) = c\, f(y)\, g(\theta)\, \exp\left[\sum_{j=1}^k \phi_j(\theta) \sum_{i=1}^n h_j(y_i)\right]. \qquad (2.53)$$

Either way you can see that $\{\sum_{i=1}^n h_1(y_i), \ldots, \sum_{i=1}^n h_k(y_i)\}$ is a set of sufficient statistics for θ under this sampling model, because the likelihood $l(\theta|y)$ depends on y only through the values of $\{h_1, \ldots, h_k\}$.

Now here's the theorem about the conjugate prior: if the likelihood $l(\theta|y)$ is of the form (53), then in searching for a conjugate prior $p(\theta)$—that is, a prior of the same functional form as the likelihood—you can see directly what will work:

$$p(\theta) = c\, g(\theta)^{\tau_0} \, \exp\left[\sum_{j=1}^{k} \phi_j(\theta)\, \tau_j \right], \tag{2.54}$$

for some $\tau = (\tau_0, \ldots, \tau_k)$.

With this choice the posterior for θ will be

$$p(\theta|y) = c\, g(\theta)^{1+\tau_0} \, \exp\left\{ \sum_{j=1}^{k} \phi_j(\theta) \left[\tau_j + \sum_{i=1}^{n} h_j(y) \right] \right\}, \tag{2.55}$$

which is indeed of the same form (in θ) as (53).

As a first example, with $s = \sum_{i=1}^{n} y_i$, the Bernoulli/binomial likelihood in (41) can be written

$$\begin{aligned} l(\theta|y) &= c\, \theta^s (1-\theta)^{n-s} \\ &= c\,(1-\theta)^n \left(\frac{\theta}{1-\theta} \right)^s \\ &= c\,(1-\theta)^n \exp\left[s \, \log\left(\frac{\theta}{1-\theta} \right) \right], \end{aligned} \tag{2.56}$$

which shows (a) that this sampling distribution is a member of the exponential family with $k = 1,\ g(\theta) = (1-\theta)^n,\ \phi_1(\theta) = \log\left(\frac{\theta}{1-\theta}\right)$ (**NB** the natural parameterization, and the basis of logistic regression), and $h_1(y_i) = y_i$, and (b) that $\sum_{i=1}^{n} h_1(y_i) = s$ is sufficient for θ.

Then (54) says that the conjugate prior for the Bernoulli/binomial likelihood is

$$\begin{aligned} p(\theta) &= c\,(1-\theta)^{n\tau_0} \exp\left[\tau_1 \log\left(\frac{\theta}{1-\theta} \right) \right] \\ &= c\, \theta^{\alpha-1} (1-\theta)^{\beta-1} = \text{Beta}(\alpha, \beta) \end{aligned} \tag{2.57}$$

for some α and β, as we've already seen is true.

2.4 Integer-valued outcomes

Case Study: *Hospital length of stay for birth of premature babies.* As a small part of a study I worked on at the Rand Corporation in the late 1980s, we obtained data on a random sample of $n = 14$ women who came to a hospital in Santa Monica, CA, in 1988 to give birth to premature babies.

One (integer-valued) outcome of interest was y = length of hospital stay (LOS).

Here's a preliminary look at the data in an excellent freeware statistical package called `R` (see `http://www.r-project.org/` for more details and instructions on how to download the package).

```
greco 2740> R

R : Copyright 2005, The R Foundation for Statistical Computing
Version 2.1.0 Patched (2005-05-12), ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> y

 [1] 1 2 1 1 1 2 2 4 3 6 2 1 3 0

> sort( y )

 [1] 0 1 1 1 1 1 2 2 2 2 3 3 4 6

> table( y )

0 1 2 3 4 6
1 5 4 2 1 1
```

```
 > stem( y, scale = 2 )

  The decimal point is at the |

  0 | 0
  1 | 00000
  2 | 0000
  3 | 00
  4 | 0
  5 |
  6 | 0

 > mean( y )

 [1] 2.071429

 > sd( y )

 [1] 1.54244

 > q( )

 Save workspace image? [y/n/c]: y
 rosalind 1777>
```

One possible model for non-negative integer-valued outcomes is the Poisson distribution

$$P(Y_i = y_i|\lambda) = \left\{ \begin{array}{cc} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} & \text{for } y_i = 0, 1, \ldots \\ 0 & \text{otherwise} \end{array} \right\}, \qquad (2.58)$$

for some $\lambda > 0$.

As usual `Maple` can be used to work out the mean and variance of this distribution:

```
> assume( lambda > 0 );

> p := ( y, lambda ) -> lambda^y * exp( - lambda ) / y!;
```

```
                                             y
                                       lambda  exp(-lambda)
                   p := (y, lambda) -> --------------------
                                                y!

> simplify( sum( p( y, lambda ), y = 0 .. infinity ) );

                                       1

> simplify( sum( y * p( y, lambda ), y = 0 .. infinity ) );

                                    lambda~

> simplify( sum( ( y - lambda )^2 * p( y, lambda ),
    y = 0 .. infinity ) );

                                    lambda~
```

Thus if $Y \sim \text{Poisson}(\lambda), E(Y) = V(Y) = \lambda$, which people sometimes express by saying that the variance-to-mean ratio (VTMR) for the Poisson is 1.

`R` can be used to check informally whether the Poisson is a good fit to the LOS data:

```
> dpois( 0:7, mean( y ) )
[1] 0.126005645 0.261011693 0.270333539 0.186658872 0.096662630
[6] 0.040045947 0.013825386 0.004091186

> print( n <- length( y ) )

[1] 14

> table( y ) / n
         0          1          2          3          4          6
0.07142857 0.35714286 0.28571429 0.14285714 0.07142857 0.07142857

> cbind( c( dpois( 0:6, mean( y ) ),
    1 - sum( dpois( 0:6, mean( y ) ) ) ),
    apply( outer( y, 0:7, ’==’ ), 2, sum ) / n )
```

```
            [,1]       [,2]
[1,] 0.126005645 0.07142857
[2,] 0.261011693 0.35714286
[3,] 0.270333539 0.28571429
[4,] 0.186658872 0.14285714
[5,] 0.096662630 0.07142857
[6,] 0.040045947 0.00000000
[7,] 0.013825386 0.07142857
[8,] 0.005456286 0.00000000
```

The second column in the above table records the values of the Poisson probabilities for $\lambda = 2.07$, the mean of the y_i, and the third column is the empirical relative frequencies; informally the fit is reasonably good.

Another informal check comes from the fact that the sample mean and variance are 2.07 and $1.542^2 \doteq 2.38$, which are reasonably close.

Exchangeability. As with the AMI mortality case study, before the data arrive I recognize that my uncertainty about the Y_i is exchangeable, and you would expect from a generalization of the binary-outcomes version of de Finetti's Theorem that the structure of a plausible Bayesian model for the data would then be

$$\begin{array}{rcll} \theta & \sim & p(\theta) & \text{(prior)} \\ (Y_i|\theta) & \stackrel{\text{IID}}{\sim} & F(\theta) & \text{(likelihood)}, \end{array} \tag{2.59}$$

where θ is some parameter (vector) and $F(\theta)$ is some parametric family of distributions on the non-negative integers indexed by θ.

Thus, in view of the preliminary examination of the data above, a plausible Bayesian model for these data is

$$\begin{array}{rcll} \lambda & \sim & p(\lambda) & \text{(prior)} \\ (Y_i|\lambda) & \stackrel{\text{IID}}{\sim} & \text{Poisson}(\lambda) & \text{(likelihood)}, \end{array} \tag{2.60}$$

where λ is a positive real number.

NB (1) This approach to model-building involves a form of cheating, because we've used the data twice: once to choose the model, and again to draw conclusions conditional on the chosen model.

The result in general can be a failure to assess and propagate model uncertainty (Draper 1995).

(2) Frequentist modeling often employs this same kind of cheating in specifying the likelihood function.

(3) There are two Bayesian ways out of this dilemma: cross-validation and Bayesian non-parametric/semi-parametric methods.

The latter is beyond the scope of this course; I'll give examples of the former later.

To get more practice with Bayesian calculations I'm going to ignore the model uncertainty problem for now and pretend that somehow we knew that the Poisson was a good choice.

The likelihood function in model (2.60) is

$$\begin{aligned} l(\lambda|y) &= c \, p_{Y_1,\ldots,Y_n}(y_1,\ldots,y_n|\lambda) \\ &= c \prod_{i=1}^{n} p_{Y_i}(y_i|\lambda) \\ &= c \prod_{i=1}^{n} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} \\ &= c \, \lambda^s \, e^{-n\lambda}, \end{aligned} \tag{2.61}$$

where $y = (y_1,\ldots,y_n)$ and $s = \sum_{i=1}^{n} y_i$; here $\left(\prod_{i=1}^{n} y_i!\right)^{-1}$ can be absorbed into the generic positive c because it doesn't involve λ.

Thus (as was true in the Bernoulli model) $s = \sum_{i=1}^{n} y_i$ is sufficient for λ in the Poisson model, and we can write $l(\lambda|s)$ instead of $l(\lambda|y)$ if we want.

If a conjugate prior $p(\lambda)$ for λ exists it must be such that the product $p(\lambda)\, l(\lambda|s)$ has the same mathematical form as $p(\lambda)$.

Examination of (61) reveals that the same trick works here as with Bernoulli data, namely taking the prior to be of the same form as the likelihood:

$$p(\lambda) = c \, \lambda^{\alpha-1} e^{-\beta\lambda} \tag{2.62}$$

for some $\alpha > 0, \beta > 0$—this is the Gamma distribution $\lambda \sim \Gamma(\alpha, \beta)$ for $\lambda > 0$ (see Gelman et al., 2003, Appendix A).

As usual `Maple` can work out the normalizing constant:

```
> assume( lambda > 0, alpha > 0, beta > 0 );

> p1 := ( lambda, alpha, beta ) -> lambda^( alpha - 1 ) *
    exp( - beta * lambda );
```

```
      p1 := (lambda, alpha, beta) ->
               (alpha - 1)
         lambda            exp(-beta lambda)

> simplify( integrate( p1( lambda, alpha, beta ),
    lambda = 0 .. infinity ) );

                              (-alpha~)
                         beta~          GAMMA(alpha~)
```

Thus $c^{-1} = \beta^{-\alpha}\,\Gamma(\alpha)$ and the proper definition of the Gamma distribution is

$$\text{If } \lambda \sim \Gamma(\alpha, \beta) \text{ then } p(\lambda) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\,\lambda^{\alpha-1}\,e^{-\beta\,\lambda} \tag{2.63}$$

for $\alpha > 0, \beta > 0$.

As usual `Maple` can also be used to explore the behavior of this family of distributions as a function of its inputs α and β:

```
> p := ( lambda, alpha, beta ) -> beta^alpha *
    lambda^( alpha - 1 ) * exp( - beta * lambda ) /
    GAMMA( alpha );

p := (lambda, alpha, beta) ->

      alpha       (alpha - 1)
  beta      lambda            exp(-beta lambda)
  ---------------------------------------------
                  GAMMA(alpha)

> plotsetup( x11 );

> plot( { p( lambda, 1, 1 ), p( lambda, 2, 1 ),
    p( lambda, 3, 1 ), p( lambda, 6, 1 ) },
    lambda = 0 .. 14, color = black );
```

α evidently controls the shape of the Gamma family.

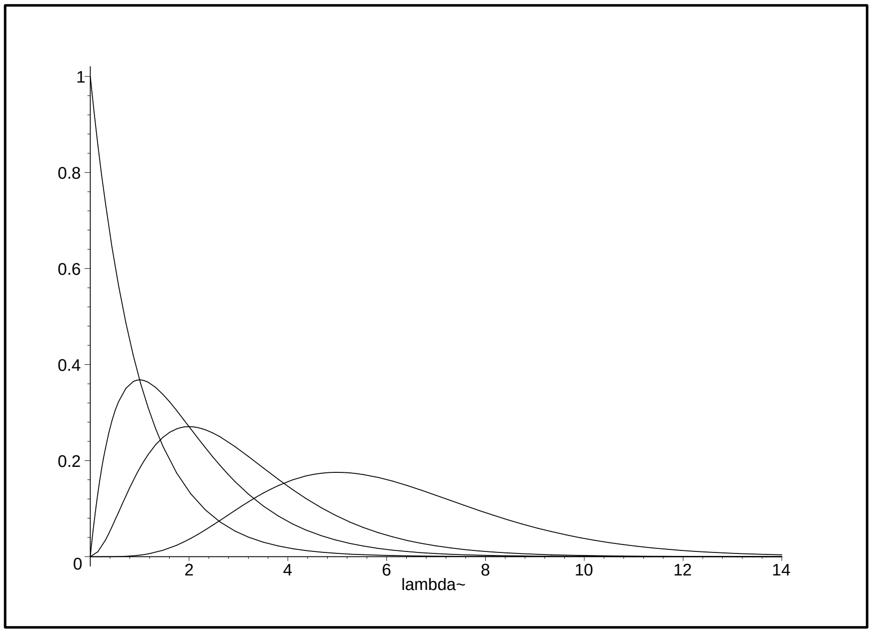


Figure 2.11: *Gamma*(α, β) *densities with* $\beta = 1$ *and* $\alpha = \{1, 2, 3, 6\}$*; the distributions change shape, becoming more Gaussian, as* α *increases.*

When $\alpha = 1$ the Gamma distributions have a special form which you'll probably recognize—they're the exponential distributions $\mathcal{E}(\beta)$: for $\beta > 0$

$$\text{If } \lambda \sim \mathcal{E}(\beta) \text{ then } p(\lambda) = \left\{ \begin{array}{cc} \beta \, e^{-\beta \, \lambda} & \text{for } \lambda > 0 \\ 0 & \text{otherwise} \end{array} \right\}. \tag{2.64}$$

```
> plot( { p( lambda, 2, 1 ), p( lambda, 2, 2 ),
    p( lambda, 2, 3 ) }, lambda = 0 .. 7, color = black );
```

In the Gamma family the parameter β controls the spread or scale of the distribution.

Definition Given a random quantity y whose density $p(y|\sigma)$ depends on a parameter $\sigma > 0$, if it's possible to express $p(y|\sigma)$ in the form $\frac{1}{\sigma} \, f(\frac{y}{\sigma})$, where $f(\cdot)$ is a function which does not depend on y or σ, then σ is called a scale parameter for the parametric family p.

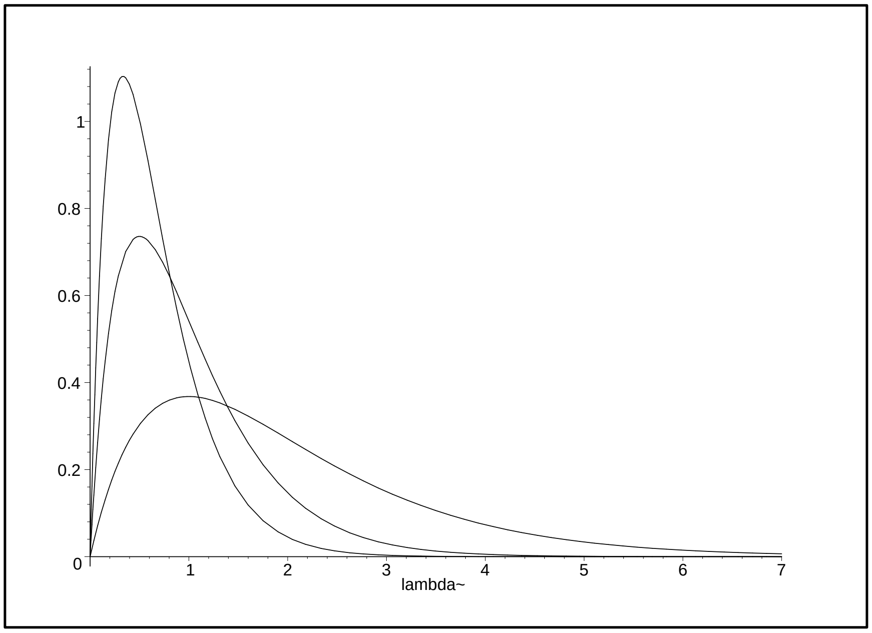


Figure 2.12: *Gamma*(α, β) *densities with* $\alpha = 2$ *and* $\beta = \{1, 2, 3\}$*; as* β *goes up the spread of the distributions decreases.*

Letting $f(t) = e^{-t}$ and taking $\sigma = \frac{1}{\beta}$, you can see that the Gamma family can be expressed in this way, so $\frac{1}{\beta}$ is a scale parameter for the Gamma distribution.

As usual `Maple` can also work out the mean and variance of this family:

```
> simplify( integrate( p( lambda, alpha, beta ),
    lambda = 0 .. infinity ) );

                                     1

> simplify( integrate( lambda * p( lambda, alpha, beta ),
    lambda = 0 .. infinity ) );

                                  alpha~
                                  ------
                                  beta~
```

```
> simplify( integrate( ( lambda - alpha / beta )^2 *
    p( lambda, alpha, beta ), lambda = 0 .. infinity ) );

                                    alpha~
                                    ------
                                         2
                                    beta~
```

Thus if $\lambda \sim \Gamma(\alpha, \beta)$ then $E(\lambda) = \frac{\alpha}{\beta}$ and $V(\lambda) = \frac{\alpha}{\beta^2}$.

Conjugate updating is now straightforward: with $y = (y_1, \ldots, y_n)$ and $s = \sum_{i=1}^n y_i$, by Bayes' Theorem

$$\begin{aligned} p(\lambda|y) &= c\,p(\lambda)\,l(\lambda|y) \\ &= c\,\left(c\,\lambda^{\alpha-1}\,e^{-\beta\lambda}\right)\left(c\,\lambda^s\,e^{-n\lambda}\right) \\ &= c\,\lambda^{(\alpha+s)-1}\,e^{-(\beta+n)\lambda}, \end{aligned} \tag{2.65}$$

and the resulting distribution is just $\Gamma(\alpha + s, \beta + n)$.

Conjugate Poisson analysis. This can be summarized as follows:

$$\left\{ \begin{array}{c} (\lambda|\alpha, \beta) \sim \Gamma(\alpha, \beta) \\ (Y_i|\lambda) \overset{\text{IID}}{\sim} \text{Poisson}(\lambda), \\ i = 1, \ldots, n \end{array} \right\} \rightarrow (\lambda|s) \sim \Gamma(\alpha^*, \beta^*), \tag{2.66}$$

where $(\alpha^*, \beta^*) = (\alpha + s, \beta + n)$ and $s = \sum_{i=1}^n y_i$ is a sufficient statistic for λ in this model.

The posterior mean of λ here is evidently $\frac{\alpha^*}{\beta^*} = \frac{\alpha+s}{\beta+n}$, and the prior and data means are $\frac{\alpha}{\beta}$ and $\bar{y} = \frac{s}{n}$, so (as was the case in the Bernoulli model) the posterior mean can be written as a weighted average of the prior and data means:

$$\frac{\alpha + s}{\beta + n} = \left(\frac{\beta}{\beta + n}\right)\left(\frac{\alpha}{\beta}\right) + \left(\frac{n}{\beta + n}\right)\left(\frac{s}{n}\right). \tag{2.67}$$

Thus the prior sample size n_0 in this model is just β (which makes sense given that $\frac{1}{\beta}$ is the scale parameter for the Gamma distribution), and the prior acts like a dataset consisting of β observations with mean $\frac{\alpha}{\beta}$.

LOS data analysis. Suppose that, before the current data set is scheduled to arrive, I know little about the mean length of hospital stay of women giving birth to premature babies.

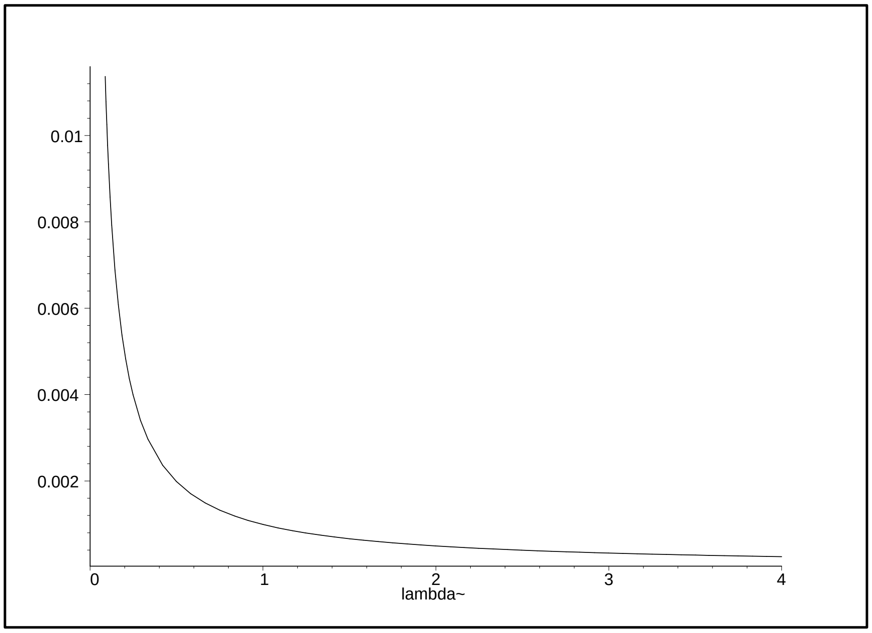


Figure 2.13: *The Gamma*(ϵ, ϵ) *prior with* $\epsilon = 0.001$.

Then for my prior on λ I'd like to specify a member of the $\Gamma(\alpha, \beta)$ family which is relatively flat in the region in which the likelihood function is appreciable.

The $\Gamma(\epsilon, \epsilon)$ **prior.** A convenient and fairly all-purpose default choice of this type is $\Gamma(\epsilon, \epsilon)$ for some small ϵ like 0.001.

When used as a prior this distribution has prior sample size ϵ; it also has mean 1, but that usually doesn't matter when ϵ is tiny.

```
> plot( p( lambda, 0.001, 0.001 ), lambda = 0 .. 4,
    color = black );
```

With the LOS data $s = 29$ and $n = 14$, so the likelihood for λ is like a $\Gamma(30, 14)$ density, which has mean $\frac{30}{14} \doteq 2.14$ and SD $\sqrt{\frac{30}{14^2}} \doteq 0.39$.

Thus by the Empirical Rule the likelihood is appreciable in the range $(\text{mean} \pm 3\,\text{SD}) \doteq (2.14 \pm 1.17) \doteq (1.0, 3.3)$, and you can see from the plot above that the prior is indeed relatively flat in this region.

From the Bayesian updating in (66), with a $\Gamma(0.001, 0.001)$ prior the posterior is $\Gamma(29.001, 14.001)$.

LOS data analysis. It's useful, in summarizing the updating from prior through likelihood to posterior, to make a table that records measures of center and spread at each point along the way.

For example, the $\Gamma(0.001, 0.001)$ prior, when regarded (as usual) as a density for λ, has mean 1.000 and SD $\sqrt{1000} \doteq 31.6$ (i.e., informally, as far as we're concerned, before the data arrive λ could be anywhere between 0 and (say) 100).

And the $\Gamma(29.001, 14.001)$ posterior has mean $\frac{29.001}{14.001} \doteq 2.071$ and SD $\sqrt{\frac{29.001}{14.001^2}} \doteq 0.385$, so after the data have arrived we know quite a bit more than before.

There are two main ways to summarize the likelihood—Fisher's approach based on maximizing it, and the Bayesian approach based on regarding it as a density and integrating it—and it's instructive to compute them both and compare.

The likelihood-integrating approach (which, at least in one-parameter problems, is essentially equivalent to Fisher's (1935) attempt at so-called fiducial inference) treats the $\Gamma(30, 14)$ likelihood as a density for λ, with mean $\frac{30}{14} \doteq 2.143$ and SD $\sqrt{\frac{30}{14^2}} \doteq 0.391$.

As for the likelihood-maximizing approach, from (61) the log likelihood function is

$$ll(\lambda|y) = ll(\lambda|s) = \log\big(c\,\lambda^s e^{-n\lambda}\big) = c + s\log\lambda - n\lambda, \qquad (2.68)$$

and this is maximized as usual (check that it's the max) by setting the derivative equal to 0 and solving:

$$\frac{\partial}{\partial\lambda} ll(\lambda|s) = \frac{s}{\lambda} - n = 0 \quad \text{iff} \quad \lambda = \hat{\lambda}_{\text{MLE}} = \frac{s}{n} = \bar{y}. \qquad (2.69)$$

Since the MLE $\hat{\lambda}_{\text{MLE}}$ turns out to be our old friend the sample mean $\bar{y}$, you might be tempted to conclude immediately that $\widehat{SE}\left(\hat{\lambda}_{\text{MLE}}\right) = \frac{\hat{\sigma}}{\sqrt{n}}$, where $\hat{\sigma} = 1.54$ is the sample SD, and indeed it's true in repeated sampling that $V\big(\bar{Y}\big) = \frac{V(Y_1)}{n}$; but the Poisson distribution has variance $V(Y_1) = \lambda$, so that $\sqrt{V\big(\bar{Y}\big)} = \frac{\sqrt{\lambda}}{\sqrt{n}}$, and there's no guarantee in the Poisson model that the best way to estimate $\sqrt{\lambda}$ in this standard error calculation is with the sample SD

$\hat{\sigma}$ (in fact we have a strong hint from the above MLE calculation that the sample variance is irrelevant to the estimation of λ in the Poisson model).

The right (large-sample) likelihood-based standard error for $\hat{\lambda}_{\text{MLE}}$, using the Fisher information logic we examined earlier, is obtained from the following calculation:

$$\begin{aligned}
\frac{\partial^2}{\partial \lambda^2} \log l(\lambda|y) &= -\frac{s}{\lambda^2}, \quad \text{so} \\
\hat{I}\left(\hat{\lambda}_{\text{MLE}}\right) &= \left[-\frac{\partial^2}{\partial \lambda^2} \log l(\lambda|y) \right]_{\lambda=\hat{\lambda}_{\text{MLE}}} \\
&= \left(\frac{s}{\lambda^2} \right)_{\lambda=\bar{y}} = \frac{s}{\bar{y}^2} = \frac{n}{\bar{y}}, \quad \text{and} \\
\hat{V}\left(\hat{\lambda}_{\text{MLE}}\right) &= \hat{I}^{-1}\left(\hat{\lambda}_{\text{MLE}}\right) = \frac{\bar{y}}{n} = \frac{\hat{\lambda}_{\text{MLE}}}{n}.
\end{aligned} \tag{2.70}$$

So in this case study Fisher's likelihood-maximizing approach would estimate λ by $\hat{\lambda}_{\text{MLE}} = \bar{y} = \frac{29}{14} \doteq 2.071$, with a give-or-take of $\widehat{SE}\left(\hat{\lambda}_{\text{MLE}}\right) = \frac{\sqrt{\hat{\lambda}_{\text{MLE}}}}{\sqrt{n}} = \frac{1.44}{\sqrt{14}} \doteq 0.385$.

All of this may be summarized in the following table:

	Prior	Likelihood Maximizing	 Integrating	Posterior
Mean/Estimate	1.00	2.071	2.143	2.071
SD/SE	31.6	0.385	0.391	0.385

The discrepancies between the likelihood-maximizing and likelihood-integrating columns in this table would be smaller with a larger sample size and would tend to 0 as $n \rightarrow \infty$.

The prior-likelihood-posterior plot comes out like this:

```
> plot( { p( lambda, 0.001, 0.001 ), p( lambda, 30, 14 ),
    p( lambda, 29.001, 14.001 ) }, lambda = 0 .. 5,
    color = black );
```

For interval estimation in the maximum-likelihood approach the best we could do, using the technology I've described to you so far, would be to appeal to the CLT (even though n is only 14) and use $\hat{\lambda}_{\text{MLE}} \pm 1.96\, \widehat{SE}(\hat{\lambda}_{\text{MLE}}) \doteq 2.071 \pm (1.96)(0.385) \doteq (1.316, 2.826)$ as an approximate 95% confidence interval for λ.

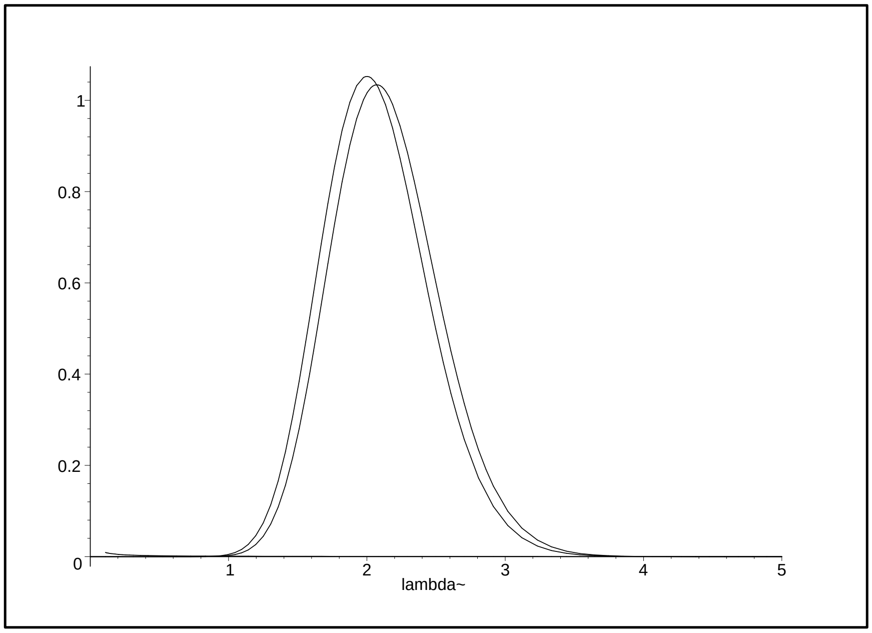


Figure 2.14: *Prior (almost flat, barely visible toward the left of the plot), likelihood (the right-most density), and posterior (the central density) distributions in the length of stay case study.*

You can see from the previous plot that the likelihood function is skewed, so a more careful method (e.g., the bootstrap; Efron 1979) would be needed to create a better interval estimate from the likelihood point of view.

Some trial and error with `Maple` can be used to find the lower and upper limits of the central 95% posterior interval for λ:

```
> evalf( Int( p( lambda, 29.001, 14.001 ),
    lambda = 0 .. 1.316 ) );

                                        .01365067305

> evalf( Int( p( lambda, 29.001, 14.001 ), lambda = 0 .. 1.4 ) );

                                        .02764660367
```

```
  > evalf( Int( p( lambda, 29.001, 14.001 ),
      lambda = 0 .. 1.387 ) );

                                   .02495470339

  > evalf( Int( p( lambda, 29.001, 14.001 ),
      lambda = 2.826 .. infinity ) );

                                   .03403487851

  > evalf( Int( p( lambda, 29.001, 14.001 ),
      lambda = 2.890 .. 5 ) );

                                   .02505306648

  > evalf( Int( p( lambda, 29.001, 14.001 ),
      lambda = 2.890 .. infinity ) );

                                   .02505307631
```

Thus a 95% (central) posterior interval for λ, given a diffuse prior, runs from 1.387 to 2.890, and is (correctly) asymmetric around the posterior mean of 2.071.

R can be used to work out the limits of this interval even more readily:

```
> help( qgamma )
GammaDist               package:base               R Documentation

The Gamma Distribution

Description:

     Density, distribution function, quantile function and random
     generation for the Gamma distribution with parameters ‘shape’
     and ‘scale’.

Usage:

     dgamma(x, shape, scale=1, log = FALSE)
     pgamma(q, shape, scale=1, lower.tail = TRUE, log.p = FALSE)
     qgamma(p, shape, scale=1, lower.tail = TRUE, log.p = FALSE)
     rgamma(n, shape, scale=1)

Arguments:

    x, q: vector of quantiles.

       p: vector of probabilities.

       n: number of observations.

shape, scale: shape and scale parameters.

log, log.p: logical; if TRUE, probabilities p are given as log(p).

lower.tail: logical; if TRUE (default), probabilities are
          P[X <= x], otherwise, P[X > x].

Details:

     If ‘scale’ is omitted, it assumes the default value of ‘1’.

     The Gamma distribution with parameters ‘shape’ = a and
     ‘scale’ = s has density

               f(x)= 1/(s^a Gamma(a)) x^(a-1) e^-(x/s)

     for x > 0, a > 0 and s > 0. The mean and variance are
     E(X) = a*s and Var(X) = a*s^2.

> qgamma( 0.025, 29.001, 1 / 14.001 )

[1] 1.387228

> qgamma( 0.975, 29.001, 1 / 14.001 )

[1] 2.890435
```

`Maple` or `R` can also be used to obtain the probability content, according to the posterior distribution, of the approximate 95% (large-sample) likelihood-based interval:

```
  > evalf( Int( p( lambda, 29.001, 14.001 ),
      lambda = 1.316 .. 2.826 ) );

                                        .9523144484
```

So the maximization approach has led to decent approximations here (later I'll give examples where maximum likelihood doesn't do so well in small samples).

Predictive distributions in this model can be computed by `Maple` in the usual way: for instance, to compute $p(y_{n+1}|y)$ for $y = (y_1, \ldots, y_n)$ we want to evaluate

$$\begin{aligned} p(y_{n+1}|y) &= \int_0^\infty p(y_{n+1}, \lambda|y)\, d\lambda \\ &= \int_0^\infty p(y_{n+1}|\lambda, y)\, p(\lambda|y)\, d\lambda \qquad (2.71) \\ &= \int_0^\infty p(y_{n+1}|\lambda)\, p(\lambda|y)\, d\lambda \\ &= \int_0^\infty \frac{\lambda^{y_{n+1}} e^{-\lambda}}{y_{n+1}!} \frac{(\beta^*)^{\alpha^*}}{\Gamma(\alpha^*)} \lambda^{\alpha^*-1} e^{-\beta^* \lambda}\, d\lambda, \\ &= \frac{(\beta^*)^{\alpha^*}}{\Gamma(\alpha^*)\, y_{n+1}!} \int_0^\infty \lambda^{(\alpha^*+y_{n+1})-1} e^{-(\beta^*+1)\lambda}\, d\lambda, \end{aligned}$$

where $\alpha^* = \alpha + s$ and $\beta^* = \beta + n$; in these expressions y_{n+1} is a non-negative integer.

```
> assume( astar > 0, bstar > 0, yf > 0 );

> simplify( bstar^astar * int( lambda^( astar + yf - 1 ) *
    exp( - ( bstar + 1 ) * lambda ), lambda = 0 .. infinity ) /
    ( GAMMA( astar ) * yf! ) );

          astar~             (-astar~ - yf~)
      bstar~       (bstar~ + 1)                GAMMA(astar~ + yf~)
      ------------------------------------------------------------
                      GAMMA(astar~) GAMMA(yf~ + 1)
```

Predictive distributions. A bit of rearranging then gives that for $y_{n+1} = 0, 1, \ldots,$

$$p(y_{n+1}|y) = \frac{\Gamma(\alpha^* + y_{n+1})}{\Gamma(\alpha^*)\,\Gamma(y_{n+1}+1)} \left(\frac{\beta^*}{\beta^*+1} \right)^{\alpha^*} \left(\frac{1}{\beta^*+1} \right)^{y_{n+1}} . \qquad (2.72)$$

This is called the Poisson-Gamma distribution, because (71) is asking us to take a mixture (weighted average) of Poisson distributions, using probabilities from a Gamma distribution as the mixing weights.

(72) is a generalization of the negative binomial distribution (e.g., Johnson and Kotz 1994), which you may have encountered in your earlier probability study.

`Maple` can try to get simple expressions for the mean and variance of this distribution:

```
> pg := ( y, alpha, beta ) -> GAMMA( alpha + y ) *
    ( beta / ( beta + 1 ) )^alpha * ( 1 / ( beta + 1 ) )^y /
    ( GAMMA( alpha ) * GAMMA( y + 1 ) );

    pg := (y, alpha, beta) ->

                       /  beta  \alpha /    1    \y
      GAMMA(alpha + y) |--------|      |--------|
                       \beta + 1/      \beta + 1/
      ---------------------------------------------
                 GAMMA(alpha) GAMMA(y + 1)

> simplify( sum( pg( y, alpha, beta ), y = 0 .. infinity ) );

                                        1

> simplify( sum( y * pg( y, alpha, beta ), y = 0 .. infinity ) );

                                      alpha
                                      -----
                                      beta
```

So the mean of the distribution in (72) is $E(y_{n+1}|y) = \frac{\alpha^*}{\beta^*}$.

```
> simplify( sum( ( y - alpha / beta )^2 * pg( y, alpha, beta ),
    y = 0 .. infinity ) );

     2 /  beta  \alpha                      alpha - beta
alpha  |--------|      hypergeom([alpha, - ------------, ... ],
       \beta + 1/                               beta

       alpha    alpha      1         /     2
    [- -----, - -----], --------)  /  beta
       beta     beta    beta + 1  /
```

Maple has failed to realize that this expression may be considerably simplified: Bernardo and Smith (1994) note that the variance of the distribution in (72) is just

$$V(y_{n+1}|y) = \frac{\alpha^*}{\beta^*}\left(1 + \frac{1}{\beta^*}\right). \tag{2.73}$$

Inference and prediction. This provides an interesting contrast between inference and prediction: we've already seen in this model that the posterior mean and variance of λ are $\frac{\alpha^*}{\beta^*} = \frac{\alpha+s}{\beta+n}$ and $\frac{\alpha^*}{(\beta^*)^2} = \frac{\alpha+s}{(\beta+n)^2}$, respectively.

	Posterior	
Quantity	Mean	Variance
λ	$\frac{\alpha+s}{\beta+n}$	$\frac{\alpha+s}{(\beta+n)^2} = \frac{\alpha+s}{\beta+n}\left(0 + \frac{1}{\beta+n}\right)$
y_{n+1}	$\frac{\alpha+s}{\beta+n}$	$\frac{\alpha+s}{\beta+n}\left(1 + \frac{1}{\beta+n}\right)$

Thus λ (the inferential objective) and y_{n+1} (the predictive objective) have the same posterior mean, but the posterior variance of y_{n+1} is much larger, as can be seen by the following argument.

(1) Denoting by μ the mean of the population from which the Y_i are thought of as (like) a random sample, when n is large α and β will be small in relation to s and n, respectively, and the ratio $\bar{y} = \frac{s}{n}$ should more and more closely approach μ—thus for large n,

$$E(\lambda|y) = E(y_{n+1}|y) \doteq \mu. \tag{2.74}$$

(2) For the Poisson distribution the (population) mean μ and variance σ^2 are equal, meaning that for large n the ratio $\frac{\alpha+s}{\beta+n}$ will be close both to μ and to σ^2.

Thus for large n,

$$V(\lambda|y) \doteq \frac{\sigma^2}{n} \quad \text{but} \quad V(y_{n+1}|y) \doteq \sigma^2. \tag{2.75}$$

An informal way to restate (75) is to say that accurate prediction of new data is an order of magnitude harder (in powers of n) than accurate inference about population parameters.

Bayesian model-checking with predictive distributions. One way to check a model like (58) is as follows.

```
for ( i in 1:n ) {
```

Temporarily set aside observation y_i, obtaining a new dataset $y_{-i} = (y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n)$ with $(n-1)$ observations.

Use the current Bayesian model applied to y_{-i} to predict y_i, and summarize the extent to which the actual value of y_i is surprising in view of this predictive distribution.

```
}
```

One possible measure of surprise is predictive z–scores:

$$z_i = \frac{y_i - E[y_i|y_{-i}]}{\sqrt{V[y_i|y_{-i}]}}. \tag{2.76}$$

Compare the surprise measure with its expected behavior if the model had been "correct" (e.g., $z = (z_1, \ldots, z_n)$ should have mean 0 and SD 1).

Example: the LOS data. Here's some `R` code to carry out this program on the LOS data.

```
> poisson.gamma <- function( y, alpha, beta ) {

    log.density <- lgamma( alpha + y ) + alpha *
      log( beta / ( beta + 1 ) ) + y * log( 1 / ( beta + 1 ) ) -
      lgamma( alpha ) - lgamma( y + 1 )

    return( exp( log.density ) )

  }
```

```
> print( y <- sort( y ) )

 [1] 0 1 1 1 1 1 2 2 2 2 3 3 4 6

> print( y.current <- y[ -1 ] )

 [1] 1 1 1 1 1 2 2 2 2 3 3 4 6

> print( n.current <- length( y.current ) )

[1] 13

> alpha <- beta <- 0.001

> print( s.current <- sum( y.current ) )

[1] 29

> print( alpha.star <- alpha + s.current )

[1] 29.001

> print( beta.star <- beta + n.current )

[1] 13.001

> print( pg.current <- poisson.gamma( 0:9, alpha.star,
    beta.star ) )

 [1] 0.116595340 0.241509997 0.258750854 0.190975293 0.109124354
 [6] 0.051442223 0.020820977 0.007435744 0.002389956 0.000701781
```

This predictive distribution is plotted in Figure 2.15

```
> postscript( "pg1.ps" )

> plot( 0:9, pg.current, type = 'n', xlab = 'y',
    ylab = 'Density' )
```

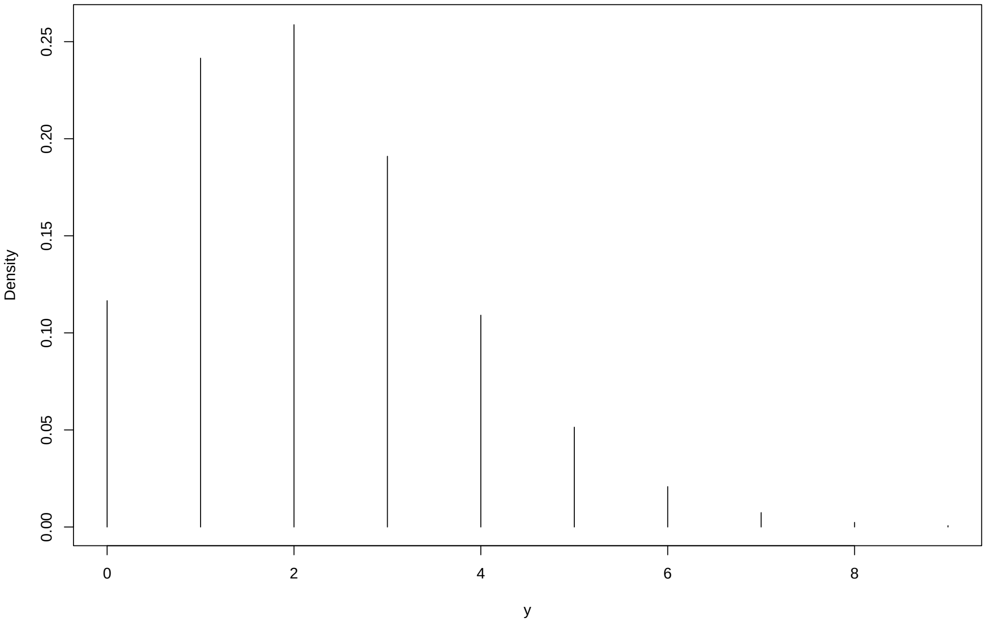


Figure 2.15: *The Poisson-Gamma predictive distribution with 0 omitted.*

```
> for ( i in 0:9 ) {

    segments( i, 0, i, pg.current[ i + 1 ] )

  }

> dev.off( )

null device
          1
```

The omitted observed value of 0 is not too unusual in this predictive distribution.

The following R code loops through the whole dataset to get the predictive z–scores.

```
alpha <- beta <- 0.001
```

```
z <- rep( 0, n )

for ( i in 1:n ) {

  y.current <- y[ -i ]

  n.current <- length( y.current )

  s.current <- sum( y.current )

  alpha.star <- alpha + s.current

  beta.star <- beta + n.current

  predictive.mean.current <- alpha.star / beta.star

  predictive.SD.current <- sqrt( ( alpha.star / beta.star ) *
    ( 1 + 1 / beta.star ) )

  z[ i ] <- ( y[ i ] - predictive.mean.current ) /
    predictive.SD.current

}
```

And the predictive z–scores are:

```
> z

 [1] -1.43921925 -0.75757382 -0.75757382 -0.75757382 -0.75757382
 [6] -0.75757382 -0.05138023 -0.05138023 -0.05138023 -0.05138023
[11]  0.68145253  0.68145253  1.44329065  3.06513271

> mean( z )

[1] 0.03133708

> sqrt( var( z ) )

[1] 1.155077
```

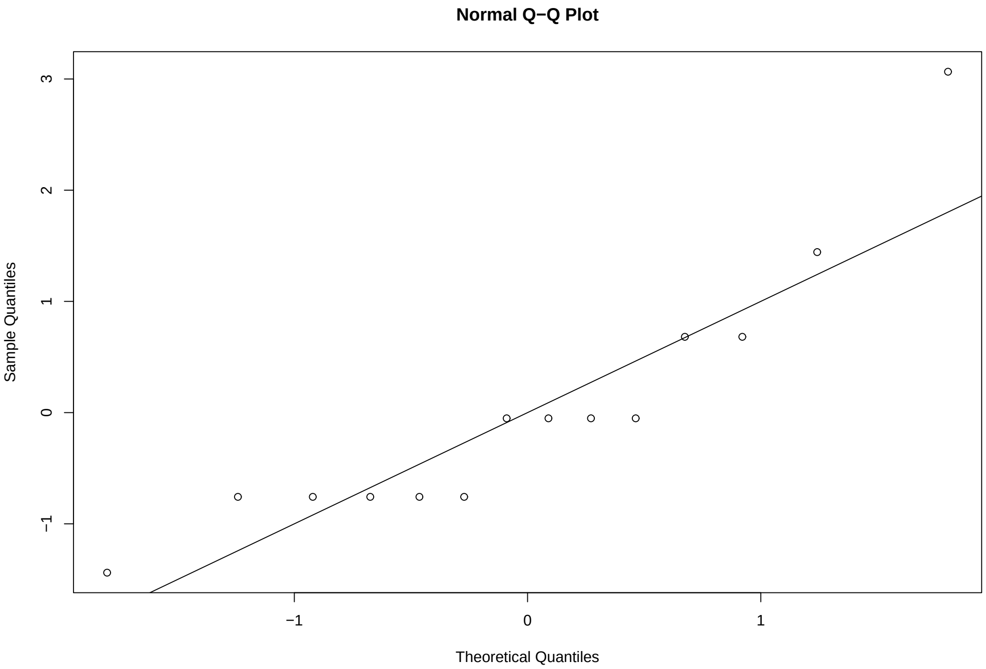


Figure 2.16: *Normal qqplot of the predictive z–scores.*

```
> postscript( "pg2.ps" )

> qqnorm( z )

> abline( 0, 1 )
```

The 14 predictive z–scores have mean 0.03 (about right) and SD 1.16 (close enough to 1 when sampling variability is considered?), and the normal qqplot above shows that the only really surprising observation in the data, as far as the Poisson model was concerned, is the value of 6, which has a z–score of 3.07.

NB The figure above is only a crude approximation to the right qqplot, which would have to be created by simulation; even so it's enough to suggest how the model might be improved.

I would conclude informally (a) that the Poisson is a decent model for these data, but (b) if you wanted to expand the model in a direction suggested

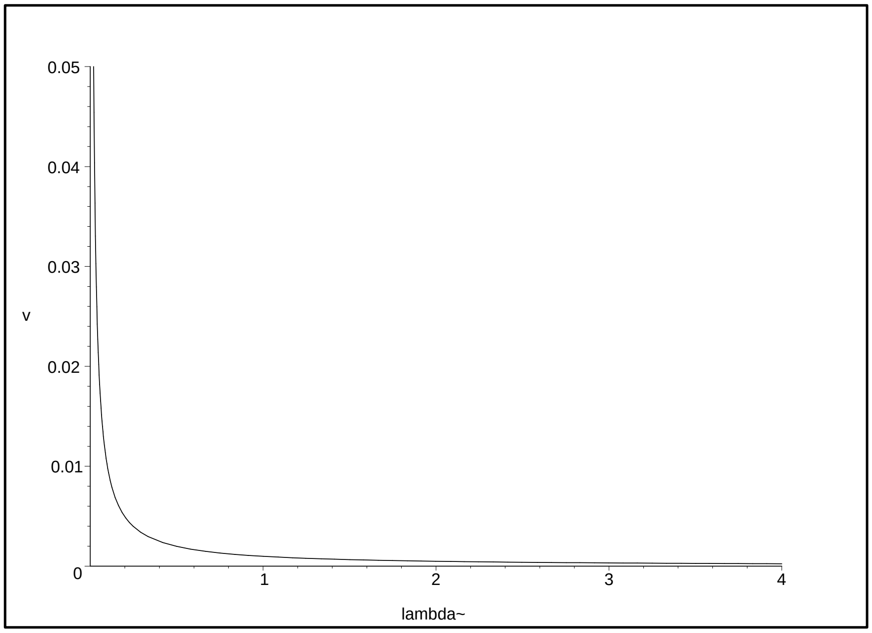


Figure 2.17: *The Gamma*$(0.001, 0.001)$ *prior for* $\lambda \in (0, 4)$.

by this diagnostic you should look for a model with extra-Poisson variation: the sample VTMR in this dataset was about 1.15.

Diffuse priors in the LOS case study. In specifying a diffuse prior for λ in the LOS case study, several alternatives to $\Gamma(\epsilon, \epsilon)$ might occur to you, including $\Gamma(1, \epsilon), \Gamma(\alpha, \beta)$ for some large α (like 20, to get a roughly normal prior) and small β (like 1, to have a small prior sample size), and $U(0, C)$ for some cutoff C (like 4) chosen to avoid truncation of the likelihood function, where $U(a, b)$ denotes the uniform distribution on (a, b).

```
> plot( p( lambda, 0.001, 0.001 ), lambda = 0 .. 4, v = 0 .. 0.05,
    color = black );
```

```
> plot( p( lambda, 1.0, 0.001 ), lambda = 0 .. 4, color = black );
```

$\Gamma(1, \epsilon)$ doesn't look promising initially as a flat prior, but that's a consequence of `Maple`'s default choice of vertical axis.

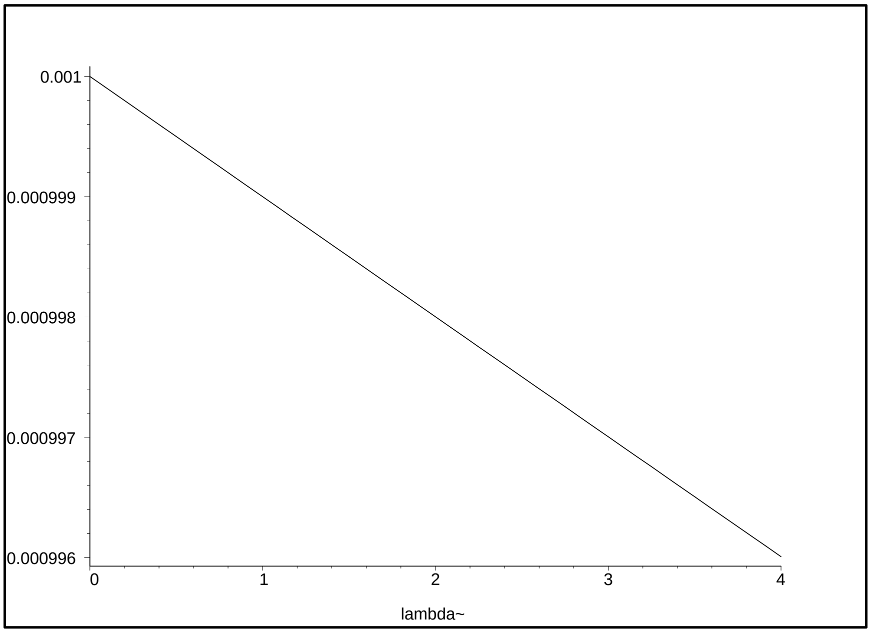


Figure 2.18: *The Gamma*$(1.0, 0.001)$ *prior for* $\lambda \in (0, 4)$*, with* ***Maple****'s default choice of vertical axis.*

```
> plot( p( lambda, 1.0, 0.001 ), lambda = 0 .. 4, v = 0 .. 0.05,
    color = black );
```

```
> plot( p( lambda, 20, 1 ), lambda = 0 .. 4, color = black );
```

```
> plot( p( lambda, 20, 1 ), lambda = 0 .. 40, color = black );
```

$\Gamma(20, 1)$ does indeed look not far from Gaussian, and at first it may appear that it is indeed relatively flat in the region where the likelihood is appreciable ($\lambda \in (1.0, 3.3)$), but we'll see below that it's actually rather more informative than we intend.

Recalling that the mean and SD of a $\Gamma(\alpha, \beta)$ random quantity are $\frac{\alpha}{\beta}$ and $\sqrt{\frac{\alpha}{\beta^2}}$, respectively, and that when used as a prior with the Poisson likelihood

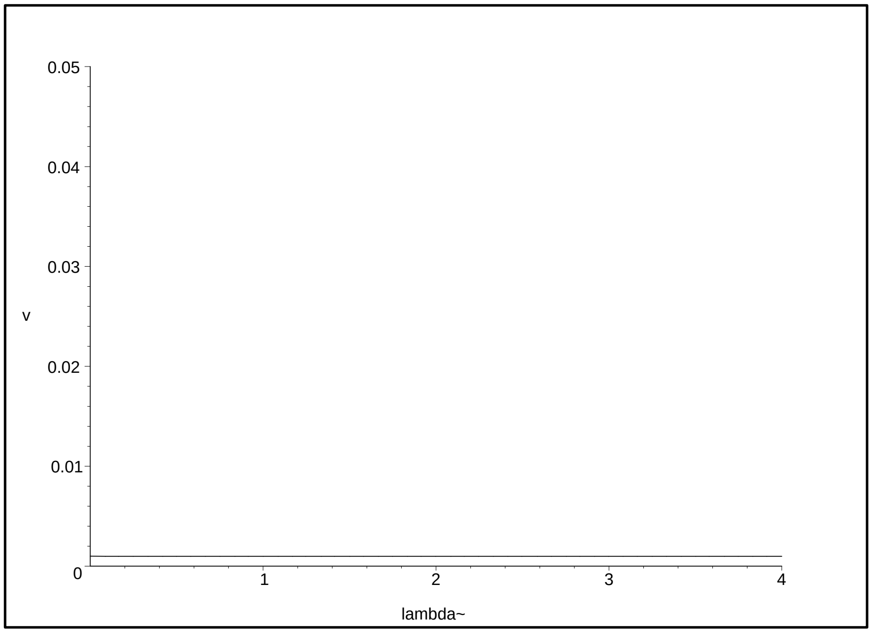


Figure 2.19: *The Gamma*$(1.0, 0.001)$ *prior for* $\lambda \in (0, 4)$*, with the same vertical axis as in Figure 2.17.*

the $\Gamma(\alpha, \beta)$ distribution acts like a dataset with prior sample size β, you can construct the following table:

Prior				Posterior			
	$\beta =$						
α	Sample Size	Mean	SD	α^*	β^*	Mean	SD
0.001	0.001	1	31.6	29.001	14.001	2.071	0.385
1	0.001	1000	1000	30	14.001	2.143	0.391
20	1	20	4.47	49	15	3.267	0.467
20	0.001	20000	4472	49	14.001	3.500	0.500
$U(0, C)$ for $C > 4$		$\frac{C}{2}$	$\frac{C}{\sqrt{12}}$	30	14	2.143	0.391

The $\Gamma(1, \epsilon)$ prior leads to an analysis that's essentially equivalent to the integrated likelihood (fiducial) approach back on p. 72, and the $U(0, C)$ prior for $C > 4$ (say) produces similar results: $U(0, C)$ yields the $\Gamma(s + 1, n)$ posterior truncated to the right of C (and this truncation has no effect if you choose C big enough).

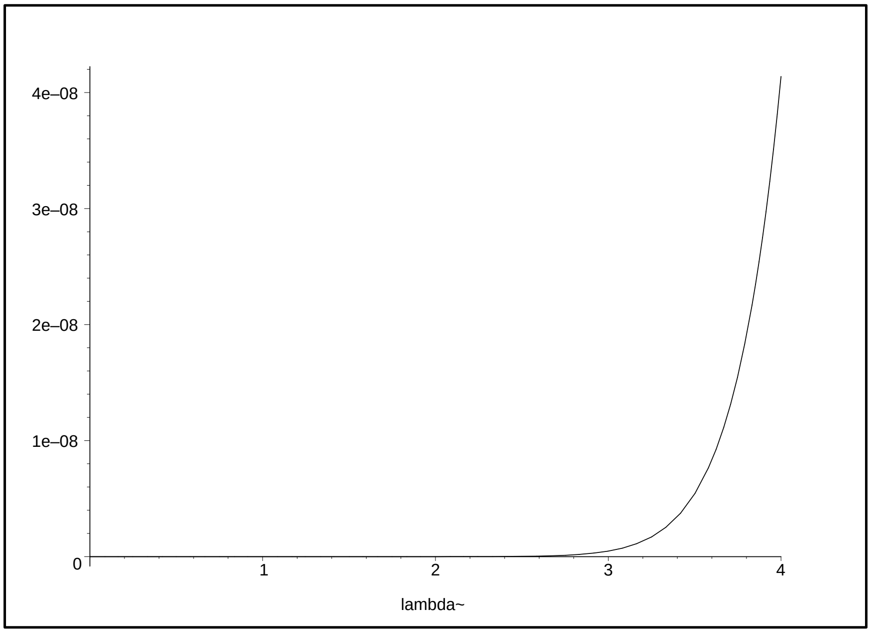


Figure 2.20: *The* $Gamma\,(20, 1)$ *prior, with the same horizontal and vertical axes as in the previous two figures.*

You might say that the $U(0, C)$ distribution has a prior sample size of 0 in this analysis, and its prior mean $\frac{C}{2}$ and SD $\frac{C}{\sqrt{12}}$ (both of which can be made arbitrarily large by letting C grow without bound) are irrelevant (an example of how intuition can change when you depart from the class of conjugate priors).

```
> plot( { p( lambda, 29.001, 14.001 ), p( lambda, 30, 14.001 ),
    p( lambda, 49, 15 ), p( lambda, 49, 14.001 ) },
    lambda = 0 .. 6, color = black );
```

The moral is that with only $n = 14$ observations, some care is needed (e.g., through pre-posterior analysis) to achieve a prior that doesn't affect the posterior very much, if that's your goal.

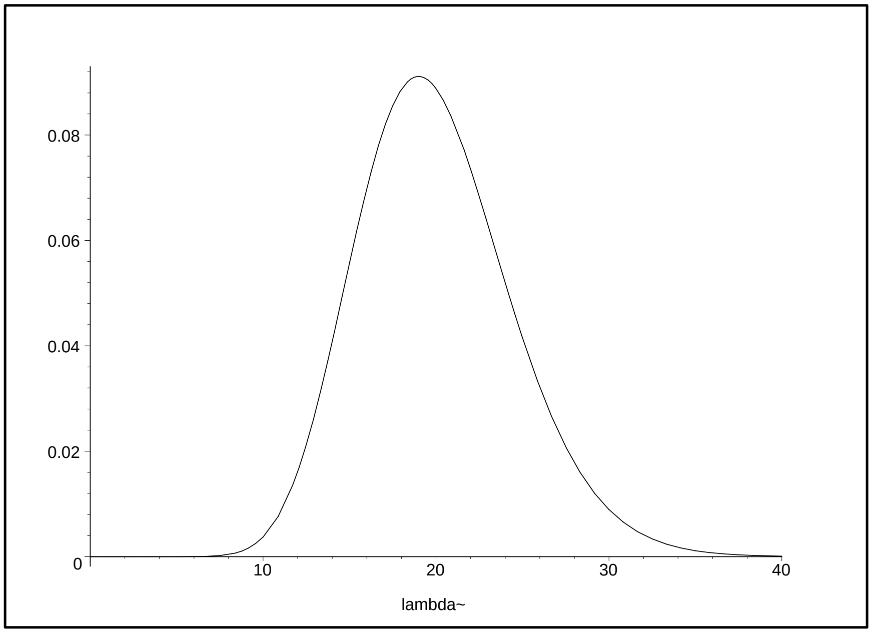


Figure 2.21: *The Gamma*$(20, 1)$ *prior, for* $\lambda \in (0, 40)$.

2.5 Continuous outcomes

For continuous outcomes there's an analogue of de Finetti's Theorem that's equally central to Bayesian model-building (e.g., Bernardo and Smith, 1994):

de Finetti's Theorem for Continuous Outcomes. If $Y_1, Y_2, \ldots$ is an infinitely exchangeable sequence of real-valued random quantities with probability measure p, there exists a probability measure Q over $\mathcal{D}$, the space of all distribution functions on R, such that the joint distribution function of $Y_1, \ldots, Y_n$ has the form

$$p(y_1, \ldots, y_n) = \int_{\mathcal{D}} \prod_{i=1}^{n} F(y_i) \, dQ(F), \tag{2.77}$$

where $Q(F) \overset{P}{=} \lim_{n \to \infty} p(F_n)$ and F_n is the empirical distribution function based on $Y_1, \ldots, Y_n$.

In other words, exchangeability of real-valued observables is equivalent to

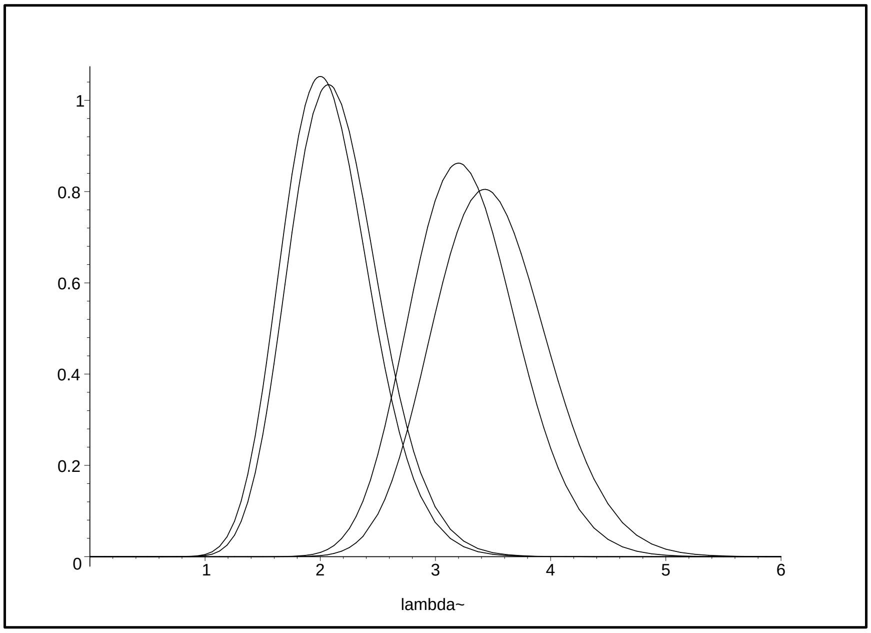


Figure 2.22: *The four posteriors arising from the five priors in Table xxx.*

the hierarchical model

$$\begin{aligned} F &\sim p(F) && \text{(prior)} \\ (Y_1, \ldots, Y_n | F) &\stackrel{\text{IID}}{\sim} F && \text{(likelihood)} \end{aligned} \tag{2.78}$$

for some prior distribution p on the set $\mathcal{D}$ of all possible distribution functions.

This prior makes the continuous form of de Finetti's Theorem considerably harder to apply: to take the elicitation task seriously is to try to specify a probability distribution on a function space (F is in effect an infinite-dimensional parameter).

(NB This task is not unique to Bayesians; it's equivalent to asking "Where does the likelihood come from?" in frequentist analyses of observational data.)

What people often do in practice is to appeal to considerations that narrow down the field, such as an *a priori* judgment that the Y_i ought to be symmetrically distributed about a measure of center μ, and then try to use a fairly rich parametric family satisfying (e.g.) the symmetry restriction as a substitute for all of $\mathcal{D}$.

Strictly speaking you're not supposed to look at the Y_i while specifying your prior on $\mathcal{D}$—this can lead to a failure to fully assess and propagate model uncertainty—but not doing so can permit the data to surprise you in ways that would make you want to go back and revise your prior (an example of Cromwell's Rule in action).

As mentioned earlier, I'll suggest two potential ways out of this dilemma, based on out-of-sample predictive validation (the model-checking in the LOS data above was an example of this) and Bayesian nonparametrics.

Case Study: *Measurement of physical constants.* What used to be called the National Bureau of Standards (NBS) in Washington, DC, conducts extremely high precision measurement of physical constants, such as the actual weight of so-called check-weights that are supposed to serve as reference standards (like the official kg).

In 1962–63, for example, $n = 100$ weighings (listed below) of a block of metal called NB10, which was supposed to weigh exactly 10g, were made under conditions as close to IID as possible (Freedman et al., 1998).

Value	375	392	393	397	398	399	400	401
Frequency	1	1	1	1	2	7	4	12
Value	402	403	404	405	406	407	408	409
Frequency	8	6	9	5	12	8	5	5
Value	410	411	412	413	415	418	423	437
Frequency	4	1	3	1	1	1	1	1

NB10 modeling. Q: (a) How much does NB10 really weigh? (b) How certain are you given the data that the true weight of NB10 is less than (say) 405.25? And (c) How accurately can you predict the 101st measurement?

The graph below is a normal qqplot of the 100 measurements $y = (y_1, \ldots, y_n)$, which have a mean of $\bar{y} = 404.6$ (the units are micrograms below 10g) and an SD of $s = 6.5$.

Evidently it's plausible in answering these questions to assume symmetry of the "underlying distribution" F in de Finetti's Theorem.

One standard choice, for instance, is the Gaussian:

$$\begin{array}{rcl} (\mu, \sigma^2) & \sim & p(\mu, \sigma^2) \\ (Y_i | \mu, \sigma^2) & \stackrel{\text{IID}}{\sim} & N\big(\mu, \sigma^2\big) \, . \end{array} \tag{2.79}$$

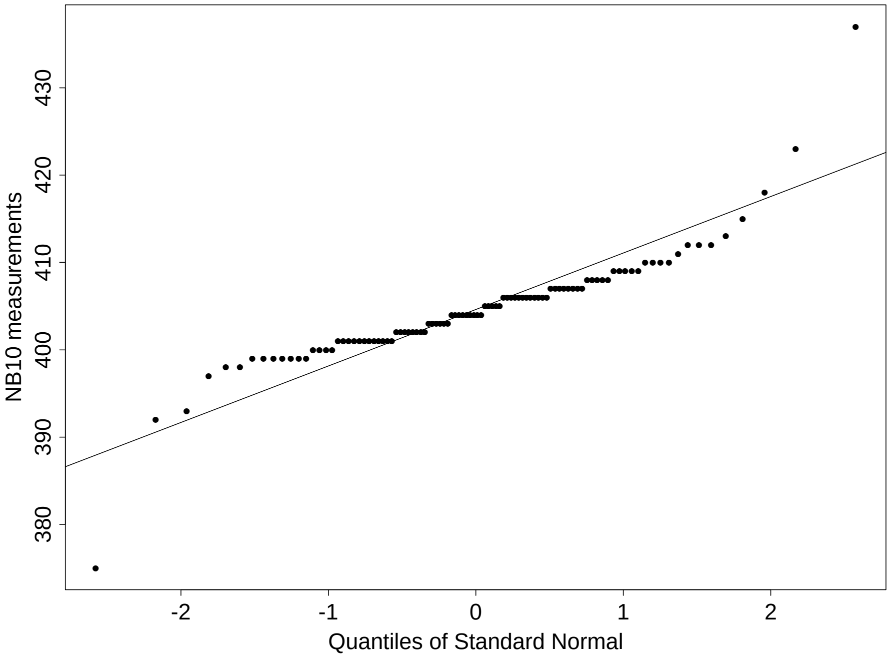


Figure 2.23: *Normal qqplot of the 100 NB10 measurements.*

Here $N(\mu, \sigma^2)$ is the familiar normal density

$$p(y_i|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\frac{y_i - \mu}{\sigma} \right)^2 \right]. \tag{2.80}$$

Gaussian modeling. Even though you can see from the previous graph that (79) is not a good model for the NB10 data, I'm going to fit it to the data for practice in working with the normal distribution from a Bayesian point of view (later we'll improve upon the Gaussian).

(79) is more complicated than the models in the AMI and LOS case studies because the parameter θ here is a vector: $\theta = (\mu, \sigma^2)$.

To warm up for this new complexity let's first consider a cut-down version of the model in which we pretend that σ is known to be $\sigma_0 = 6.5$ (the sample SD).

This simpler model is then

$$\left\{ \begin{array}{ccc} \mu & \sim & p(\mu) \\ (Y_i|\mu) & \stackrel{\text{IID}}{\sim} & N(\mu, \sigma_0^2) \end{array} \right\}. \tag{2.81}$$

The likelihood function in this model is

$$\begin{aligned} l(\mu|y) &= \prod_{i=1}^n \frac{1}{\sigma_0\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma_0^2}(y_i-\mu)^2 \right] \\ &= c \exp\left[-\frac{1}{2\sigma_0^2} \sum_{i=1}^n (y_i-\mu)^2 \right] \qquad (2.82) \\ &= c \exp\left[-\frac{1}{2\sigma_0^2} \left(\sum_{i=1}^n y_i^2 - 2\,\mu \sum_{i=1}^n y_i + n\mu^2 \right) \right] \\ &= c \exp\left[-\frac{1}{2\left(\frac{\sigma_0^2}{n}\right)} (\mu-\bar{y})^2 \right]. \end{aligned}$$

Thus the likelihood function, when thought of as a density for μ, is a normal distribution with mean $\bar{y}$ and SD $\frac{\sigma_0}{\sqrt{n}}$.

Notice that this SD is the same as the frequentist standard error for $\bar{Y}$ based on an IID sample of size n from the $N(\mu, \sigma_0^2)$ distribution.

(82) also shows that the sample mean $\bar{y}$ is a sufficient statistic for μ in model (81).

In finding the conjugate prior for μ it would be nice if the product of two normal distributions is another normal distribution, because that would demonstrate that the conjugate prior is normal.

Suppose therefore, to see where it leads, that the prior for μ is (say) $p(\mu) = N\big(\mu_0, \sigma_\mu^2\big)$.

Then Bayes' Theorem would give

$$\begin{aligned} p(\mu|y) &= c\,p(\mu)\,l(\mu|y) \qquad (2.83) \\ &= c \exp\left[-\frac{1}{2\sigma_\mu^2}(\mu-\mu_0)^2 \right] \exp\left[-\frac{n}{2\sigma_0^2}(\mu-\bar{y})^2 \right] \\ &= c \exp\left\{ -\frac{1}{2} \left[\frac{(\mu-\mu_0)^2}{\sigma_\mu^2} + \frac{n(\mu-\bar{y})^2}{\sigma_0^2} \right] \right\}, \end{aligned}$$

and we want this to be of the form

$$\begin{aligned} p(\mu|y) &= c \exp\left\{ -\frac{1}{2} \left[A(\mu - B)^2 + C \right] \right\} \\ &= c \exp\left\{ -\frac{1}{2} \left[A\mu^2 - 2AB\mu + (AB^2 + C) \right] \right\} \end{aligned} \tag{2.84}$$

for some B, C, and $A > 0$.

`Maple` can help see if this works:

```
> collect( ( mu - mu0 )^2 / sigmamu^2 +
    n * ( mu - ybar )^2 / sigma0^2, mu );


/   1          n   \   2   /     mu0        n ybar \
|-------- + -------| mu  + |-2 -------- - 2 -------| mu
|       2         2|       |          2           2|
\sigmamu    sigma0 /       \   sigmamu      sigma0 /

          2          2
      mu0      n ybar
  + -------- + -------
           2         2
    sigmamu    sigma0
```

Matching coefficients for A and B (we don't really care about C) gives

$$A = \frac{1}{\sigma_\mu^2} + \frac{n}{\sigma_0^2} \quad \text{and} \quad B = \frac{\frac{\mu_0}{\sigma_\mu^2} + \frac{n\bar{y}}{\sigma_0^2}}{\frac{1}{\sigma_\mu^2} + \frac{n}{\sigma_0^2}}. \tag{2.85}$$

Since $A > 0$ this demonstrates two things: (1) the conjugate prior for μ in model (81) is normal, and (2) the conjugate updating rule (when σ_0 is assumed known) is

$$\left\{ \begin{array}{c} \mu \sim N\big(\mu_0, \sigma_\mu^2\big) \\ (Y_i|\mu) \overset{\text{IID}}{\sim} N(\mu, \sigma_0^2)\,, \\ i = 1, \ldots, n \end{array} \right\} \rightarrow (\mu|y) = (\mu|\bar{y}) = N\big(\mu_*, \sigma_*^2\big)\,, \tag{2.86}$$

where the posterior mean and variance are given by

$$\mu_* = B = \frac{\left(\frac{1}{\sigma_\mu^2}\right)\mu_0 + \left(\frac{n}{\sigma_0^2}\right)\bar{y}}{\frac{1}{\sigma_\mu^2} + \frac{n}{\sigma_0^2}} \quad \text{and} \quad \sigma_*^2 = A^{-1} = \frac{1}{\frac{1}{\sigma_\mu^2} + \frac{n}{\sigma_0^2}}. \tag{2.87}$$

It becomes useful in understanding the meaning of these expressions to define the precision of a distribution, which is just the reciprocal of its variance: whereas the variance and SD scales measure uncertainty, the precision scale quantifies information about an unknown.

With this convention (87) has a series of intuitive interpretations, as follows:

• The prior, considered as an information source, is Gaussian with mean μ_0, variance σ_μ^2, and precision $\frac{1}{\sigma_\mu^2}$, and when viewed as a data set consists of n_0 (to be determined below) observations;

• The likelihood, considered as an information source, is Gaussian with mean $\bar{y}$, variance $\frac{\sigma_0^2}{n}$, and precision $\frac{n}{\sigma_0^2}$, and when viewed as a data set consists of n observations;

• The posterior, considered as an information source, is Gaussian, and the posterior mean is a weighted average of the prior mean and data mean, with weights given by the prior and data precisions;

• The posterior precision (the reciprocal of the posterior variance) is just the sum of the prior and data precisions (this is why people invented the idea of precision—on this scale knowledge about μ in model (81) is additive); and

• Rewriting μ_* as

$$\mu_* = \frac{\left(\frac{1}{\sigma_\mu^2}\right)\mu_0 + \left(\frac{n}{\sigma_0^2}\right)\bar{y}}{\frac{1}{\sigma_\mu^2} + \frac{n}{\sigma_0^2}} = \frac{\left(\frac{\sigma_0^2}{\sigma_\mu^2}\right)\mu_0 + n\bar{y}}{\frac{\sigma_0^2}{\sigma_\mu^2} + n}, \qquad (2.88)$$

you can see that the prior sample size is

$$n_0 = \frac{\sigma_0^2}{\sigma_\mu^2} = \frac{1}{\left(\frac{\sigma_\mu}{\sigma_0}\right)^2}, \qquad (2.89)$$

which makes sense: the bigger σ_μ is in relation to σ_0, the less prior information is being incorporated in the conjugate updating (86).

Bayesian inference with multivariate θ. Returning now to (79) with σ^2 unknown, (as mentioned above) this model has a ($p = 2$)-dimensional parameter vector $\theta = (\mu, \sigma^2)$.

When $p > 1$ you can still use Bayes' Theorem directly to obtain the joint posterior distribution,

$$\begin{aligned} p(\theta|y) &= p(\mu, \sigma^2|y) = c\,p(\theta)\,l(\theta|y) \\ &= c\,p(\mu, \sigma^2)\,l(\mu, \sigma^2|y), \end{aligned} \qquad (2.90)$$

where $y = (y_1, \ldots, y_n)$, although making this calculation directly requires a p-dimensional integration to evaluate the normalizing constant c; for example, in this case

$$\begin{aligned} c &= [p(y)]^{-1} = \left(\iint p(\mu, \sigma^2, y) \, d\mu \, d\sigma^2 \right)^{-1} \\ &= \left(\iint p(\mu, \sigma^2) \, l(\mu, \sigma^2 | y) \, d\mu \, d\sigma^2 \right)^{-1} . \end{aligned} \tag{2.91}$$

Usually, however, you'll be more interested in the marginal posterior distributions, in this case $p(\mu|y)$ and $p(\sigma^2|y)$.

Obtaining these requires p integrations, each of dimension $(p-1)$, a process that people refer to as marginalization or integrating out the nuisance parameters—for example,

$$p(\mu|y) = \int_0^\infty p(\mu, \sigma^2|y) \, d\sigma^2 \, . \tag{2.92}$$

Predictive distributions also involve a p-dimensional integration: for example, with $y = (y_1, \ldots, y_n)$,

$$\begin{aligned} p(y_{n+1}|y) &= \iint p(y_{n+1}, \mu, \sigma^2|y) \, d\mu \, d\sigma^2 \qquad (2.93) \\ &= \iint p(y_{n+1}|\mu, \sigma^2) \, p(\mu, \sigma^2|y) \, d\mu \, d\sigma^2. \end{aligned}$$

And, finally, if you're interested in a function of the parameters, you have some more hard integrations ahead of you.

For instance, suppose you wanted the posterior distribution for the coefficient of variation $\lambda = g_1(\mu, \sigma^2) = \frac{\sqrt{\sigma^2}}{\mu}$ in model (79).

Then one fairly direct way to get this posterior (e.g., Bernardo and Smith, 1994) is to (a) introduce a second function of the parameters, say $\eta = g_2(\mu, \sigma^2)$, such that the mapping $f = (g_1, g_2)$ from (μ, σ^2) to (λ, η) is invertible; (b) compute the joint posterior for (λ, η) through the usual change-of-variables formula

$$p(\lambda, \eta|y) = p_{\mu, \sigma^2}\big[f^{-1}(\lambda, \eta)|y\big] \; |J_{f^{-1}}(\lambda, \eta)| \, , \tag{2.94}$$

where $p_{\mu,\sigma^2}(\cdot, \cdot|y)$ is the joint posterior for μ and σ^2 and $|J_{f^{-1}}|$ is the determinant of the Jacobian of the inverse transformation; and (c) marginalize in λ by integrating out η in $p(\lambda, \eta|y)$, in a manner analogous to (92).

Here, for instance, $\eta = g_2(\mu, \sigma^2) = \mu$ would create an invertible f, with inverse defined by $(\mu = \eta, \sigma^2 = \lambda^2 \eta^2)$; the Jacobian determinant comes out $2\lambda\eta^2$ and (94) becomes $p(\lambda, \eta | y) = 2\lambda\eta^2 \, p_{\mu,\sigma^2}(\eta, \lambda^2\eta^2 | y)$.

This process involves two integrations, one (of dimension p) to get the normalizing constant that defines (94) and one (of dimension $(p-1)$) to get rid of η.

You can see that when p is a lot bigger than 2 all these integrals may create severe computational problems—this has been the big stumbling block for applied Bayesian work for a long time.

More than 200 years ago Laplace (1774)—perhaps the second applied Bayesian in history (after Bayes himself)—developed, as one avenue of solution to this problem, what people now call Laplace approximations to high-dimensional integrals of the type arising in Bayesian calculations (see, e.g., Tierney and Kadane, 1986).

Starting in the next case study after this one, we'll use another, computationally intensive, simulation-based approach: Markov chain Monte Carlo (MCMC).

Back to model (79). The conjugate prior for $\theta = (\mu, \sigma^2)$ in this model (e.g., Gelman et al., 2003) turns out to be most simply described hierarchically:

$$\begin{aligned} \sigma^2 &\sim \text{SI-}\chi^2(\nu_0, \sigma_0^2) \\ (\mu | \sigma^2) &\sim N\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right). \end{aligned} \tag{2.95}$$

Here saying that $\sigma^2 \sim \text{SI-}\chi^2(\nu_0, \sigma_0^2)$, where SI stands for scaled inverse, amounts to saying that the precision $\tau = \frac{1}{\sigma^2}$ follows a scaled χ^2 distribution with parameters ν_0 and σ_0^2.

The scaling is chosen so that σ_0^2 can be interpreted as a prior estimate of σ^2, with ν_0 the prior sample size of this estimate (i.e., think of a prior data set with ν_0 observations and sample SD σ_0).

Since χ^2 is a special case of the Gamma distribution, SI-χ^2 must be a special case of the inverse Gamma family—its density (see Gelman et al., 2003, Appendix A) is

$$\begin{aligned} \sigma^2 &\sim \text{SI-}\chi^2(\nu_0, \sigma_0^2) \leftrightarrow \qquad (2.96) \\ p(\sigma^2) &= \frac{\left(\frac{1}{2}\nu_0\right)^{\frac{1}{2}\nu_0}}{\Gamma\left(\frac{1}{2}\nu_0\right)} \left(\sigma_0^2\right)^{\frac{1}{2}\nu_0} \left(\sigma^2\right)^{-\left(1+\frac{1}{2}\nu_0\right)} \exp\left(\frac{-\nu_0 \, \sigma_0^2}{2\sigma^2}\right). \end{aligned}$$

As may be verified with `Maple`, this distribution has mean (provided that $\nu_0 > 2$) and variance (provided that $\nu_0 > 4$) given by

$$E\left(\sigma^2\right) = \frac{\nu_0}{\nu_0 - 2}\sigma_0^2 \quad \text{and} \quad V\left(\sigma^2\right) = \frac{2\nu_0^2}{(\nu_0-2)^2(\nu_0-4)}\sigma_0^4. \tag{2.97}$$

The parameters μ_0 and κ_0 in the second level of the prior model (95), $(\mu|\sigma^2) \sim N\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right)$, have simple parallel interpretations to those of σ_0^2 and ν_0: μ_0 is the prior estimate of μ, and κ_0 is the prior effective sample size of this estimate.

The likelihood function in model (79), with both μ and σ^2 unknown, is

$$\begin{aligned} l(\mu, \sigma^2|y) &= c \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(y_i-\mu)^2\right] \\ &= c \left(\sigma^2\right)^{-\frac{1}{2}n} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i-\mu)^2\right] \\ &= c \left(\sigma^2\right)^{-\frac{1}{2}n} \exp\left[-\frac{1}{2\sigma^2}\left(\sum_{i=1}^n y_i^2 - 2\,\mu\sum_{i=1}^n y_i + n\mu^2\right)\right]. \end{aligned} \tag{2.98}$$

The expression in brackets in the last line of (98) is

$$\begin{aligned} [\ \cdot\ ] &= -\frac{1}{2\sigma^2}\left[\sum_{i=1}^n y_i^2 + n(\mu-\bar{y})^2 - n\bar{y}^2\right] \\ &= -\frac{1}{2\sigma^2}\left[n(\mu-\bar{y})^2 + (n-1)s^2\right], \end{aligned} \tag{2.99}$$

where $s^2 = \frac{1}{n-1}\sum_{i=1}^n (y_i - \bar{y})^2$ is the sample variance. Thus

$$l(\mu, \sigma^2|y) = c \left(\sigma^2\right)^{-\frac{1}{2}n} \exp\left\{-\frac{1}{2\sigma^2}\left[n(\mu-\bar{y})^2 + (n-1)s^2\right]\right\},$$

and it's clear that the vector $(\bar{y}, s^2)$ is sufficient for $\theta = (\mu, \sigma^2)$ in this model, i.e., $l(\mu, \sigma^2|y) = l(\mu, \sigma^2|\bar{y}, s^2)$.

`Maple` can be used to make 3D and contour plots of this likelihood function with the NB10 data:

```
> l := ( mu, sigma2, ybar, s2, n ) -> sigma2^( - n / 2 ) *
    exp( - ( n * ( mu - ybar )^2 + ( n - 1 ) * s2 ) /
    ( 2 * sigma2 ) );

l := (mu, sigma2, ybar, s2, n) ->


                                           2
          (- 1/2 n)           n (mu - ybar)  + (n - 1) s2
    sigma2          exp(- 1/2 ----------------------------)
                                        sigma2

> plotsetup( x11 );

> plot3d( l( mu, sigma2, 404.6, 42.25, 100 ), mu = 402.6 .. 406.6,
    sigma2 = 25 .. 70 );
```

You can use the mouse to rotate 3D plots and get other useful views of them:

The projection or shadow plot of μ looks a lot like a normal (or maybe a t) distribution.

And the shadow plot of σ^2 looks a lot like a Gamma (or maybe an inverse Gamma) distribution.

```
> plots[ contourplot ]( 10^100 * l( mu, sigma2, 404.6, 42.25,
    100 ), mu = 402.6 .. 406.6, sigma2 = 25 .. 70,
    color = black );
```

The contour plot shows that μ and σ^2 are uncorrelated in the likelihood distribution, and the skewness of the marginal distribution of σ^2 is also evident.

Posterior analysis. Having adopted the conjugate prior (95), what I'd like next is simple expressions for the marginal posterior distributions $p(\mu|y)$ and $p(\sigma^2|y)$ and for predictive distributions like $p(y_{n+1}|y)$.

Fortunately, in model (79) all of the integrations (such as (92) and (93)) may be done analytically (see, e.g., Bernardo and Smith 1994), yielding the following results:

$$\begin{aligned} (\sigma^2|y,\mathcal{G}) &\sim \text{SI-}\chi^2(\nu_n,\sigma_n^2), \\ (\mu|y,\mathcal{G}) &\sim t_{\nu_n}\left(\mu_n,\frac{\sigma_n^2}{\kappa_n}\right), \quad \text{and} \end{aligned} \tag{2.100}$$

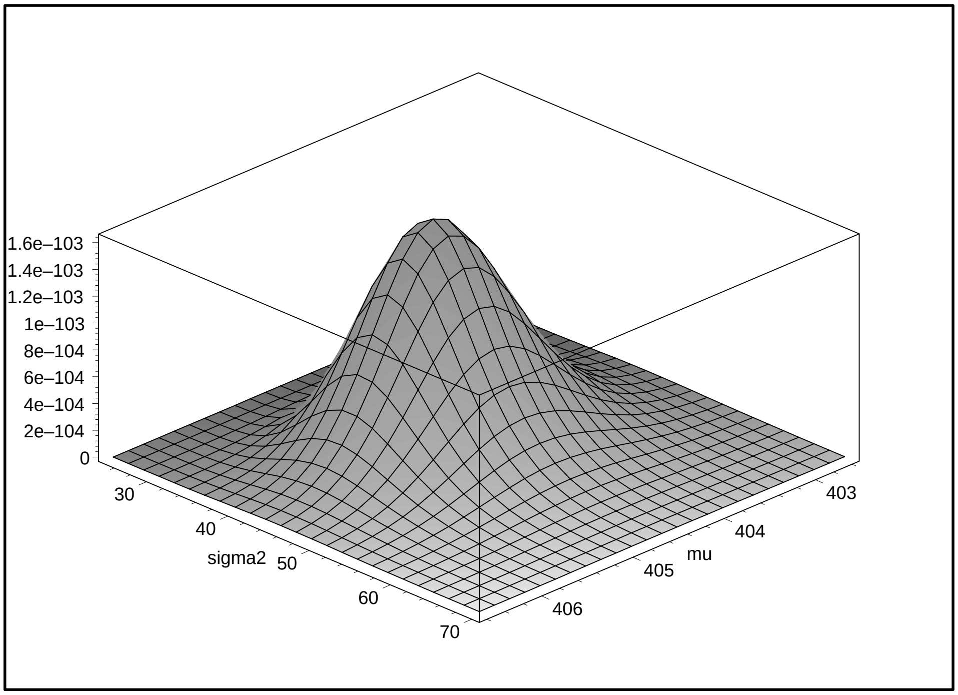


Figure 2.24: *3D plot of the Gaussian likelihood function with both μ and σ^2 unknown.*

$$(y_{n+1}|y,\mathcal{G}) \;\sim\; t_{\nu_n}\left(\mu_n, \frac{\kappa_n+1}{\kappa_n}\sigma_n^2\right).$$

NB10 Gaussian analysis. In the above expressions

$$\begin{aligned} \nu_n &= \nu_0 + n, \\ \sigma_n^2 &= \frac{1}{\nu_n}\left[\nu_0\sigma_0^2 + (n-1)s^2 + \frac{\kappa_0 n}{\kappa_0+n}(\bar{y}-\mu_0)^2\right], \\ \mu_n &= \frac{\kappa_0}{\kappa_0+n}\,\mu_0 + \frac{n}{\kappa_0+n}\,\bar{y}, \quad \text{and} \\ \kappa_n &= \kappa_0 + n, \end{aligned} \tag{2.101}$$

$\bar{y}$ and s^2 are the usual sample mean and variance of y, and $\mathcal{G}$ denotes the assumption of the Gaussian model.

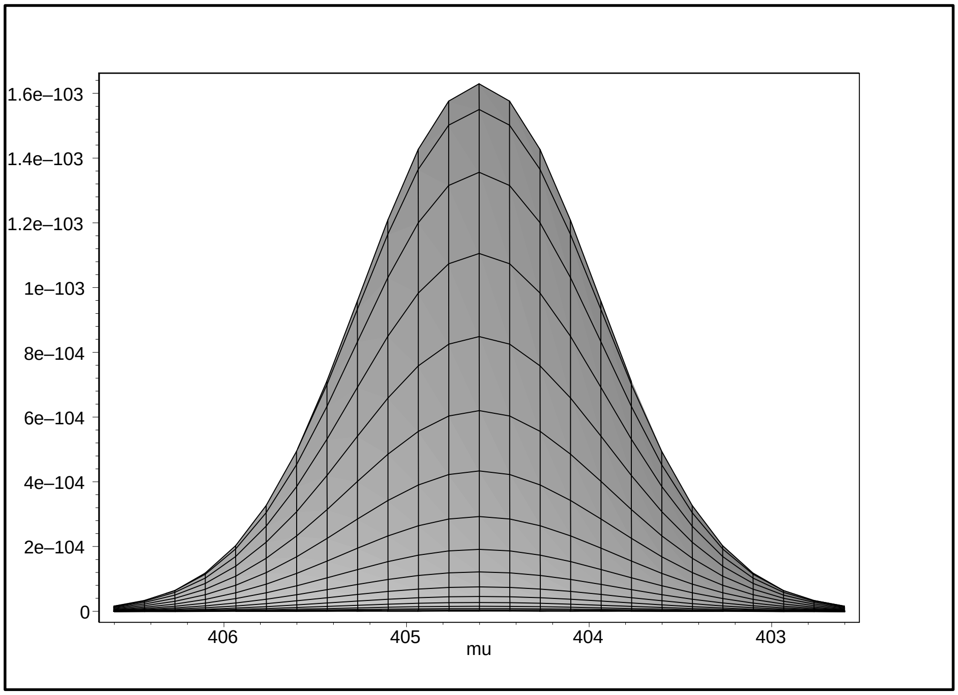


Figure 2.25: *Shadow plot of the Gaussian likelihood along the μ axis.*

Here $t_\nu(\mu, \sigma^2)$ is a scaled version of the usual t_ν distribution, i.e., $W \sim t_\nu(\mu, \sigma^2) \iff \frac{W-\mu}{\sigma} \sim t_\nu$.

The scaled t distribution (see, e.g., Gelman et al., 2003, Appendix A) has density

$$\eta \sim t_\nu(\mu, \sigma^2) \leftrightarrow p(\eta) = \frac{\Gamma\left[\frac{1}{2}(\nu+1)\right]}{\Gamma\left(\frac{1}{2}\nu\right)\sqrt{\nu\pi\sigma^2}} \left[1 + \frac{1}{\nu\sigma^2}(\eta-\mu)^2\right]^{-\frac{1}{2}(\nu+1)} . \qquad (2.102)$$

This distribution has mean μ (as long as $\nu > 1$) and variance $\frac{\nu}{\nu-2}\sigma^2$ (as long as $\nu > 2$).

Notice that, as with all previous conjugate examples, the posterior mean is again a weighted average of the prior mean and data mean, with weights determined by the prior sample size and the data sample size:

$$\mu_n = \frac{\kappa_0}{\kappa_0 + n}\,\mu_0 + \frac{n}{\kappa_0 + n}\,\bar{y}. \qquad (2.103)$$

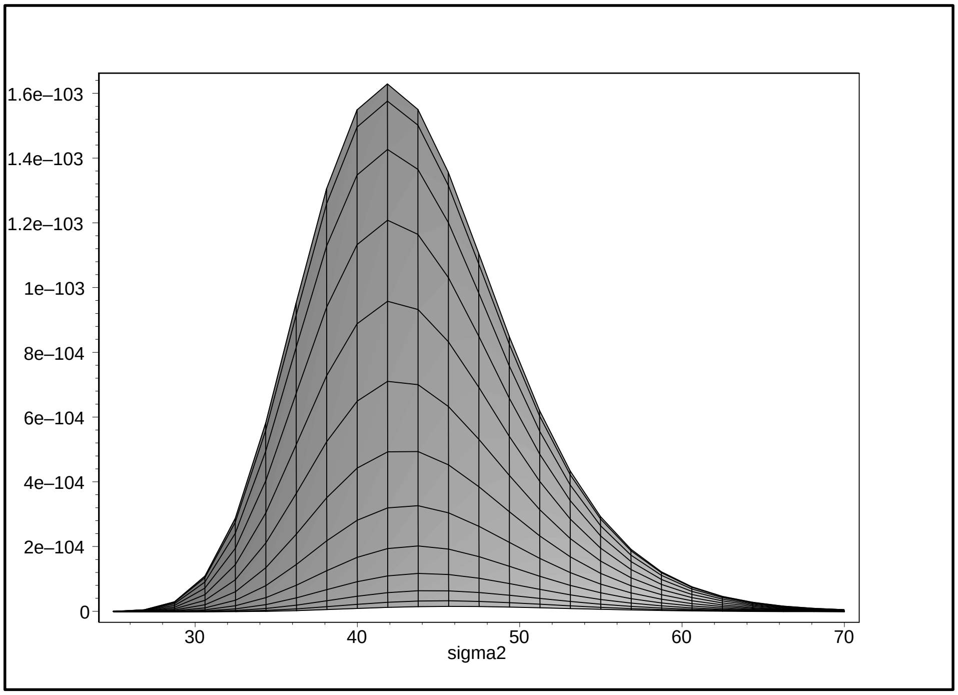


Figure 2.26: *Shadow plot of the Gaussian likelihood along the σ^2 axis.*

NB10 Gaussian Analysis. *Question (a):* I don't know anything about what NB10 is supposed to weigh (down to the nearest microgram) or about the accuracy of the NBS's measurement process, so I want to use a diffuse prior for μ and σ^2.

Considering the meaning of the hyperparameters, to provide little prior information I want to choose both ν_0 and κ_0 close to 0.

Making them exactly 0 would produce an improper prior distribution (which doesn't integrate to 1), but choosing positive values as close to 0 as you like yields a proper and highly diffuse prior.

You can see from (100, 101) that the result is then

$$(\mu|y,\mathcal{G}) \sim t_n\left[\bar{y}, \frac{(n-1)s^2}{n^2}\right] \doteq N\left(\bar{y}, \frac{s^2}{n}\right), \tag{2.104}$$

i.e., with diffuse prior information (as with the Bernoulli model in the AMI

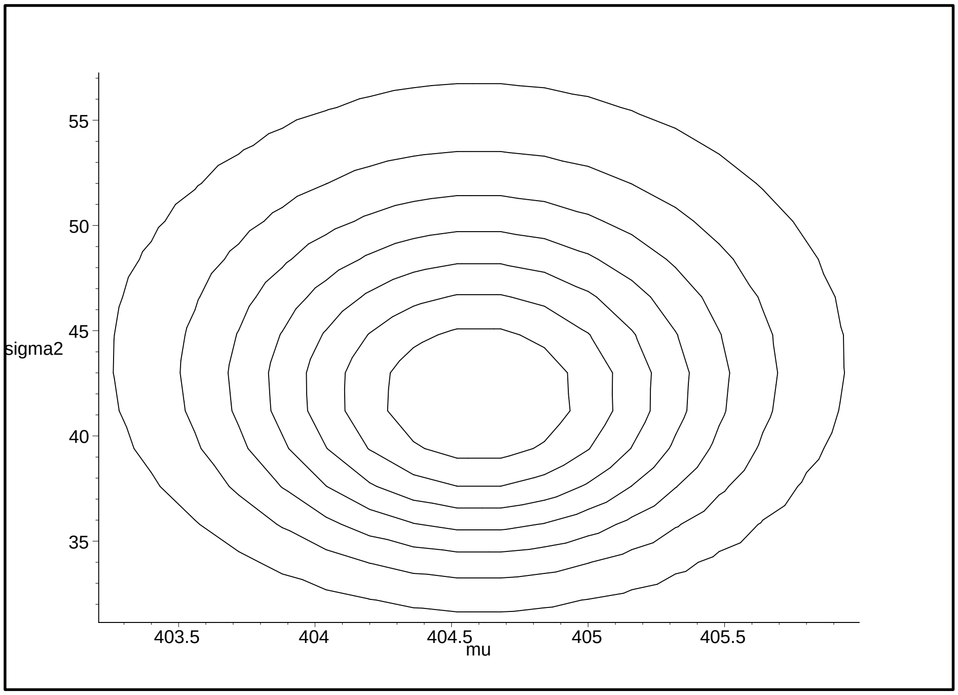


Figure 2.27: *Contour plot of the Gaussian likelihood, which is like looking at Figure 2.24 from above.*

case study) the 95% central Bayesian interval virtually coincides with the usual frequentist 95% confidence interval $\bar{y} \pm t_{n-1}^{.975} \frac{s}{\sqrt{n}} = 404.6 \pm (1.98)(0.647) = (403.3, 405.9)$.

Thus both {frequentists who assume $\mathcal{G}$} and {Bayesians who assume $\mathcal{G}$ with a diffuse prior} conclude that NB10 weighs about 404.6μg below 10g, give or take about 0.65μg.

Question (b). If interest focuses on whether NB10 weighs less than some value like 405.25, when reasoning in a Bayesian way you can answer this question directly: the posterior distribution for μ is shown below, and $P_B(\mu < 405.25 | y, \mathcal{G}, \text{diffuse prior}) \doteq .85$, i.e., your betting odds in favor of the proposition that $\mu < 405.25$ are about 5.5 to 1.

When reasoning in a frequentist way $P_F(\mu < 405.25)$ is undefined; about

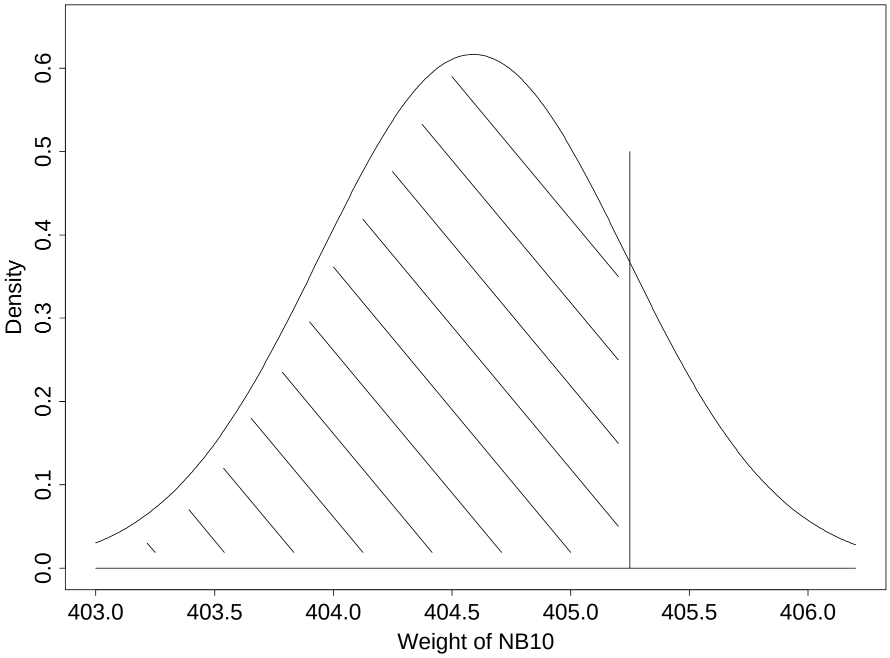


Figure 2.28: *Marginal posterior distribution for μ, with shaded region corresponding to $P_B(\mu < 405.25 \,|\, y, \mathcal{G}$, diffuse prior).*

the best you can do is to test $H_0 \colon \mu < 405.25$, for which the p-value would (approximately) be $p = P_{F,\mu=405.25}(\bar{y} > 405.59) = 1 - .85 = .15$, i.e., insufficient evidence to reject H_0 at the usual significance levels (note the connection between the p-value and the posterior probability, which arises in this example because the null hypothesis is one-sided).

NB The significance test tries to answer a different question: in Bayesian language it looks at $P(\bar{y}|\mu)$ instead of $P(\mu|\bar{y})$.

Many people find the latter quantity more interpretable.

Question (c). We saw earlier that in this model

$$(y_{n+1}|y, \mathcal{G}) \sim t_{\nu_n}\left[\mu_n, \frac{\kappa_n + 1}{\kappa_n}\sigma_n^2\right], \qquad (2.105)$$

and for n large and ν_0 and κ_0 close to 0 this is $(y_{n+1}|y, \mathcal{G}) \stackrel{\cdot}{\sim} N(\bar{y}, s^2)$, i.e., a 95% posterior predictive interval for y_{n+1} is $(392, 418)$.

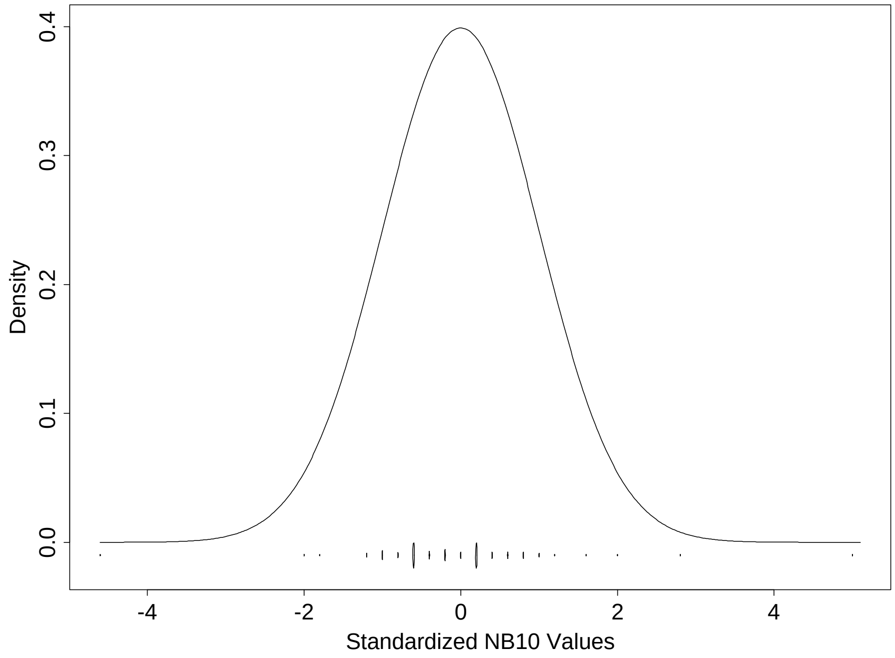


Figure 2.29: *Standardized predictive distribution* $p(y_{n+1} \,|\, y, \mathcal{G}, \textit{diffuse prior})$*, with the standardized data values superimposed.*

Model expansion. A standardized version of this predictive distribution is plotted below, with the standardized NB10 data values superimposed.

It's evident from this plot (and also from the normal qqplot given earlier) that the Gaussian model provides a poor fit for these data—the three most extreme points in the data set in standard units are $-4.6, 2.8$, and 5.0.

With the symmetric heavy tails indicated in these plots, in fact, the empirical CDF looks quite a bit like that of a t distribution with a rather small number of degrees of freedom.

This suggests revising the previous model by expanding it: embedding the Gaussian in the t family and adding a parameter k for tail-weight.

Unfortunately there's no standard closed-form conjugate choice for the prior on k.

A more flexible approach to computing is evidently needed—this is where Markov chain Monte Carlo methods (our next main topic) come in.

Postscript on the exponential family. Two more examples of the use of the exponential family:

(1) An example of a non-regular exponential family: suppose (as in the case study in homework 3 problem 2) that a reasonable model for the data is to take the observed values $(y_i|\theta)$ to be conditionally IID from the uniform distribution $U(0,\theta)$ on the interval $(0,\theta)$ for unknown θ:

$$p(y_1|\theta) = \left\{ \begin{array}{cc} \frac{1}{\theta} & \text{for } 0 < y_1 < \theta \\ 0 & \text{otherwise} \end{array} \right\} = \frac{1}{\theta} I(0,\theta), \qquad (2.106)$$

where $I(A) = 1$ if A is true and 0 otherwise.

θ in this model is called a range-restriction parameter; such parameters are fundamentally different from location and scale parameters (like the mean μ and variance σ^2 in the $N(\mu,\sigma^2)$ model, respectively) or shape parameters (like the degrees of freedom ν in the t_ν model).

(106) is an example of (51) with $c = 1, f_1(y) = 1, g_1(\theta) = \frac{1}{\theta}, h_1(y) = 0$, and $\phi_1(\theta) =$ anything you want (e.g., 1), but only when the set $\mathcal{Y} = (0,\theta)$ is taken to depend on θ.

(Truncated distributions with unknown truncation point also lead to non-regular exponential families.)

As you'll see in homework 3, inference in non-regular exponential families is similar in some respects to the story when the exponential family is regular, but there are some important differences too.

(2) For an example with $p > 1$, take $\theta = (\mu,\sigma^2)$ with the Gaussian likelihood:

$$\begin{aligned} l(\theta|y) &= \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2}(y_i-\mu)^2\right] \qquad (2.107) \\ &= c\left(\sigma^2\right)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2}\left(\sum_{i=1}^n y_i^2 \right.\right. \\ &\qquad \left.\left. -2\mu\sum_{i=1}^n y_i + n\mu^2\right)\right]. \end{aligned}$$

This is of the form (2.53) with $k = 2,\ f(y) = 1, g(\theta) = (\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{n\mu^2}{2\sigma^2}\right)$, $\phi_1(\theta) = -\frac{1}{2\sigma^2}, \phi_2(\theta) = \frac{\mu}{\sigma^2}, h_1(y_i) = y_i^2$, and $h_2(y_i) = y_i$, which shows that $[h_1(y) = \sum_{i=1}^n y_i^2, h_2(y) = \sum_{i=1}^n y_i]$ or equivalently $(\bar{y}, s^2)$ is sufficient for θ.

Some unpleasant algebra then demonstrates that an application of the conjugate prior theorem (54) in the exponential family leads to (95) as the conjugate prior for the Gaussian likelihood when both μ and σ^2 are unknown.

2.6 Appendix on hypothesis testing

Setup: Controlled (phase III) clinical trial of new versus old treatment, with n (human) subjects randomized, $\frac{n}{2}$ to old, $\frac{n}{2}$ to new, n (fairly) large.

$\theta =$ the mean difference (new $-$ old), on the most important outcome of interest (scaled [without loss of generality] so that large values are better than small), in the population $\mathcal{P}$ of subjects judged exchangeable with those in the trial.

(This is like imagining that the n trial subjects were randomly sampled from $\mathcal{P}$ [of course this is typically not how subjects are actually enlisted in the trial] and then randomized to new or old, which gives θ a causal interpretation as the mean improvement per person caused by receiving the new treatment instead of the old.)

As Spiegelhalter et al. note (Section 4.2), two frequentist schools of inference about θ developed in the twentieth century:

- The Fisherian approach, which has two parts:

(a) Point and interval estimates of θ based on the likelihood function; and

(b) Summarization of the evidence against a null hypothesis like H_0: $\theta = 0$ via P-values (the chance, if the null is true, of getting data as extreme as, or more extreme than, what you got).

- The Neyman-Pearson approach, which also has two parts:

(c) Testing H_0: $\theta = 0$ against H_1: $\theta \neq 0$ by developing rules (as a function of n) that reject H_0 with a pre-specified Type I error probability α (the chance of incorrectly rejecting H_0), and then (having first specified α) choosing n so that the Type II error probability β (the chance of incorrectly failing to reject H_0) is no more than some pre-specified threshold when θ actually is some pre-specified positive value θ_1 (this is equivalent to choosing n so that the power $(1 - \beta)$ of the test is not less than a pre-specified threshold when $\theta = \theta_1$); and

(d) Constructing a confidence interval for θ with some pre-specified confidence level $100(1 - \gamma)\%$.

As Spiegelhalter et al. note, in practice a combined frequentist approach has somehow evolved in which clinical trials are often designed from the

Neyman-Pearson point of view (c) but then summarized with Fisherian P-values (b) as measures of evidence against H_0.

From a Bayesian point of view this approach perversely emphasizes the worst of both the Fisherian and Neyman-Pearson schools, by failing to focus on the most scientifically relevant summary of any given trial: an (interval) estimate of θ on the scale of the clinically most important outcome variable (recall de Finetti's Bayesian emphasis on predicting data values on the scales on which they're measured).

A good rule of thumb: don't wander off onto the probability scale (as P-values do) when you can stay on the data scale (as interval estimates do), because it's harder to think about whether probabilities are important scientifically ("Is $P = 0.03$ small enough?") than it is to think about whether changes on the main outcome scale of interest are scientifically relevant ("Would it positively affect this hypertensive patient's health for her mean systolic blood pressure over time to go down by 10 mmHg?").

Standard example: I've run my trial and the P-value comes out 0.02, which is "small enough to publish"; but can I tell from this whether the difference I've found is clinically meaningful?

In a two-tailed test of H_0: $\theta = 0$ against H_1: $\theta \neq 0$ I can work backwards from $P = 0.02$ to figure out that the value of the standard test statistic

$$z = \frac{\overline{\text{new}} - \overline{\text{old}}}{\widehat{SE}\left(\overline{\text{new}} - \overline{\text{old}}\right)} \tag{2.108}$$

that gave rise to $P = 0.02$ was ± 2.3 (taking n to be large), but (1) I can't even tell from the P-value whether the new treatment was better or worse than the old, (2) the thing I really want to know to judge the practical significance of this finding is the numerator of (108), (3) the thing I really want to know to judge the statistical significance of this finding is the denominator of (108), and (4) the P-value has thrown away crucial information by (in effect) specifying only the ratio of (2) and (3) rather than their separate, and separately important, values.

If I have to work out the numerator and denominator of (108) separately to pin down both the practical and statistical significance of my result, both of which are key scientific summaries, then what's the point of calculating P at all?

Why not dispense with it altogether and go directly to the (e.g., 95%) interval estimate

$$\left(\overline{\text{new}} - \overline{\text{old}}\right) \pm 2\,\widehat{SE}\left(\overline{\text{new}} - \overline{\text{old}}\right)? \tag{2.109}$$

(This is a large-n approximation to the Bayesian solution to the inference problem when prior information is diffuse.)

For me the above argument demolishes the use of P-values in inference (although in part 4 I will make better use of them in diagnostic checking of a statistical model, which is another task altogether.)

The Fisherian point and interval estimates (a) and the Neyman-Pearson confidence intervals (d) are much more in keeping with the scientifically compelling idea of staying on the data scale, but they have the following two drawbacks in relation to the Bayesian approach:

- They fail to incorporate relevant prior information about θ when it's available, and

- They don't necessarily work very well (i.e., they don't necessarily live up to their advertised frequentist properties) when the likelihood function is heavily skewed and/or when n is small.

As Spiegelhalter et al. note (section 6.3), the standard testing of H_0: $\theta = 0$ against $H_1 : \theta \neq 0$ is often naive in a realistic clinical trial setting: (to paraphrase these authors) increased costs, toxicity, etc. will often mean that a particular level of mean improvement θ_S would be necessary for the new treatment to be considered clinically superior, and the new treatment will often not be considered clinically inferior unless the true benefit were less than some threshold θ_I.

To paraphrase Spiegelhalter et al., the Bayesian interval-estimation approach then leads to one of six conclusions, corresponding to the six possibilities in the figure on the previous page:

A: We are confident that the old treatment is clinically superior (if, say, the 95% central posterior interval for θ lies entirely below θ_I);

B: The new treatment is not superior, but the treatments could be clinically equivalent;

C: We are substantially uncertain as to the two treatments ("equipoise");

$C+$: We are confident that the two treatments are clinically equivalent;

D: The old treatment is not superior, but the treatments could be clinically equivalent;

E: We are confident that the new treatment is clinically superior.

This leads to an arguably better approach to sample size determination in experimental design than the usual way the Neyman-Pearson significance-level-and-power approach is employed: plan your experiment so that if many investigators were to compare new and old using the same protocol, a high percentage of them (this corresponds to $(1-\alpha)$) would find that their central

95% posterior intervals for θ lie in region E if indeed the true value of θ is some pre-specified $\theta^* > \theta_S$ (this corresponds to $(1 - \beta)$).

Because θ^* will tend to be larger than typical values of θ_1 in the Neyman-Pearson part (c) approach above, this will require increased sample sizes, but this will help to combat the empirical (and regrettable) tendency of the Neyman-Pearson part (c) approach to produce false positives rather more often than we would like (Spiegelhalter et al., Section 3.10):

2.7 Problems

1. (the Exchange Paradox) You are playing the following game against an opponent, with a referee also taking part. The referee has two envelopes (numbered 1 and 2 for the sake of this problem, but when the game is played the envelopes have no markings on them), and (without you or your opponent seeing what she does) she puts $\$m$ in envelope 1 and $\$2m$ in envelope 2 for some $m > 0$ (treat m as continuous in this problem even though in practice it would have to be rounded to the nearest dollar or penny). You and your opponent each get one of the envelopes at random. You open your envelope secretly and find $\$x$ (your opponent also looks secretly in her envelope), and the referee then asks you if you want to trade envelopes with your opponent. You reason that if you trade, you will get either $\$\frac{x}{2}$ or $\$2x$, each with probability $\frac{1}{2}$. This makes the expected value of the amount of money you'll get if you trade equal to $\left(\frac{1}{2}\right)\left(\frac{\$x}{2}\right) + \left(\frac{1}{2}\right)(\$2x) = \frac{\$5x}{4}$, which is greater than the $\$x$ you currently have, so you offer to trade. The paradox is that your opponent is capable of making exactly the same calculation. How can the trade be advantageous for both of you?

   The point of this problem is to demonstrate that the above reasoning is flawed from a Bayesian point of view; the conclusion that trading envelopes is always optimal is based on the assumption that there is no information obtained by observing the contents of the envelope you get, and this assumption can be seen to be false when you reason in a Bayesian way. At a moment in time before the game begins, let $p(m)$ be your prior distribution on the amount of money M the referee will put in envelope 1, and let X be the amount of money you will find in your envelope when you open it (when the game is actually played, the observed x, of course, will be data that can be used to decrease your uncertainty about M).

(a) Explain why the setup of this problem implies that $P(X = m|M = m) = P(X = 2m|M = m) = \frac{1}{2}$, and use this to show that

$$\begin{aligned} P(M = x|X = x) &= \frac{p(x)}{p(x) + p\left(\frac{x}{2}\right)} \quad \text{and} \\ P\left(M = \frac{x}{2} \middle| X = x\right) &= \frac{p\left(\frac{x}{2}\right)}{p(x) + p\left(\frac{x}{2}\right)}. \end{aligned} \tag{2.110}$$

Demonstrate from this that the expected value of the amount Y of money in your opponent's envelope, given than you've found \$$x$ in the envelope you've opened, is

$$E(Y|X = x) = \frac{p(x)}{p(x) + p\left(\frac{x}{2}\right)} 2x + \frac{p\left(\frac{x}{2}\right)}{p(x) + p\left(\frac{x}{2}\right)} \frac{x}{2}. \tag{2.111}$$

(b) Suppose that for you in this game, money and utility coincide (or at least suppose that utility is linear in money for you with a positive slope). Use Bayesian decision theory, through the principle of maximizing expected utility, to show that you should offer to trade envelopes only if

$$p\left(\frac{x}{2}\right) < 2p(x). \tag{2.112}$$

If you and two friends (one of whom would serve as the referee) were to actually play this game with real money in the envelopes, it would probably be the case that small amounts of money are more likely to be chosen by the referee than big amounts, which makes it interesting to explore condition (3) for prior distributions that are decreasing (that is, for which $p(m_2) < p(m_1)$ for $m_2 > m_1$). Make a sketch of what condition (3) implies for a decreasing p. One possible example of a continuous decreasing family of priors on M is the *exponential* distribution, with density (4) below, indexed by the parameter λ which represents the mean of the distribution. Identify the set of conditions in this family of priors, as a function of x and λ, under which it's optimal for you to trade. Does the inequality you obtain in this way make good intuitive sense (in terms of both x and λ)? Explain briefly.

Extra credit: Looking carefully at the correct argument in paragraph 2 of this problem, identify precisely the point at which the argument in the first paragraph breaks down, and specify what someone who believes the argument in paragraph 1 is implicitly assuming about p.

2. (a) (exchangeability) Suppose Y_1 and Y_2 are identically distributed Bernoulli random variables with success probability $0 < \theta < 1$. Show that independence of Y_1 and Y_2 implies exchangeability but not conversely. The simplest way to do this is to specify their joint distribution by making a 2×2 table cross-tabulating Y_1 against Y_2, labeling all of the probabilities symbolically. What does this table have to look like in terms of θ if Y_1 and Y_2 are independent? What about when they're exchangeable? (In the latter case you'll have to choose a new symbol for some of the relevant probabilities.)

   Extra credit: See if you can quantify how far away from independence Y_1 and Y_2 can be (in some sense of distance in the space of possible joint distributions) and still be exchangeable.

   (b) Can you give another simple example, involving a comparison of random sampling with and without replacement from a finite population, of a set of random variables that are exchangeable but not independent? Explain briefly.

3. (Bayesian conjugate inference in the exponential distribution) In a consulting project that one of my Ph.D. students and I worked on at the University of Bath in England in the late 1990s, a researcher from the Department of Electronic and Electrical Engineering (EEE) at Bath wanted help in analyzing some data on failure times for a particular kind of metal wire (in this problem, failure time was defined to be the number of times the wire could be mechanically stressed by a machine at a given point along the wire before it broke). The $n = 14$ raw data values y_i in one part of his experiment, arranged in ascending order, were

495	541	1461	1555	1603	2201	2750
3468	3516	4319	6622	7728	13159	21194

Probably the simplest model for failure time data is the *exponential* distribution $\mathcal{E}(\lambda)$:

$$(y_i|\lambda) \stackrel{\text{IID}}{\sim} p(y_i|\lambda) = \left\{ \begin{array}{cc} \frac{1}{\lambda}\exp(-\frac{y_i}{\lambda}) & y_i > 0 \\ 0 & \text{otherwise} \end{array} \right\} \tag{2.113}$$

for some $\lambda > 0$. (**NB** This distribution can be parameterized either in terms of λ or $\frac{1}{\lambda}$; whenever it occurs in print you need to be careful which parameterization is in use.)

(a) To see if this model fits the data above, you can make an *exponential probability plot*, analogous to a Gaussian quantile-quantile plot to check for normality. In fact the idea works for more or less any distribution: you plot

$$y_{(i)} \quad \text{versus} \quad F^{-1}\left(\frac{i-0.5}{n}\right), \tag{2.114}$$

where $y_{(i)}$ are the sorted y values and F is the CDF of the distribution (the 0.5 is in there to avoid problems at the edges of the data). In so doing you're graphing the data values against an approximation of *what you would have expected for the data values if the CDF of the* y_i *really had been* F, so the plot should resemble the 45° line if the fit is good.

(i) Show that the inverse CDF of the $\mathcal{E}(\lambda)$ distribution (parameterized as in equation (4)) is given by

$$F_Y(y|\lambda) = p \iff y = F^{-1}(p) = -\lambda \log(1-p). \tag{2.115}$$

(ii) To use equation (6) to make the plot we need a decent estimate of λ. Show that the maximum likelihood estimate of λ in this model is $\hat{\lambda}_{\text{MLE}} = \bar{y}$, the sample mean, and use this (in `Maple`, or freehand, or with whatever other software you might like) to make an exponential probability plot of the 14 data values above. Informally, does the exponential model appear to provide a good fit to the data? Explain briefly.

(b) (exponential family and conjugate prior; prior to posterior updating)

(i) Show that the exponential sampling model (4) is a member of the one-parameter exponential family, and use this to show that the conjugate family for the $\mathcal{E}(\lambda)$ likelihood (parameterized as in (4)) is the set of *Inverse Gamma* distributions $\Gamma^{-1}(\alpha, \beta)$ for $\alpha > 0, \beta > 0$ (**NB** $W \sim \Gamma^{-1}(\alpha, \beta)$ just means that $\frac{1}{W} \sim \Gamma(\alpha, \beta)$; see Table A.1 from Appendix A in Gelman et al. for details): $\lambda \sim \Gamma^{-1}(\alpha, \beta)$ if and only if

$$p(\lambda) = \left\{ \begin{array}{cc} \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{-(\alpha+1)} \exp\left(-\frac{\beta}{\lambda}\right) & \lambda > 0 \\ 0 & \text{otherwise} \end{array} \right\}. \qquad (2.116)$$

(ii) By directly using Bayes' Theorem (and ignoring constants), show that the prior-to-posterior updating rule in this model is

$$\left\{ \begin{array}{ccc} \lambda & \sim & \Gamma^{-1}(\alpha, \beta) \\ (Y_i|\lambda) & \stackrel{\text{IID}}{\sim} & \mathcal{E}(\lambda) \end{array} \right\} \Longrightarrow (\lambda|y) \sim \Gamma^{-1}(\alpha + n, \beta + n\bar{y}). \qquad (2.117)$$

(iii) It turns out that the mean and variance of the $\Gamma^{-1}(\alpha, \beta)$ distribution are $\frac{\beta}{\alpha-1}$ and $\frac{\beta^2}{(\alpha-1)^2(\alpha-2)}$, respectively (as long as $\alpha > 2$). Use this to write down an explicit formula which shows that the posterior mean is a weighted average of the prior and sample means, and deduce from this formula that $n_0 = (\alpha - 1)$ is the prior effective sample size. Note also from the formula for the likelihood in this problem that, when thought of as a distribution in λ, it's equivalent to a constant times the $\Gamma^{-1}(n - 1, n\bar{y})$ distribution.

(c) The guy from EEE has prior information from another experiment he judges to be comparable to this one: from this other experiment the prior for λ should have a mean of about $\mu_0 = 4500$ and an SD of about $\sigma = 1800$.

(i) Show that this corresponds to a $\Gamma^{-1}(\alpha_0, \beta_0)$ prior with (α_0, β_0) $= (8.25, 32625)$, and therefore to a prior sample size of about 7.

(ii) The next step is to work on filling in the entries in the following table:

		Likelihood		
	Prior	Maximizing	Integrating	Posterior
Mean/Estimate	4500			
SD/SE	1800			

Show that the Fisher information provided by the MLE in this model is

$$\hat{I}\left(\hat{\lambda}_{\text{MLE}}\right) = \frac{n}{\bar{y}^2}, \tag{2.118}$$

so that a large-sample standard error for the MLE is

$$\widehat{SE}\left(\hat{\lambda}_{\text{MLE}}\right) = \frac{\bar{y}}{\sqrt{n}}. \tag{2.119}$$

In the "Likelihood Maximizing" column put the numerical values of the MLE and its standard error; in the "Likelihood Integrating" column put the mean and SD of the likelihood function, interpreted as the $\Gamma^{-1}(n-1, n\bar{y})$ distribution; and in the "Posterior" column put the posterior mean and SD using the (α_0, β_0) and data values above. By examining the formulas for the relevant quantities, show that the discrepancies between the "Likelihood Maximizing" and "Likelihood Integrating" columns in this table will diminish as n increases.

(iii) What kind of compromise, if any, gives rise to the posterior SD as a function of the prior and likelihood SDs, at least approximately? Explain briefly.

(iv) Make a plot with `Maple`, or an approximate freehand sketch, of the prior, likelihood ($\Gamma^{-1}(n-1, n\bar{y})$), and posterior distributions on the same graph, and summarize what all of this has to say about the failure times of the metal wire samples with which the problem began.

(d) (comparing Bayesian and [large-sample] maximum likelihood interval estimates) From the Fisher information calculation above, an approximate 95% interval estimate for λ in this model based on the (large-sample) likelihood approach has the form

$$\bar{y} \pm 1.96 \frac{\bar{y}}{\sqrt{n}}. \tag{2.120}$$

By using the numerical integration features in `Maple` I've computed the endpoints of 95% central intervals based both on the posterior distribution in (c) and the $\Gamma^{-1}(n-1, n\bar{y})$ likelihood distribution, obtaining $(3186, 7382)$ and $(3369, 10201)$, respectively. (**NB** (a, b) is a $100(1-\alpha)\%$ central interval for a real-valued parameter θ with respect to an inferential density $p(\theta)$ if $\int_{-\infty}^{a} p(\theta)\, d\theta$ $= \int_{b}^{\infty} p(\theta)\, d\theta = \frac{\alpha}{2}$.) Compute the (large-sample) likelihood interval (11) above on this dataset and explain briefly why it's not directly comparable to the 95% posterior interval. In what way does your plot of the likelihood function in (c) suggests that the central likelihood interval might be better than interval (11) for a value of n as small as the one in this problem? Explain briefly.

Extra credit: Compute the predictive distribution for the next observation Y_{n+1} given $y = (y_1, \ldots, y_n)$ in model (8). Apply this to the data set on page 2 with the largest observation (21194) set aside, using a diffuse Inverse Gamma prior (e.g., pick that member of the Inverse Gamma family that has mean 1 and precision ϵ for some small ϵ like 0.001, by analogy with the $\Gamma(\epsilon, \epsilon)$ prior), and compute a numerical measure of how surprising this observation is under the exponential model. How strongly, if at all, do your calculations call into question this model for these data? Explain briefly.

4. (Bayesian transformation of variables) Continuing problem 2.3, let's again consider the $n = 14$ failure time values y_i given in the statement of that problem, for which we saw that a reasonable (initial) model is based on the exponential distribution for the y_i,

$$\left\{ \begin{array}{ccc} \lambda & \sim & \Gamma^{-1}(\alpha, \beta) \\ (y_i|\lambda) & \stackrel{\text{IID}}{\sim} & \mathcal{E}(\lambda) \end{array} \right\} \Longrightarrow (\lambda|y) \sim \Gamma^{-1}(\alpha + n, \beta + n\bar{y}). \quad (2.121)$$

Here, as before, (i) $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, (ii) the sampling distribution for the y_i is given by

$$(y_i|\lambda) \stackrel{\text{IID}}{\sim} p(y_i|\lambda) = \left\{ \begin{array}{cc} \frac{1}{\lambda}\exp(-\frac{y_i}{\lambda}) & y_i > 0 \\ 0 & \text{otherwise} \end{array} \right\} \quad (2.122)$$

for some $\lambda > 0$, and (iii) the conjugate prior for λ is

$$\lambda \sim \Gamma^{-1}(\alpha, \beta) \quad \Longleftrightarrow \quad p(\lambda) = \left\{ \begin{array}{cc} \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{-(\alpha+1)}\exp\left(-\frac{\beta}{\lambda}\right) & \lambda > 0 \\ 0 & \text{otherwise} \end{array} \right\}. \quad (2.123)$$

In that problem I mentioned that the exponential model can either be parameterized in terms of λ or $\frac{1}{\lambda}$. In this problem we'll explore what happens when you're more interested in $\eta = g(\lambda) = \frac{1}{\lambda}$ than in λ itself.

(a) Use the **change-of-variables formula** derived below to show that the prior and posterior distributions for η are $\Gamma(\alpha, \beta)$ and $\Gamma(\alpha + n, \beta + n\bar{y})$, respectively (which justifies the name *inverse gamma* for the distribution of λ).

(b) Write out the likelihood function in terms of η instead of λ (just substitute η everywhere you see $\frac{1}{\lambda}$), and use `Maple` (or some other environment of your choosing) to plot the prior, likelihood, and posterior distributions for η on the same graph, using the data and prior values given in problem 3.

(c) Use the fact that the $\Gamma(\alpha, \beta)$ distribution has mean $\frac{\alpha}{\beta}$ and variance $\frac{\alpha}{\beta^2}$ to numerically compute the prior, likelihood, and posterior means and SDs for η (you don't have to give the likelihood-maximizing summaries if you don't want to; it's enough to give results based on the likelihood-integrating approach). Is the posterior mean a weighted average of the prior and data means in this model, and if so what interpretation would you give to α and β in the $\Gamma(\alpha, \beta)$ prior for η? Explain briefly.

The change-of-variables formula. Consider a real-valued continuous random variable Y with CDF $F_Y(y) = P(Y \leq y)$ and density $f_Y(y)$, related as usual to the CDF by $F_Y(y) = \int_{-\infty}^{y} f_Y(t)\, dt$ and $f_Y(y) = \frac{d}{dy} F_Y(y)$. Suppose you're interested mainly in a random variable X which is a transformed version of Y: $X = h(Y)$ for some invertible (strictly monotonic) function h. Such functions have to be either strictly increasing or decreasing; as a first case assume the former. Then the CDF of X, $F_X(x) = P(X \leq x)$, satisfies

$$\begin{aligned} F_X(x) &= P(X \leq x) = P[h(Y) \leq x] \qquad (2.124) \\ &= P[Y \leq h^{-1}(x)] = F_Y\left[h^{-1}(x)\right], \end{aligned}$$

from which the density of X is

$$f_X(x) = \frac{d}{dx} F_X(x) = \frac{d}{dx} F_Y\left[h^{-1}(x)\right] \qquad (2.125)$$

$$= \; f_Y\left[h^{-1}(x)\right] \frac{d}{dx} h^{-1}(x) = f_Y\left[h^{-1}(x)\right] \left| \frac{d}{dx} h^{-1}(x) \right| ,$$

the last equality holding because h, and therefore h^{-1}, are strictly increasing (and therefore both have positive derivatives). Similarly, if h is strictly decreasing,

$$\begin{aligned} F_X(x) &= P(X \le x) = P[h(Y) \le x] \qquad (2.126) \\ &= P[Y \ge h^{-1}(x)] = 1 - F_Y\left[h^{-1}(x)\right] , \end{aligned}$$

from which the density of X is

$$f_X(x) = \frac{d}{dx} F_X(x) = \frac{d}{dx} F_Y\left[h^{-1}(x)\right] = f_Y\left[h^{-1}(x)\right] \left[-\frac{d}{dx} h^{-1}(x) \right] . \qquad (2.127)$$

But since h is strictly decreasing, so is h^{-1}, and both therefore have negative derivatives, so that

$$-\frac{d}{dx} h^{-1}(x) = \left| \frac{d}{dx} h^{-1}(x) \right| . \qquad (2.128)$$

Thus the conclusion is that in either case

$$f_X(x) = f_Y\left[h^{-1}(x)\right] \left| \frac{d}{dx} h^{-1}(x) \right| , \qquad (2.129)$$

which is the **change-of-variables** formula. (Since $y = h^{-1}(x)$, a simple mnemonic for this formula, using a slightly old-fashioned notation for derivatives, is $f_X(x)\,|dx| = f_Y(y)\,|dy|$.)

5. (Inference with the uniform distribution) Paleobotanists estimate the moment in the remote past when a given species became extinct by taking cylindrical, vertical core samples well below the earth's surface and looking for the last occurrence of the species in the fossil record, measured in meters above the point P at which the species was known to have first emerged. Letting $\{y_i, i = 1, \ldots, n\}$ denote a sample of such distances above P at a random set of locations, the model $(y_i|\theta) \stackrel{\text{IID}}{\sim} \text{Uniform}(0, \theta)$ $(*)$ emerges from simple and plausible assumptions. In this model the unknown $\theta > 0$ can be used, through carbon dating, to estimate the species extinction time. This problem is about

Bayesian inference for θ in model $(*)$, and it will be seen that some of our usual intuitions (derived from the Bernoulli, Poisson, and Gaussian case studies) do not quite hold in this case.

The marginal sampling distribution of a single observation y_i in this model may be written

$$p(y_i|\theta) = \left\{ \begin{array}{cc} \frac{1}{\theta} & \text{if } 0 \le y_i \le \theta \\ 0 & \text{otherwise} \end{array} \right\} = \frac{1}{\theta} \, I \left(0 \le y_i \le \theta \right), \tag{2.130}$$

where $I(A) = 1$ if A is true and 0 otherwise.

(a) Use the fact that $\{0 \le y_i \le \theta$ for all $i = 1, \ldots, n\}$ if and only if $\{m = \max \left(y_1, \ldots y_n \right) \le \theta\}$ to show that the likelihood function in this model is

$$l(\theta|y) = \theta^{-n} I(\theta \ge m). \tag{2.131}$$

Briefly explain why this demonstrates that m is sufficient for θ in this model.

(b) As we have seen in this chapter, the maximum likelihood estimator (MLE) of a parameter θ is the value of θ (which will be a function of the data) that maximizes the likelihood function, and this maximization is usually performed by setting the derivative of the likelihood (or log likelihood) function to 0 and solving. Show by means of a rough sketch of the likelihood function in (a) that m is the maximum likelihood estimator (MLE) of θ, and briefly explain why the usual method for finding the MLE fails in this case.

(c) A positive quantity θ follows the **Pareto** distribution (written $\theta \sim \text{Pareto}(\alpha, \beta)$) if, for parameters $\alpha, \beta > 0$, it has density

$$p(\theta) = \left\{ \begin{array}{cc} \alpha \, \beta^{\alpha} \, \theta^{-(\alpha+1)} & \text{if } \theta \ge \beta \\ 0 & \text{otherwise} \end{array} \right\}. \tag{2.132}$$

This distribution has mean $\frac{\alpha\beta}{\alpha-1}$ (if $\alpha > 1$) and variance $\frac{\alpha\beta^2}{(\alpha-1)^2(\alpha-2)}$ (if $\alpha > 2$). With the likelihood function viewed as (a constant multiple of) a density for θ, show that equation (2.131) corresponds to the Pareto$(n-1, m)$ distribution. Show further that if the prior distribution for θ is taken to be (12), under the model $(*)$ above the posterior distribution is $p(\theta|y) = \text{Pareto}\,[\alpha + n, \max(\beta, m)]$, thereby demonstrating that the Pareto distribution is conjugate to the Uniform$(0, \theta)$ likelihood.

(d) In an experiment conducted in the Antarctic in the 1980s to study a particular species of fossil ammonite, the following was a linearly rescaled version of the data obtained, in ascending order: $y = (y_1, \ldots, y_n) = (0.4, 1.0, 1.5, 1.7, 2.0, 2.1, 2.8, 3.2, 3.7, 4.3, 5.1)$. Prior information equivalent to a Pareto prior specified by the choice $(\alpha, \beta) = (2.5, 4)$ was available. Plot the prior, likelihood, and posterior distributions arising from this data set on the same graph, explicitly identifying the three curves, and briefly discuss what this picture implies about the updating of information from prior to posterior in this case.

(e) Make a table summarizing the mean and standard deviation (SD) for the prior (Pareto(α, β)), likelihood (Pareto$(n-1, m)$), and posterior (Pareto$[\alpha + n, \max(\beta, m)]$) distributions, using the (α, β) choices and the data in part (d) above (as in problem 1, it's enough to do this using the likelihood-integrating approach). In Bayesian updating the posterior mean is usually (at least approximately) a weighted average of the prior and likelihood means (with weights between 0 and 1), and the posterior SD is typically smaller than either the prior or likelihood SDs. Are each of these behaviors true in this case? Explain briefly.

(f) You've shown in (c) that the posterior for θ based on a sample of size n in model $(*)$ is $p(\theta|y) = \text{Pareto}\,[\alpha + n, \max(\beta, m)]$. Write down a symbolic expression for the posterior variance of θ in terms of (α, β, m, n). When considered as a function of n, what's unusual about this expression in relation to the findings in our previous case studies in this course? Explain briefly.

6. (Inference for the variance in the Gaussian model with known mean) As we saw in problem 4 in Chapter 1, American football experts provide a *point spread* for every football game before it occurs, as a measure of the difference in ability between the two teams (and taking account of where the game will be played). For example, if Denver is a 3.5–point favorite to defeat San Francisco, the implication is that betting on whether Denver's final score minus 3.5 points exceeds or falls short

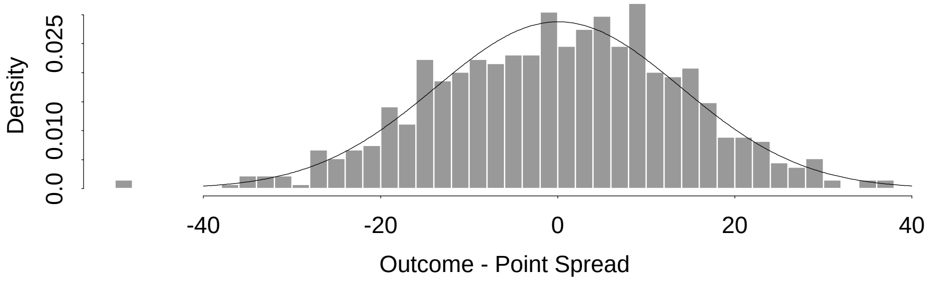


Figure 2.30: *Differences d_i between observed and predicted American football scores, 1981–1984.*

of San Francisco's final score is an even-money proposition. Figure 1 below (based on data from Gelman et al. 2003) presents a histogram of the differences $d =$ (actual outcome – point spread) for a sample of $n = 672$ professional football games in the early 1980s, with a normal density superimposed having the same mean $\bar{d} = 0.07$ and standard deviation (SD) $s = 13.86$ as the sample. You can see from this figure that the model $(D_i|\sigma^2) \overset{\text{IID}}{\sim} N(0, \sigma^2)$ is reasonable for the observed differences d_i (at least as a starting point in the modeling).

(a) Write down the likelihood and log likelihood functions for σ^2 in this model. Show that $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n d_i^2$, which takes the value 191.8 with the data in Figure 1, is both sufficient and the maximum likelihood estimator (MLE) for σ^2. Plot the log likelihood function for σ^2 in the range from 160 to 240 with these data, briefly explaining why it should be slightly skewed to the right.

(b) The conjugate prior for σ^2 in this model is the *scaled inverse chi-square* distribution,

$$\sigma^2 \sim \chi^{-2}(\nu_0, \sigma_0^2), \quad \text{i.e.,} \quad p(\sigma^2) = c\,\left(\sigma^2\right)^{-\left(\frac{\nu_0}{2}+1\right)} \exp\left(-\frac{\nu_0 \sigma_0^2}{2\sigma^2}\right), \tag{2.133}$$

where ν_0 is the prior sample size and σ_0^2 is a prior estimate of σ^2. In an attempt to be "non-informative" people sometimes work

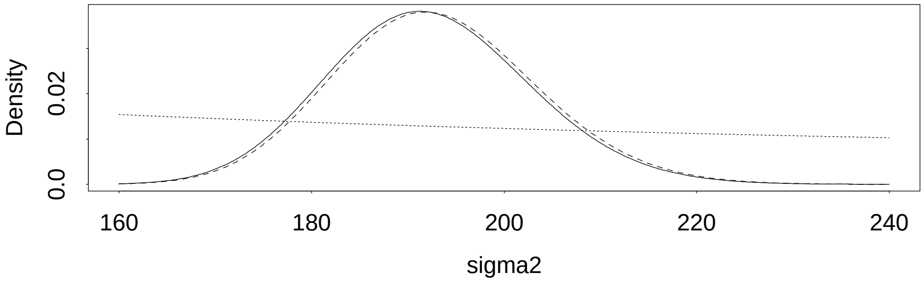


Figure 2.31: *Prior, likelihood, and posterior densities for σ^2 with the football data of Figure 2.30.*

with a version of (13) obtained by letting $\nu_0 \rightarrow 0$, namely $p(\sigma^2) = c_0 \, (\sigma^2)^{-1}$. The resulting prior is *improper* in that it integrates to ∞, but it turns out that posterior inferences will be sensible nonetheless (even with sample sizes as small as $n = 1$). Show that with this prior, the posterior distribution is $\chi^{-2}(n, \hat{\sigma}^2)$.

Figure 2 below plots the prior, likelihood, and posterior densities on the same graph using the data in Figure 1 and taking $c_0 = 2.5$ for convenience in the plot. Get `R` (or some equivalent environment) to reproduce this figure (**NB** `Maple` has a hard time doing this). You'll need to be careful to use the correct normalizing constant c in (13), which can be found either in the lecture notes or in Appendix A of Gelman et al. (2003); and because the data values in this example lead to astoundingly large and small numbers on the original scale, it's necessary to do all possible computations on the log scale and wait to transform back to the original scale until the last possible moment (you'll need to use the built-in function `lgamma` in `R`, or something like it in your favorite environment). Explicitly identify the three curves, and briefly discuss what this plot implies about the updating of information from prior to posterior in this case.

Chapter 3

Simulation-based computation

3.1 IID sampling

Computation via conjugate analysis (part 2 of the lecture notes) produces closed-form results (good) but is limited in scope to a fairly small set of models for which straightforward conjugate results are possible (bad).

This was a severe limitation for Bayesians for almost 250 years (from the 1750s to the 1980s).

Over the past 10 years the Bayesian community has "discovered" and developed an entirely new computing method, Markov chain Monte Carlo (MCMC) ("discovered" because the physicists first figured it out about 50 years ago: Metropolis and Ulam, 1949; Metropolis et al., 1953).

We've seen that the central Bayesian practical challenge is the computation of high-dimensional integrals.

People working on the first atom bomb in World War II faced a similar challenge, and noticed that digital computers (which were then passing from theory (Turing, 1943) to reality) offered an entirely new approach to solving the problem.

The idea (Metropolis and Ulam, 1949) was based on the observation that anything you want to know about a probability distribution can be learned to arbitrary accuracy by sampling from it.

Suppose, for example, that you're interested in a posterior distribution $p(\theta|y)$ which cannot be worked with (easily) in closed form, and initially (to keep things simple) think of θ as a scalar (real number) rather than vector.

Four things of direct interest to you about $p(\theta|y)$ would be

• its mean $\mu = E(\theta|y)$ and standard deviation $\sigma = \sqrt{V(\theta|y)}$,

• its shape (basically you'd like to be able to trace out (an estimate of) the entire density curve), and

• one or more of its quantiles (e.g., to construct a 95% central posterior interval for θ you need to know the 2.5% and 97.5% quantiles, and sometimes the posterior median (the 50th percentile) is of interest too).

Suppose you could take an arbitrarily large random sample from $p(\theta|y)$, say $\theta_1^*, \ldots, \theta_m^*$.

Then each of the above four aspects of $p(\theta|y)$ can be estimated from the θ^* sample:

• $\hat{E}(\theta|y) = \bar{\theta}^* = \frac{1}{m} \sum_{j=1}^m \theta_j^*$,

• $\sqrt{\hat{V}(\theta|y)} = \sqrt{\frac{1}{m-1} \sum_{j=1}^m \left(\theta_j^* - \bar{\theta}^* \right)^2}$,

• the density curve can be estimated by a histogram or kernel density estimate, and

• percentiles can be estimated by counting how many of the θ^* values fall below a series of specified points—e.g., to find an estimate of the 2.5% quantile you solve the equation

$$\hat{F}_\theta(t) = \frac{1}{m} \sum_{j=1}^m I(\theta_j^* \le t) = 0.025 \qquad (3.1)$$

for t, where $I(A)$ is the indicator function (1 if A is true, otherwise 0).

These are called Monte Carlo estimates of the true summaries of $p(\theta|y)$ because they're based on the controlled use of chance.

Theory shows that with large enough m, each of the Monte Carlo (or simulation-based) estimates can be made arbitrarily close to the truth with arbitrarily high probability, under some reasonable assumptions about the nature of the random sampling.

One way to achieve this, of course, is to make the sampling IID (this turns out to be sufficient but not necessary—see below).

If, for example, $\bar{\theta}^* = \frac{1}{m} \sum_{j=1}^m \theta_j^*$ is based on an IID sample of size m from $p(\theta|y)$, we can use the frequentist fact that in repeated sampling $V\left(\bar{\theta}^*\right) = \frac{\sigma^2}{m}$, where (as above) σ^2 is the variance of $p(\theta|y)$, to construct a Monte Carlo standard error (MCSE) for $\bar{\theta}^*$:

$$\widehat{SE}\left(\bar{\theta}^*\right) = \frac{\hat{\sigma}}{\sqrt{m}}, \qquad (3.2)$$

where $\hat{\sigma}$ is the sample SD of the θ^* values.

This can be used, possibly after some preliminary experimentation, to decide on m, the Monte Carlo sample size, which later we'll call the length of the monitoring run.

An IID example. Consider the posterior distribution

$$p(\lambda|y) = \Gamma(\lambda; 29.001, 14.001)$$

in the LOS example in part 2.

We already know that the posterior mean of λ in this example is $\frac{29.001}{14.001} \doteq 2.071$; let's see how well the Monte Carlo method does in estimating this known truth.

Here's an `R` function to construct Monte Carlo estimates of the posterior mean and MCSE values for these estimates.

```
gamma.sim <- function( m, alpha, beta, n.sim, seed ) {

  set.seed( seed )

  theta.out <- matrix( 0, n.sim, 2 )

  for ( i in 1:n.sim ) {

    theta.sample <- rgamma( m, alpha, 1 / beta )

    theta.out[ i, 1 ] <- mean( theta.sample )

    theta.out[ i, 2 ] <- sqrt( var( theta.sample ) / m )

 }

  return( theta.out )

}
```

This function simulates, `n.sim` times, the process of taking an IID sample of size m from the $\Gamma(\alpha, \beta)$ distribution and calculating $\bar{\theta}^*$ and $\widehat{SE}\left(\bar{\theta}^*\right)$.

```
rosalind 296> R
```

```
R : Copyright 2005, The R Foundation
  for Statistical Computing
Version 2.1.0 Patched (2005-05-12), ISBN 3-900051-07-0

> m <- 1000

> alpha <- 29.001

> beta <- 14.001

> n.sim <- 500

> seed <- c( 6425451, 9626954 )

> theta.out <- gamma.sim( m, alpha, beta, n.sim, seed )

# This took about 1 second at 550 Unix MHz.

> theta.out[ 1:10, ]

          [,1]       [,2]
 [1,] 2.082105 0.01166379
 [2,] 2.072183 0.01200723
 [3,] 2.066756 0.01247277
 [4,] 2.060785 0.01200449
 [5,] 2.078591 0.01212440
 [6,] 2.050640 0.01228875
 [7,] 2.071706 0.01182579
 [8,] 2.063158 0.01176577
 [9,] 2.058440 0.01186379
[10,] 2.068976 0.01220723
```

The $\bar{\theta}^*$ values fluctuate around the truth with a give-or-take of about 0.012, which agrees well with the theoretical SE $\frac{\sigma}{\sqrt{m}} = \frac{\sqrt{\alpha}}{\beta\sqrt{m}} \doteq 0.01216$ (recall that the variance of a Gamma distribution is $\frac{\alpha}{\beta^2}$).

```
> postscript( "gamma-sim1.ps" )

> theta.bar <- theta.out[ , 1 ]
```

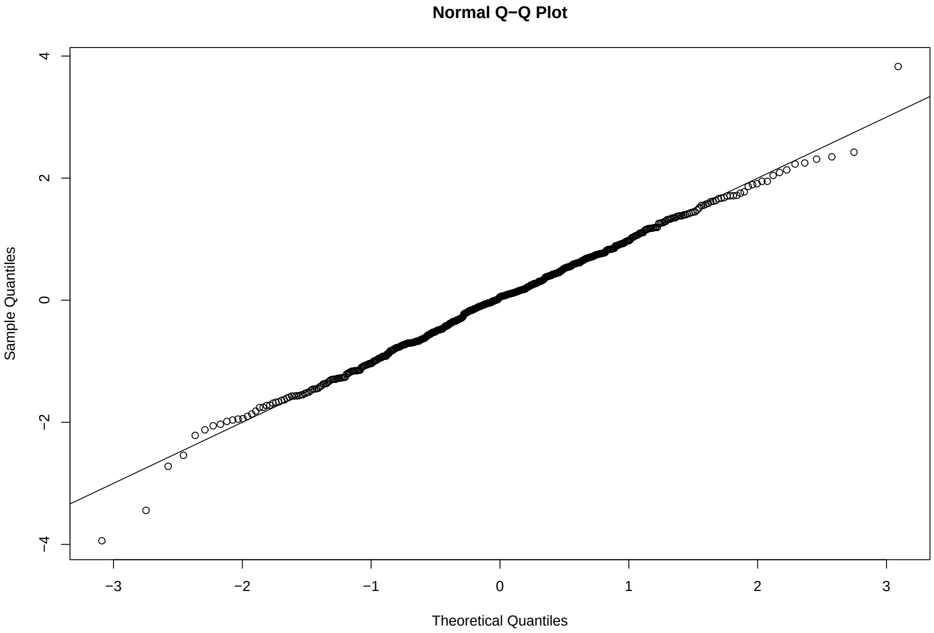


Figure 3.1: *Normal qqplot of the 500 $\bar{\theta}^*$ values.*

```
> qqnorm( ( theta.bar - mean( theta.bar ) ) /
    sqrt( var( theta.bar ) ) )

> abline( 0, 1 )

> dev.off( )

null device
          1
```

Each of the $\bar{\theta}^*$ values is the mean of $m = 1,000$ IID draws, so (by the CLT) the distribution of the random variable $\bar{\theta}^*$ should be closely approximated by a Gaussian.

```
> truth <- alpha / beta

> theta.bar.SE <- theta.out[ , 2 ]

> qnorm( 0.025 )
```

```
[1] -1.959964

> sum( ( theta.bar - 1.96 * theta.bar.SE < truth ) *
    ( truth < theta.bar + 1.96 * theta.bar.SE ) ) / n.sim

[1] 0.972
```

Thus we can use frequentist ideas to work out how big m needs to be to have any desired Monte Carlo accuracy for $\bar{\theta}^*$ as an estimate of the posterior mean $E(\theta|y)$.

In practice, with $p(\theta|y)$ unknown, you would probably take an initial sample (of size $m = 1,000$, say) and look at the MCSE to decide how big m really needs to be.

```
> theta.bar <- gamma.sim( m, alpha, beta, 1, seed )

> theta.bar
         [,1]       [,2]
[1,] 2.082105 0.01166379
```

(1) Suppose you wanted the MCSE of $\bar{\theta}^*$ to be (say) $\epsilon = 0.001$. Then you could solve the equation

$$\frac{\hat{\sigma}}{\sqrt{m}} = \epsilon \quad \leftrightarrow \quad m = \frac{\sigma^2}{\epsilon^2}, \tag{3.3}$$

which says (unhappily) that the required m goes up as the square of the posterior SD and as the inverse square of ϵ.

The previous calculation shows that $\frac{\hat{\sigma}}{\sqrt{1000}} \doteq 0.01166379$, from which $\hat{\sigma} \doteq 0.3688414$, meaning that to get $\epsilon = 0.001$ you need a sample of size $\frac{0.3688414^2}{0.001^2} \doteq 136,044 \doteq 136\text{k}$ (!).

(2) Suppose instead that you wanted $\bar{\theta}^*$ to differ from the true posterior mean μ by no more than ϵ_1 with Monte Carlo probability at least $(1 - \epsilon_2)$:

$$P\left(\left|\bar{\theta}^* - \mu\right| \le \epsilon_1\right) \ge 1 - \epsilon_2, \tag{3.4}$$

where $P(\cdot)$ here is based on the (frequentist) Monte Carlo randomness inherent in $\bar{\theta}^*$.

We know from the CLT and the calculations above that in repeated sampling $\bar{\theta}^*$ is approximately normal with mean μ and variance $\frac{\sigma^2}{m}$; this leads to the inequality

$$m \geq \frac{\sigma^2 \left[\Phi^{-1} \left(1 - \frac{\epsilon_2}{2} \right) \right]^2}{\epsilon_1^2}, \tag{3.5}$$

where $\Phi^{-1}(q)$ is the place on the standard normal curve where $100q\%$ of the area is to the left of that place (the qth quantile of the standard normal distribution).

(5) is like (3) except that the value of m from (3) has to be multiplied by $\left[\Phi^{-1} \left(1 - \frac{\epsilon_2}{2} \right) \right]^2$, which typically makes the required sample sizes even bigger.

For example, with $\epsilon_1 = 0.001$ and $\epsilon_2 = 0.05$—i.e., to have at least 95% Monte Carlo confidence that reporting the posterior mean as 2.071 will be correct to about four significant figures—(5) says that you would need a monitoring run of at least $136,044(1.959964)^2 \doteq 522,608 \doteq 523\text{k}$ (!).

(On the other hand, this sounds like a long monitoring run but only takes about 2.5 seconds at 550 Unix MHz on a `SunBlade 100`, yielding $\left[\bar{\theta}^*, \widehat{SE} \left(\bar{\theta}^* \right) \right] = (2.0709, 0.00053)$.)

It's evident from calculations like these that people often report simulation-based answers with numbers of significant figures far in excess of what is justified by the actual accuracy of the Monte Carlo estimates.

A Closer Look at IID Sampling. I was able to easily perform the above simulation study because `R` has a large variety of built-in functions like `rgamma` for pseudo-random-number generation.

How would you go about writing such functions yourself?

There are a number of general-purpose methods for generating random numbers (I won't attempt a survey here); the one we need to look closely at, to understand the algorithms that arise later in this section, is rejection sampling (von Neumann, 1951), which is often one of the most computationally efficient ways to make IID draws from a distribution.

3.1.1 Rejection sampling

Example. In the spring of 1993 a survey was taken of bicycle and other traffic in the vicinity of the University of California, Berkeley, campus (Gelman, Carlin et al. 2003).

As part of this survey 10 city blocks on residential streets with bike routes were chosen at random from all such blocks at Berkeley; on one of those blocks

n vehicles were observed on a randomly chosen Tuesday afternoon from 3 to 4pm, and s of them were bicycles.

To draw inferences about the underlying proportion θ of bicycle traffic (PBT) on blocks similar to this one at times similar to Tuesday afternoons from 3 to 4pm, it's natural (as in the AMI mortality case study) to employ the model

$$\left\{ \begin{array}{c} \theta \sim \text{Beta}(\alpha_0, \beta_0) \\ (S|\theta) \sim \text{Binomial}(n, \theta) \end{array} \right\} \rightarrow (\theta|s) \sim \text{Beta}(\alpha_0 + s, \beta_0 + n - s), \tag{3.6}$$

provided that whatever prior information I have about θ can be meaningfully captured in the Beta family.

After reflection I realize that I'd be quite surprised if the PBT in residential city blocks with bike routes in Berkeley on Tuesday afternoons from 3 to 4pm was less than 5% or greater than 50%.

Making this operational by assuming that in the prior $p(0.05 \leq \theta \leq 0.5) = 0.9$, and putting half of the remaining prior probability in each of the left and right tails of the Beta distributions, yields (via numerical methods similar to those in the AMI case study) $(\alpha_0, \beta_0) = (2.0, 6.4)$ (this Beta distribution has prior mean and SD 0.24 and 0.14, respectively).

In the city block in question the data came out $(n, s) = (74, 16)$, so that the data mean was 0.216, and the posterior is then $\text{Beta}(\alpha_0 + s, \beta_0 + n - s) = \text{Beta}(18.0, 64.4)$.

Pretend for the sake of illustration of rejection sampling that you didn't know the formulas for the mean and SD of a Beta distribution, and suppose that you wanted to use IID Monte Carlo sampling from the $\text{Beta}(\alpha_0 + s, \beta_0 + n - s)$ posterior to estimate the posterior mean.

Here's von Neumann's basic idea: suppose the target density $p(\theta|y)$ is difficult to sample from, but you can find an integrable envelope function $G(\theta|y)$ such that (a) G dominates p in the sense that $G(\theta|y) \geq p(\theta|y) \geq 0$ for all θ and (b) the density g obtained by normalizing G—later to be called the proposal distribution—is easy and fast to sample from.

Then to get a random draw from p, make a draw θ^* from g instead and accept or reject it according to an acceptance probability $\alpha_R(\theta^*|y)$; if you reject the draw, repeat this process until you accept.

von Neumann showed that the choice

$$\alpha_R(\theta^*|y) = \frac{p(\theta^*|y)}{G(\theta^*|y)} \tag{3.7}$$

correctly produces IID draws from p, and you can intuitively see that he's right by the following argument.

Making a draw from the posterior distribution of interest is like choosing a point at random (in two dimensions) under the density curve $p(\theta|y)$ in such a way that all possible points are equally likely, and then writing down its θ value.

If you instead draw from G so that all points under G are equally likely, to get correct draws from p you'll need to throw away any point that falls between p and G, and this can be accomplished by accepting each sampled point θ^* with probability $\frac{p(\theta^*|y)}{G(\theta^*|y)}$, as von Neumann said.

A summary of this method is as follows.

Algorithm (rejection sampling). To make m IID draws at random from the density $p(\theta|y)$ for real-valued θ, select an integrable envelope function G—which when normalized to integrate to 1 is the proposal distribution g—such that $G(\theta|y) \geq p(\theta|y) \geq 0$ for all θ; define the acceptance probability $\alpha_R(\theta^*|y) = \frac{p(\theta^*|y)}{G(\theta^*|y)}$; and

```
Initialize $t \leftarrow 0$
Repeat {
    Sample $\theta^* \sim g(\theta|y)$
    Sample $u \sim$ Uniform$(0, 1)$
    If $u \leq \alpha_R(\theta^*|y)$ then
        { $\theta_{t+1} \leftarrow \theta^*$;  $t \leftarrow (t+1)$ }
}
until $t = m$.
```

(3.8)

choosing a θ^* from g locates the horizontal coordinate according to G; choosing a u as above is equivalent to picking a point at random vertically on the line segment from $(\theta^*, 0)$ to $(\theta^*, G(\theta^*))$ and seeing whether it's below p or not

The figure below demonstrates this method on the Beta$(18.0, 64.4)$ density arising in the Beta-Bernoulli example above.

Rejection sampling permits considerable flexibility in the choice of envelope function; here, borrowing an idea from Gilks and Wild (1992), I've noted that the relevant Beta density is log concave (a real-valued function is log concave if its second derivative on the log scale is everywhere non-positive),

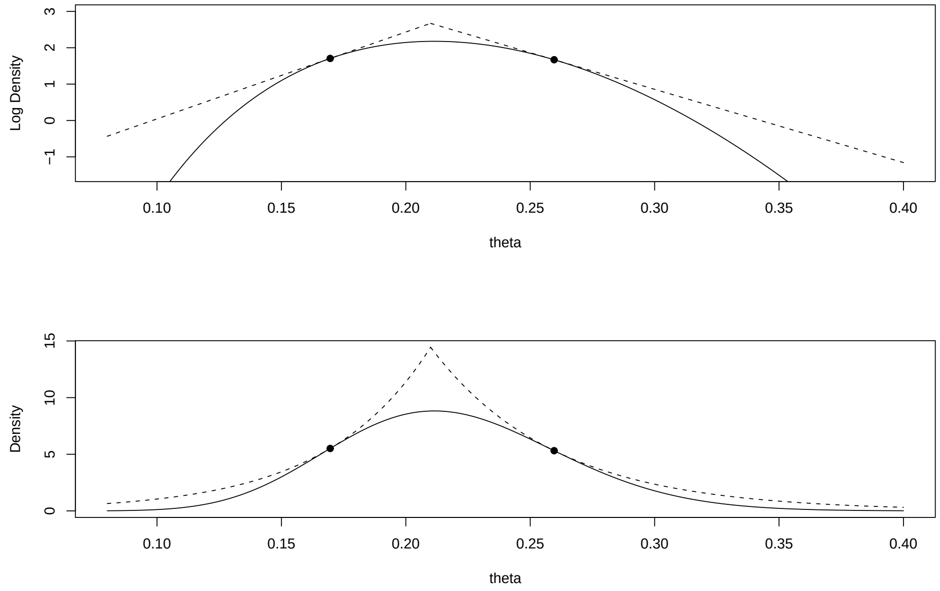


Figure 3.2: *Top panel: piecewise linear rejection sampling envelope function for the log posterior; bottom panel: top panel transformed back to the density scale.*

meaning that it's easy to construct an envelope on that scale in a piecewise linear fashion, by choosing points on the log density and constructing tangents to the curve at those points.

The simplest possible such envelope involves two line segments, one on either side of the mode.

The optimal choice of the tangent points would maximize the marginal probability of acceptance of a draw in the rejection algorithm, which can be shown to be

$$\left[\int G(\theta) \, d\theta \right]^{-1} ; \qquad (3.9)$$

in other words, you should minimize the area under the (un-normalized) envelope function subject to the constraint that it dominates the target density $p(\theta|y)$ (which makes eminently good sense).

Here this optimum turns out to be attained by locating the two tangent points at about 0.17 and 0.26, as in the figure above; the resulting acceptance probability of about 0.75 could clearly be improved by adding more tangents.

Piecewise linear envelope functions on the log scale are a good choice because the resulting envelope density on the raw scale is a piecewise set of scaled exponential distributions (see the bottom panel in the figure above), from which random samples can be taken quickly.

A preliminary sample of $m_0 = 500$ IID draws from the Beta$(18.0, 64.4)$ distribution using the above rejection sampling method yields $\bar{\theta}^* = 0.2197$ and $\hat{\sigma} = 0.04505$, meaning that the posterior mean has already been estimated with an MCSE of only $\frac{\hat{\sigma}}{\sqrt{m_0}} = 0.002$ even with just 500 draws.

Suppose, however, that—as in equation (4) above—I want $\bar{\theta}^*$ to differ from the true posterior mean μ by no more than some (perhaps even smaller) tolerance ϵ_1 with Monte Carlo probability at least $(1 - \epsilon_2)$; then equation (5) tells me how long to monitor the simulation output.

For instance, to pin down three significant figures (sigfigs) in the posterior mean in this example with high Monte Carlo accuracy I might take $\epsilon_1 = 0.0005$ and $\epsilon_2 = 0.05$, which yields a recommended IID sample size of $\frac{(0.04505^2)(1.96)^2}{0.0005^2} \doteq 31,200$.

So I take another sample of 30,700 (which is virtually instantaneous at 550 Unix MHz) and merge it with the 500 draws I already have; this yields $\bar{\theta}^* = 0.21827$ and $\hat{\sigma} = 0.04528$, meaning that the MCSE of this estimate of μ is $\frac{0.04528}{\sqrt{31200}} \doteq 0.00026$.

I might announce that I think $E(\theta|y)$ is about 0.2183, give or take about 0.0003, which accords well with the true value 0.2184.

Of course, other aspects of $p(\theta|y)$ are equally easy to monitor; for example, if I want a Monte Carlo estimate of $p(\theta \leq q|y)$ for some q, as noted above I just work out the proportion of the sampled θ^* values that are no larger than q.

Or, even better, I recall that $P(A) = E[I(A)]$ for any event or proposition A, so to the Monte Carlo dataset (see p. 26 below) consisting of 31,200 rows and one column (the θ^*_t) I add a column monitoring the values of the derived variable which is 1 whenever $\theta^*_t \leq q$ and 0 otherwise; the mean of this derived variable is the Monte Carlo estimate of $p(\theta \leq q|y)$, and I can attach an MCSE to it in the same way I did with $\bar{\theta}^*$.

By this approach, for instance, the Monte Carlo estimate of $p(\theta \leq 0.15|y)$ based on the 31,200 draws examined above comes out $\hat{p} = 0.0556$ with an MCSE of 0.0013.

Percentiles are typically harder to pin down with equal Monte Carlo accuracy (in terms of sigfigs) than means or SDs, because the 0/1 scale on which they're based is less information-rich than the θ^* scale itself; if I wanted an MCSE for $\hat{p}$ of 0.0001 I would need an IID sample of more than 5 million draws (which would still only take a few seconds at contemporary workstation speeds).

Beyond rejection sampling: IID sampling is not necessary. Nothing in the Metropolis-Ulam idea of Monte Carlo estimates of posterior summaries requires that these estimates be based on IID samples from the posterior.

This is lucky, because in practice it's often difficult, particularly when θ is a vector of high dimension (say k), to figure out how to make such an IID sample, via rejection sampling or other methods (e.g., imagine trying to find an envelope function for $p(\theta|y)$ when k is 10 or 100 or 1,000).

Thus it's necessary to relax the assumption that $\theta_j^* \overset{\text{IID}}{\sim} p(\theta|y)$, and to consider samples $\theta_1^*, \ldots, \theta_m^*$ that form a time series: a series of draws from $p(\theta|y)$ in which θ_j^* may depend on $\theta_{j'}^*$ for $j' < j$.

In their pioneering paper Metropolis et al. (1953) allowed for serial dependence of the θ_j^* by combining von Neumann's idea of rejection sampling (which had itself only been published a few years earlier in 1951) with concepts from Markov chains, a subject in the theory of stochastic processes.

Combining Monte Carlo sampling with Markov chains gives rise to the name now used for this technique for solving the Bayesian high-dimensional integration problem: Markov chain Monte Carlo (MCMC).

3.2 Markov chains

Markov chains. A stochastic process is just a collection of random variables $\{\theta_t^*, t \in T\}$ for some index set T, usually meant to stand for time.

In practice T can be either discrete, e.g., $\{0, 1, \ldots\}$, or continuous, e.g., $[0, \infty)$.

Markov chains are a special kind of stochastic process that can either unfold in discrete or continuous time—we'll talk here about discrete-time Markov chains, which is all you need for MCMC.

The possible values that a stochastic process can take on are collectively called the state space S of the process—in the simplest case S is real-valued and can also either be discrete or continuous.

Intuitively speaking, a Markov chain (e.g., Feller, 1968; Roberts, 1996; Gamerman, 1997) is a stochastic process unfolding in time in such a way that the past and future states of the process are independent given the present state—in other words, to figure out where the chain is likely to go next you don't need to pay attention to where it's been, you just need to consider where it is now.

More formally, a stochastic process $\{\theta^*_t, t \in T\},\ T = \{0, 1, \ldots\}$, with state space S is a Markov chain if, for any set $A \in S$,

$$P(\theta^*_{t+1} \in A|\theta^*_0, \ldots, \theta^*_t) = P(\theta^*_{t+1} \in A|\theta^*_t). \tag{3.10}$$

The theory of Markov chains is harder mathematically if S is continuous (e.g., Tierney, 1996), which is what we need for MCMC with real-valued parameters, but most of the main ideas emerge with discrete state spaces, and I'll assume discrete S in the intuitive discussion here.

Example. For a simple example of a discrete-time Markov chain with a discrete state space, imagine a particle that moves around on the integers $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$, starting at 0 (say).

Wherever it is at time t—say at i—it tosses a (3-sided) coin and moves to $(i-1)$ with probability p_1, stays at i with probability p_2, and moves to $(i+1)$ with probability p_3, for some $0 < p_1, p_2, p_3 < 1$ with $p_1 + p_2 + p_3 = 1$—these are the transition probabilities for the process.

This is a random walk (on the integers), and it's clearly a Markov chain.

Nice behavior. The most nicely-behaved Markov chains satisfy three properties:

- They're irreducible, which basically means that no matter where it starts the chain has to be able to reach any other state in a finite number of iterations with positive probability;
- They're aperiodic, meaning that for all states i the set of possible sojourn times, to get back to i having just left it, can have no divisor bigger than 1 (this is a technical condition; periodic chains still have some nice properties, but the nicest chains are aperiodic).
- They're positive recurrent, meaning that (a) for all states i, if the process starts at i it will return to i with probability 1, and (b) the expected length of waiting time til the first return to i is finite.

Notice that this is a bit delicate: wherever the chain is now, we insist that it must certainly come back here, but we don't expect to have to wait forever for this to happen.

The random walk defined above is clearly irreducible and aperiodic, but it may not be positive recurrent (depending on the p_i): it's true that it has positive probability of returning to wherever it started, but (because S is unbounded) this probability may not be 1, and on average you may have to wait forever for it to return.

We can fix this by bounding S: suppose instead that $S = \{-k, -(k-1), \dots, -1, 0, 1, \dots, k\}$, keeping the same transition probabilities except rejecting any moves outside the boundaries of S.

This bounded random walk now satisfies all three of the nice properties.

The value of nice behavior. Imagine running the bounded random walk for a long time, and look at the distribution of the states it visits—over time this distribution should settle down (converge) to a kind of limiting, steady-state behavior.

This can be demonstrated by simulation, for instance in R, and using the bounded random walk as an example:

```
rw.sim <- function( k, p, theta.start, n.sim, seed ) {

  set.seed( seed )

  theta <- rep( 0, n.sim + 1 )

  theta[ 1 ] <- theta.start

  for ( i in 1:n.sim ) {

    theta[ i + 1 ] <- move( k, p, theta[ i ] )

  }

  return( table( theta ) )

}

move <- function( k, p, theta ) {

  repeat {

    increment <- sample( x = c( -1, 0, 1 ), size = 1, prob = p )

    theta.next <- theta + increment
    if ( abs( theta.next ) <= k ) {

      return( theta.next )

      break

    }

  }

}

rosalind 17> R

R : Copyright 2005, The R Foundation
Version 2.1.0 Patched (2005-05-12), ISBN 3-900051-07-0

> p <- c( 1, 1, 1 ) / 3

> k <- 5

> theta.start <- 0

> seed <- c( 6425451, 9626954 )

> rw.sim( k, p, theta.start, 10, seed )

theta
0 1 2
5 5 1

> rw.sim( k, p, theta.start, 100, seed )

-2 -1  0  1  2  3  4  5
 7  9 16 17 23 14  8  7

> rw.sim( k, p, theta.start, 1000, seed )

 -5  -4  -3  -2  -1   0   1   2   3   4   5
 65 115 123 157 148 123 106  82  46  21  15
```

```
> rw.sim( k, p, theta.start, 10000, seed )

  -5   -4   -3   -2   -1    0    1    2    3    4    5
 581  877  941  976  959 1034 1009  982 1002  959  681

> rw.sim( k, p, theta.start, 100000, seed )

  -5   -4   -3   -2   -1    0    1    2    3    4    5
6515 9879 9876 9631 9376 9712 9965 9749 9672 9352 6274

> rw.sim( k, p, theta.start, 1000000, seed )

   -5    -4    -3    -2    -1     0     1     2     3     4     5
65273 98535 97715 96708 95777 96607 96719 96361 96836 95703 63767
```

You can see that the distribution of where the chain has visited is converging to something close to uniform on $\{-5, -4, \ldots, 4, 5\}$, except for the effects of the boundaries.

Letting q_1 denote the limiting probability of being in one of the 9 non-boundary states $(-4, -3, \ldots, 3, 4)$ and q_2 be the long-run probability of being in one of the 2 boundary states $(-5, 5)$, on grounds of symmetry you can guess that q_1 and q_2 should satisfy

$$9q_1 + 2q_2 = 1 \quad \text{and} \quad q_1 = \frac{3}{2} q_2, \tag{3.11}$$

from which $(q_1, q_2) = \left(\frac{3}{31}, \frac{2}{31}\right) \doteq (0.096774, 0.064516)$.

Based on the run of 1,000,001 iterations above we would estimate these probabilities empirically as

$$\left[\frac{98535 + \ldots + 95703}{(9)(1000001)}, \frac{65273 + 63767}{(2)(1000001)}\right] \doteq (0.096773, 0.064520)\,.$$

It should also be clear that the limiting distribution does not depend on the initial value of the chain:

```
> rw.sim( k, p, 5, 100000, seed )

  -5   -4   -3   -2   -1    0    1    2    3    4    5
6515 9879 9876 9624 9374 9705 9959 9738 9678 9365 6288
```

Of course, you get a different limiting distribution with a different choice of (p_1, p_2, p_3):

```
> p <- c( 0.2, 0.3, 0.5 )

> rw.sim( k, p, 0, 10, seed )

0 1 2 3
1 3 4 3

> rw.sim( k, p, 0, 100, seed )

 0  1  2  3  4  5
 1  3  6 13 30 48

> rw.sim( k, p, 0, 1000, seed )

  0   1   2   3   4   5
  1  18  71 157 336 418

> rw.sim( k, p, 0, 10000, seed )

  -5   -4   -3   -2   -1    0    1    2    3    4    5
   5   16   19   30   28   74  215  583 1344 3470 4217

> rw.sim( k, p, 0, 100000, seed )

   -5    -4    -3    -2    -1     0     1     2     3     4     5
    5    22    53   132   302   834  2204  5502 13489 34460 42998

> rw.sim( k, p, 0, 1000000, seed )
  -5  -4  -3   -2   -1    0     1     2      3      4      5
  61 198 511 1380 3398 8591 22117 54872 137209 343228 428436
```

Stationary distributions. A positive recurrent and aperiodic chain is called ergodic, and it turns out that such chains possess a unique stationary (or equilibrium, or invariant) distribution π, characterized by the relation

$$\pi(j) = \sum_i \pi(i) P_{ij}(t) \tag{3.12}$$

for all states j and times $t \geq 0$, where $P_{ij}(t) = P(\theta^*_t = j | \theta^*_{t-1} = i)$ is the transition matrix of the chain.

Informally, the stationary distribution characterizes the behavior that the chain will settle into after it's been run for a long time, regardless of its initial state.

The point of all of this. Given a parameter vector θ and a data vector y, the Metropolis et al. (1953) idea is to simulate random draws from the posterior distribution $p(\theta|y)$, by constructing a Markov chain with the following three properties:

- It should have the same state space as θ,
- It should be easy to simulate from, and
- Its equilibrium distribution should be $p(\theta|y)$.

If you can do this, you can run the Markov chain for a long time, generating a huge sample from the posterior, and then use simple descriptive summaries (means, SDs, correlations, histograms or kernel density estimates) to extract any features of the posterior you want.

There is a fourth desirable condition as well:

- It should not be necessary to work out the normalizing constant for $p(\theta|y)$ to implement the algorithm, which is equivalent to saying that $p(\theta|y)$ should appear in the calculations only through ratios of the form $\frac{p(\theta|y)}{p(\theta'|y)}$.

The Ergodic Theorem. The mathematical fact that underpins this strategy is the ergodic theorem: if the Markov chain $\{\theta^*_t\}$ is ergodic and f is any real-valued function for which $E_\pi|f(\theta)|$ is finite, then with probability 1 as $m \rightarrow \infty$

$$\frac{1}{m} \sum_{t=1}^{m} f(\theta^*_t) \rightarrow E_\pi[f(\theta)] = \sum_i f(i)\, \pi(i), \tag{3.13}$$

in which the right side is just the expectation of $f(\theta)$ under the stationary distribution π.

In plain English this means that—as long as the stationary distribution is $p(\theta|y)$—you can learn (to arbitrary accuracy) about things like posterior means, SDs, and so on just by waiting for stationarity to kick in and monitoring thereafter for a long enough period.

Of course, as Roberts (1996) notes, the theorem is silent on the two key practical questions it raises: how long you have to wait for stationarity, and how long to monitor after that.

A third practical issue is what to use for the initial value θ^*_0: intuitively the closer θ^*_0 is to the center of $p(\theta|y)$ the less time you should have to wait for stationarity.

The standard way to deal with waiting for stationarity is to (a) run the chain from a good starting value θ_0^* for b iterations, until equilibrium has been reached, and (b) discard this initial burn-in period.

All of this motivates the topic of MCMC diagnostics, which are intended to answer the following questions:

• What should I use for the initial value θ_0^*?

• How do I know when I've reached equilibrium? (This is equivalent to asking how big b should be.)

• Once I've reached equilibrium, how big should m be, i.e., how long should I monitor the chain to get posterior summaries with decent accuracy?

3.2.1 The Monte Carlo and MCMC datasets

The basis of the Monte Carlo approach to obtaining numerical approximations to posterior summaries like means and SDs is the (weak) Law of Large Numbers: with IID sampling the Monte Carlo estimates of the true summaries of $p(\theta|y)$ are consistent, meaning that they can be made arbitrarily close to the truth with arbitrarily high probability as the number of monitoring iterations $m \rightarrow \infty$.

Before we look at how Metropolis et al. attempted to achieve the same goal with a non-IID Monte Carlo approach, let's look at the practical consequences of switching from IID to Markovian sampling.

Running the IID rejection sampler on the Berkeley PBT example above for a total of m monitoring iterations would produce something that might be called the Monte Carlo dataset, with one row for each iteration and one column for each monitored quantity; in that example it might look like this (MCSEs in parenthesis):

Iteration	θ	$I(\theta \leq 0.15)$
1	$\theta_1^* = 0.244$	$I_1^* = 0$
2	$\theta_2^* = 0.137$	$I_2^* = 1$
$\vdots$	$\vdots$	$\vdots$
$m = 31,200$	$\theta_m^* = 0.320$	$I_m^* = 0$
Mean	0.2183 (0.003)	0.0556 (0.0013)
SD	0.04528	—
Density Trace	(like the bottom plot on p. 14)	—

Running the Metropolis sampler on the same example would produce something that might be called the MCMC dataset.

It would have a similar structure as far as the columns are concerned, but the rows would be divided into three phases:

- Iteration 0 would be the value(s) used to initialize the Markov chain;
- Iterations 1 through b would be the burn-in period, during which the chain reaches its equilibrium or stationary distribution (as mentioned above, iterations 0 through b are generally discarded); and
- Iterations $(b+1)$ through $(b+m)$ would be the monitoring run, on which summaries of the posterior (means, SDs, density traces, ...) will be based.

In the Berkeley PBT example the MCMC dataset might look like this:

Iteration	Phase	θ	$I(\theta \le 0.15)$
0	Initialization	$\theta_0^* = 0.200$	—
1	Burn-in	$\theta_1^* = 0.244$	—
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$b = 500$	Burn-in	$\theta_b^* = 0.098$	—
$(b+1) = 501$	Monitoring	$\theta_{b+1}^* = 0.275$	$I_{b+1}^* = 0$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(b+m) = 31,700$	Monitoring	$\theta_{b+m}^* = 0.120$	$I_{b+m}^* = 1$
Mean	(Monitoring	0.2177 (0.009)	0.0538 (0.004)
SD	Phase	0.04615	—
Density Trace	Only)	(like the bottom plot on p. 14)	—

Think of iteration number i in the Monte Carlo sampling process as a discrete index of time t, so that the columns of the MC and MCMC datasets can be viewed as time series.

An important concept from time series analysis is autocorrelation: the autocorrelation ρ_k of a stationary time series θ_t^* at lag k (see, e.g., Chatfield (1996)) is $\frac{\gamma_k}{\gamma_0}$, where γ_k is $C(\theta_t^*, \theta_{t-k}^*)$, the covariance of the series with itself k iterations in the past—this measures the degree to which the time series at any given moment depends on its past history.

IID draws from $p(\theta|y)$ correspond to white noise: a time series with zero autocorrelations at all lags.

This is the behavior of the columns in the MC data set on p. 26, produced by ordinary rejection sampling.

Because of the Markov character of the columns of the MCMC data set on p. 27, each column, when considered as a time series, will typically have non-zero autocorrelations, and because Markov chains use their present values to decide where to go next it shouldn't surprise you to hear that the typical behavior will be (substantial) positive autocorrelations—in other words, every time you get another draw from the Markov chain you get some new information about the posterior and a rehash of old information mixed in.

It's a marvelous result from time series analysis (the Ergodic Theorem for Markov chains on p. 25 is an example of this fact) that all of the usual descriptive summaries of the posterior are still consistent as long as the columns of the MCMC data set form stationary time series.

In other words, provided that you can achieve the four goals back on p. 24 which Metropolis et al. set for themselves, and provided that you only do your monitoring after the Markov chain has reached equilibrium, the MCMC approach and the IID Monte Carlo approach are equally valid (they both get the right answers), but they may well differ on their efficiency (the rate per iteration, or per CPU second, at which they learn about the posterior may not be the same); and if, as is typically true, the columns of the MCMC dataset have positive autocorrelations, this will translate into slower learning (larger MCSEs) than with IID sampling (compare the MCSEs on pages 26 and 27).

3.3 The Metropolis algorithm

Metropolis et al. were able to create what people would now call a successful MCMC algorithm by the following means (see the excellent book edited by Gilks et al. (1996) for many more details about the MCMC approach).

Consider the rejection sampling method given above in (8) as a mechanism for generating realizations of a time series (where as above time indexes iteration number).

At any time t in this process you make a draw θ^* from the proposal distribution $g(\theta|y)$ (the normalized version of the envelope function G) and either accept a "move" to θ^* or reject it, according to the acceptance probability $\frac{p(\theta^*|y)}{G(\theta^*|y)}$; if accepted the process moves to θ^*, if not you draw again and discard the rejected draws until you do make a successful move.

As noted above, the stochastic process thus generated is an IID (white noise) series of draws from the target distribution $p(\theta|y)$.

Metropolis et al. had the following beautifully simple idea for how this may be generalized to situations where IID sampling is difficult: they allowed the proposal distribution at time t to depend on the current value θ_t of the process, and then—to get the right stationary distribution—if a proposed move is rejected, instead of discarding it the process is forced to stay where it is for one iteration before trying again.

The resulting process is a Markov chain, because (a) the draws are now dependent but (b) all you need to know in determining where to go next is where you are now.

3.3.1 Metropolis-Hastings sampling

Letting θ_t stand for where you are now and θ^* for where you're thinking of going, in this approach there is enormous flexibility in the choice of the proposal distribution $g(\theta^*|\theta_t, y)$, even more so than in ordinary rejection sampling.

The original Metropolis et al. idea was to work with symmetric proposal distributions, in the sense that $g(\theta^*|\theta_t, y) = g(\theta_t|\theta^*, y)$, but Hastings (1970) pointed out that this could easily be generalized; the resulting method is the Metropolis-Hastings (MH) algorithm.

Building on the Metropolis et al. results, Hastings showed that you'll get the correct stationary distribution $p(\theta|y)$ for your Markov chain by making the following choice for the acceptance probability:

$$\alpha_{MH}(\theta^*|\theta_t, y) = \min \left\{ 1, \frac{\frac{p(\theta^*|y)}{g(\theta^*|\theta_t,y)}}{\frac{p(\theta_t|y)}{g(\theta_t|\theta^*,y)}} \right\}. \tag{3.14}$$

It turns out that the proposal distribution $g(\theta^*|\theta_t, y)$ can be virtually anything and you'll get the right equilibrium distribution using the acceptance probability (14); see, e.g., Roberts (1996) and Tierney (1996) for the mild regularity conditions necessary to support this statement.

A summary of the method is on the next page.

It's instructive to compare (15) with (8) to see how heavily the MH algorithm borrows from ordinary rejection sampling, with the key difference that the proposal distribution is allowed to change over time.

Notice how (14) generalizes von Neumann's acceptance probability ratio $\frac{p(\theta^*|y)}{G(\theta^*|y)}$ for ordinary rejection sampling: the crucial part of the new MH acceptance probability becomes the ratio of two von-Neumann-like ratios, one for where you are now and one for where you're thinking of going (it's equivalent to work with g or G since the normalizing constant cancels in the ratio).

Algorithm (Metropolis-Hastings sampling). To construct a Markov chain whose equilibrium distribution is $p(\theta|y)$, choose a proposal distribution $g(\theta^*|\theta_t, y)$, define the acceptance probability $\alpha_{MH}(\theta^*|\theta_t, y)$ by (14), and

```
    Initialize $\theta_0$; $t \leftarrow 0$
    Repeat {
        Sample $\theta^* \sim g(\theta|\theta_t, y)$
        Sample $u \sim$ Uniform$(0, 1)$
        If $u \le \alpha_{MH}(\theta^*|\theta_t, y)$ then $\theta_{t+1} \leftarrow \theta^*$
            else $\theta_{t+1} \leftarrow \theta_t$
        $t \leftarrow (t + 1)$
    }
```

(3.15)

When the proposal distribution is symmetric in the Metropolis sense, the acceptance probability ratio reduces to $\frac{p(\theta^*|y)}{p(\theta_t|y)}$, which is easy to motivate intuitively: whatever the target density is at the current point θ_t, you want to visit points of higher density more often and points of lower density less often, and it turns out that (14) does this for you in the natural and appropriate way.

As an example of the MH algorithm in action, consider a Gaussian model with known mean μ and unknown variance σ^2 applied to the NB10 data in part 2 of the lecture notes.

The likelihood function for σ^2, derived from the sampling model $(Y_i|\sigma^2) \stackrel{\text{IID}}{\sim} N(\mu, \sigma^2)$ for $i = 1, \ldots, n$, is

$$\begin{aligned} l(\sigma^2|y) &= c \prod_{i=1}^{n} (\sigma^2)^{-\frac{1}{2}} \exp\left[-\frac{(y_i - \mu)^2}{2\sigma^2} \right] \qquad (3.16) \\ &= c\,(\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{\sum_{i=1}^{n} (y_i - \mu)^2}{2\sigma^2} \right]. \end{aligned}$$

This is recognizable as a member of the Scaled Inverse χ^2 family $\chi^{-2}(\nu, s^2)$ (e.g., Gelman, Carlin et al. (2003)) of distributions, which (as we saw in part 2 of the lecture notes) is a rescaled version of the Inverse Gamma family chosen so that s^2 is an estimate of σ^2 based upon ν "observations."

You can now convince yourself that if the prior for σ^2 in this model is taken to be $\chi^{-2}(\nu, s^2)$, then the posterior for σ^2 will also be Scaled Inverse χ^2: with this choice of prior

$$p(\sigma^2|y) = \chi^{-2}\left[\nu + n, \frac{\nu s^2 + \sum_{i=1}^n (y_i - \mu)^2}{\nu + n}\right]. \qquad (3.17)$$

This makes good intuitive sense: the prior estimate s^2 of σ^2 receives ν votes and the sample estimate $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n (y_i - \mu)^2$ receives n votes in the posterior weighted average estimate $\frac{\nu s^2 + n\hat{\sigma}^2}{\nu + n}$.

Equation (17) provides a satisfying closed-form solution to the Bayesian updating problem in this model (e.g., it's easy to compute posterior moments analytically, and you can use numerical integration or well-known approximations to the CDF of the Gamma distribution to compute percentiles).

For illustration purposes suppose instead that you want to use MH sampling to summarize this posterior.

Then your main choice as a user of the algorithm is the specification of the proposal distribution (PD) $g(\sigma^2|\sigma_t^2, y)$.

The goal in choosing the PD is getting a chain that mixes well (moves freely and fluidly among all of the possible values of $\theta = \sigma^2$), and nobody has (yet) come up with a sure-fire strategy for always succeeding at this task.

Having said that, here are two basic ideas that often tend to promote good mixing:

(1) Pick a PD that looks like a somewhat overdispersed version of the posterior you're trying to sample from (e.g., Tierney (1996)).

Some work is naturally required to overcome the circularity inherent in this choice (if I fully knew $p(\theta|y)$ and all of its properties, why would I be using this algorithm in the first place?).

(2) Set up the PD so that the expected value of where you're going to move to (θ^*), given that you accept a move away from where you are now (θ_t), is to stay where you are now: $E_g(\theta^*|\theta_t, y) = \theta_t$.

That way, when you do make a move, there will be an approximate left-right balance, so to speak, in the direction you move away from θ_t, which will encourage rapid exploration of the whole space.

Using idea (1), a decent choice for the PD in the Gaussian model with unknown variance might well be the Scaled Inverse χ^2 distribution: $g(\sigma^2|\sigma_t^2, y) = \chi^{-2}(\nu_*, \sigma_*^2)$.

This distribution has mean $\frac{\nu_*}{\nu_* - 2}\,\sigma_*^2$ for $\nu_* > 2$.

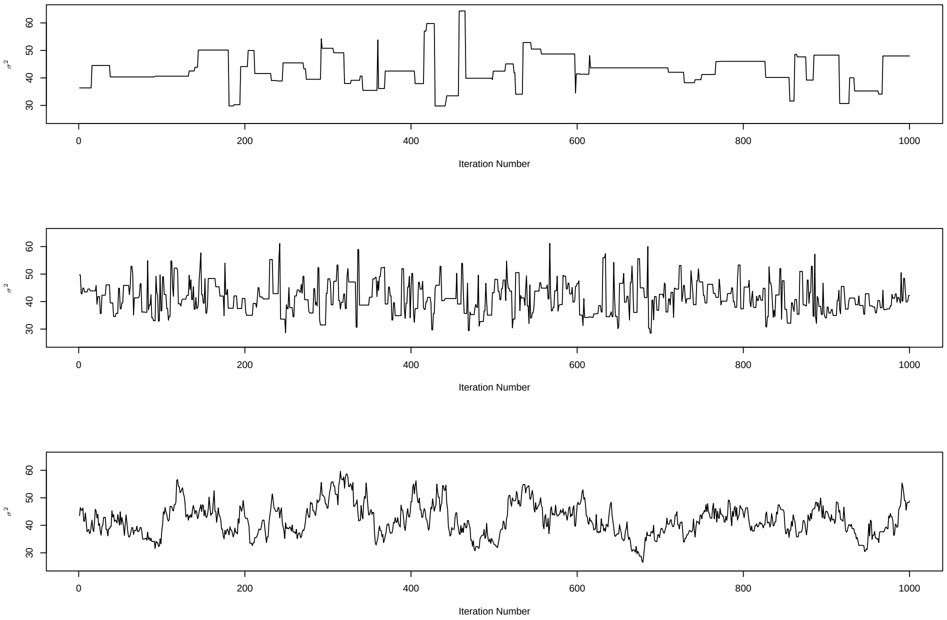


Figure 3.3: *Output of the Metropolis sampler with a Scaled Inverse* χ^2 *proposal distribution with three values of the tuning constant* $\nu_* = \{2.5, 20, 500\}$ *(reading from top to bottom).*

To use idea (2), then, I can choose any ν_* greater than 2 that I want, and as long as I take $\sigma_*^2 = \frac{\nu_* - 2}{\nu_*} \sigma_t^2$ that will center the PD at σ_t^2 as desired.

So I'll use

$$g\big(\sigma^2 | \sigma_t^2, y\big) = \chi^{-2} \left(\nu_*, \frac{\nu_* - 2}{\nu_*} \sigma_t^2 \right). \tag{3.18}$$

This leaves ν_* as a kind of potential tuning constant—the hope is that I can vary ν_* to improve the mixing of the chain.

The above figure (motivated by an analogous plot in Gilks et al. (1996)) presents time series traces of some typical output of the MH sampler with $\nu_* = (2.5, 20, 500)$.

The acceptance probabilities with these values of ν_* are $(0.07, 0.44, 0.86)$, respectively.

The SD of the $\chi^{-2} \left(\nu_*, \frac{\nu_* - 2}{\nu_*} \sigma_t^2 \right)$ distribution is proportional to $\frac{\nu_*^2}{(\nu_*^2 - 2)^2 \sqrt{\nu_* - 4}}$, which decreases as ν_* increases, and this turns out to be crucial: when the proposal distribution SD is too large (small ν_*, as in the top panel in the figure), the algorithm tries to make big jumps around θ space (good), but almost all of them get rejected (bad), so there are long periods of no movement at all, whereas when the PD SD is too small (large ν_*; see the bottom panel of the figure), the algorithm accepts most of its proposed moves (good), but they're so tiny that it takes a long time to fully explore the space (bad).

Gelman, Roberts, et al. (1995) have shown that in simple problems with approximately normal target distributions, the optimal acceptance rate for MH samplers like the one illustrated here is about 44% when the vector of unknowns is one-dimensional, and this can serve as a rough guide: you can modify the proposal distribution SD until the acceptance rate is around the Gelman et al. target figure.

The central panel of the figure displays the best possible MH behavior in this problem in the family of PDs chosen.

Even with this optimization you can see that the mixing is not wonderful, but contemporary computing speeds enable huge numbers of draws to be collected in a short period of time, compensating for the comparatively slow rate at which the MH algorithm learns about the posterior distribution of interest.

In this example the unknown quantity $\theta = \sigma^2$ was real-valued, but there's nothing in the MH method that requires this; in principle it works equally well when θ is a vector of any finite dimension (look back at the algorithm in (15) to verify this).

Notice, crucially, that to implement this algorithm you only need to know how to calculate $p(\theta|y)$ up to a constant multiple, since any such constant will cancel in computing the acceptance probability (15)—thus you're free to work with unnormalized versions of $p(\theta|y)$, which is a great advantage in practice.

There's even more flexibility in this algorithm than might first appear: it's often possible to identify a set A of auxiliary variables—typically these are latent (unobserved) quantities—to be sampled along with the parameters, which have the property that they improve the mixing of the MCMC output (even though extra time is spent in sampling them).

When the set (θ, A) of quantities to be sampled is a vector of length k, there is additional flexibility: you can block update all of (θ, A) at once, or with appropriate modifications of the acceptance probability you can divide (θ, A) up into components, say $(\theta, A) = (\lambda_1, \ldots, \lambda_l)$, and update the components one at a time (as Metropolis et al. originally proposed in 1953).

The idea in this component-by-component version of the algorithm, which

Gilks et al. (1996) call single-component MH sampling, is to have k different proposal distributions, one for each component of θ.

Each iteration of the algorithm (indexed as usual by t) has k steps, indexed by i; at the beginning of iteration t you scan along, updating λ_1 first, then λ_2, and so on until you've updated λ_k, which concludes iteration t.

Let $\lambda_{t,i}$ stand for the current state of component i at the end of iteration t, and let λ_{-i} stand for the (θ, A) vector with component i omitted (the notation gets awkward here; it can't be helped).

The proposal distribution $g_i(\lambda_i^* | \lambda_{t,i}, \lambda_{t,-i}, y)$ for component i is allowed to depend on the most recent versions of all components of (θ, A); here $\lambda_{t,-i}$ is the current state of λ_{-i} after step $(i-1)$ of iteration t is finished, so that components 1 through $(i-1)$ have been updated but not the rest.

3.3.2 Gibbs sampling

The acceptance probability for the proposed move to λ_i^* that creates the correct equilibrium distribution turns out to be

$$\alpha_{MH}(\lambda_i^* | \lambda_{t,-i}, \lambda_{t,i}, y) = \min \left[1, \frac{p(\lambda_i^* | \lambda_{t,-i}, y) \, g_i(\lambda_{t,i} | \lambda_i^*, \lambda_{t,-i}, y)}{p(\lambda_{t,i} | \lambda_{t,-i}, y) \, g_i(\lambda_i^* | \lambda_{t,i}, \lambda_{t,-i}, y)} \right]. \qquad (3.19)$$

The distribution $p(\lambda_i | \lambda_{-i}, y)$ appearing in (19), which is called the full conditional distribution for λ_i, has a natural interpretation: it represents the posterior distribution for the relevant portion of (θ, A) given y and the rest of (θ, A).

The full conditional distributions act like building blocks in constructing the complete posterior distribution $p(\theta|y)$, in the sense that any multivariate distribution is uniquely determined by its set of full conditionals (Besag (1974)).

An important special case of single-component MH sampling arises when the proposal distribution $g_i(\lambda_i^* | \lambda_{t,i}, \lambda_{t,-i}, y)$ for component i is chosen to be the full conditional $p(\lambda_i^* | \lambda_{t,-i}, y)$ for λ_i: you can see from (19) that when this choice is made a glorious cancellation occurs and the acceptance probability is 1.

This is Gibbs sampling, independently (re)discovered by Geman and Geman (1984): the Gibbs recipe is to sample from the full conditionals and accept all proposed moves.

Even though it's just a version of MH, Gibbs sampling is important enough to merit a summary of its own.

Single-element Gibbs sampling, in which each real-valued coordinate ($\theta_1, \ldots, \theta_k$) gets updated in turn, is probably the most frequent way Gibbs sampling gets used, so that's what I'll summarize ((20) details Gibbs sampling in the case with no auxiliary variables A, but the algorithm works equally well when θ is replaced by (θ, A) in the summary).

Algorithm (Single-element Gibbs sampling). To con- struct a Markov chain whose equilibrium distribution is $p(\theta|y)$ with $\theta = (\theta_1, \ldots, \theta_k)$,

```
    Initialize $\theta^*_{0,1}, \ldots, \theta^*_{0,k}$; $t \leftarrow 0$
    Repeat {
        Sample $\theta^*_{t+1,1} \sim p(\theta_1|y, \theta^*_{t,2}, \theta^*_{t,3}, \theta^*_{t,4}, \ldots, \theta^*_{t,k})$
        Sample $\theta^*_{t+1,2} \sim p(\theta_2|y, \theta^*_{t+1,1}, \theta^*_{t,3}, \theta^*_{t,4}, \ldots, \theta^*_{t,k})$
        Sample $\theta^*_{t+1,3} \sim p(\theta_3|y, \theta^*_{t+1,1}, \theta^*_{t+1,2}, \theta^*_{t,4}, \ldots, \theta^*_{t,k})$
          ⋮         ⋮         ⋮         ⋮         ⋮
          ⋮
        Sample $\theta^*_{t+1,k} \sim p(\theta_k|y, \theta^*_{t+1,1}, \theta^*_{t+1,2}, \theta^*_{t+1,3}, \ldots, \theta^*_{t+1,k-1})$
        $t \leftarrow (t+1)$
    }
```
(3.20)

We noted from the predictive plot toward the end of part 2 of the lecture notes that the Gaussian model for the NB10 data was inadequate: the tails of the data distribution are too heavy for the Gaussian.

It was also clear from the normal qqplot that the data are symmetric.

This suggests thinking of the NB10 data values y_i as like draws from a t distribution with fairly small degrees of freedom ν.

One way to write this model is

$$\begin{array}{rcl} (\mu, \sigma^2, \nu) & \sim & p(\mu, \sigma^2, \nu) \\ (y_i|\mu, \sigma^2, \nu) & \stackrel{\text{IID}}{\sim} & t_\nu(\mu, \sigma^2), \end{array} \tag{3.21}$$

where $t_\nu(\mu, \sigma^2)$ denotes the scaled t-distribution with mean μ, scale parameter σ^2, and shape parameter ν.

This distribution has variance $\sigma^2 \left(\frac{\nu}{\nu-2} \right)$ for $\nu > 2$ (so that shape and scale are mixed up, or confounded in $t_\nu(\mu, \sigma^2)$) and may be thought of as the distribution of the quantity $\mu + \sigma\, e$, where e is a draw from the standard t distribution that is tabled at the back of all introductory statistics books.

Model expansion. However, a better way to think about model (21) is as follows.

Example: the NB10 Data. Recall from the posterior

It's a fact from basic distribution theory, probably of more interest to Bayesians than frequentists, that the t distribution is an Inverse Gamma mixture of Gaussians.

This just means that to generate a t random quantity you can first draw from an Inverse Gamma distribution and then draw from a Gaussian conditional on what you got from the Inverse Gamma.

(As noted in homework 2, $\lambda \sim \Gamma^{-1}(\alpha, \beta)$ just means that $\lambda^{-1} = \frac{1}{\lambda} \sim \Gamma(\alpha, \beta)$).

In more detail, $(y|\mu, \sigma^2, \nu) \sim t_\nu(\mu, \sigma^2)$ is the same as the hierarchical model

$$\begin{array}{rcl} (\lambda|\nu) & \sim & \Gamma^{-1}\left(\frac{\nu}{2}, \frac{\nu}{2}\right) \\ (y|\mu, \sigma^2, \lambda) & \sim & N\left(\mu, \lambda\, \sigma^2\right). \end{array} \tag{3.22}$$

Putting this together with the conjugate prior for μ and σ^2 we looked at earlier in the Gaussian model gives the following HM for the NB10 data:

$$\begin{array}{rcl} \nu & \sim & p(\nu) \\ \sigma^2 & \sim & \text{SI-}\chi^2\left(\nu_0, \sigma_0^2\right) \\ \left(\mu|\sigma^2\right) & \sim & N\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right) \\ (\lambda_i|\nu) & \overset{\text{IID}}{\sim} & \Gamma^{-1}\left(\frac{\nu}{2}, \frac{\nu}{2}\right) \\ \left(y_i|\mu, \sigma^2, \lambda_i\right) & \overset{\text{indep}}{\sim} & N\left(\mu, \lambda_i\, \sigma^2\right). \end{array} \tag{3.23}$$

Remembering also from introductory statistics that the Gaussian distribution is the limit of the t family as $\nu \to \infty$, you can see that the idea here has been to expand the Gaussian model by embedding it in the richer t family, of which it's a special case with $\nu = \infty$.

Model expansion is often the best way to deal with uncertainty in the modeling process: when you find deficiencies of the current model, embed it in a richer class, with the model expansion in directions suggested by the deficiencies (we'll also see this method in action again later).

Implementing Gibbs: the MCMC dataset. Imagine trying to do Gibbs sampling on model (21), with the parameter vector $\theta = (\mu, \sigma^2, \nu)$.

Carrying out the iterative program described in (20) above would produce the following MCMC Dataset:

Iteration	Phase	μ	σ^2	ν
0	Initializing	μ_0	σ_0^2	ν_0
1	Burn-In	$\mu_1(y, \sigma_0^2, \nu_0)$	$\sigma_1^2(y, \mu_1, \nu_0)$	$\nu_1(y, \mu_1, \sigma_1^2)$
2	Burn-In	$\mu_2(y, \sigma_1^2, \nu_1)$	$\sigma_2^2(y, \mu_2, \nu_1)$	$\nu_1(y, \mu_2, \sigma_2^2)$
·	·	·	·	·
b	Burn-In	μ_b	σ_b^2	ν_b
$(b+1)$	Monitoring	μ_{b+1}	σ_{b+1}^2	ν_{b+1}
$(b+2)$	Monitoring	μ_{b+2}	σ_{b+2}^2	ν_{b+2}
·	·	·	·	·
$(b+m)$	Monitoring	μ_{b+m}	σ_{b+m}^2	ν_{b+m}

Looking at iterations 1 and 2 you can see that, in addition to y, the sampler makes use only of parameter values in the current row and the previous row (this illustrates the Markov character of the samples).

As we've seen above, at the end of the $(b+m)$ iterations, if you want (say) the marginal posterior for μ, $p(\mu|y)$, all you have to do is take the m values $\mu_{b+1}, \dots, \mu_{b+m}$ and summarize them in any ways that interest you: their sample mean is your simulation estimate of the posterior mean of μ, their sample histogram (or, better, their kernel density trace) is your simulation estimate of $p(\mu|y)$, and so on.

Practical issues: implementation details. (1) How do you figure out the full conditionals, and how do you sample from them?

(2) What should you use for initial values?

(3) How large should b and m be?

(4) More generally, how do you know when the chain has reached equilibrium?

Questions (3–4) fall under the heading of MCMC diagnostics, which I'll cover a bit later, and I'll address question (2) in the case studies below.

Computing the full conditionals. For a simple example of working out the full conditional distributions, consider the conjugate Gaussian model we looked at earlier:

$$\begin{aligned} \sigma^2 &\sim \text{SI-}\chi^2(\nu_0, \sigma_0^2) \\ (\mu|\sigma^2) &\sim N\left(\mu_0, \frac{\sigma^2}{\kappa_0}\right) \\ (Y_i|\mu, \sigma^2) &\overset{\text{IID}}{\sim} N\big(\mu, \sigma^2\big) \, . \end{aligned} \qquad (3.24)$$

The full conditional distribution for μ in this model is $p(\mu|\sigma^2, y)$, considered as a function of μ for fixed σ^2 and y—but this is just

$$\begin{aligned} p(\mu|\sigma^2, y) &= \frac{p(\mu, \sigma^2, y)}{p(\sigma^2, y)} \\ &= c\, p(\mu, \sigma^2, y) \qquad (3.25) \\ &= c\, p(\sigma^2)\, p(\mu|\sigma^2)\, p(y|\mu, \sigma^2) \\ &= c \, \exp\left[-\frac{\kappa_0}{2\sigma^2} (\mu - \mu_0)^2 \right] \prod_{i=1}^n \exp\left[-\frac{1}{2\sigma^2} (y_i - \mu)^2 \right]. \end{aligned}$$

Full conditionals. From this

$$p(\mu|\sigma^2, y) = c \, \exp\left[-\frac{\kappa_0}{2\sigma^2} (\mu - \mu_0)^2 \right] \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2 \right].$$

Expanding out the squares, collecting powers of μ, and completing the square in μ gives

$$p(\mu|\sigma^2, y) = c \, \exp\left[-\frac{\kappa_0 + n}{2\sigma^2} \left(\mu - \frac{\kappa_0 \mu_0 + n\bar{y}}{\kappa_0 + n} \right)^2 \right], \qquad (3.26)$$

from which it's clear that the full conditional for μ in model (24) is

$$(\mu|\sigma^2, y) \sim N\left(\frac{\kappa_0 \mu_0 + n\bar{y}}{\kappa_0 + n}, \frac{\sigma^2}{\kappa_0 + n} \right). \qquad (3.27)$$

Similarly, the full conditional for σ^2 in this model, $p(\sigma^2|\mu, y)$, considered as a function of σ^2 for fixed μ and y, is just

$$\begin{aligned} p(\sigma^2|\mu, y) &= \frac{p(\sigma^2, \mu, y)}{p(\mu, y)} \\ &= c\, p(\sigma^2, \mu, y) \qquad (3.28) \\ &= c\, p(\sigma^2)\, p(\mu|\sigma^2)\, p(y|\mu, \sigma^2) \\ &= c\, \left(\sigma^2\right)^{-\left(1 + \frac{1}{2}\nu_0\right)} \exp\left(\frac{-\nu_0\, \sigma_0^2}{2\sigma^2} \right) \cdot \\ &\qquad \left(\sigma^2\right)^{-\frac{1}{2}} \exp\left[-\frac{\kappa_0}{2\sigma^2} (\mu - \mu_0)^2 \right] \cdot \\ &\qquad \left(\sigma^2\right)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2 \right]. \end{aligned}$$

When this is simplified you get

$$p(\sigma^2|\mu, y) = c \, \left(\sigma^2\right)^{-\left(1+\frac{\nu_0+1+n}{2}\right)} \, \exp\left[-\frac{\nu_0\sigma_0^2 + \kappa_0(\mu-\mu_0)^2 + n s_\mu^2}{2\sigma^2}\right],$$

where $s_\mu^2 = \frac{1}{n}\sum_{i=1}^n (y_i - \mu)^2$.

From the form of this distribution it becomes clear that

$$(\sigma^2|\mu, y) \sim \text{SI-}\chi^2\left(\nu_0 + 1 + n, \frac{\nu_0\sigma_0^2 + \kappa_0(\mu-\mu_0)^2 + n s_\mu^2}{\nu_0 + 1 + n}\right). \tag{3.29}$$

Thus in conjugate situations the full conditional distributions have conjugate forms, which are tedious but straightforward to compute.

Both the directness and the tedium of this calculation suggest that it should be possible to write a computer program to work out the full conditionals for you, and indeed at least two such programs now exist:

- `BUGS`, a fairly general-purpose Gibbs sampling program produced by David Spiegelhalter and others at the MRC Biostatistics Unit in Cambridge, UK (Spiegelhalter et al., 1997), and
- `MLwiN`, a program that does both maximum-likelihood and Bayesian calculations in hierarchical (multilevel) models (Rasbash et al. 2000).

`BUGS` runs under Unix or DOS in a wide variety of hardware configurations, and a `Windows` version called `WinBUGS` is also available; we'll look here at both Unix `BUGS` and `WinBUGS` (together with `MLwiN` if there's time).

`BUGS` and `WinBUGS` are available for free downloading at

`www.mrc-bsu.cam.ac.uk/bugs`;

`MLwiN` has a nominal charge and can be downloaded from the web page of the Multilevel Models Project,

`multilevel.ioe.ac.uk`

3.3.3 Why the Metropolis algorithm works

Here's a sketch of the crucial part of the proof, based on an argument in Gamerman (1997), of the validity of the Metropolis algorithm, in the case of a discrete (finite or countably infinite) state space S (see chapter 1 in Gilks et al. 1996 for a proof sketch when S is continuous).

It will be helpful in looking at the proof sketch to specialize the Markov chain notation we've been using so far to the case of discrete state spaces, as follows.

A stochastic process $\{\theta_t^*, t \in T\}, T = \{0, 1, \ldots\}$ on a discrete state space S is a Markov chain iff

$$P(\theta_{t+1}^* = y | \theta_t^* = x, \theta_{t-1}^* = x_{n-1}, \ldots, \theta_0^* = x_0) = P(\theta_{t+1}^* = y | \theta_t^* = x) \quad (3.30)$$

for all $t = 0, 1, \ldots$ and $x_0, \ldots, x_{t-1}, x, y \in S$.

In general $P(\theta_{t+1}^* = y | \theta_t^* = x)$ depends on x, y, and t, but if the probability of transitioning from x to y at time t is constant in t things will clearly be simpler; such chains are called homogeneous (confusingly, some sources call them stationary, but that terminology seems well worth avoiding).

The random walk described earlier is obviously a homogeneous Markov chain, and so are any Markov chains generated by the MH algorithm; I'll assume homogeneity in what follows.

Under homogeneity it makes sense to talk about the transition probability

$$P(x, y) = P(\theta_{t+1}^* = y | \theta_t^* = x) \quad \text{for all } t, \quad (3.31)$$

which satisfies

$$P(x, y) \geq 0 \text{ for all } x, y \in S \quad \text{and} \quad \sum_{y \in S} P(x, y) = 1 \text{ for all } x \in S. \quad (3.32)$$

When S is discrete a transition matrix P can be defined with element (i, j) given by $P(x_i, x_j)$, where x_i is the ith element in S according to whatever numbering convention you want to use (the second part of (32) implies that the row sums of such a matrix are always 1; this is the defining condition for a stochastic matrix).

Suppose the chain is initialized at time 0 by making a draw from a probability distribution $\pi_0(x) = P(\theta_0^* = x)$ on S (deterministically starting it at some point x_0 is a special case of this); then the probability distribution $\pi_1(y)$ for where it will be at time 1 is

$$\begin{aligned} \pi_1(y) &= P(\theta_1^* = y) \\ &= \sum_{x \in S} P(\theta_0^* = x, \theta_1^* = y) \\ &= \sum_{x \in S} P(\theta_0^* = x) \, P(\theta_1^* = y | \theta_0^* = x) \qquad (3.33) \\ &= \sum_{x \in S} \pi_0(x) \, P(x, y), \end{aligned}$$

which can be written in vector and matrix notation as

$$\pi_1 = \pi_0 \, P, \tag{3.34}$$

where π_0 and π_1 are regarded as row vectors.

Then by the same reasoning

$$\pi_2 = \pi_1 \, P = (\pi_0 \, P) P = \pi_0 \, P^2, \tag{3.35}$$

and in general

$$\pi_t = \pi_0 \, P^t. \tag{3.36}$$

For simple Markov chains this can be used to work out the long-run behavior of the chain as $t \rightarrow \infty$, but this becomes algebraically prohibitive as the transition behavior of the chain increases in complexity.

In any case for ergodic Markov chains the limiting behavior $\pi(y)$ is independent of π_0 and turns out to be characterized by the relation

$$\pi(y) = \sum_{x \in S} \pi(x) \, P(x, y), \quad \text{or} \quad \pi = \pi P, \tag{3.37}$$

which defines the stationary distribution π of the chain.

As we've seen above, the hard bit in verifying the validity of the Metropolis algorithm is demonstrating that the Markov chain created by running the algorithm has the correct stationary distribution, namely the target posterior $p(\theta|y)$; one way to do this is the following.

It's possible to imagine running any homogeneous Markov chain $\{\theta_t^*, t = 0, 1, \ldots\}$ with transition probabilities $P(x, y)$ backwards in time.

This new reverse-time stochastic process can be shown also to be a Markov chain, although it may not be homogeneous.

If it is homogeneous, and if in addition the reverse-time process has the same transition probabilities as the original process, the Markov chain is said to be reversible; all such chains satisfy the detailed balance equation

$$\pi(x) \, P(x, y) = \pi(y) \, P(y, x) \text{ for all } x, y \in S. \tag{3.38}$$

It turns out that if there's a distribution π satisfying (38) for an irreducible Markov chain, then the chain is positive recurrent (and therefore ergodic) and reversible, and its stationary distribution is π (sum (38) over y to get (37)).

In other words, if you're trying to create an ergodic Markov chain and you want it to have some target stationary distribution π, one way to achieve this goal is to ensure that the chain is irreducible and that its transition probabilities $P(x, y)$ satisfy detailed balance with respect to the target π.

Any reasonable proposal distribution in the Metropolis algorithm will yield an irreducible Markov chain, so the interesting bit is to verify detailed balance; the argument proceeds as follows.

Consider a given target distribution p_x on S; we're trying to construct a Markov chain with stationary distribution π such that $\pi(x) = p_x$ for all $x \in S$.

The Metropolis algorithm—(15), with the special case of the acceptance probabilities (14) reducing to the simpler form $\min\left[1, \frac{p(\theta^*|y)}{p(\theta_t|y)}\right]$ by the assumption of a symmetric proposal distribution—actually involves two related Markov chains: the (less interesting) chain that you could create by accepting all proposed moves, and the (more interesting) chain created by the actual algorithm.

Let $Q(x, y)$ be any irreducible transition matrix on S such that $Q(x, y) = Q(y, x)$ for all $x, y \in S$; this is the transition matrix for the (less interesting) chain induced by the proposal distribution.

Define the (more interesting) chain $\{\theta_t^*, t = 0, 1, \ldots\}$ (the actual Metropolis chain) as having transitions from x to y proposed according to $Q(x, y)$, except that the proposed value for θ_{t+1}^* is accepted with probability $\min\left(1, \frac{p_y}{p_x}\right)$ and rejected otherwise, leaving the chain in state x.

The transition probabilities $P(x, y)$ for the Metropolis chain are as follows: for $y \neq x$, and denoting by A_{xy} the event that the proposed move from x to y is accepted,

$$\begin{aligned} P(x, y) &= P\left(\theta_{t+1}^* = y | \theta_t^* = x\right) \\ &= P\left(\theta_{t+1}^* = y, A_{xy} | \theta_t^* = x\right) + P\left(\theta_{t+1}^* = y, \text{not } A_{xy} | \theta_t^* = x\right) \\ &= P\left(\theta_{t+1}^* = y | A_{xy}, \theta_t^* = x\right)\, P(A_{xy} | \theta_t^* = x) \qquad (3.39) \\ &= Q(x, y) \min\left(1, \frac{p_y}{p_x}\right). \end{aligned}$$

A similar calculation shows that for $y = x$

$$P(x, x) = Q(x, x) + \sum_{y \neq x} Q(x, y) \left[1 - \min\left(1, \frac{p_y}{p_x}\right)\right], \qquad (3.40)$$

but this is not needed to show detailed balance because (38) is trivially satisfied when $y = x$.

When $y \neq x$ there are two cases: $p_y \geq p_x > 0$ (I'll give details in this case) and $0 < p_y < p_x$ (the other case follows analogously).

If $p_y \geq p_x$, note that $\min\left(1, \frac{p_y}{p_x}\right) = 1$ and

$$\min\left(1, \frac{p_x}{p_y}\right) p_y = \min\left(p_y, \frac{p_x}{p_y} p_y\right) = \min(p_y, p_x) = p_x;$$

then

$$\begin{aligned} p_x\, P(x,y) &= p_x\, Q(x,y) \min\left(1, \frac{p_y}{p_x}\right) = p_x\, Q(x,y) \\ &= p_x\, Q(y,x) = Q(y,x) \min\left(1, \frac{p_x}{p_y}\right) p_y \qquad (3.41) \\ &= p_y\, P(y,x) \end{aligned}$$

and the proof of detailed balance, and with it the validity of the Metropolis algorithm, is complete.

3.3.4 Directed acyclic graphs

BUGS achieves its generality by means of two ideas:

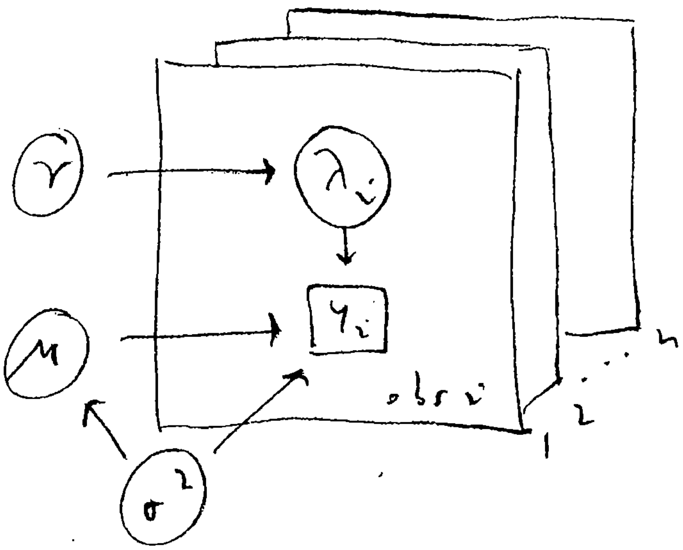
(1) Viewing Bayesian models as directed (acyclic) graphs (DAGs).

The conditional independence nature of Bayesian hierarchical models—in which quantities in the model depend on things one layer higher in the hierarchy but no higher (e.g., in the NB10 t model (23) the y_i depend on $(\mu, \sigma^2, \lambda_i)$ but not on ν)—lends itself to thinking of all quantities in such models as nodes in a directed graph.

A DAG can be thought of as a picture in which known and unknown quantities are represented either by squares (for knowns) or circles (for unknowns), connected by arrows (from the parents to the children) that indicate the direction of the stochastic dependence.

The acyclic assumption means that by following the directions of the arrows it's impossible to return to a node once you've left it, and stacked sheets indicate repetition (e.g., across conditionally IID data values).

Here's a DAG for the NB10 model based on the t distribution.



Adaptive rejection sampling. (2) Employing adaptive-rejection sampling (Gilks and Wild, 1992) to generate the random draws from the full conditional distributions, when they don't have simple recognizable forms.

As we've seen, rejection sampling is a general method for sampling from a given density $p(\theta|y)$, which requires an envelope function G which dominates p (chosen so that $G(\theta|y) \geq p(\theta|y)$ for all θ).

A restatement of the algorithm for normalized G (e.g., Ripley 1987) is

```
Repeat {
  Sample a point theta from G ( . | y );
  Sample a Uniform( 0, 1 ) random variable U;
  If U <= p ( theta | y ) / G ( theta | y ) accept theta;
  }
until one theta is accepted.
```

If $p(\theta|y)$ is expensive to evaluate, time can be saved by identifying squeezing functions $a(\theta|y)$ and $b(\theta|y)$ with $b(\theta|y) \leq p(\theta|y) \leq a(\theta|y)$; to use these, replace the acceptance step above (line 4 in the algorithm) by

```
If U > a( theta | y ) / G( theta | y ) reject theta;
  else if U <= b( theta | y ) / G( theta | y ) accept theta;
  else if U <= p( theta | y ) / G( theta | y ) accept theta.
```

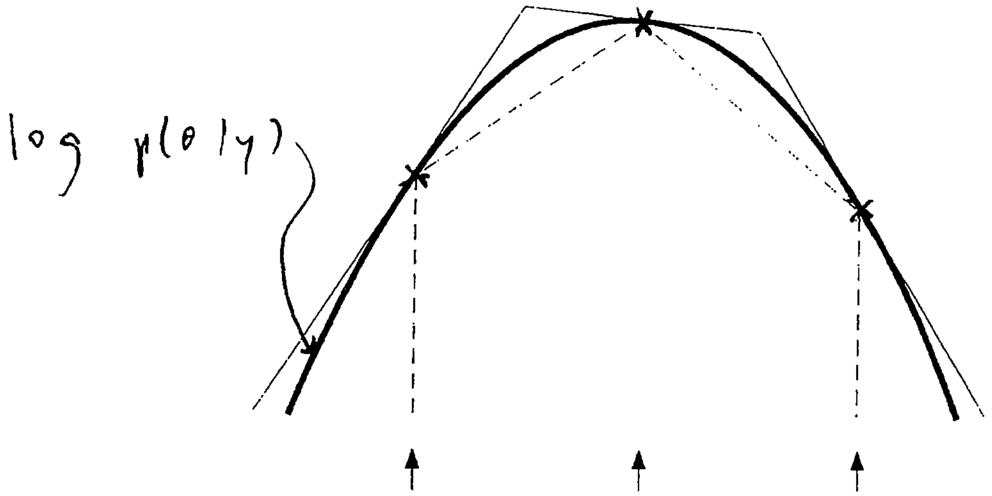
Adaptive rejection sampling (ARS; Gilks and Wild 1992) is a relatively efficient method of adaptive envelope construction that works as a basis for Gibbs sampling if all of the full conditional densities are log concave (formally, a function $p(\theta|y)$ of a vector argument θ is log concave if the determinant of

$$\frac{d^2 \log g}{dy\, dy^T} \tag{3.42}$$

is non-positive).

For univariate θ the idea (see the figure on p. 14) is that an envelope function $\log G_S(\theta|y)$ can be constructed on the log scale by drawing tangents to $\log p(\theta|y)$ at each point in a given set of θ values S.

An envelope between any two adjacent points is then constructed from the tangents at each end of the interval defined by the points:



the tangents form the envelope function on the log scale, & the secants form a lower squeezing function

The envelope is linear on the log scale, so rejection sampling on the original scale is performed with scaled exponential distributions (as noted earlier, this can be done efficiently), and you get a lower squeezing function for free.

The useful thing about this idea is that the envelope can be constructed adaptively, by adding points to S as new θ are sampled—thus the envelope improves as more samples are drawn.

`BUGS` uses a hierarchy of methods to sample from the full conditionals: it first tries to verify conjugacy; if that fails it then tries to verify log concavity of the full conditionals and uses ARS if so; and if that fails "classic" `BUGS` quits and `winBUGS` switches over to (non-Gibbs) Metropolis-Hastings sampling.

Log concavity includes many, but not all, distributions occurring in standard models, e.g., a uniform $U(a, b)$ prior on the degrees of freedom parameter ν in the NB10 t model fails log-concavity.

In classic `BUGS` such distributions must be discretized (`BUGS` allows discrete variables to have 500 possible values, which generally leads to quite accurate approximations).

Running classic `BUGS`. You make four kinds of files:

(1) a program file, with suffix `.bug`, containing the specification of your model;

(2) one or more data files, with suffix `.dat`;

(3) an initial values file, with suffix `.in`; and

(4) a command file with suffix `.cmd`, containing instructions that specify the burn-in and monitoring phases.

Here's the data file in the NB10 example.

```
list( y = c(409., 400., 406., 399., 402., 406., 401., 403.,
  401., 403., 398., 403., 407., 402., 401., 399., 400., 401.,
                   [ several lines omitted ]
  401., 407., 412., 375., 409., 406., 398., 406., 403., 404.),
  grid = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                   [ several lines omitted ]
    1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 )
```

And here are the `BUGS` program (`.bug`) and initial values (`.in`) files in the NB10 example.

```
model nb10;

const
```

```
  n = 100, g = 100;

var

  mu, tau, u, grid[ g ], nu, y[ n ], sigma;

data in "nb10.dat";
inits in "nb10.in";

{

  mu ~ dnorm( 0.0, 1.0E-6 );
  tau ~ dgamma( 0.001, 0.001 );  # specifying the
  u ~ dcat( grid[ ] );           # prior distributions
  nu <- 2.0 + u / 10.0;

  for ( i in 1:n ) {
                                 # specifying the
    y[ i ] ~ dt( mu, tau, nu );  # likelihood

  }
                                 # defining any other
  sigma <- 1.0 / sqrt( tau );    # quantities to be
                                 # monitored
}
             Initial values

list( mu = 404.59, u = 30, tau = 0.04,
  seed = 90915314 )
```

Implementation details. Here are two BUGS command (.cmd) files in the NB10 example.

```
compile( "nb10-1.bug" )    |   compile( "nb10-1.bug" )
update( 1000 )             |   update( 2000 )
monitor( mu )              |   monitor( mu, 8 )
monitor( sigma )           |   monitor( sigma, 8 )
monitor( nu )              |   monitor( nu, 8 )
update( 5000 )             |   update( 40000 )
q( )                       |   q( )
```

Some Details. (1) The priors: (a) I want to use a diffuse prior for μ, since I don't know anything about the true weight of NB10 *a priori*.

The phrase `mu ~ dnorm( 0.0, 1.0E-6 )` in `BUGS`-speak means that μ has a Gaussian prior with mean 0 and precision 10^{-6}, i.e.,

$$\text{SD} = 1/\sqrt{\text{precision}} = 1,000,$$

i.e., as far as I'm concerned *a priori* μ could be just about anywhere between $-3,000$ and $3,000$.

(b) Similarly I want a diffuse prior for σ^2, or equivalently for the precision $\tau = \frac{1}{\sigma^2}$.

As we saw in the Poisson LOS case study, one popular conventional choice is $\tau \sim \Gamma(\epsilon, \epsilon)$ for a small ϵ like 0.001, which in `BUGS`-speak is said `tau ~ dgamma( 0.001, 0.001 )`.

This distribution is very close to flat over an extremely wide range of the interval $(0, \infty)$, although it does have a nasty spike at 0 (as $\tau \downarrow 0, \Gamma(\epsilon, \epsilon)(\tau) \uparrow \infty$).

As noted earlier, the idea behind diffuse priors is to make them approximately constant in the region in which the likelihood is appreciable.

For this purpose it's useful to remember what the frequentist answers for μ and σ would be, at least in the Gaussian model we looked at earlier.

Recall that the 95% confidence interval (CI) for μ came out $(403.3, 405.9)$, so you can guess that the likelihood for μ would be non-negligible in the range from (say) 402 to 407.

Diffuse priors. As for σ (or σ^2 or τ), in the model $(Y_i|\mu, \sigma^2) \stackrel{\text{IID}}{\sim} N(\mu, \sigma^2)$, it's a standard result from frequentist distribution theory that in repeated sampling

$$\frac{(n-1)s^2}{\sigma^2} \sim \chi^2_{n-1}, \tag{3.43}$$

where $s^2 = \frac{1}{n-1}\sum_{i=1}^n (y_i - \bar{y})^2$ is random and σ^2 is fixed, from which

$$P_f\left[A \leq \frac{(n-1)s^2}{\sigma^2} \leq B\right] = 0.99 \tag{3.44}$$

for A, B such that

$$P_f\big(\chi^2_{n-1} \leq A\big) = P_f\big(\chi^2_{n-1} \geq B\big) = 0.005. \tag{3.45}$$

Thus, using Neyman's confidence trick,

$$P_f\left[\frac{(n-1)s^2}{B} \le \sigma^2 \le \frac{(n-1)s^2}{A}\right] = 0.99; \qquad (3.46)$$

in other words, $\left[\frac{(n-1)s^2}{B}, \frac{(n-1)s^2}{A}\right]$ is a 99% confidence interval for σ^2.

With the NB10 data $n = 100$ and $s^2 = 41.82$, and you can use R to do this analysis:

```
> y
 [1] 409 400 406 399 402 406 401 403 401 403 398 403 407 402 401
[16] 399 400 401 405 402 408 399 399 402 399 397 407 401 399 401
[31] 403 400 410 401 407 423 406 406 402 405 405 409 399 402 407
[46] 406 413 409 404 402 404 406 407 405 411 410 410 410 401 402
[61] 404 405 392 407 406 404 403 408 404 407 412 406 409 400 408
[76] 404 401 404 408 406 408 406 401 412 393 437 418 415 404 401
[91] 401 407 412 375 409 406 398 406 403 404

> print( n <- length( y ) )

[1] 100

> print( s2 <- var( y ) )

[1] 41.8201

> qchisq( 0.005, 99 )

[1] 66.5101

> qchisq( 0.995, 99 )

[1] 138.9868

> ( n - 1 ) * s2 / qchisq( 0.995, 99 )

[1] 29.78837

> ( n - 1 ) * s2 / qchisq( 0.005, 99 )
```

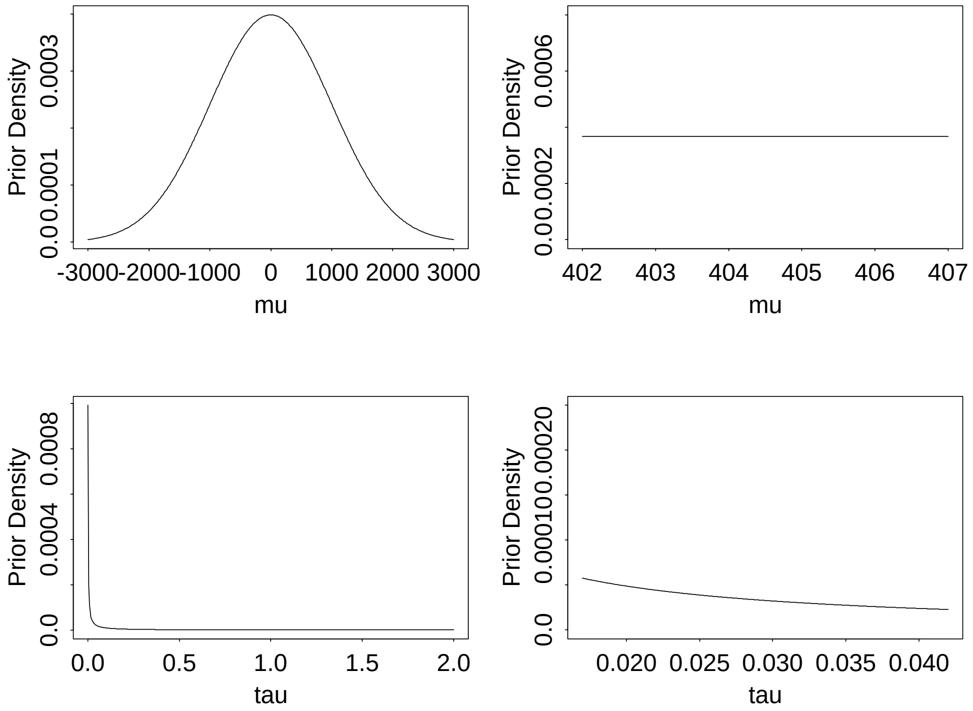


Figure 3.4: *Priors for μ (top) and τ (bottom), plotted globally (left) and locally in the region in which the likelihood is appreciable (right).*

```
[1] 62.24904

> qchisq( 0.005, 99 ) / ( ( n - 1 ) * s2 )

[1] 0.01606451

> qchisq( 0.995, 99 ) / ( ( n - 1 ) * s2 )

[1] 0.03357015
```

So the conclusion is that the likelihood for $\tau = \frac{1}{\sigma^2}$ should be non-negligible roughly in the region from about 0.015 to 0.035.

The figure below plots the prior distributions for μ and τ and verifies their diffuseness in the relevant regions.

1. (c) As for the prior on ν, you can tell from the normal qqplot of the NB10 data that the degrees of freedom parameter in the underlying t distribution is fairly small.

I'm going to use a uniform $U(c_1, c_2)$ prior, where c_1 is small but not too small (as noted earlier, with $\nu < 2$ the variance is infinite, which is not realistic as a model for actual data) and c_2 is big enough not to truncate the likelihood function (experience tells me that $c_2 = 12$ will suffice; this can also be determined via MCMC experimentation).

Classic `BUGS` can't figure out how to sample from a continuous $U(c_1, c_2)$ prior on ν, however, so instead I've used a discrete uniform prior on a $g =$ 100–point grid from 2.1 to 12.0 in steps of 0.1 (that's what `u ~ dcat( grid[ ] ); nu <- 2.0 + u / 10.0;` does when `grid[ ]` is a vector of 100 1s).

`WinBUGS` has a more elegant solution to this problem which we'll look at later.

(2) Initial Values. I can make fairly decent guesses at all the parameters as starting values for the Markov chain:

(a) The sample mean is 404.59, which should be close to the posterior mean for μ in the t model;

(b) I'm just going to guess that ν is around 5, which is specified by taking $u = 30$.

(c) Earlier I said that $V[t_\nu(\mu, \sigma^2)] = \sigma^2 \left(\frac{\nu}{\nu - 2} \right)$, so with $\nu \doteq 5$ and a sample variance of 41.82 you get $\tau = \frac{1}{\sigma^2} \doteq 0.04$.

A Running Strategy. With a problem like this with relatively few parameters, I often start off with a burn-in of 1,000 and a monitoring run of 5,000 and then look at the MCMC diagnostics (to be covered below).

The left-hand part of the table at the top of page 54 shows the `BUGS` commands that carry out this run.

You can either type in these commands interactively one at a time at the keyboard or put them in a `.cmd` file and run `BUGS` in the background (this is useful when you're interested in simulating the Bayesian analysis of many similar datasets for research purposes; the latest release of `WinBUGS` now also has this capability).

This run took about 5 minutes on a not particularly fast workstation (a `SunBlade` 150 running `Solaris Unix` at 600 Mhz), which is actually fairly slow for a 3-parameter problem (the discrete grid sampling for ν slows things down a lot).

```
  rosalind 61> bugs
```

```
Welcome to BUGS on 20 th Feb 2003  at 16:38:29
BUGS : Copyright (c) 1992 .. 1995 MRC Biostatistics Unit.
All rights reserved.
Version 0.603 for unix systems.
For general release: please see documentation for disclaimer.
The support of the Economic and Social Research Council (UK)
is gratefully acknowledged.

Bugs>compile( "nb10-1.bug" )

model nb10;

  [here BUGS just echoes the model shown on page 53]

}

Parsing model declarations.
Loading data value file(s).
Loading initial value file(s).
Parsing model specification.
Checking model graph for directed cycles.
Generating code.
Generating sampling distributions.
Checking model specification.
Choosing update methods.
compilation took  00:00:00

Bugs> update( 1000 )

      time for    1000   updates was  00:00:47

Bugs>monitor( mu )

Bugs>monitor( sigma )

Bugs>monitor( nu )

Bugs>update( 5000 )
```

```
      time for    5000   updates was  00:03:56
Bugs>q( )       # (output file created; more about this later)
```

3.3.5 Practical MCMC monitoring and convergence diagnostics

Remember questions (3) and (4) awhile ago?—(3) How large should b and m be? (4) More generally, how do you know when the chain has reached equilibrium?

A large body of research has grown up just in the last eight years or so to answer these questions (some good reviews are available in Gelman et al. 2003, Gilks et al. 1995, and Cowles and Carlin 1996).

The theoretical bottom line is unpleasant: you can't ever be sure you've reached equilibrium, in the sense that every MCMC diagnostic invented so far has at least one example in which it failed to diagnose problems.

However, a collection of four of the best diagnostics has been brought together in a set of `R` functions called `CODA` by Best, Cowles, and Vines (1995) (downloadable from the `R` web site).

I will briefly discuss each of these in the context of the NB10 analysis.

Geweke (1992) proposed a simple diagnostic based on time series ideas.

Thinking of each column of the MCMC dataset as a time series (with iterations indexing time), he reasoned that, if the chain were in equilibrium, the means of the first (say) 10% and the last (say) 50% of the iterations should be nearly equal.

His diagnostic is a z-score for testing this equality, with a separate value for each quantity being monitored: Geweke z-scores a lot bigger than 2 in absolute value indicate that the mean level of the time series is still drifting, even after whatever burn-in you've already done.

```
GEWEKE CONVERGENCE DIAGNOSTIC (Z-score):
========================================

Iterations used = 1002:6001     Fraction in
Thinning interval = 1             1st window = 0.1
Sample size per chain = 5000    Fraction in
                                2nd window = 0.5

-+----------+-------------+-
 | VARIABLE |   bugs1     |
 | ======== |    =====     |
 |          |              |
 | mu       |    2.39      |
 | nu       |    1.78      |
 | sigma    |    1.14      |
 |          |              |
-+----------+-------------+-
```

Here for run 1 with the NB10 data (the left-hand set of commands in the table on p. 54) there is some evidence of nonstationarity with a burn-in of only 1,000 (although a z-value of 2.4 is not overwhelming).

Gelman-Rubin (1992) have suggested a diagnostic that looks for multimodality of the posterior distribution.

If the posterior has (say) two major modes that are far away from each other in parameter space, and you initialize the chain near one of the modes, you may never find the other one.

The idea is to run the chain two or more times from widely-dispersed starting points and see if you always converge to the same place.

Gelman and Rubin do what amounts to an analysis of variance within and between the chains, looking for evidence of large variability between them.

Gelman-Rubin shrink factors. "This comparison is used to estimate the factor by which the scale parameter of the marginal posterior distribution of each [quantity being monitored] might [shrink] if the chain were run to infinity" (Best et al., 1995).

The output is the 50% and 97.5% quantiles of the distributions of shrink factors, one for each quantity monitored.

If these quantiles are both close to 1.0 then there is little evidence of dispersion between the distributions to which the chains are converging.

```
GELMAN AND RUBIN 50% AND 97.5% SHRINK FACTORS:
==============================================

Iterations used for diagnostic  = 2501:5000
Thinning interval = 1
Sample size per chain = 5000

-+----------+----------------------------+-
 | VARIABLE |  Point est. 97.5% quantile  |
 | ======== |  ========== ==============  |
 |          |                             |
 | mu       |  1.00       1.00            |
 | nu       |  1.00       1.01            |
 | sigma    |  1.00       1.00            |
 |          |                             |
-+----------+-----------------------------+-
```

Here, with initial values as different as $(\mu, \tau, \nu) = (405.0, 0.1823, 5.0)$ and $(402.0, 0.03, 11.0)$ there is no evidence of multimodality at all.

(To be really safe I should run a number of additional chains—Gelman and Rubin (1992) give advice on how to generate the set of initial values to try—but with even modest sample sizes (like $n = 100$) the posterior in t models is unimodal so there would be no point in this case.)

Raftery-Lewis dependence factors. Raftery and Lewis (1992) suggested a diagnostic that directly helps to answer question (3)—How do you pick b and m?

The answer to this question depends on how accurate you want your posterior summaries to be, so Raftery and Lewis require you to input three values:

(a) Which quantiles of the marginal posteriors are you most interested in?

Usually the answer is the 2.5% and 97.5% points, since they're the basis of a 95% interval estimate.

(b) How close to the nominal levels would you like the estimated quantiles to be?

The `CODA` default is 0.5%, e.g., if the left-hand value of your 95% interval is supposed to be at the 2.5% point of the distribution, `CODA` will recommend a length of monitoring run so that the actual level of this quantile will be between 2.0% and 3.0%.

(NB This is sometimes more, and often less, Monte Carlo accuracy than you really need.)

(c) With what minimum probability do you want to achieve these accuracy goals? The default is 95%.

Having input these values, the output is of five kinds for each quantity monitored:

(a) A recommended thinning interval. When the Gibbs sampler is performing poorly people say the output is not mixing well, and what they mean is that the Markovian nature of the time series for each quantity has led to large positive serial autocorrelations in time, e.g., μ_{1000} depends highly on μ_{999}, μ_{998}, and so on.

This is another way to say that the random draws in the simulation process are not moving around the parameter space quickly.

When this happens, one way to reduce the autocorrelation is to run the chain a lot longer and only record every kth iteration—this is the thinning interval (NB this only reduces the autocorrelation of the saved MCMC data set; the underlying Markov chain is of course unaffected by this).

(b) A recommended length of burn-in to use, above and beyond whatever you've already done.

(c) A recommended total length of run N (including burn-in) to achieve the desired accuracy.

(d) A lower bound N_{min} on run length—what the minimum would have needed to be if the quantity in question had an IID time series instead of an autocorrelated series.

(e) And finally, the ratio $I = N/N_{min}$, which Raftery and Lewis call the dependence factor—values of I near 1 indicate good mixing.

```
RAFTERY AND LEWIS CONVERGENCE DIAGNOSTIC:
=========================================

Iterations used = 1001:6000
Thinning interval = 1
Sample size per chain = 5000

Quantile = 0.025
Accuracy = +/- 0.005
Probability = 0.95

-+----------+---------------------------------------------------+-
 |          |  Thin  Burn-in  Total   Lower bound  Dependence   |
 | VARIABLE |  (k)     (M)    (N)        (Nmin)    factor (I)   |
 | ======== |  ====  =======  =====   ===========  ==========   |
 |          |                                                   |
 | mu       |  1     3        4533    3746         1.21         |
 | nu       |  3     18       39720   3746         10.6         |
 | sigma    |  3     12       13308   3746         3.55         |
 |          |                                                   |
-+----------+---------------------------------------------------+-
```

Here μ is mixing well—5,000 iterations are sufficient to achieve the default accuracy goal—but σ and (especially) ν require longer monitoring periods: the recommendation is to run for about 40,000 iterations and store every third.

Heidelberger-Welch Diagnostic. Heidelberger and Welch (1983) propose a diagnostic approach that uses the Cramér-von Mises statistic to test for stationarity.

If overall stationarity fails for a given quantity being monitored, `CODA` discards the first 10% of the series for that quantity and recomputes the C-vonM statistic, continuing in this manner until only the final 50% of the data remain.

If stationarity still fails with the last half of the data then `CODA` reports overall failure of the stationarity test.

`CODA` also computes a half-width test, which tries to judge whether the portion of the series that passed the stationarity test is sufficient to estimate the posterior mean with a particular default accuracy (NB this default is often not stringent enough for careful numerical work).

Here the table below shows that the first run with the NB10 data clears the Heidelberger-Welch hurdle with ease.

Autocorrelations and Cross-correlations. `CODA` also computes the autocorrelations for each monitored quantity at lags from 1 to 50 and the cross-correlations between all of the variables.

As mentioned previously, the autocorrelation at lag k of a time series $\{\theta_t^*, t = 1, \ldots, m\}$ (e.g., Chatfield 1996) measures the extent to which the series at time $(t+k)$ and at time t are linearly related, for $k = 1, 2, \ldots$.

```
HEIDELBERGER AND WELCH STATIONARITY AND INTERVAL HALFWIDTH TESTS:
=================================================================

Precision of halfwidth test = 0.1

-+----------+-------------------------------------------------+-
 |          | Stationarity  # of iters.  # of iters.  C-vonM   |
 | VARIABLE |     test        to keep    to discard    stat.   |
 | ======== | ============  ===========  ===========  ======   |
 |          |                                                 |
 | mu       | passed        5000         0            0.126    |
 | nu       | passed        5000         0            0.349    |
 | sigma    | passed        5000         0            0.176    |
 |          |                                                  |
-+----------+--------------------------------------------------+-
 |          | Halfwidth                       |
 | VARIABLE |   test      Mean    Halfwidth   |
 | ======== | =========   ====    =========   |
 |          |                                 |
 | mu       | passed      404.00  0.0160      |
 | nu       | passed        3.75  0.1500      |
 | sigma    | passed        3.89  0.0344      |
 |          |                                 |
-+----------+---------------------------------+-
```

The usual sample estimate of this quantity is

$$r_k = \frac{c_k}{c_0}, \text{ where } c_k = \frac{1}{m-k} \sum_{t=1}^{m-k} \left(\theta_t^* - \bar{\theta^*} \right) \left(\theta_{t+k}^* - \bar{\theta^*} \right) \tag{3.47}$$

and $\bar{\theta^*} = \frac{1}{m} \sum_{t=1}^{m} \theta_t^*$.

The cross-correlation at lag k of two time series $\{ \theta_t^*, t = 1, \ldots, m \}$ and $\{ \eta_t^*, t = 1, \ldots, m \}$ measures the extent to which the first series at time $(t+k)$ and the second at time t are linearly related, for $k = 1, 2, \ldots$.

A natural sample estimate of this quantity is

$$\begin{aligned} r_{\theta\eta}(k) &= \frac{c_{\theta\eta}(k)}{\sqrt{c_{\theta\theta}(0) c_{\eta\eta}(0)}}, \quad \text{where} \\ c_{\theta\eta}(k) &= \frac{1}{m-k} \sum_{t=1}^{m-k} \left(\theta_t^* - \bar{\theta^*} \right) \left(\eta_{t+k}^* - \bar{\eta^*} \right). \end{aligned} \tag{3.48}$$

```
LAGS AND AUTOCORRELATIONS WITHIN EACH CHAIN:
============================================


-+---------+------------+-------------------------------+-
 | Chain   | Variable   | Lag 1     Lag 10     Lag 50   |
 | =====   | ========   | =====     ======     ======   |
 |         |            |                               |
-+---------+------------+-------------------------------+-
 | bugs1   | mu         | 0.29400   0.00118   -0.01010 |
 |         | nu         | 0.97200   0.78900    0.32100 |
 |         | sigma      | 0.62100   0.30300    0.10800 |
 |         |           |                             |
-+---------+-----------+-----------------------------+-

CROSS-CORRELATION MATRIX:
=========================

-+----------+------------------------------+-
 | VARIABLE |   mu       nu       sigma     |
 | ======== |                               |
 |          |                               |
 | mu       |   1.0000                      |
 | nu       |   0.0946   1.0000             |
 | sigma    |   0.0534   0.5540   1.0000    |
 |          |                               |
-+----------+------------------------------+-
```

You can see (a) that the series for ν is especially strongly autocorrelated, and (b) that ν and σ are fairly strongly positively correlated, which connects with the observation earlier about confounding of scale and shape in the t family.

Diagnostic and Summary Plots. The figure below presents four plots that are useful as MCMC diagnostics and for graphical summaries of posterior distributions, in the case of the parameter ν with run 1 from the NB10 data.

The upper left panel is a time series trace, which documents the poor mixing that has been evident from several of the numerical diagnostics.

The lower left panel is a plot of the autocorrelation function (ACF) for ν, and the lower right panel plots the partial autocorrelation function (PACF).

One of the most common behaviors observed in time series in general, and in the output of MCMC samplers in particular, is that of an autoregressive process.

Letting e_t denote an IID (or white-noise or purely random) process with mean 0 and variance σ_e^2, the time series θ_t^* is said to be an autoregressive process of order p (AR_p) if

$$\theta_t^* = \alpha_1 \theta_{t-1}^* + \ldots + \alpha_p \theta_{t-p}^* + e_t. \tag{3.49}$$

Equation (49) is like a multiple regression model except that θ_t^* is being regressed on past values of itself instead of on other predictor variables; this gives rise to the term autoregressive.

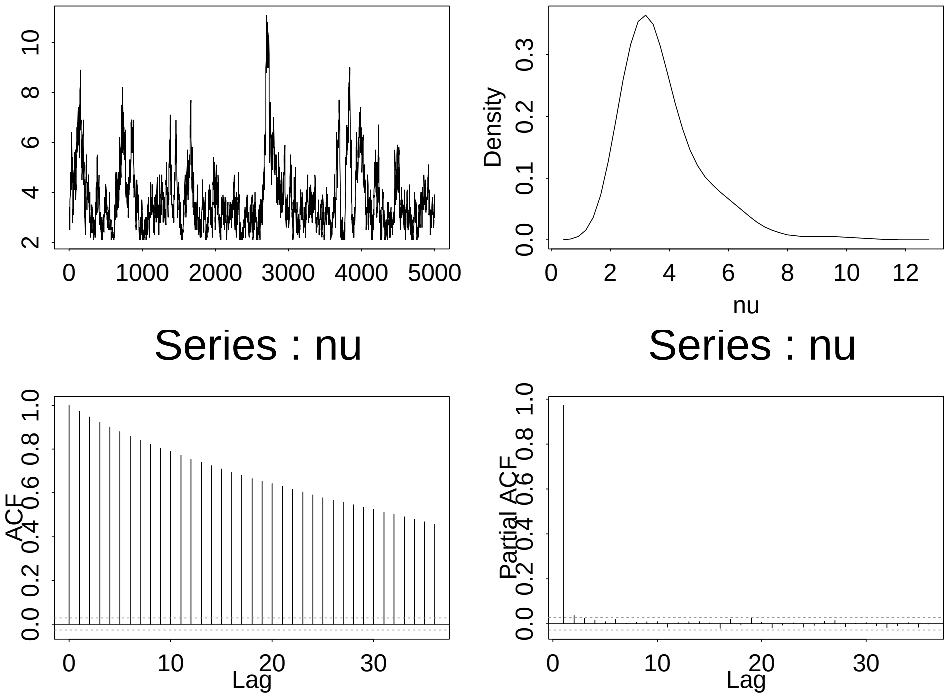


Figure 3.5: *MCMC four-plot for ν in the NB10 t model.*

The *partial autocorrelation function* (PACF) measures the excess correlation between θ^*_t and θ^*_{t+k} not accounted for by the autocorrelations $r_1, \ldots,$ r_{k-1}, and is useful in diagnosing the order of an AR_p process: if θ^*_t is AR_p then the PACF at lags $1, \ldots, p$ will be significantly different from 0 and then close to 0 at lags larger than p.

The lower right-hand plot above shows the characteristic single spike at lag 1 which diagnoses an AR_1 series (the dotted lines in the ACF and PACF plots represent 2 standard error traces around 0, indicating how big an ACF or PACF value needs to be to be significantly different from 0).

This is reinforced by the ACF plot: if θ^*_t is AR_1 with positive first-order autocorrelation ρ_1 then the autocorrelation function should show a slow geometric decay (a ski-slope shape), which it clearly does in this case.

We would conclude that the Gibbs sampling output for ν, when thought of as a time series, behaves like an AR_1 process with first-order autocorrelation roughly $r_1 = 0.972$ (from the table above).

MCMC Accuracy. Suppose that θ_t^* is a stationary time series with underlying true mean μ_θ and variance σ_θ^2.

3.3.6 MCMC accuracy

It can be shown that if $\{\theta_t^*, t = 1, \ldots, m\}$ is AR_1 with first-order autocorrelation ρ_1 then in repeated sampling the uncertainty about μ_θ on the basis of the sample mean $\bar{\theta}^*$ is quantified by

$$V\left(\bar{\theta}^*\right) = \frac{\sigma_\theta^2}{m} \left(\frac{1 + \rho_1}{1 - \rho_1} \right). \tag{3.50}$$

Thus if you want to use MCMC to estimate the posterior mean of a given quantity θ with sufficient accuracy that the standard error of the Monte Carlo mean estimate $\bar{\theta}^*$ based on a monitoring run of length m is no larger than a specified tolerance T, and the MCMC output θ^* behaves like an AR_1 series with first-order autocorrelation ρ_1, you would need m to satisfy

$$\widehat{SE}\left(\bar{\theta}^*\right) = \frac{\hat{\sigma}_\theta}{\sqrt{m}} \sqrt{\frac{1 + \hat{\rho}_1}{1 - \hat{\rho}_1}} \leq T, \tag{3.51}$$

from which

$$m \geq \frac{\hat{\sigma}_\theta^2}{T^2} \left(\frac{1 + \hat{\rho}_1}{1 - \hat{\rho}_1} \right). \tag{3.52}$$

This formula explains why monitoring runs with MCMC often need to be quite long: as $\rho_1 \rightarrow 1$ the required $m \rightarrow \infty$.

For example, we've seen that $\hat{\rho}_1 = r_1$ for ν in the NB10 t model is +0.972, and we'll see below that the sample mean and SD based on the output for ν are roughly 3.628 and 1.161, respectively.

If you wanted to be able to report the posterior mean of ν to 3–significant-figure accuracy (3.63) with reasonably high Monte Carlo probability, you would want T to be on the order of 0.01, giving an enormous monitoring run:

$$m \geq \left(\frac{1.161}{0.01} \right)^2 \left(\frac{1 + 0.972}{1 - 0.972} \right) \doteq (13,479)(70.4) \doteq 949,322 \tag{3.53}$$

This is much larger than the Raftery-Lewis default recommendation above (there is no conflict in this fact; the two diagnostics are focusing on different posterior summaries).

Note from (52) that if you could figure out how to sample in an IID manner from the posterior for θ you would only need $m_{\text{IID}} \ge \frac{\hat{\sigma}_\theta^2}{T^2}$, which in this case is about 13,500 draws.

The term $\left(\frac{1+\hat{\rho}_1}{1-\hat{\rho}_1} \right)$ in (52) represents the amount by which m_{IID} would need to be multiplied to get the same accuracy from MCMC output—it's natural to call this the sample size inflation factor (SSIF), which for ν comes out a whopping 70.4.

The upper right panel in the diagnostic plots above gives a density trace for ν, which shows a mode at about 3 degrees of freedom and a long right-hand tail.

Round 2. From all of this I decided to run the chain again with the `BUGS` commands in the right-hand part of the table on page 54: a burn-in of 2,000 and a monitoring run of 40,000, thinning the output by writing out to disk only every 8th draw (thus ending up with 5,000 stored values).

The MCMC diagnostics were much better: Raftery-Lewis total N recommendations all less than 5,000, all other summaries fine.

All the parameters are mixing well now, so numerical posterior summaries are worth making, as in the table below.

	Posterior	Posterior	95%
Parameter	Mean	SD	Interval
μ	404.3	0.4641	(403.4, 405.2)
ν	3.63	1.16	(2.2, 6.6)
σ	3.873	0.4341	(3.100, 4.778)

I read in three files—the model, the data, and the initial values—and used the `Specification Tool` from the `Model` menu to `check` the model, `load` the data, `compile` the model, `load` the initial values, and `generate` additional initial values for uninitialized nodes in the graph.

I then used the `Sample Monitor Tool` from the `Inference` menu to `set` the `mu`, `sigma`, `nu`, and `y.new` nodes, and clicked on `Dynamic Trace` plots for `mu` and `nu`.

Then choosing the `Update Tool` from the `Model` menu, specifying 2000 in the `updates` box, and clicking `update` permitted a burn-in of 2,000 iterations to occur with the time series traces of the two parameters displayed in real time.

After minimizing the `model`, `data`, and `inits` windows and killing the `Specification Tool` (which are no longer needed until the model is respecified), I typed 10000 in the `updates` box of the `Update Tool` and clicked

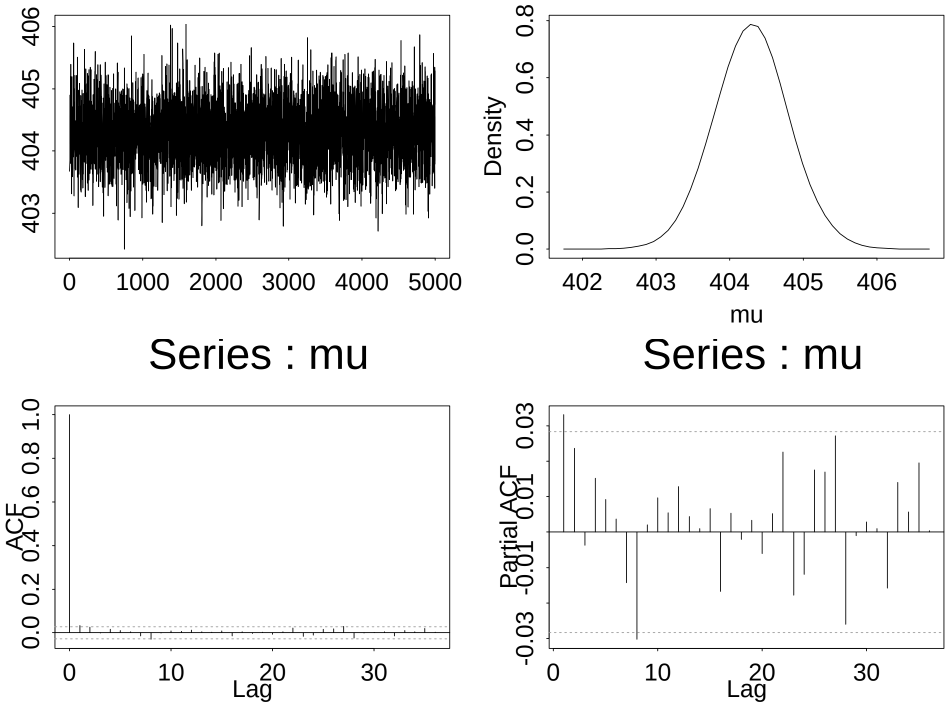


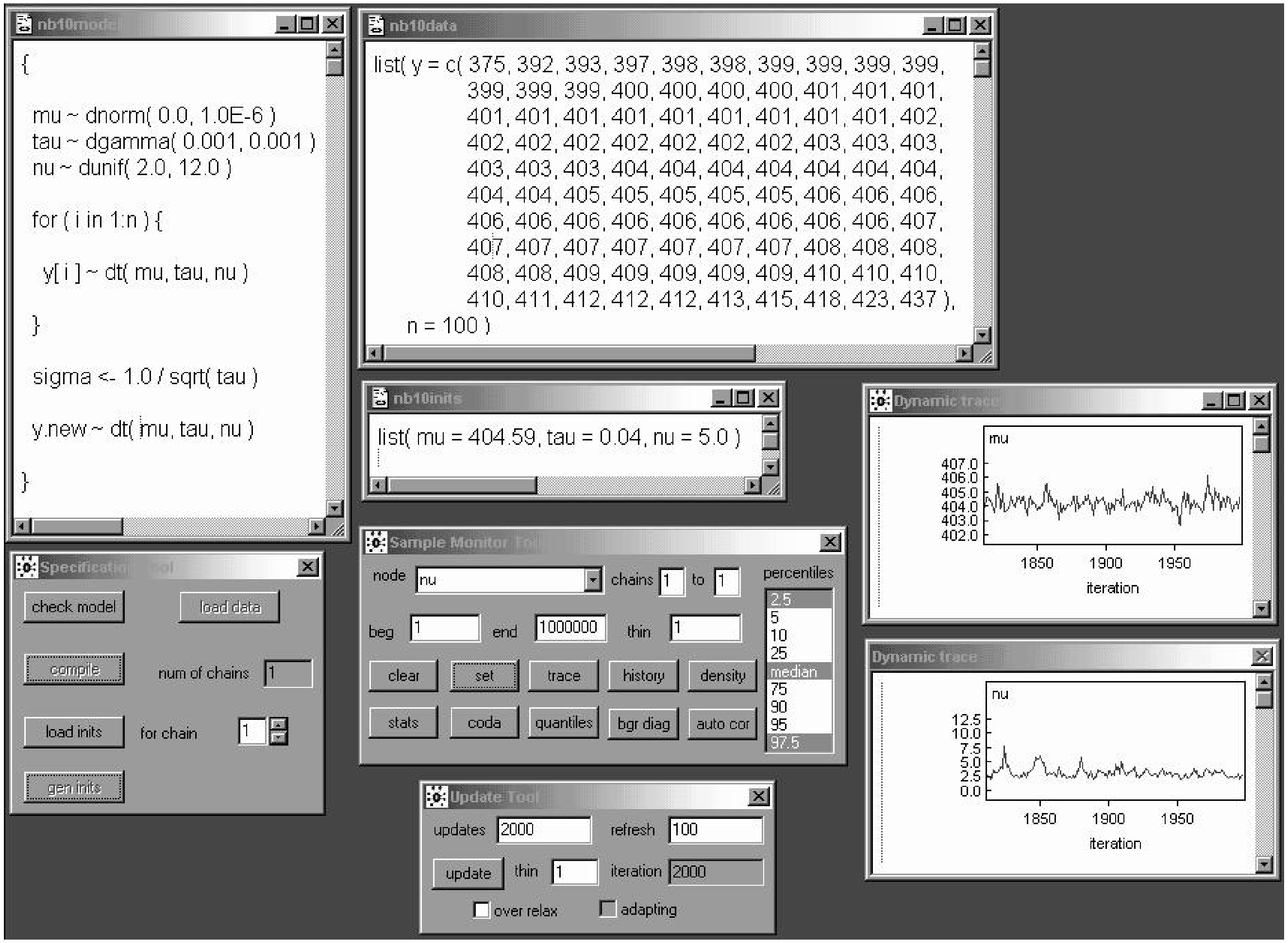
Figure 3.6: *MCMC four-plot for μ.*

`update` to generate a monitoring run of 10,000 iterations (you can watch the updating of `mu` and `nu` dynamically to get an idea of the mixing, but this slows down the sampling).

After killing the `Dynamic Trace` window for `nu` (to concentrate on `mu` for now), in the `Sample Monitor Tool` I selected `mu` from the pull-down menu, set the `beg` and `end` boxes to 2001 and 12000, respectively (to summarize only the monitoring part of the run), and clicked on `history` to get the time series trace of the monitoring run, `density` to get a kernel density trace of the 10,000 iterations, `stats` to get numerical summaries of the monitored iterations, `quantiles` to get a trace of the cumulative estimates of the 2.5%, 50% and 97.5% points in the estimated posterior, and `autoC` to get the autocorrelation function.

You can see that the output for μ is mixing fairly well—the ACF looks like that of an AR_1 series with first-order serial correlation of only about 0.3.

σ is mixing less well: its ACF looks like that of an AR_1 series with first-



```
{

 mu ~ dnorm( 0.0, 1.0E-6 )
 tau ~ dgamma( 0.001, 0.001 )
 nu ~ dunif( 2.0, 12.0 )

 for ( i in 1:n ) {

   y[ i ] ~ dt( mu, tau, nu )

 }

 sigma <- 1.0 / sqrt( tau )

 y.new ~ dt( mu, tau, nu )

}
```

nb10data

```
list( y = c( 375, 392, 393, 397, 398, 398, 399, 399, 399, 399,
             399, 399, 399, 400, 400, 400, 400, 401, 401, 401,
             401, 401, 401, 401, 401, 401, 401, 401, 401, 402,
             402, 402, 402, 402, 402, 402, 402, 403, 403, 403,
             403, 403, 403, 404, 404, 404, 404, 404, 404, 404,
             404, 404, 405, 405, 405, 405, 405, 406, 406, 406,
             406, 406, 406, 406, 406, 406, 406, 406, 406, 407,
             407, 407, 407, 407, 407, 407, 407, 408, 408, 408,
             408, 408, 409, 409, 409, 409, 409, 410, 410, 410,
             410, 411, 412, 412, 412, 413, 415, 418, 423, 437 ),
     n = 100 )
```

nb10inits

```
list( mu = 404.59, tau = 0.04, nu = 5.0 )
```

num of chains 1

for chain 1

node nu chains 1 to 1

beg 1 end 1000000 thin 1

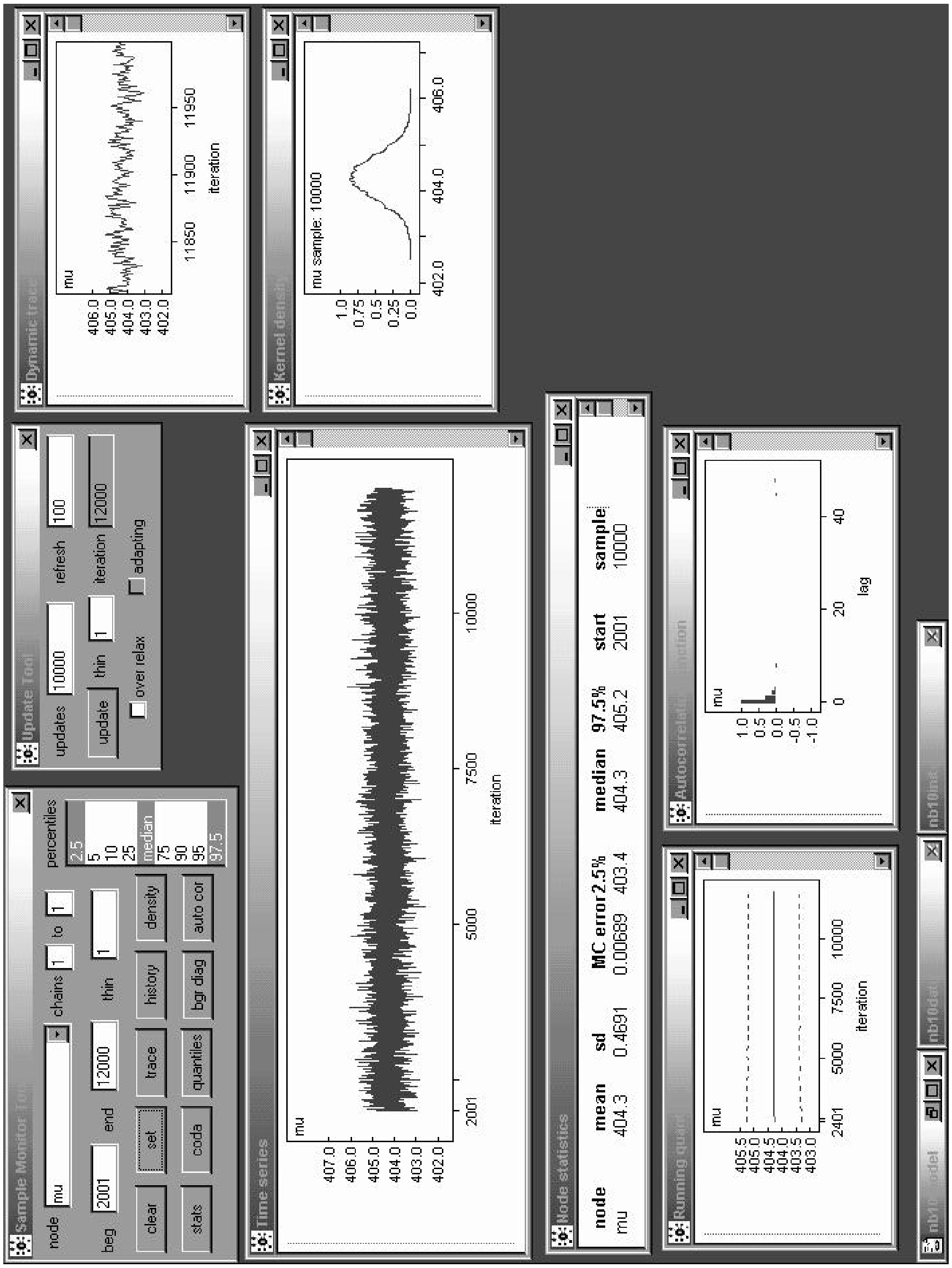
percentiles: 2.5, 5, 10, 25, median, 75, 90, 95, 97.5

updates 2000 refresh 100

thin 1 iteration 2000

□ over relax □ adapting

Figure 3.7: *WinBUGS screen, NB10 t model, with dynamic traces for μ and ν.*



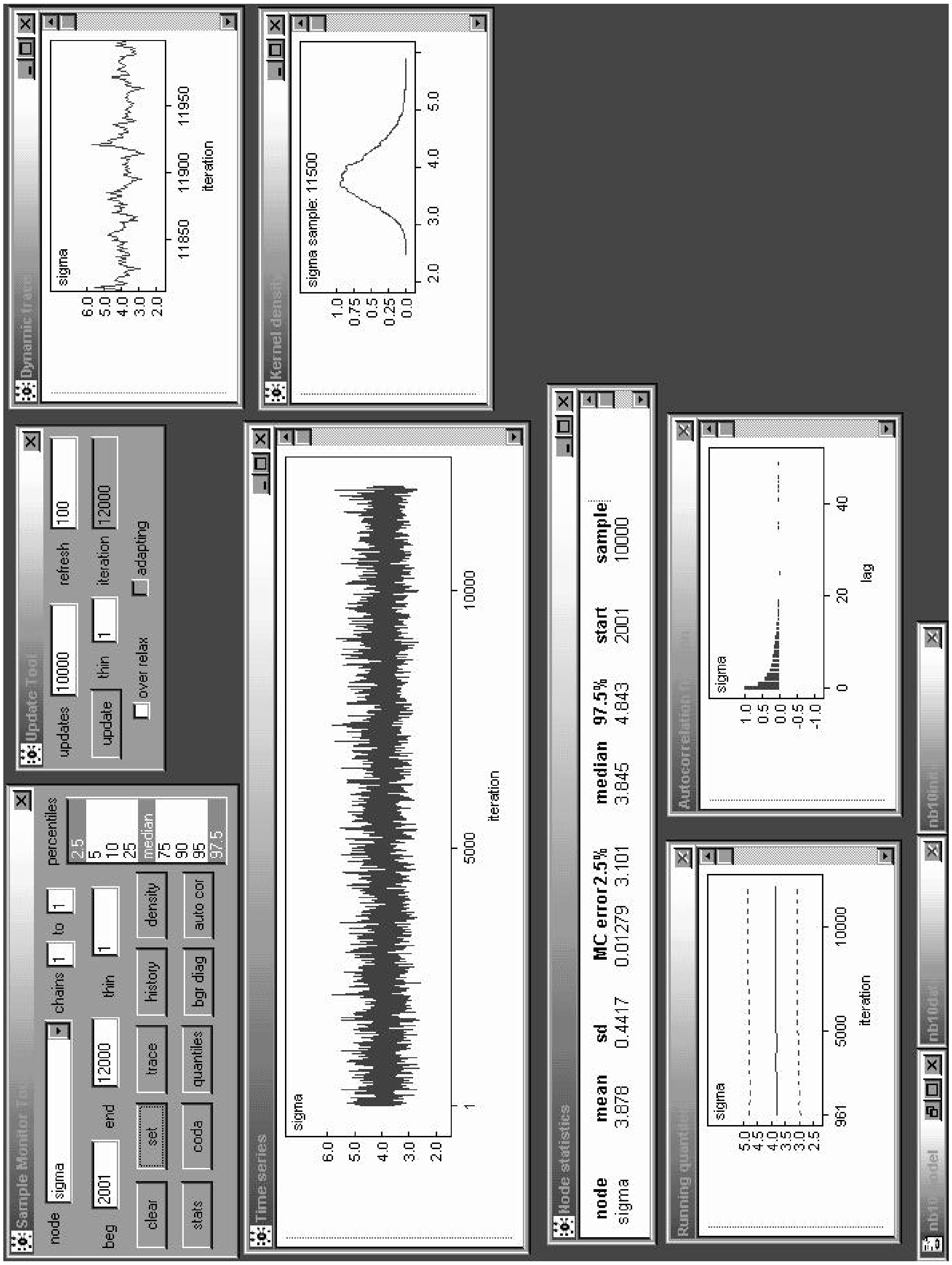
Sample Monitor Tool: node mu, chains 1 to 1, beg 2001, end 12000, thin 1

Update Tool: updates 10000, refresh 100, thin 1, iteration 12000, □ over relax, □ adapting

Node statistics

```
node  mean   sd      MC error  2.5%   median  97.5%  start  sample
mu    404.3  0.4691  0.00689   403.4  404.3   405.2  2001   10000
```

Figure 3.8: *Posterior summaries for μ after 10,000 monitoring iterations.*



Sample Monitor Tool

node sigma · chains 1 to 1 · beg 2001 · end 12000 · thin 1

Update Tool

updates 10000 · refresh 100 · thin 1 · iteration 12000 · □ over relax · □ adapting

```
node    mean    sd      MC error 2.5%   median  97.5%   start   sample
sigma   3.878   0.4417  0.01279  3.101  3.845   4.843   2001    10000
```

Figure 3.9: *Posterior summaries for σ.*

order serial correlation of about 0.6.

This means that a monitoring run of 10,000 would probably not be enough to satisfy minimal Monte Carlo accuracy goals—for example, from the `Node statistics` window the estimated posterior mean is 3.878 with an estimated MC error of 0.0128, meaning that we've not yet achieved three-significant-figure accuracy in this posterior summary.

And ν's mixing is the worst of the three: its ACF looks like that of an AR_1 series with first-order serial correlation of a bit less than +0.9.

`WinBUGS` has a somewhat complicated provision for printing out the autocorrelations; alternately, you can approximately infer $\hat{\rho}_1$ from an equation like (51) above: assuming that the `WinBUGS` people are taking the output of any MCMC chain as (at least approximately) AR_1 and using the formula

$$\widehat{SE}\left(\bar{\theta}^*\right) = \frac{\hat{\sigma}_\theta}{\sqrt{m}}\sqrt{\frac{1+\hat{\rho}_1}{1-\hat{\rho}_1}}, \tag{3.54}$$

you can solve this equation for $\hat{\rho}_1$ to get

$$\hat{\rho}_1 = \frac{m\left[\widehat{SE}\left(\bar{\theta}^*\right)\right]^2 - \hat{\sigma}_\theta^2}{m\left[\widehat{SE}\left(\bar{\theta}^*\right)\right]^2 + \hat{\sigma}_\theta^2}. \tag{3.55}$$

Plugging in the relevant values here gives

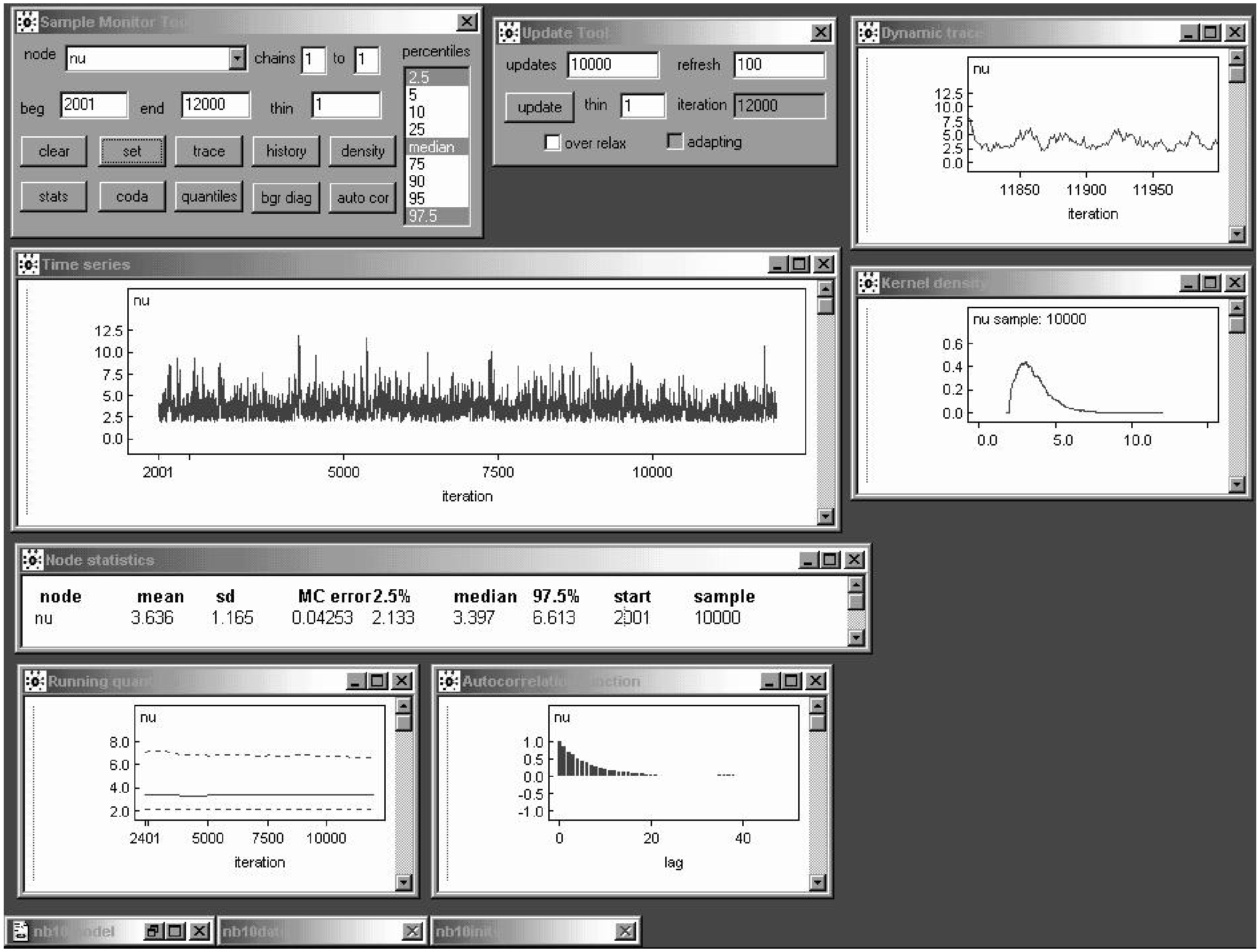
$$\hat{\rho}_1 = \frac{(10,000)(0.04253)^2 - (1.165)^2}{(10,000)(0.04253)^2 + (1.165)^2} \doteq 0.860, \tag{3.56}$$

which is smaller than the corresponding value of 0.972 generated by the `classicBUGS` sampling method (from `CODA`, page 67).

To match the `classicBUGS` strategy outlined above (page 71) I typed 30000 in the `updates` window in the `Update Tool` and hit `update`, yielding a total monitoring run of 40,000.

Remembering to type 42000 in the `end` box in the `Sample Monitoring Tool` window before going any further, to get a monitoring run of 40,000 after the initial burn-in of 2,000, the summaries below for μ are satisfactory in every way.

A monitoring run of 40,000 also looks good for σ: on this basis, and conditional on this model and prior, I think σ is around 3.87 (posterior mean, with an MCSE of 0.006), give or take about 0.44 (posterior SD), and



Sample Monitor Tool

node nu chains 1 to 1

beg 2001 end 12000 thin 1

Update Tool

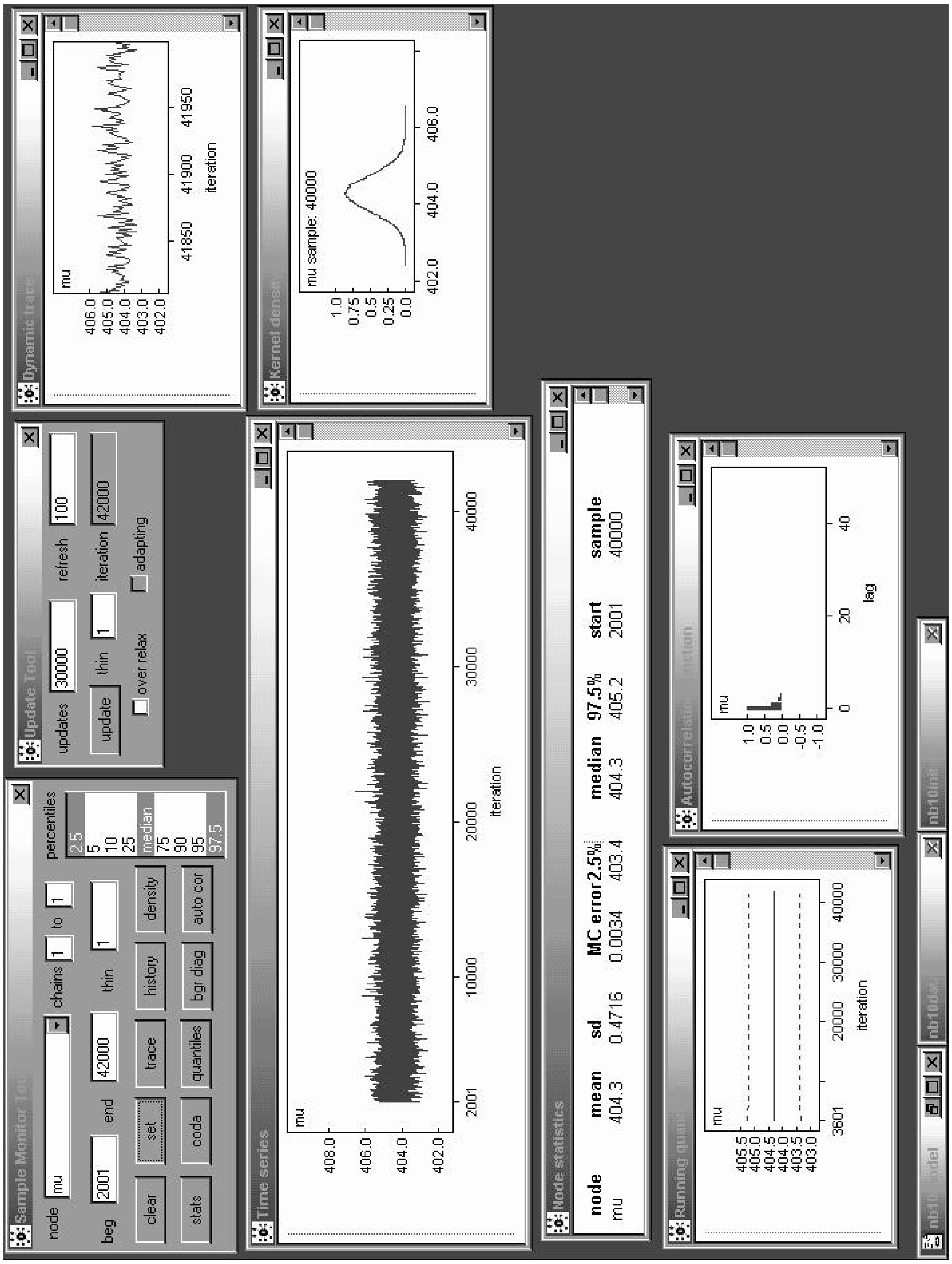
updates 10000 refresh 100

thin 1 iteration 12000

□ over relax □ adapting

```
node   mean    sd      MC error2.5%      median  97.5%   start   sample
nu     3.636   1.165   0.04253   2.133   3.397   6.613   2001    10000
```

Figure 3.10: *Posterior summaries for* ν.



Sample Monitor Tool

node mu chains 1 to 1 percentiles

beg 2001 end 42000 thin 1

percentiles
2.5
5
10
25
median
75
90
95
97.5

Update Tool

updates 30000 refresh 100

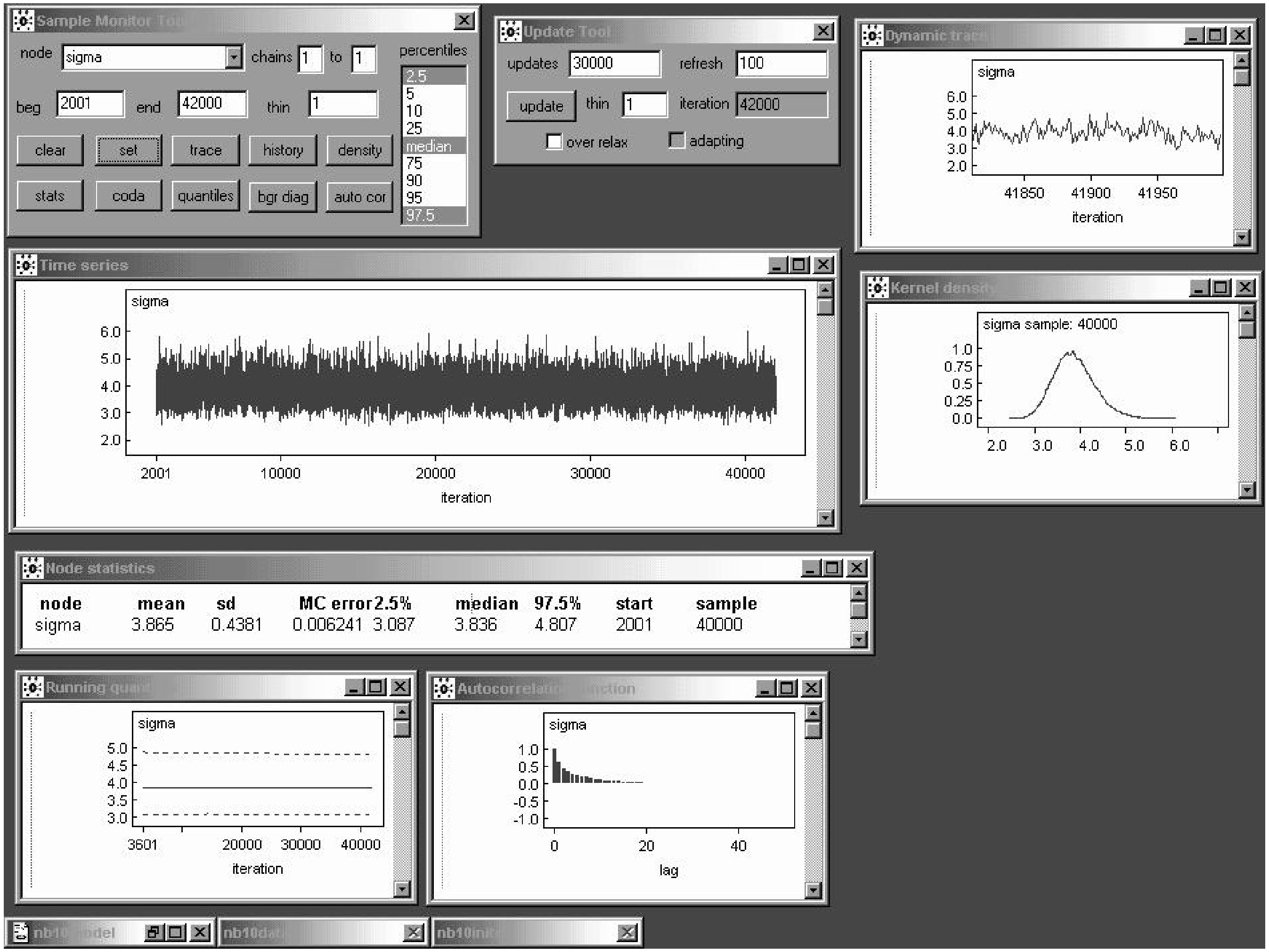
thin 1 iteration 42000

□ over relax □ adapting

Node statistics

```
node   mean    sd       MC error  2.5%    median   97.5%   start   sample
mu     404.3   0.4716   0.0034    403.4   404.3    405.2   2001    40000
```

Figure 3.11: *Posterior summaries for μ after 40,000 monitoring iterations.*



node: sigma chains: 1 to: 1
beg: 2001 end: 42000 thin: 1

updates: 30000 refresh: 100
thin: 1 iteration: 42000
□ over relax □ adapting

```
node    mean    sd      MC error2.5%     median  97.5%   start   sample
sigma   3.865   0.4381  0.006241  3.087  3.836   4.807   2001    40000
```

Figure 3.12: *Posterior summaries for σ after 40,000 monitoring iterations.*

my 95% central posterior interval for σ runs from about 3.09 to about 4.81 (the distribution has a bit of skewness to the right, which makes sense given that σ is a scale parameter).

If the real goal were ν I would use a longer monitoring run, but the main point here is μ, and we saw back on p. 67 that μ and ν are close to uncorrelated in the posterior, so this is good enough.

If you wanted to report the posterior mean of ν with an MCSE of 0.01 (to come close to 3-sigfig accuracy) you'd have to increase the length of the monitoring run by a multiplicative factor of $\left(\frac{0.02213}{0.01}\right)^2 \doteq 4.9$, which would yield a recommended length of monitoring run of about 196,000 iterations (the entire monitoring phase would take about 3 minutes at 2.0 (PC) GHz).

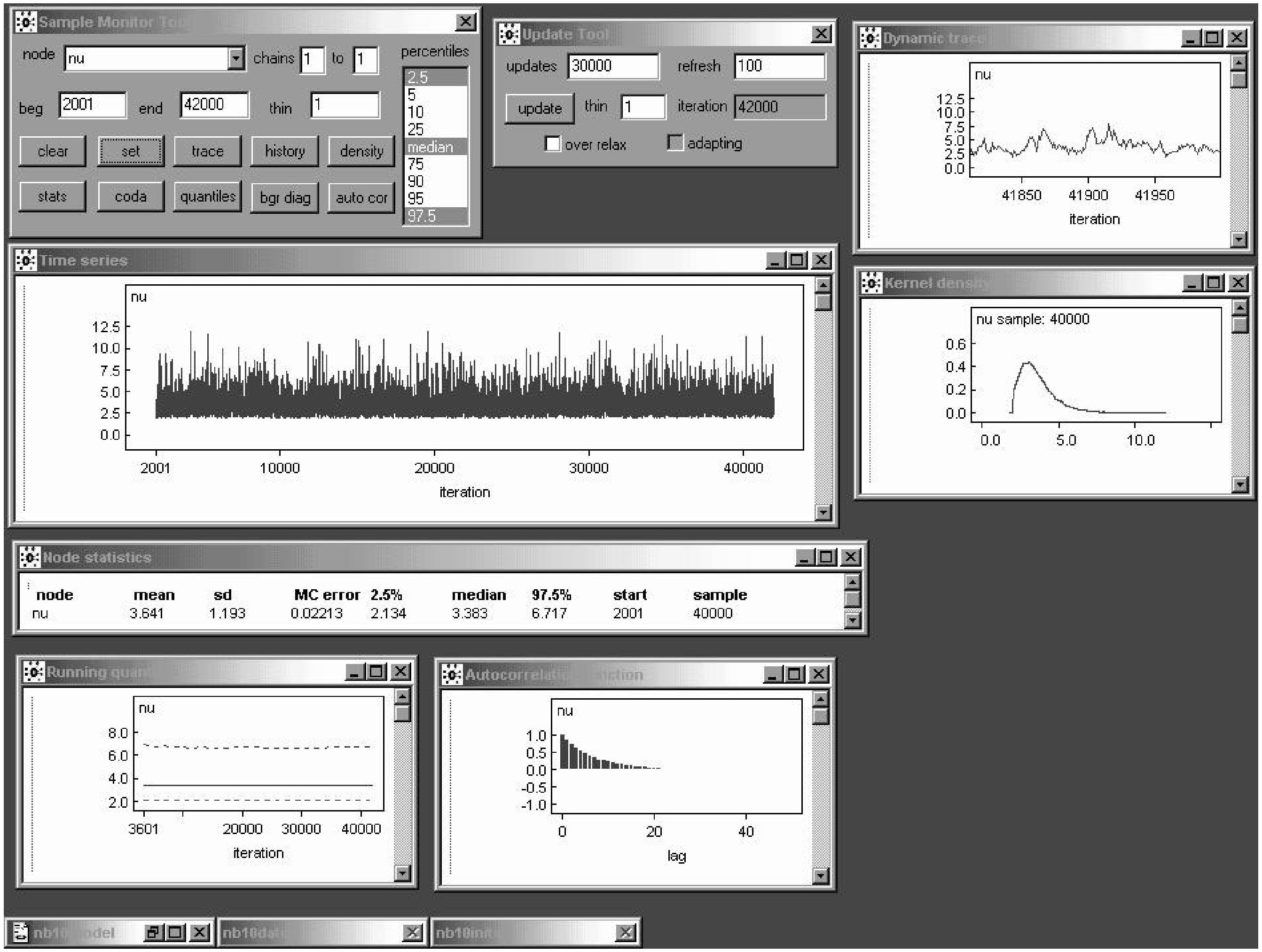
The posterior predictive distribution for y_{n+1} given $(y_1, \ldots, y_n)$ is interesting in the t model: the predictive mean and SD of 404.3 and 6.44 are not far from the sample mean and SD (404.6 and 6.5, respectively), but the predictive distribution has very heavy tails, consistent with the degrees of freedom parameter ν in the t distribution being so small (the time series trace has a few simulated values less than 300 and greater than 500, much farther from the center of the observed data than the most outlying actual observations).

Gaussian comparison. The posterior SD for μ, the only parameter directly comparable across the Gaussian and t models for the NB10 data, came out 0.47 from the t modeling, versus 0.65 with the Gaussian, i.e., the interval estimate for μ from the (incorrect) Gaussian model is about 40% wider that that from the (much better-fitting) t model.

A model uncertainty anomaly? NB Moving from the Gaussian to the t model involves a net increase in model uncertainty, because when you assume the Gaussian you're in effect saying that you know the t degrees of freedom are ∞, whereas with the t model you're treating ν as unknown. And yet, even though there's been an increase in model uncertainty, the inferential uncertainty about μ has gone down.

This is relatively rare—usually when model uncertainty increases so does inferential uncertainty (Draper 2004)—and arises in this case because of two things: (a) the t model fits better than the Gaussian, and (b) the Gaussian is actually a conservative model to assume as far as inferential accuracy for location parameters is concerned.

Two more items on MCMC accuracy. (1) A stringent but potentially useful diagnostic for deciding how long the monitoring run should be for



node nu chains 1 to 1

beg 2001 end 42000 thin 1

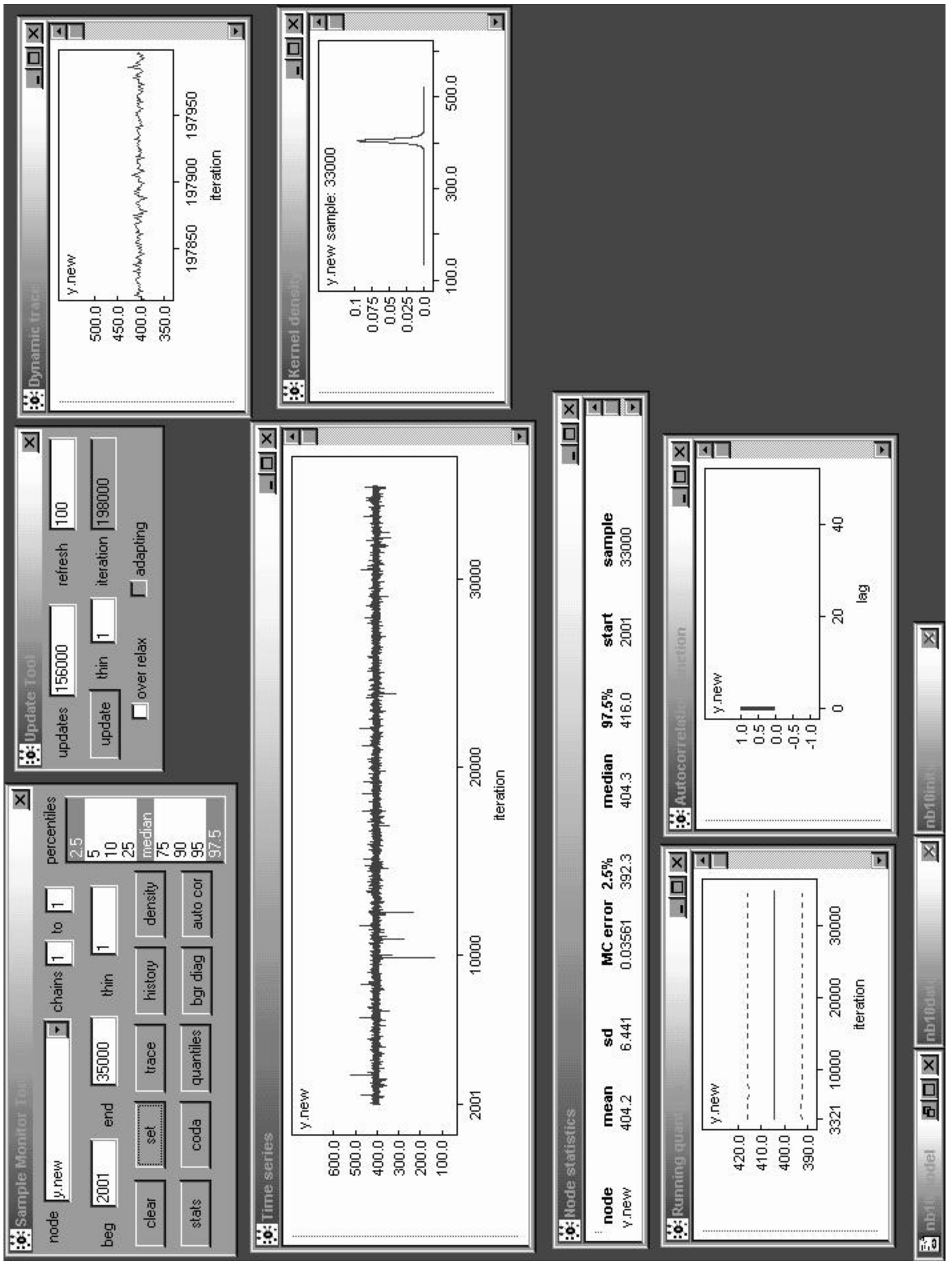
updates 30000 refresh 100

thin 1 iteration 42000

□ over relax □ adapting

```
node    mean    sd      MC error  2.5%    median  97.5%   start   sample
nu      3.641   1.193   0.02213   2.134   3.383   6.717   2001    40000
```

Figure 3.13: *Posterior summaries for ν after 40,000 monitoring iterations.*



Sample Monitor Tool

node y.new chains 1 to 1 percentiles

beg 2001 end 35000 thin 1

Update Tool

updates 156000 refresh 100

update thin 1 iteration 198000

□ over relax □ adapting

Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
y.new	404.2	6.441	0.03561	392.3	404.3	416.0	2001	33000

Figure 3.14: *Posterior summaries for y_{n+1} after 40,000 monitoring iterations.*

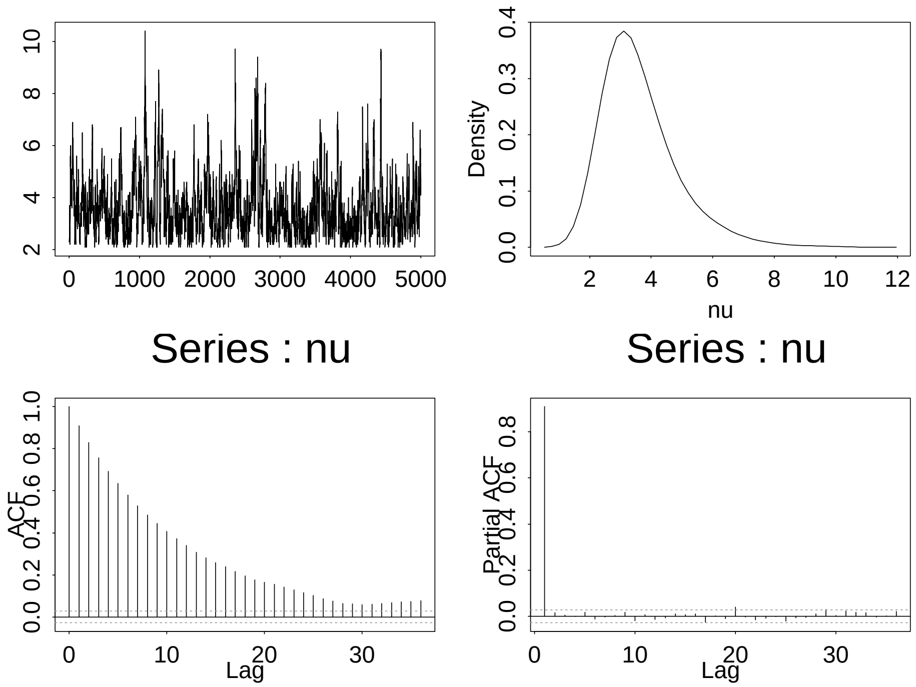


Figure 3.15: *MCMC four-plot for ν.*

a given component θ' of the parameter vector θ, if the output of your MCMC sampler for θ' behaves like an AR_1 series with first-order autocorrelation ρ_1, can be derived as follows.

Suppose, after a burn-in that's long enough to reach stationarity, you've done a preliminary monitoring run, obtaining mean $\bar{\theta}'$, SD $\hat{\sigma}_{\theta'}$, and first-order autocorrelation $\hat{\rho}_1$ as estimates of the corresponding summaries for θ'.

Writing $\theta' = a \cdot 10^b$ for $1 \leq a < 10$, if you want at least k significant figures (sigfigs) of accuracy for the posterior mean summary for θ' with Monte Carlo probability of at least $100(1-\alpha)$, you can check that you'll need

$$2\Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \widehat{SE}\left(\bar{\theta}'\right) \leq 10^{b-k+1}; \tag{3.57}$$

then substituting in the relevant expression from equation (51) above,

$$\widehat{SE}\left(\bar{\theta}'\right) = \frac{\hat{\sigma}_{\theta'}}{\sqrt{m}} \sqrt{\frac{1+\hat{\rho}_1}{1-\hat{\rho}_1}}, \tag{3.58}$$

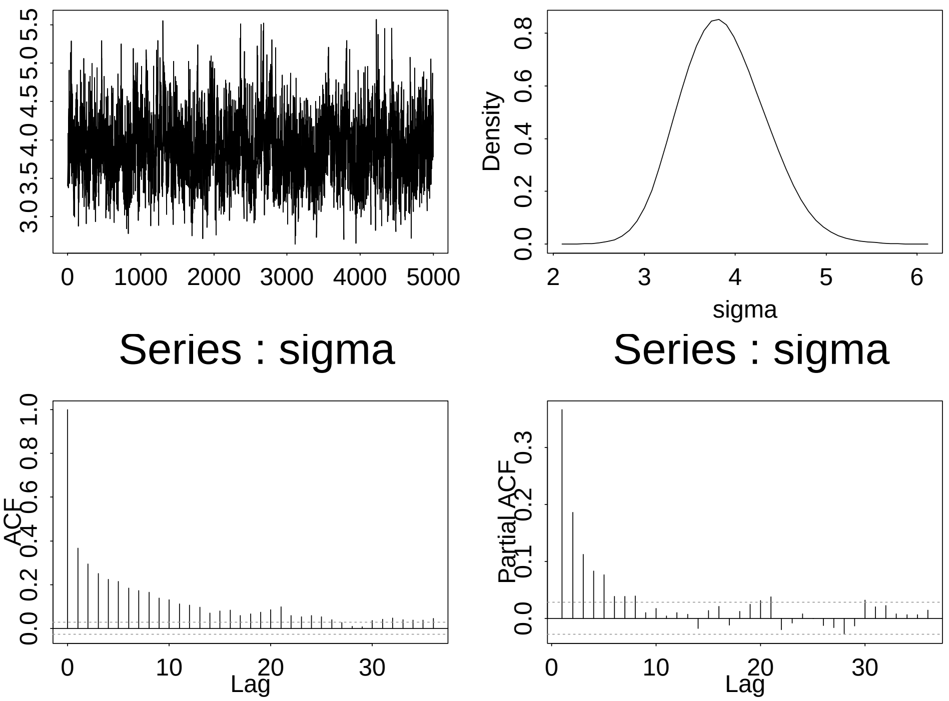


Figure 3.16: *MCMC four-plot for σ^2.*

and solving (58) for m yields

$$m \geq 4 \left[\Phi^{-1} \left(1 - \frac{\alpha}{2} \right) \right]^2 \left(\frac{\hat{\sigma}_{\theta'}}{10^{b-k+1}} \right)^2 \left(\frac{1 + \hat{\rho}_1}{1 - \hat{\rho}_1} \right). \qquad (3.59)$$

This is referred to in the `MLwiN` documentation as the Brooks-Draper diagnostic (Brooks and Draper 2002).

Comments. (a) This diagnostic is sensitive to the scale chosen by the user for reporting results, as far as choosing the target number of sigfigs is concerned.

Example. In my initial monitoring run of 5,000 iterations in the NB10 case study, the posterior mean of μ, on the micrograms below 10g scale, was $\bar{\theta}' = 404.3$ (to 4 sigfigs); the other relevant quantities for μ were as follows: posterior SD $\hat{\sigma}_{\theta'} \doteq 0.464$ and first-order autocorrelation $\hat{\rho}_1 \doteq 0.294$ (NB the MCSE for μ is already down to 0.009 with 5,000 iterations, so I already have a bit more than 4 sigfigs of accuracy).

Suppose (just for the sake of illustration; it's hard to imagine setting an accuracy goal this stringent in practice) that I want to ensure 5 sigfigs with at least 95% Monte Carlo probability for the posterior mean—write $\bar{\theta}' = 4.043 \cdot 10^2$, so that $b = 2$, take $\alpha = 0.05$ and substitute into (14) to yield

$$m \geq 4(1.96)^2 \left(\frac{0.464}{10^{2-5+1}} \right)^2 \left(\frac{1+0.294}{1-0.294} \right) \doteq 60,600. \qquad (3.60)$$

Now, if you instead subtracted 404 from all of the data values (on the micrograms below 10g scale) and made a similar MCMC run, everything would be the same as above except that your current posterior mean for μ would be 0.3 to 1 sigfig, and (with the same MCSE of 0.009) you would regard yourself as already having a bit more than 1 sigfig of accuracy from the initial monitoring run of 5,000.

Then to apply (59) to get 2 sigfigs of accuracy you would write $\bar{\theta}' = 3.0 \cdot 10^{-1}$ and obtain

$$m \geq 4(1.96)^2 \left(\frac{0.464}{10^{(-1)-2+1}} \right)^2 \left(\frac{1+0.294}{1-0.294} \right) \doteq 60,600. \qquad (3.61)$$

These two sets of results from (59) are consistent—by subtracting 404 from all of the data values you (at least temporarily) threw away 3 sigfigs—but you can see that care needs to be taken in thinking about how much accuracy you want, and this question is closely tied to the scale of measurement.

(b) Note from (59) that every time you want to add 1 new sigfig of accuracy in the posterior mean the required length of monitoring run goes up multiplicatively by $(10^1)^2 = 100$.

(2) I've concentrated so far on the MCMC accuracy of the posterior mean—what about other posterior summaries like the SD?

Suppose as above that you're interested in a given component θ' of the parameter vector θ, and that the output of your MCMC sampler for θ' behaves like an AR_1 series with first-order autocorrelation ρ_1; and suppose as above that after a burn-in that's long enough to reach stationarity, you've done a preliminary monitoring run, obtaining mean $\bar{\theta}'$, SD $\hat{\sigma}_{\theta'}$, and first-order autocorrelation $\hat{\rho}_1$ as estimates of the corresponding summaries for θ'.

Then it can be shown, in an expression analogous to (58), that if the marginal posterior for θ' is approximately Gaussian

$$\widehat{SE}(\hat{\sigma}_{\theta'}) = \frac{\hat{\sigma}_{\theta'}}{\sqrt{2m}} \sqrt{\frac{1 + \hat{\rho}_1^2}{1 - \hat{\rho}_1^2}}. \tag{3.62}$$

Note that with a parameter with MCMC output that's approximately AR_1 and roughly Gaussian this implies that

$$\frac{\widehat{SE}\left(\bar{\theta}'\right)}{\widehat{SE}(\hat{\sigma}_{\theta'})} \doteq \sqrt{\frac{2(1 + \hat{\rho}_1)^2}{1 + \hat{\rho}_1^2}}, \tag{3.63}$$

which goes from $\sqrt{2}$ to 2 as $\hat{\rho}_1$ ranges from 0 to $+1$, i.e., the mean is harder to pin down than the SD with Gaussian data (a reflection of how light the tails are).

`CODA` in `R`. If you go to `http://www.r-project.org/`, click on `CRAN` (the Comprehensive R Archive Network), click on one of the `CRAN` mirror sites, and click on `Package Sources`, you'll find a lot of contributed packages, one of which is `CODA`.

Clicking on `coda` will get you the source code for `CODA` (you can also visit `http://www-fis.iarc.fr/coda/`, a web site maintained by Martyn Plummer, the guy who ported `CODA` from `S+` to `R`).

In this way you can download the source for `R-CODA` and follow the instructions for installing it.

An easier way, if you're running `R` on a machine that's connected to the internet, is to go into `R` and just type

```
install.packages( "coda" )
```

If everything goes smoothly this will automatically install `R-CODA` on your machine (you'll need to do this on your laptop to get `CODA`; it's already installed in the Engineering School version of `R`).

Once you have it in your local library you can invoke it from inside `R` with the command

```
library( coda )
```

and you can find out what it can do with the command

```
help( package = coda )
```

The idea is to run `classicBUGS` or `WinBUGS`, store the MCMC dataset somewhere handy, go into `R`, and use `R-CODA` to read the MCMC dataset in and analyze it.

All of the MCMC diagnostics I showed you described above are available to you with this approach.

3.4 Problems

1. (The gambler's ruin; review of basic ideas in Markov chains) Consider a gambler at a casino who at each play of a game has probability $0 < p < 1$ of winning \$1 and probability $(1 - p)$ of losing \$1. If the successive plays of the game are assumed independent, the question this problem addresses is as follows: what is the probability P that if she (the gambler) starts with $\$M > \$0$ she will *break the bank* (reach $\$N > \$M$, for integer M and N; here $\$N$ represents the initial capital of the casino against which she's playing[1]) before *going broke* (reaching \$0)?

   (a) If we let Y_t denote her fortune after the tth play of the game, explain why the process $\{Y_t\}$ is a Markov chain on the state space $\{\$0, \$1, \ldots, \$N\}$, and identify the possible states the process could be in at times $t = 0, 1, \ldots$.

   (b) My intent is that this problem should be a somewhat playful environment within which you can learn more about Markov chains than you already know. Therefore, using whatever combination you like of {simulation (`R` is a good language for this), looking around on the web, reading probability books, etc.}, see how much progress you can make on the basic question posed at the beginning of the problem. A fully satisfying mathematical answer to the question would be symbolic in p, M, and N, but you'll get nearly full credit for doing a good job of answering it for a few (well-chosen) specific values of these quantities and speculating about the nature of the dependence of P on p, M, and N. Ex-

[1] For the sake of this problem let's pretend that once she reaches $\$N$ the casino judges that it has lost enough money to her that it does not wish to continue playing against her, which is what "breaking the bank" means.

plore the sensitivity of P to small changes in p, M, and N: on which of these quantities does P depend most sensitively?

(c) Let $N \to \infty$ and show that under these conditions, if $p > \frac{1}{2}$ there is a positive probability (specify it if you can) of the gambler's fortune increasing indefinitely, but if $p \leq \frac{1}{2}$ she will go broke with probability 1 against an infinitely rich adversary (this last fact is not surprising for $p < \frac{1}{2}$, but what about $p = \frac{1}{2}$?).

2. (First practice with `BUGS`) Write a classic `BUGS` or `WinBUGS` program to use Gibbs sampling to analyze the data in the length-of-stay case study, using the same Gamma prior and Poisson likelihood as in that example. Obtain MCMC approximations both for the posterior distribution of λ given the data vector y and the predictive distribution $p(y_{n+1}|y)$, and compare summaries of these distributions (means, SDs, histograms or density traces) with the theoretical conjugate results we got in the case study. You don't need to worry about MCMC diagnostics in this simple example, because Gibbs sampling when there's only one parameter amounts to IID sampling from the relevant posterior and predictive distributions. Justify your choices of initial values for the Markov chain and length of burn-in period. Use one of the formulas given in class to work out how long you need to monitor the chain to report 3-significant-figure accuracy of the posterior mean estimates for both λ and y_{n+1}, and verify that you do indeed achieve that level of accuracy (at least up to Monte Carlo noise) in your simulation. What length of monitoring run is necessary to report 3-significant-figure accuracy of the posterior **SD** estimate? Explain briefly, and report all relevant calculations (simulation or otherwise).

3. (Second practice with `BUGS`) In problem 3 of homework 2 we used conjugate inference to fit an Exponential sampling model to the wire failure data given in that problem, and you may remember noticing that the biggest data value (21194) seemed a bit large in the Exponential context, which tentatively called the Exponential distribution into question. Recalling that the basic Bayesian idea for improving a model is to *expand* it by embedding it in a richer class of models of which it's a special case, the natural thing to try is to fit a model to this data set in which the sampling distribution is Gamma (we saw in part 2 of the lecture notes that the Exponential is a special case of the $\Gamma(\alpha, \beta)$

family with $\alpha = 1$). Write a classic `BUGS` or `WinBUGS` program to use MCMC to fit the model

$$\begin{aligned} (\alpha, \beta) &\sim p(\alpha, \beta) \\ (y_i | \alpha, \beta) &\overset{\text{IID}}{\sim} \Gamma(\alpha, \beta), \quad i = 1, \ldots, n \end{aligned} \tag{3.64}$$

to the wire failure data. For this problem, by way of prior information (unlike the situation in homework 2) I'd like you to use a diffuse prior on α and β. Since they both live on $(0, \infty)$ it's natural to try independent $\Gamma(\epsilon, \epsilon)$ priors for both of them, with (as usual) a small value for ϵ like 0.001; or you could use an initial run with $\Gamma(\epsilon, \epsilon)$ priors to see where the likelihood is appreciable and then use $U(0, c_\alpha)$ and $U(0, c_\beta)$ priors for α and β, where c_α and c_β are chosen to be big enough not to truncate the likelihood but not much larger than that. Summarize the posterior distribution on α and β to an appropriate degree of Monte Carlo accuracy. Does the $\Gamma(\alpha, \beta)$ family appear to provide a better fit to the wire failure data than the Exponential sampling distribution used in homework 2? Explain briefly.

4. (Multinomial data and the Dirichlet distribution as a prior; based on Section 3.5 in Gelman et al.) In late October 1988, CBS News conducted a survey which was equivalent to a simple random sample of $n = 1,447$ American adults to learn about voter preferences in the Presidential election which was to be held a few weeks later. $y_1 = 727$ of these people supported George Bush (the elder), $y_2 = 583$ supported Michael Dukakis, and $y_3 = 137$ supported other candidates or responded "no opinion." This situation is a lot like the AMI mortality case study in class except that there are three outcome categories (Bush, Dukakis, other) instead of two (died, lived): before any data arrives you would probably agree that your uncertainty about the string of 1,447 individual outcomes from each sampled person (which is summarized by the counts $y = (y_1, y_2, y_3) = (727, 583, 137)$) is exchangeable. This leads by an easy generalization of de Finetti's representation theorem for binary outcomes to the following model for the summary counts:

$$\begin{aligned} (\theta_1, \ldots, \theta_k) &\sim p(\theta_1, \ldots, \theta_k) \\ p(y_1, \ldots, y_k | \theta_1, \ldots, \theta_k) &= c \prod_{j=1}^{k} \theta_j^{y_j}, \end{aligned} \tag{3.65}$$

where $0 < \theta_j < 1$ for all $j = 1, \ldots, k$ and $\sum_{j=1}^k \theta_j = 1$. The second line of (2) (the sampling distribution of the vector y, which defines the likelihood function) is the *multinomial* distribution, an obvious generalization of the binomial to $k > 2$ categories (in this voting problem $k = 3$). Evidently in this model the conjugate prior for the vector $\theta = (\theta_1, \ldots, \theta_k)$ is of the form

$$p(\theta_1, \ldots, \theta_k | \alpha_1, \ldots, \alpha_k) = c \prod_{j=1}^k \theta_j^{\alpha_j - 1}; \tag{3.66}$$

this distribution turns out to be well-behaved for any choice of the hyperparameter vector $\alpha = (\alpha_1, \ldots, \alpha_k)$ such that $\alpha_j > 0$ for all $j = 1, \ldots, k$. This is the *Dirichlet*(α) distribution, a kind of generalization of the Beta distribution to more than two categories. With this prior the model becomes

$$\begin{aligned} (\theta_1, \ldots, \theta_k) & \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k) \\ (y_1, \ldots, y_k | \theta_1, \ldots, \theta_k) & \sim \text{Multinomial}(n; \theta_1, \ldots, \theta_k) \end{aligned} \tag{3.67}$$

(see Appendix A in Gelman et al. for the normalizing constants). As with the Beta distribution, the α_j can clearly be seen in this model to represent prior sample sizes; in the voting example, choosing a particular $(\alpha_1, \alpha_2, \alpha_3)$ is equivalent to assuming that the prior is equivalent to a data set with α_1 preferences for Bush, α_2 for Dukakis, and α_3 for other. To create a diffuse prior, which would be a natural choice in the absence of any earlier sampling data (and even with earlier data it's not clear that voter opinion is sufficiently stable over time to make simple use of any previous polling results), we evidently want the α_j to be small; an easy choice that avoids complications with improper priors is to take $\alpha = (1, \ldots, 1)$, a kind of multivariate generalization of the uniform distribution. The main scientific interest in this problem focuses on $\gamma = (\theta_1 - \theta_2)$, the margin by which Bush is leading Dukakis.

(a) Write out the likelihood function for the vector θ in the Multinomial sampling model above, and compute the maximum likelihood estimates of the θ_i and of γ. You can either do this by (i) expressing the log likelihood as a function of θ_1, θ_2, and θ_3 and performing a constrained maximization of it using Lagrange multipliers, or (ii) substituting $\theta_3 = (1 - \theta_1 - \theta_2)$ and $y_3 = (n - y_1 - y_2)$ into the log likelihood and carrying out an unconstrained maximization in the usual way (by setting first partial derivatives to 0 and solving). Do the MLEs have reasonable intuitive forms? Explain briefly.

(extra credit) On the web or in a statistics text, read about how Fisher information generalizes when the parameter θ of interest is a vector and use this to compute approximate large-sample standard errors for the MLEs of the θ_i and of γ.

(b) Use `BUGS` or `WinBUGS` with the diffuse prior mentioned above to simulate m draws from the marginal posterior distributions for the θ_i and for γ, where m is large enough to yield results that seem accurate enough to you given the context of the problem (briefly justify your choice of m). How do the posterior means of the θ_i compare with the MLEs? Explain briefly. Report the posterior mean and SD of γ, and compare your estimated posterior density with the plot below, which is taken from Gelman et al. Use your MCMC output to estimate $p(\gamma > 0|y)$, the chance that Bush would win the election if it were held shortly after the data were gathered and the "other" (non-Bush, non-Dukakis) voters behaved appropriately (briefly explain what has to be assumed about these other voters so that $p(\gamma > 0|y)$ **is** the chance that Bush would win the election), and attach a Monte Carlo standard error to your estimate of $p(\gamma > 0|y)$. Describe your MCMC sampling strategy (mainly your starting values and the length b of your burnin run; you've already justified your choice of m) and briefly explain why you believe that this strategy has accurately extracted the posterior distribution of interest.

(c) (extra credit) Use `Maple` or some equivalent environment (or paper and pen, if you're brave) to see if you can derive a closed-form expression for $p(\gamma|y)$, and compare your mathematical result with your simulation-based findings in (a), using the actual data in this example.

5. Write your own Metropolis-Hastings sampler to analyze the data in the length-of-stay case study, using the same Gamma prior and Poisson likelihood as in that example; using your MH sampler, complete as many of the steps in problem 1 of this assignment as you have time

and patience for, and compare the results you obtained in problem 1 with Gibbs sampling. In choosing a proposal distribution for your MH sampler there are two main ways to go: you can either (i) transform λ to the log scale so that it lives on the entire real line and use (something like) a Gaussian proposal distribution for $\eta = \log(\lambda)$ (in this case you'll be using the simpler Metropolis form for the acceptance probability), or (ii) pick a proposal distribution for λ that simulates from the positive part of the real line (a natural choice would be the family of Gamma distributions; in this case you'll be using the more complicated MH form for the acceptance probability). In either (i) or (ii) you'll find that some measure of scale for the proposal distribution acts like a tuning constant that can be adjusted to achieve optimal MH Monte Carlo efficiency. If you have time it would be good to make a small study of how the MCSE of the posterior mean for λ or η depends on this tuning constant, so that you can find the optimal scaling of the proposal distribution.

Chapter 4

Bayesian model specification

4.1 Hierarchical model selection: An example with count data

Case Study: *In-home geriatric assessment (IHGA).* In an experiment conducted in the 1980s (Hendriksen et al. 1984), 572 elderly people living in a number of villages in Denmark were randomized, 287 to a control (C) group (who received standard care) and 285 to an experimental (E) group (who received standard care plus IHGA: a kind of preventive medicine in which each person's medical and social needs were assessed and acted upon individually).

One important outcome was the number of hospitalizations during the two-year life of the study (Table 4.1).

Table 4.1. Distribution of number of hospitalizations in the IHGA study over a two-year period.

	Number of Hospitalizations										
Group	0	1	2	3	4	5	6	7	n	Mean	SD
Control	138	77	46	12	8	4	0	2	287	0.944	1.24
Experimental	147	83	37	13	3	1	1	0	285	0.768	1.01

Evidently IHGA lowered the mean hospitalization rate (for these elderly Danish people, at least) by $(0.944 - 0.768) = 0.176$, which is about a $100\left(\frac{0.768-0.944}{0.944}\right) = 19\%$ reduction from the control level, a difference that's large in clinical terms.

Modeling the IHGA data. An off-the-shelf analysis of this experiment might pretend (Model 0) that the data are Gaussian,

$$\begin{aligned} \left(C_i|\mu_C,\sigma_C^2\right) &\overset{\text{IID}}{\sim} N\left(\mu_C,\sigma_C^2\right), i=1,\ldots,n_C, \\ \left(E_j|\mu_E,\sigma_E^2\right) &\overset{\text{IID}}{\sim} N\left(\mu_E,\sigma_E^2\right), j=1,\ldots,n_E, \end{aligned} \tag{4.1}$$

and use the ordinary frequentist two-independent-samples "z-machinery":

```
rosalind 15> R

R : Copyright 2005, The R Foundation
Version 2.1.0 Patched (2005-05-12), ISBN 3-900051-07-0

> C <- c( rep( 0, 138 ), rep( 1, 77 ), rep( 2, 46 ),
    rep( 3, 12 ), rep( 4, 8 ), rep( 5, 4 ), rep( 7, 2 ) )

> print( n.C <- length( C ) )

[1] 287                 # sample size in the control group

> mean( C )

[1] 0.9442509           # control group mean

> sd( C )

[1] 1.239089            # control group
                        # standard deviation (SD)
> table( C )

   0  1  2  3 4 5 7     # control group
 138 77 46 12 8 4 2     # frequency distribution

> E <- c( rep( 0, 147 ), rep( 1, 83 ), rep( 2, 37 ),
    rep( 3, 13 ),rep( 4, 3 ), rep( 5, 1 ), rep( 6, 1 ) )

> print( n.E <- length( E ) )

[1] 285                 # sample size in the
                        # experimental group
> mean( E )
```

```
[1] 0.7684211           # experimental group mean

> sd( E )

[1] 1.008268            # experimental group SD

> table( E )

  0  1  2  3 4 5 6      # experimental group
 147 83 37 13 3 1 1     # frequency distribution

> print( effect <- mean( E ) - mean( C ) )

[1] -0.1758298          # mean difference ( E - C )

> effect / mean( C )

[1] -0.1862109          # relative difference ( E - C ) / C

> SE.effect <- sqrt( var( C ) / n.C + var( E ) / n.E )

[1] 0.09442807          # standard error of the difference

> print( CI <- c( effect - 1.96 * SE.effect,
    effect + 1.96 * SE.effect ) )

[1] -0.3609 0.009249   # the 95% confidence interval from
                       # model 0 runs from -.36 to +.01
```

Deficiencies of model 0. The frequentist analysis of Model 0 is equivalent to a Bayesian analysis of the same model with diffuse priors on the control and experimental group means and SDs $(\mu_C, \sigma_C, \mu_E, \sigma_E)$, and is summarized in Table 4.2.

Table 4.2. Summary of analysis of Model 0.

	Posterior		
	Mean	SD	95% Interval
Treatment effect ($\mu_E - \mu_C$)	-0.176	0.0944	$(-0.361, 0.009)$

However, both distributions have long right-hand tails; in fact they look

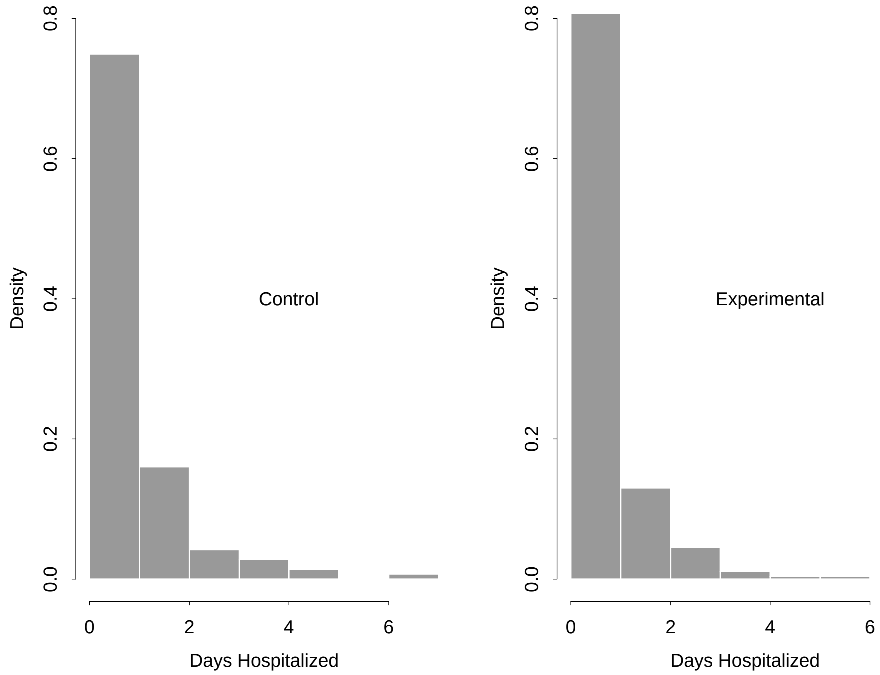


Figure 4.1: *Histograms of control and experimental numbers of hospitalizations.*

rather Poisson.

4.1.1 Poisson fixed-effects modeling

`R` code to make the histograms:

```
> x11( )                             # to open a
                                     #   graphics window
> par( mfrow = c( 1, 2 ) )          # to plot two histograms

> hist( C, nclass = 8, probability = T,
    xlab = 'Days Hospitalized', ylab = 'Density',
    xlim = c( 0, 7 ), ylim = c( 0, 0.8 ) )

> text( 4, 0.4, 'Control' )
> hist( E, nclass = 8, probability = T,
    xlab = ’Days Hospitalized’, ylab = ’Density’,
    xlim = c( 0, 7 ), ylim = c( 0, 0.8 ) )

> text( 4, 0.4, ’Experimental’ )
```

So I created a `classicBUGS` file called `poisson1.bug` that looked like this:

```
model poisson1;

const

  n.C = 287, n.E = 285;

var

  lambda.C, lambda.E, C[ n.C ], E[ n.E ], effect;

data C in "poisson-C.dat", E in "poisson-E.dat";

inits in "poisson1.in";

{

  lambda.C ~ dgamma( 0.001, 0.001 );
  lambda.E ~ dgamma( 0.001, 0.001 );

  for ( i in 1:n.C ) {

    C[ i ] ~ dpois( lambda.C );

  }

  for ( j in 1:n.E ) {

    E[ j ] ~ dpois( lambda.E );

  }
```

```
    effect <- lambda.E - lambda.C;

  }
```

poisson1.in initializes both λ_C and λ_E to 1.0; the $\Gamma(0.001, 0.001)$ priors for λ_C and λ_E are chosen (as usual to create diffuseness) to be flat in the region in which the likelihood is appreciable:

```
  > sqrt( var( C ) / n.C )

  [1] 0.07314114

  > sqrt( var( E ) / n.E )

  [1] 0.05972466

  > c( mean( C ) - 3.0 * sqrt( var( C ) / n.C ),
      mean( C ) + 3.0 * sqrt( var( C ) / n.C ) )

  [1] 0.7248275 1.1636743

  > c( mean( E ) - 3.0 * sqrt( var( E ) / n.E ),
      mean( E ) + 3.0 * sqrt( var( E ) / n.E ) )

  [1] 0.5892471 0.9475950

  > lambda.grid <- seq( 0.01, 2.0, 0.01 )

  > plot( lambda.grid, 0.001 * dgamma( lambda.grid, 0.001 ),
      type = 'l', xlab = 'Lambda', ylab = 'Density' )
```

The likelihood under the Gaussian model is concentrated for λ_C from about 0.7 to 1.2, and that for λ_E from about 0.6 to 1; you can see from the plot that across those ranges the $\Gamma(0.001, 0.001)$ prior is essentially constant.

Figure 4.3 presents part of the results of fitting the 2-independent-samples additive Poisson model detailed earlier in WinBUGS. A burn-in of 2,000 was almost instantaneous at 2.0 PC GHz and revealed good mixing for the three main quantities of interest.

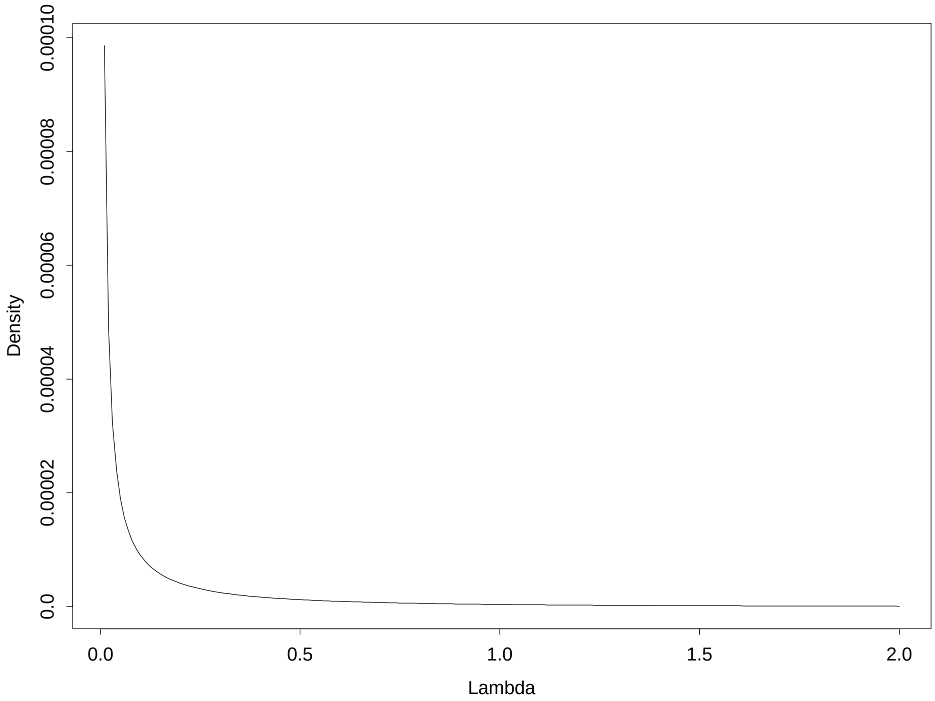
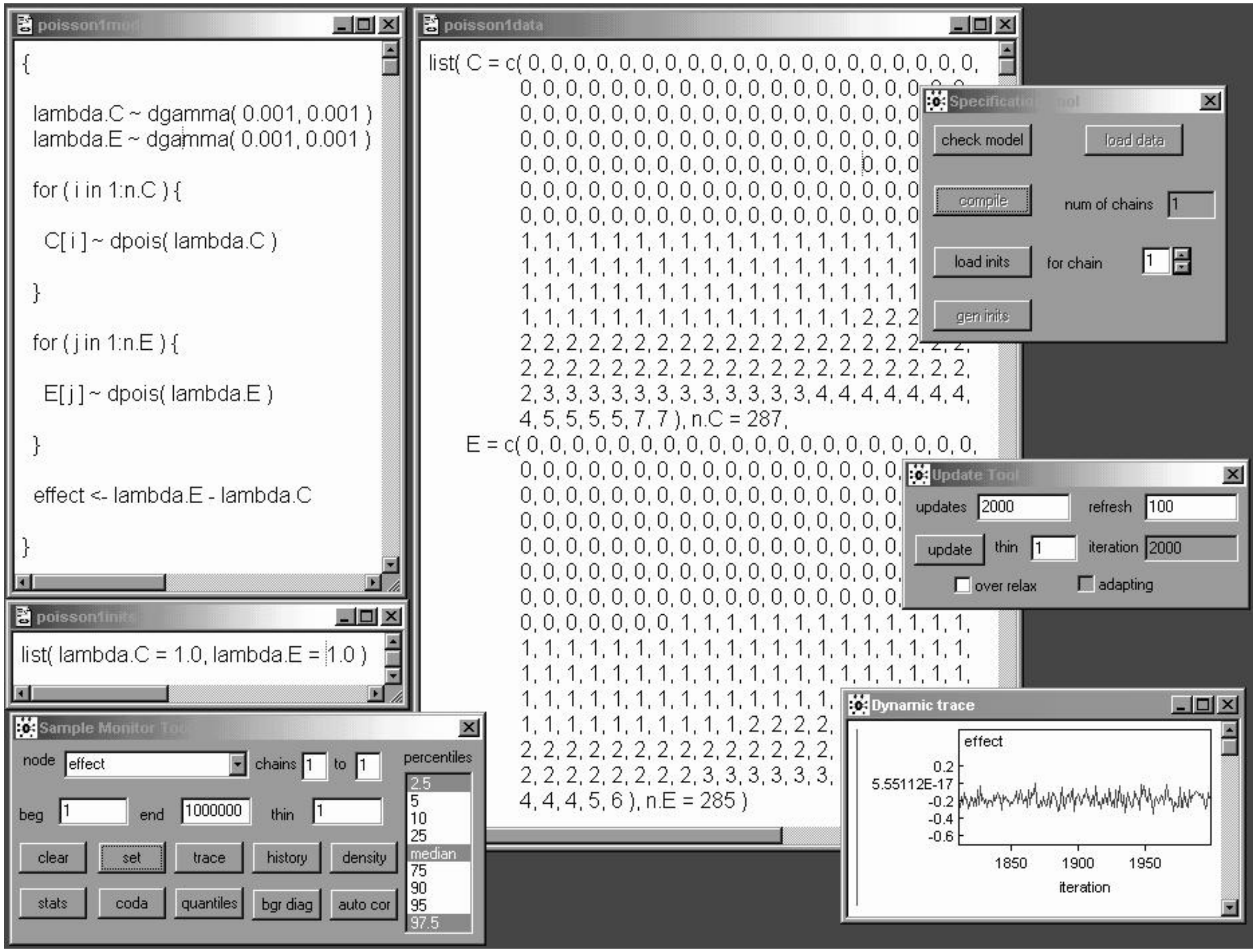


Figure 4.2: *The* $\Gamma(0.001, 0.001)$ *distribution in the region in which the likelihoods for* λ_C *and* λ_E *are appreciable.*

A monitoring run of 8,000 reveals that the effect parameter in the 2-independent-samples Poisson model is behaving like white noise, so that already with only 8,000 iterations the posterior mean has a Monte Carlo standard error of less than 0.001.

Thus a burn-in of 2,000 and a monitoring run of 8,000 yields good MCMC diagnostics and permits a comparison between model 0 (Gaussian) and model 1 (Poisson), as in Table 4.3.

Table 4.3. Comparison of inferential conclusions from models 0 and 1.



poisson1model

```
{

  lambda.C ~ dgamma( 0.001, 0.001 )
  lambda.E ~ dgamma( 0.001, 0.001 )

  for ( i in 1:n.C ) {

    C[ i ] ~ dpois( lambda.C )

  }

  for ( j in 1:n.E ) {

    E[ j ] ~ dpois( lambda.E )

  }

  effect <- lambda.E - lambda.C

}
```

poisson1data

```
list( C = c( 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4,
             4, 5, 5, 5, 5, 7, 7 ), n.C = 287,
      E = c( 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,
             4, 4, 4, 5, 6 ), n.E = 285 )
```

poisson1inits

```
list( lambda.C = 1.0, lambda.E = 1.0 )
```

Specification Tool

num of chains: 1

for chain: 1

Update Tool

updates: 2000 refresh: 100

thin: 1 iteration: 2000

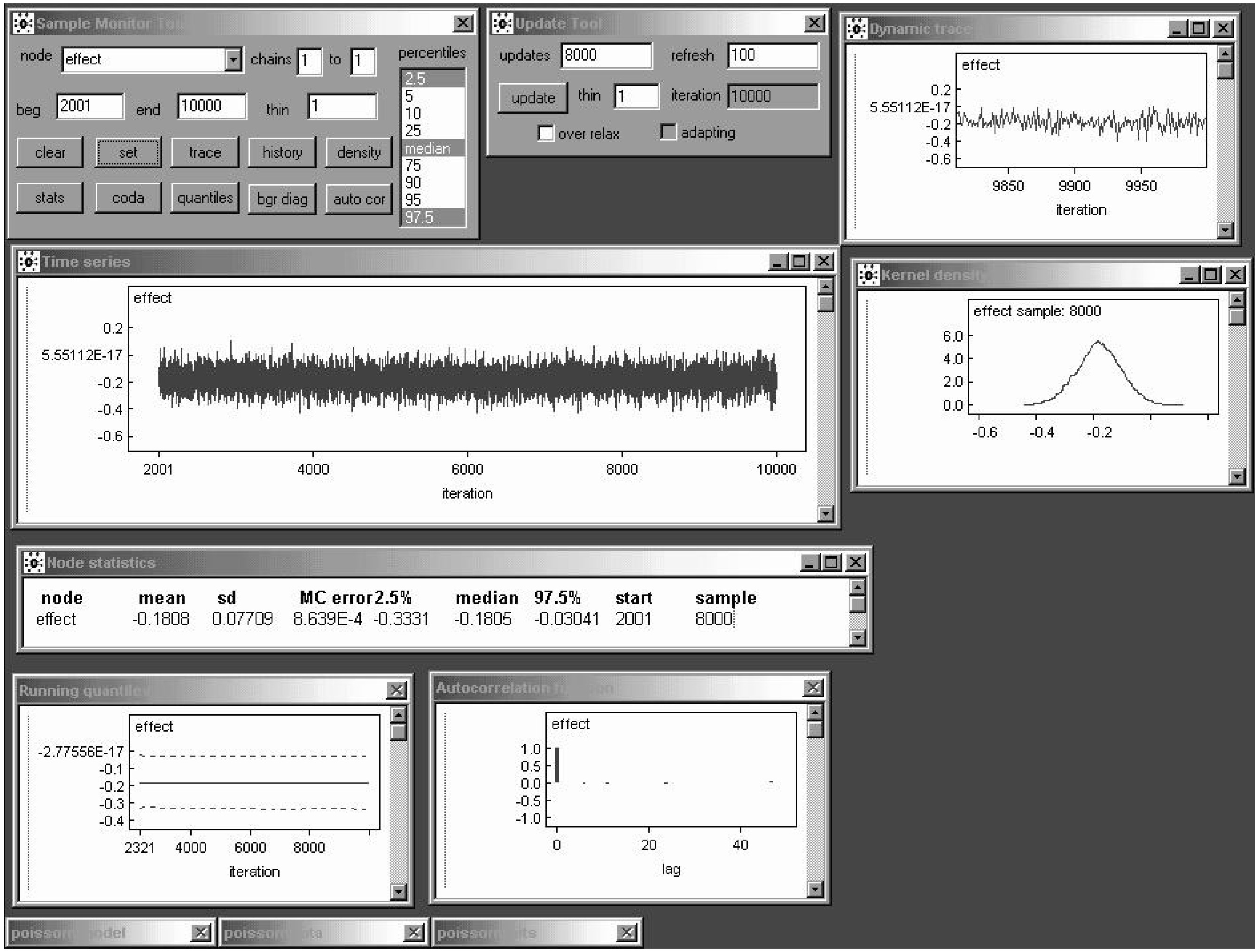
□ over relax □ adapting

Sample Monitor Tool

node: effect chains: 1 to: 1

beg: 1 end: 1000000 thin: 1

Figure 4.3: *Fitting the 2–independent-samples additive Poisson model to the IHGA data in* ***WinBUGS***.



node effect chains 1 to 1

beg 2001 end 10000 thin 1

updates 8000 refresh 100

thin 1 iteration 10000

□ over relax □ adapting

```
node    mean     sd       MC error 2.5%     median   97.5%     start  sample
effect  -0.1808  0.07709  8.639E-4 -0.3331  -0.1805  -0.03041  2001   8000
```

Figure 4.4: *Posterior monitoring for the effect parameter.*

λ_C Model	Posterior Mean	Posterior SD	Central 95% Interval
Gaussian	0.944	0.0731	$(0.801, 1.09)$
Poisson	0.943	0.0577	$(0.832, 1.06)$

λ_E Model	Posterior Mean	Posterior SD	Central 95% Interval
Gaussian	0.768	0.0597	$(0.651, 0.885)$
Poisson	0.769	0.0521	$(0.671, 0.875)$

$\Delta = \lambda_E - \lambda_C$ Model	Posterior Mean	Posterior SD	Central 95% Interval
Gaussian	-0.176	0.0944	$(-0.361, 0.009)$
Poisson	-0.174	0.0774	$(-0.325, -0.024)$

The two models produce almost identical point estimates, but the Poisson model leads to sharper inferences (e.g., the posterior SD for the treatment effect $\Delta = \lambda_E - \lambda_C$ is 22% larger in model 0 than in model 1).

4.1.2 Additive and multiplicative treatment effects

This is the same point we noticed with the NB10 data—when a location parameter is the only thing at issue, the Gaussian is a conservative modeling choice (intuitively, the Poisson gains its "extra accuracy" from the variance and the mean being equal, which permits second-moment information to help in estimating the λ values along with the usual first-moment information).

Both the Gaussian and Poisson models so far implicitly assume that the treatment effect is additive:

$$E \overset{\text{st}}{=} C + \text{effect}, \tag{4.2}$$

where $\overset{\text{st}}{=}$ means *is stochastically equal to*; in other words, apart from random variation the effect of the IHGA is to add or subtract a constant to or from each person's underlying rate of hospitalization.

However, since the outcome variable is non-negative, it is plausible that a better model for the data is

$$E \overset{\text{st}}{=} (1 + \text{effect})\, C. \tag{4.3}$$

Here the treatment effect is multiplicative—in other words, apart from random variation the effect of the IHGA is to multiply each person's underlying rate of hospitalization by a constant above or below 1.

A qqplot of the control and experimental outcome values can in some cases be helpful in choosing between additive and multiplicative models:

```
> CEqq <- qqplot( C, E, plot = F )

> table( CEqq$y, CEqq$x )

               Interpolated C values
      0 0.965  1 1.5  2 2.82 3 3.91 4 4.96 5 6.99 7
  0 137     1  9   0  0    0 0    0 0    0 0    0 0
  1   0     0 66   1 16    0 0    0 0    0 0    0 0
  2   0     0  0   0 29    1 7    0 0    0 0    0 0
E 3   0     0  0   0  0    0 4    1 7    1 0    0 0
  4   0     0  0   0  0    0 0    0 0    0 3    0 0
  5   0     0  0   0  0    0 0    0 0    0 0    1 0
  6   0     0  0   0  0    0 0    0 0    0 0    0 1

> symbols( c( 0, 0.964798, 1, 1, 1.5, 2, 2, 2.823944, 3, 3,
    3.908447, 4, 4.964813, 5, 6.985962, 7 ), c( rep( 0, 3 ),
    rep( 1, 3 ), rep( 2, 3 ), rep( 3, 4 ), 4, 5, 6 ),
    circles = c( 137, 1, 9, 66, 1, 16, 29, 1, 7, 4, 1, 7, 1,
    3, 1, 1 ), xlab = 'C', ylab = 'E' )

> abline( 0, 1 )                   # E = C (no effect)

> abline( 0, 0.793, lty = 2 )      # E = 0.816 C
                                   #   (multiplicative)

> abline( -0.174, 1, lty = 3 )     # E = C - 0.174 (additive)
```

Here, because the Poisson model has only one parameter for both location and scale, the multiplicative and additive formulations fit equally well, but the multiplicative model generalizes more readily (see below).

A multiplicative Poisson model. A simple way to write the multiplicative model is to re-express the data in the form of a regression of the outcome y on a dummy variable x which is 1 if the person was in the experimental group and 0 if he/she was in the control group:

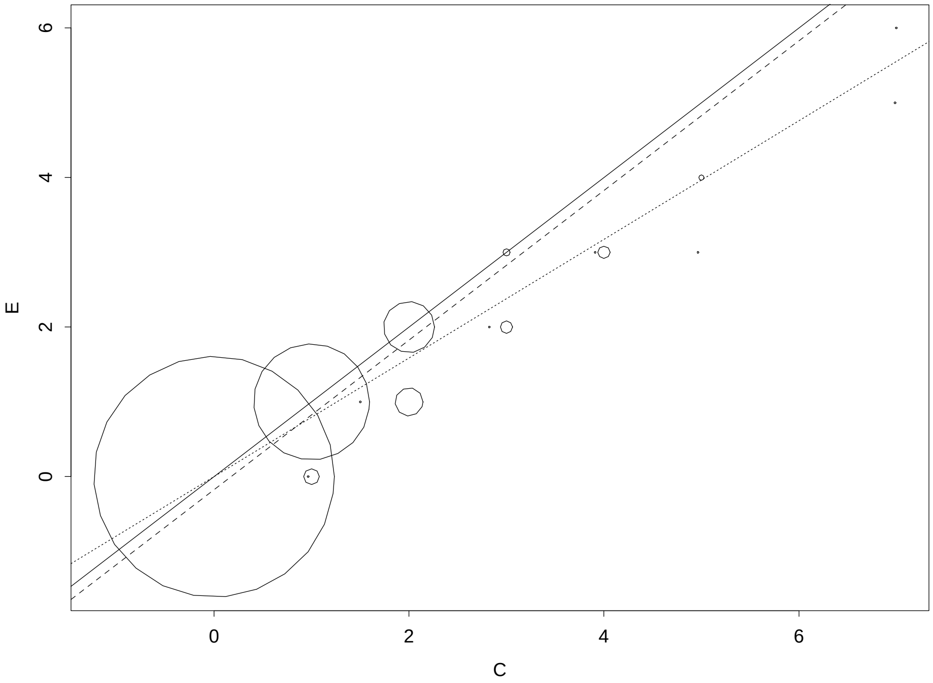


Figure 4.5: *QQplot of E versus C values, with the radii of the plotted circles proportional to the number of observations at the indicated point. The solid line corresponds to no treatment effect, the small dotted line to the best-fitting multiplicative model ($E \overset{st}{=} 0.816\,C$), and the large dotted line to the best-fitting additive model ($E \overset{st}{=} C - 0.174$).*

i	1	2	$\cdots$	287	288	289	$\cdots$	572
x_i	0	0	$\cdots$	0	1	1	$\cdots$	1
y_i	1	0	$\cdots$	2	0	3	$\cdots$	1

Then for $i = 1, \ldots, n = 572$ the multiplicative model can be written

$$\begin{array}{rcl} (y_i \,|\, \lambda_i) & \overset{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) \\ \log(\lambda_i) & = & \gamma_0 + \gamma_1 x_i \\ (\gamma_0, \gamma_1) & \sim & \text{diffuse} \end{array} \qquad (4.4)$$

In this model the control people have

$$\log(\lambda_i) = \gamma_0 + \gamma_1(0) = \gamma_0, \quad \text{i.e.,} \quad \lambda_C = e^{\gamma_0}, \tag{4.5}$$

and the experimental people have

$$\begin{aligned} \log(\lambda_i) &= \gamma_0 + \gamma_1(1) = \gamma_0 + \gamma_1, \;\; \text{i.e.,} \\ \lambda_E &= e^{\gamma_0 + \gamma_1} = e^{\gamma_0} e^{\gamma_1} = \lambda_C e^{\gamma_1}. \end{aligned} \tag{4.6}$$

Now you may remember from basic Taylor series that for γ_1 not too far from 0

$$e^{\gamma_1} \doteq 1 + \gamma_1, \tag{4.7}$$

so that finally (for γ_1 fairly near 0)

$$\lambda_E \doteq (1 + \gamma_1)\, \lambda_C, \tag{4.8}$$

which is a way of expressing equation (3) in
Poisson language.

Fitting this model in `classicBUGS` is easy:

```
model poisson2;

const

  n = 572;

var

  gamma.0, gamma.1, lambda[ n ], x[ n ], y[ n ], lambda.C,
  lambda.E, mult.effect;

data x in "poisson-x.dat", y in "poisson-y.dat";
inits in "poisson2.in";

{

  gamma.0 ~ dnorm( 0.0, 1.0E-4 );    # flat priors for
  gamma.1 ~ dnorm( 0.0, 1.0E-4 );    # gamma.0 and gamma.1

  for ( i in 1:n ) {
      log( lambda[ i ] ) <- gamma.0 + gamma.1 * x[ i ];
      y[ i ] ~ dpois( lambda[ i ] );

    }

    lambda.C <- exp( gamma.0 );
    lambda.E <- exp( gamma.0 + gamma.1 );
    mult.effect <- exp( gamma.1 );
}
```

The multiplicative Poisson model (11) takes longer to run—2,000 burn-in iterations now take about 4 seconds at 2.0 PC GHz—but still exhibits fairly good mixing, as we'll see below.

A total of 10,000 iterations (the chain started essentially in equilibrium, so the burn-in can be absorbed into the monitoring run) reveals that the multiplicative effect parameter e^{γ_1} in model (11) behaves like an AR_1 series with $\hat{\rho}_1 \doteq 0.5$, but the Monte Carlo standard error for the posterior mean is still only about 0.001 with a run of this length.

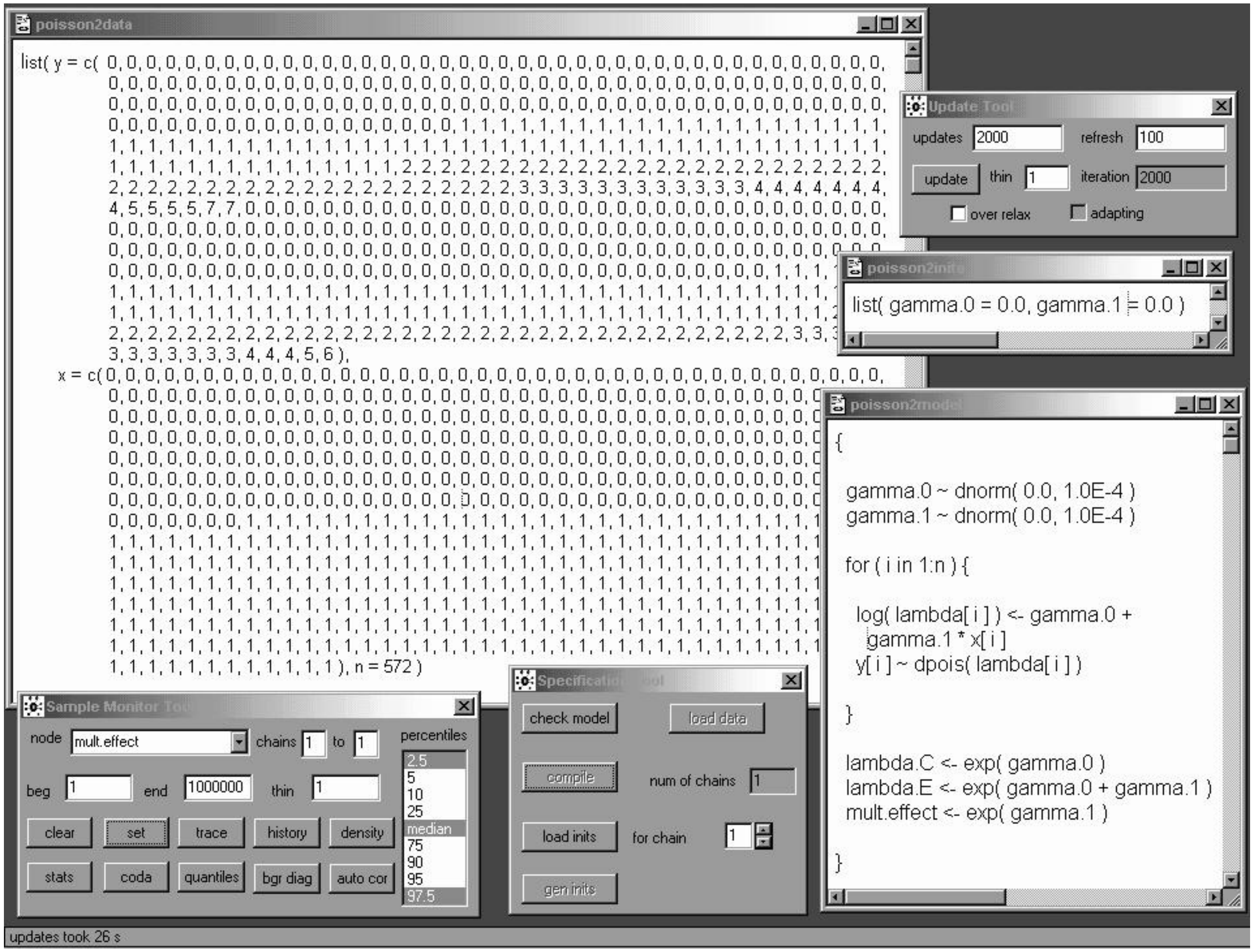
A burn-in of 2,000 and a monitoring run of 8,000 again yields good MCMC diagnostics and permits a comparison between the additive and multiplicative Poisson models, as in Table 4.4.

Comparison of inferential conclusions from the additive and multiplicative Poisson models.

λ_C Model	Posterior Mean	Posterior SD	Central 95% Interval
additive	0.943	0.0577	$(0.832, 1.06)$
multiplicative	0.945	0.0574	$(0.837, 1.06)$

λ_E Model	Posterior Mean	Posterior SD	Central 95% Interval
additive	0.769	0.0521	$(0.671, 0.875)$
multiplicative	0.768	0.0518	$(0.671, 0.872)$

effect Model	Posterior Mean	Posterior SD	Central 95% Interval
additive	-0.174	0.0774	$(-0.325, -0.024)$
multiplicative	-0.184	0.0743	$(-0.324, -0.033)$



```
list( y = c( 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4,
             4, 5, 5, 5, 5, 7, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3
             3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 5, 6 ),
        x = c( 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
             0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ), n = 572 )
```

```
list( gamma.0 = 0.0, gamma.1 = 0.0 )
```

```
{

  gamma.0 ~ dnorm( 0.0, 1.0E-4 )
  gamma.1 ~ dnorm( 0.0, 1.0E-4 )

  for ( i in 1:n ) {

    log( lambda[ i ] ) <- gamma.0 +
      gamma.1 * x[ i ]
    y[ i ] ~ dpois( lambda[ i ] )

  }

  lambda.C <- exp( gamma.0 )
  lambda.E <- exp( gamma.0 + gamma.1 )
  mult.effect <- exp( gamma.1 )

}
```

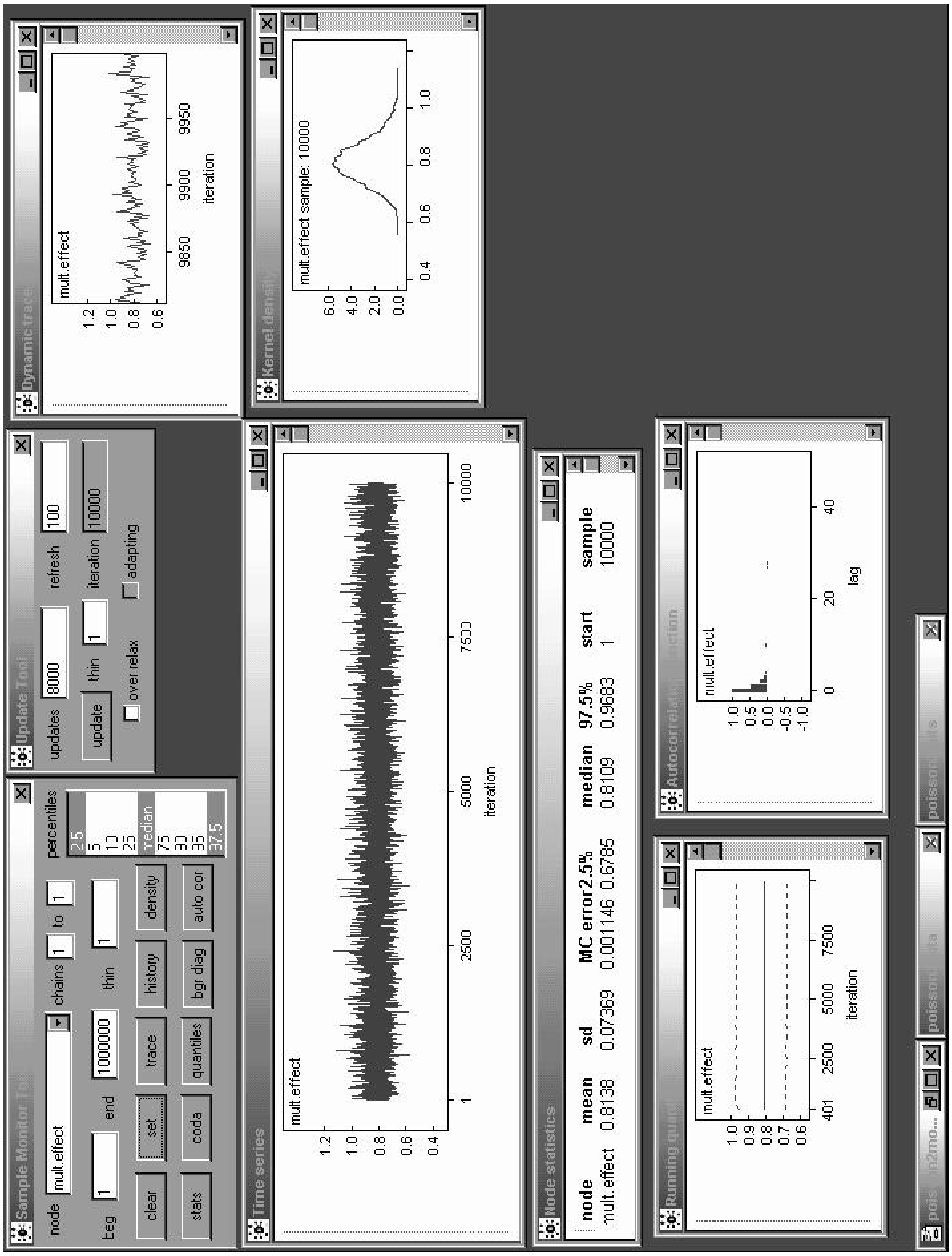
updates 2000 refresh 100
thin 1 iteration 2000
□ over relax □ adapting

num of chains 1
for chain 1

node mult.effect
chains 1 to 1
beg 1 end 1000000 thin 1

updates took 26 s

Figure 4.6: *Fitting a multiplicative-effect Poisson model to the IHGA data in WinBUGS.*



Sample Monitor Tool

node mult.effect chains 1 to 1

beg 1 end 1000000 thin 1

Update Tool

updates 8000 refresh 100

thin 1 iteration 10000

□ over relax □ adapting

Node statistics

```
node        mean    sd      MC error  2.5%    median  97.5%   start  sample
mult.effect 0.8138  0.07369 0.001146  0.6785  0.8109  0.9683  1      10000
```

Figure 4.7: *Monitoring the multiplicative effect parameter.*

With this model it is as if the experimental people's average underlying rates of hospitalization have been multiplied by 0.82,
give or take about 0.07.

The additive and multiplicative effects are similar here, because both are not too far from zero.

Extra-Poisson variability. However, none of this has verified that the Poisson model is reasonable for these data—the histograms show that the Gaussian model is clearly unreasonable, but the diagnostic plots in `WinBUGS` and `CODA` only check on the adequacy of the MCMC sampling, not the model.

In fact we had a good clue that the data are not Poisson back on page 2: as noted in part 2, the Poisson(λ) distribution has mean λ and also variance λ—in other words, the variance-to-mean-ratio (VTMR) for the Poisson is 1. But

```
> var( C ) / mean( C )
[1] 1.62599

> var( E ) / mean( E )
[1] 1.322979
```

i.e., the data exhibit extra-Poisson variability (VTMR > 1).

This actually makes good sense if you think about it, as follows.

The Poisson model assumes that everybody in the control group has the same underlying rate λ_C of hospitalization, and similarly everybody in the experimental group has the same rate λ_E.

In reality it's far more reasonable to think that each person has his/her own underlying rate of hospitalization that depends on baseline health status, age, and various other things.

Now Hendriksen forgot to measure (or at least to report on) these other variables (he may have hoped that the randomization would balance them between C and E)—the only predictor we have is x, the experimental status dummy variable—so the best we can do is to lump all of these other unobserved predictor variables together into a kind of "error" term e.

This amounts to expanding the second Poisson model (11) above: for $i = 1, \ldots, n = 572$ the new model is

$$\begin{array}{rcl} (y_i \,|\, \lambda_i) & \stackrel{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) \\ \log(\lambda_i) & = & \gamma_0 + \gamma_1 x_i + e_i \end{array} \qquad (4.9)$$

$$\begin{array}{rcl} e_i & \overset{\text{IID}}{\sim} & N\big(0, \sigma_e^2\big) \\ \big(\gamma_0, \gamma_1, \sigma_e^2\big) & \sim & \text{diffuse.} \end{array}$$

4.1.3 Random-effects Poisson regression modeling

The Gaussian choice for the error distribution is conventional, not dictated by the science of the problem (although if there were a lot of unobserved predictors hidden inside the e_i their weighted sum would be close to normal by the Central Limit Theorem).

Model (16) is an expansion of the earlier model (11) because you can obtain model (11) from (16) by setting $\sigma_e^2 = 0$, whereas with (16) we're letting σ_e^2 vary and learning about it from the data.

The addition of the random effects e_i to the model is one way to address the extra-Poisson variability: this model would be called a lognormal mixture of Poisson distributions (or a random effects Poisson regression (REPR) model) because it's as if each person's λ is drawn from a lognormal distribution and then his/her number of hospitalizations y is drawn from a Poisson distribution with his/her λ, and this mixing process will make the variance of y bigger than its mean.

The new WinBUGS model is

```
{

  gamma.0 ~ dnorm( 0.0, 1.0E-4 )
  gamma.1 ~ dnorm( 0.0, 1.0E-4 )
  tau.e ~ dgamma( 0.001, 0.001 )

  for ( i in 1:n ) {

    e[ i ] ~ dnorm( 0.0, tau.e )
    log( lambda[ i ] ) <- gamma.0 + gamma.1 * x[ i ] +
      e[ i ]
    y[ i ] ~ dpois( lambda[ i ] )

  }

  lambda.C <- exp( gamma.0 )
  lambda.E <- exp( gamma.0 + gamma.1 )
  mult.effect <- exp( gamma.1 )
  sigma.e <- 1.0 / sqrt( tau.e )

}
```

I again use a diffuse $\Gamma(\epsilon, \epsilon)$ prior (with $\epsilon = 0.001$) for the precision τ_e of the random effects.

With a model like that in equation (16), there are n random effects e_i that need to be sampled as nodes in the graph (the e_i play the role of auxiliary variables in the MCMC) along with the fixed effects (γ_0, γ_1) and the variance parameter σ_e^2.

In earlier releases of the software, at least, this made it more crucial to give `WinBUGS` good starting values.

Here `WinBUGS` release 1.3 has figured out that random draws like $1.66 \cdot 10^{-316}$ result from the generic (and quite poor) initial values $(\gamma_0, \gamma_1, \tau_e) = (0.0, 0.0, 1.0)$ and has refused to continue sampling.

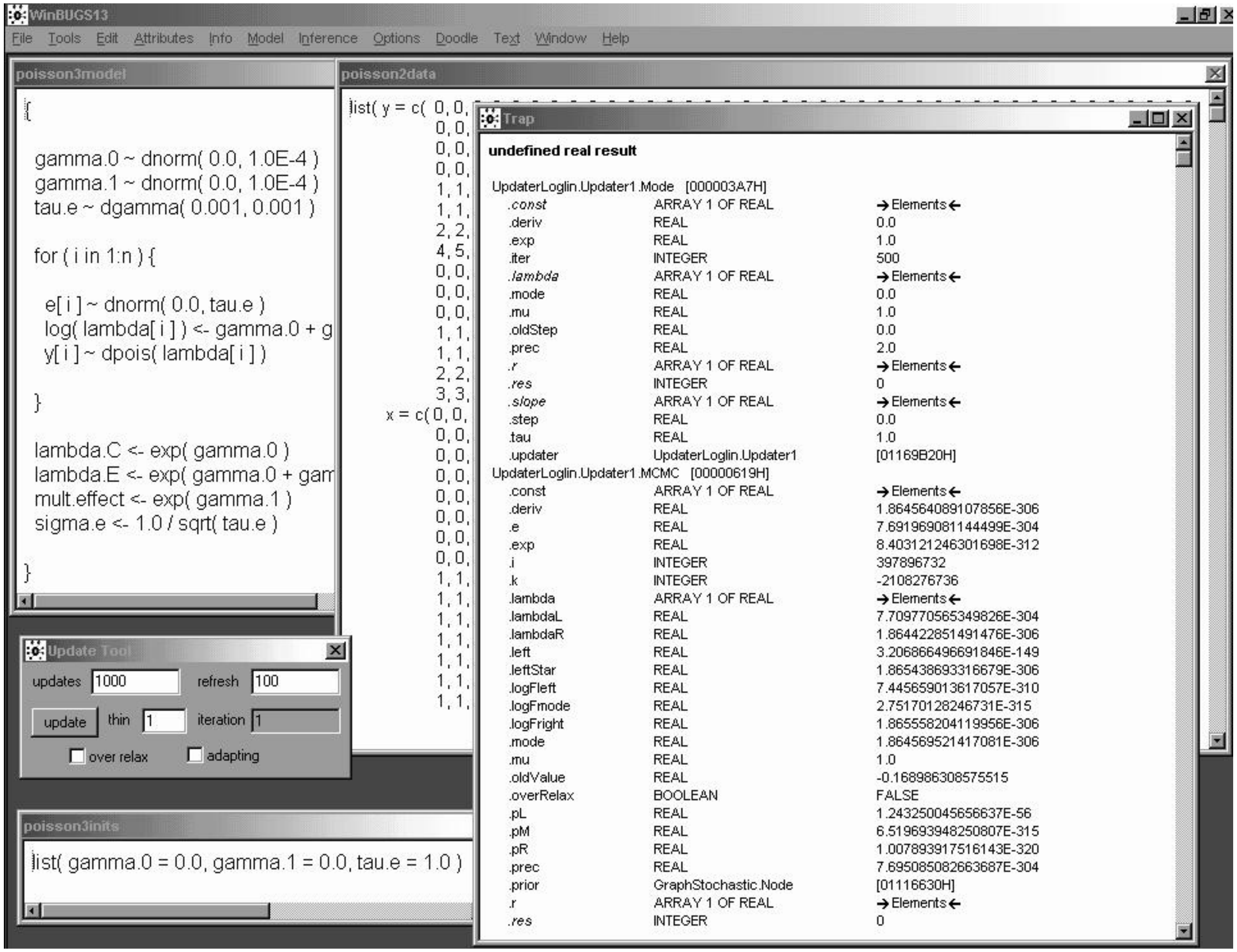
Sensitivity to initial values. Warning: `WinBUGS` can fail, particularly in random-effects models, when you give it initial values that are not very close to the final posterior means; an example in release 1.3 is the REPR model (16) on the IHGA data with the "generic" starting values $(\gamma_0, \gamma_1, \tau_e) = (0.0, 0.0, 1.0)$.

When this problem arises there are two ways out in `WinBUGS`: trial and error, or a calculation (see below).

NB `MLwiN` does not have this problem—it gets its starting values from maximum likelihood (the mode of the likelihood function is often a decent approximation to the mean or mode of the posterior).

Technical note. To get a decent starting value for τ_e in model (16) you can calculate as follows: renaming the random effects η_i to avoid confusion with the number e, (1) $V(y_i) = V[E(y_i\,|\eta_i)] + E[V(y_i\,|\eta_i)]$, where (2) $(y_i\,|\eta_i) \sim \text{Poisson}\left(e^{\gamma_0+\gamma_1 x_i+\eta_i}\right)$, so $E(y_i\,|\eta_i) = V(y_i\,|\eta_i) = e^{\gamma_0+\gamma_1 x_i+\eta_i}$. Then (3) $V[E(y_i\,|\eta_i)] = V(e^{\gamma_0+\gamma_1 x_i+\eta_i}) = e^{2(\gamma_0+\gamma_1 x_i)} V(e^{\eta_i})$ and $E[V(y_i\,|\eta_i)] = E(e^{\gamma_0+\gamma_1 x_i+\eta_i}) = e^{\gamma_0+\gamma_1 x_i} E(e^{\eta_i})$. Now (4) e^{η_i} is lognormal with mean 0 and variance σ_e^2 on the log scale, so $E(e^{\eta_i}) = e^{\frac{1}{2}\sigma_e^2}$ and $V(e^{\eta_i}) = e^{\sigma_e^2}\left(e^{\sigma_e^2} - 1\right)$, yielding finally $V(y_i) = e^{2(\gamma_0+\gamma_1 x_i)+\frac{1}{2}\sigma_e^2} + e^{\gamma_0+\gamma_1 x_i+\sigma_e^2}\left(e^{\sigma_e^2} - 1\right)$. (5) Plugging in $x_i = 0$ for the C group, whose sample variance is 1.54, and using the value $\gamma_0 = -0.29$ from runs with previous models, gives an equation for σ_e^2 that can be solved numerically, yielding $\sigma_e^2 \doteq 0.5$ and $\tau_e \doteq 2$.

Interestingly, `WinBUGS` release 1.4 is able to sample successfully with the


WinBUGS13
File Tools Edit Attributes Info Model Inference Options Doodle Text Window Help
poisson3model
poisson2data
Trap
Update Tool
poisson3inits

```
{

 gamma.0 ~ dnorm( 0.0, 1.0E-4 )
 gamma.1 ~ dnorm( 0.0, 1.0E-4 )
 tau.e ~ dgamma( 0.001, 0.001 )

 for ( i in 1:n ) {

   e[ i ] ~ dnorm( 0.0, tau.e )
   log( lambda[ i ] ) <- gamma.0 + g
   y[ i ] ~ dpois( lambda[ i ] )

 }

 lambda.C <- exp( gamma.0 )
 lambda.E <- exp( gamma.0 + gam
 mult.effect <- exp( gamma.1 )
 sigma.e <- 1.0 / sqrt( tau.e )

}
```

```
list( y = c( 0,0,
             0,0,
             0,0,
             0,0,
             1,1,
             1,1,
             2,2,
             4,5,
             0,0,
             0,0,
             0,0,
             1,1,
             1,1,
             2,2,
             3,3,
      x = c( 0,0,
             0,0,
             0,0,
             0,0,
             0,0,
             0,0,
             0,0,
             0,0,
             1,1,
             1,1,
             1,1,
             1,1,
             1,1,
             1,1,
             1,1,
```

```
undefined real result

UpdaterLoglin.Updater1.Mode   [000003A7H]
   .const        ARRAY 1 OF REAL            →Elements←
   .deriv        REAL                       0.0
   .exp          REAL                       1.0
   .iter         INTEGER                    500
   .lambda       ARRAY 1 OF REAL            →Elements←
   .mode         REAL                       0.0
   .mu           REAL                       1.0
   .oldStep      REAL                       0.0
   .prec         REAL                       2.0
   .r            ARRAY 1 OF REAL            →Elements←
   .res          INTEGER                    0
   .slope        ARRAY 1 OF REAL            →Elements←
   .step         REAL                       0.0
   .tau          REAL                       1.0
   .updater      UpdaterLoglin.Updater1     [01169B20H]
UpdaterLoglin.Updater1.MCMC   [00000619H]
   .const        ARRAY 1 OF REAL            →Elements←
   .deriv        REAL                       1.864564089107856E-306
   .e            REAL                       7.691969081144499E-304
   .exp          REAL                       8.403121246301698E-312
   .i            INTEGER                    397896732
   .k            INTEGER                    -2108276736
   .lambda       ARRAY 1 OF REAL            →Elements←
   .lambdaL      REAL                       7.709770565349826E-304
   .lambdaR      REAL                       1.864422851491476E-306
   .left         REAL                       3.206866496691846E-149
   .leftStar     REAL                       1.865438693316679E-306
   .logFleft     REAL                       7.445659013617057E-310
   .logFmode     REAL                       2.75170128246731E-315
   .logFright    REAL                       1.865558204119956E-306
   .mode         REAL                       1.864569521417081E-306
   .mu           REAL                       1.0
   .oldValue     REAL                       -0.168986308575515
   .overRelax    BOOLEAN                    FALSE
   .pL           REAL                       1.24325004565663 7E-56
   .pM           REAL                       6.519693948250807E-315
   .pR           REAL                       1.007893917516143E-320
   .prec         REAL                       7.695085082663687E-304
   .prior        GraphStochastic.Node       [01116630H]
   .r            ARRAY 1 OF REAL            →Elements←
   .res          INTEGER                    0
```

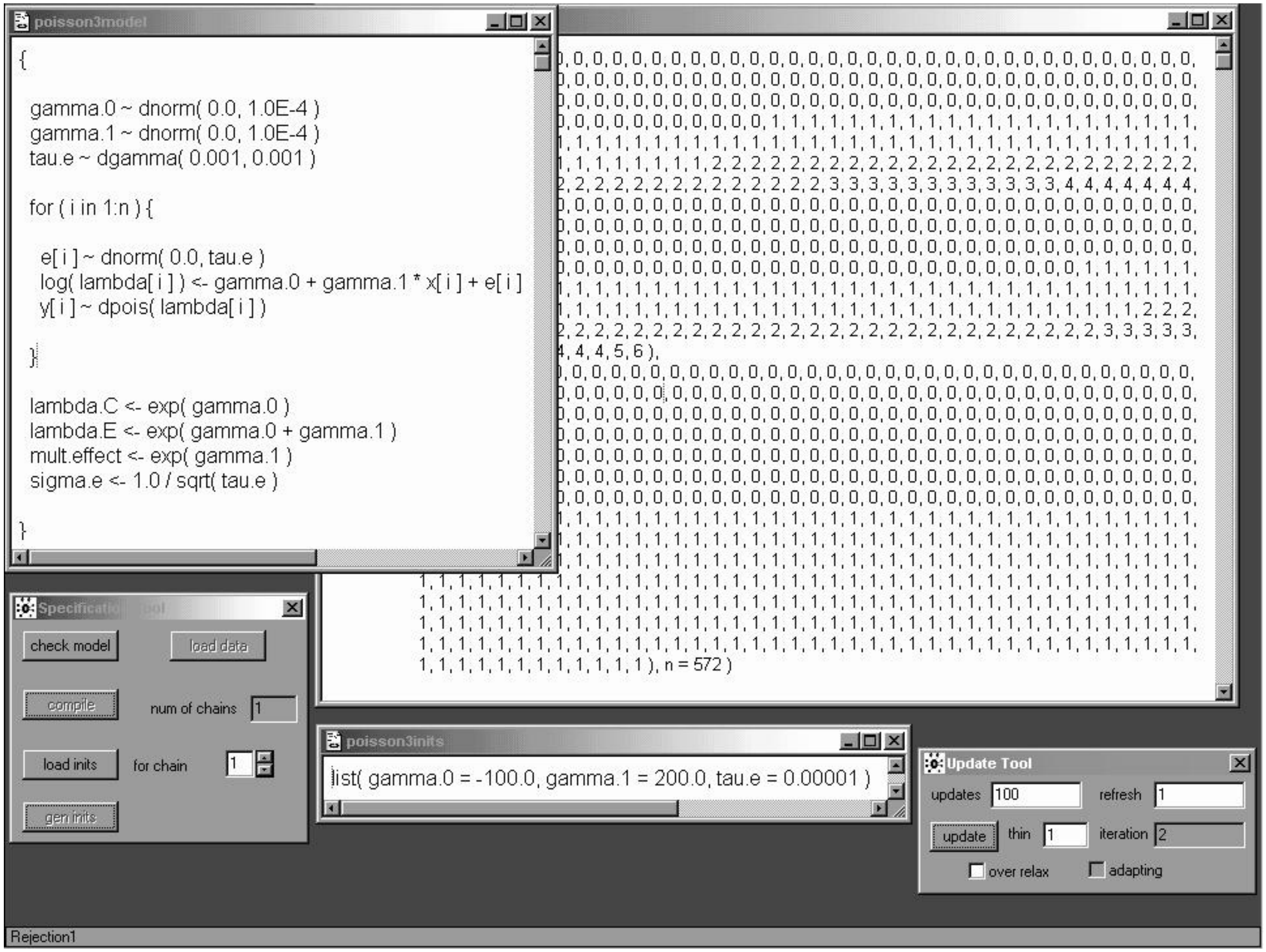
updates 1000 refresh 100

update thin 1 iteration 1

□ over relax □ adapting

```
list( gamma.0 = 0.0, gamma.1 = 0.0, tau.e = 1.0 )
```

Figure 4.8: *Bad initial values yielding the dreaded* ***Trap*** *window.*



```
{

 gamma.0 ~ dnorm( 0.0, 1.0E-4 )
 gamma.1 ~ dnorm( 0.0, 1.0E-4 )
 tau.e ~ dgamma( 0.001, 0.001 )

 for ( i in 1:n ) {

  e[ i ] ~ dnorm( 0.0, tau.e )
  log( lambda[ i ] ) <- gamma.0 + gamma.1 * x[ i ] + e[ i ]
  y[ i ] ~ dpois( lambda[ i ] )

 }

 lambda.C <- exp( gamma.0 )
 lambda.E <- exp( gamma.0 + gamma.1 )
 mult.effect <- exp( gamma.1 )
 sigma.e <- 1.0 / sqrt( tau.e )

}
```

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3,
4, 4, 4, 5, 6 ),
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ), n = 572 )
```

num of chains 1

for chain 1

```
list( gamma.0 = -100.0, gamma.1 = 200.0, tau.e = 0.00001 )
```

updates 100 refresh 1

thin 1 iteration 2

□ over relax □ adapting

Rejection1

Figure 4.9: *WinBUGS release 1.4 will refuse to sample with truly absurd initial values.*

generic starting values $(\gamma_0, \gamma_1, \tau_e) = (0.0, 0.0, 1.0)$, although of course a longer burn-in period would be needed when they're used; you have to try truly absurd initial values to get it to fall over, and when it does so the error message ("`Rejection1`") in the lower left corner is more discreet.

With a better set of initial values—$(\gamma_0, \gamma_1, \tau_e) = (-0.058, -0.21, 2.0)$, obtained from (a) the earlier Poisson models (in the case of the regression parameters γ_j) and (b) either a calculation like the one on the bottom of page 29 or trial and error—`WinBUGS` is able to make progress, although this model takes a fairly long time to fit in release 1.4: a burn-in of 1,000 takes 11 seconds at 1.0 PC GHz (the code runs about twice as fast in release 1.3 for some reason).

A monitoring run of 5,000 iterations reveals that the random effects make everything mix more slowly: λ_C (this page) and λ_E and the multiplicative effect (next page) all behave like AR_1 series with $\hat{\rho}_1 \doteq 0.7$, 0.5, and 0.6, respectively.

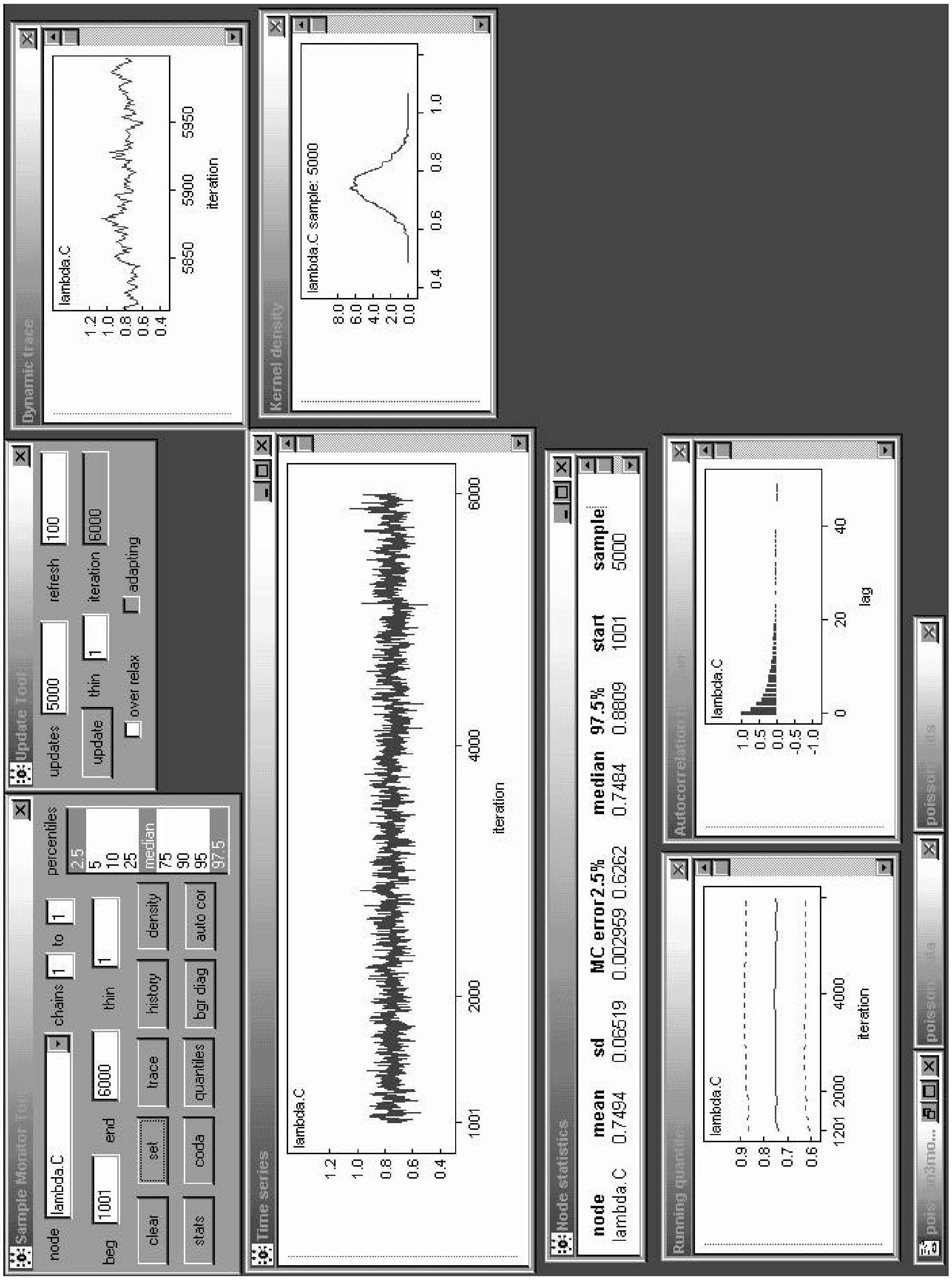
Learning about σ_e in this model is slow: its autocorrelation function is that of an AR_1 with a high value of $\hat{\rho}_1$ (equation (55) on page 76 of part 3 of the lecture notes gives $\hat{\rho}_1 \doteq 0.92$).

The MCSE of the posterior mean for σ_e based on 5,000 draws is 0.005182; to get this down to (say) 0.001 I need to increase the length of the monitoring run by a factor of $\left(\frac{0.005182}{0.001}\right)^2 \doteq 26.9$, meaning a total run of about $(26.9)(5,000) \doteq 134,000$ iterations (this takes about half an hour at 1 PC GHz).

There is clear evidence that σ_e is far from 0—its posterior mean and SD are estimated as 0.675 (with an MCSE of about 0.001 after 134,000 iterations) and 0.074, respectively—meaning that the model expansion from (11) to (16) was amply justified.

(Another way to achieve the goal of describing the extra-Poisson variability would be to fit different negative binomial distributions to the observed counts in the C and E groups—the negative binomial is a gamma mixture of Poissons, and the gamma and lognormal distributions often fit long-tailed data about equally well, so you would not be surprised to find that the two approaches give similar results.)

Table 4.5. Comparison of inferential conclusions about the multiplicative effect parameter e^{γ_1} from the fixed-effects and random-effects Poisson regression models.



Sample Monitor Tool

node lambda.C chains 1 to 1

beg 1001 end 6000 thin 1

percentiles: 2.5, 5, 10, 25, median, 75, 90, 95, 97.5

Update Tool

updates 5000 refresh 100

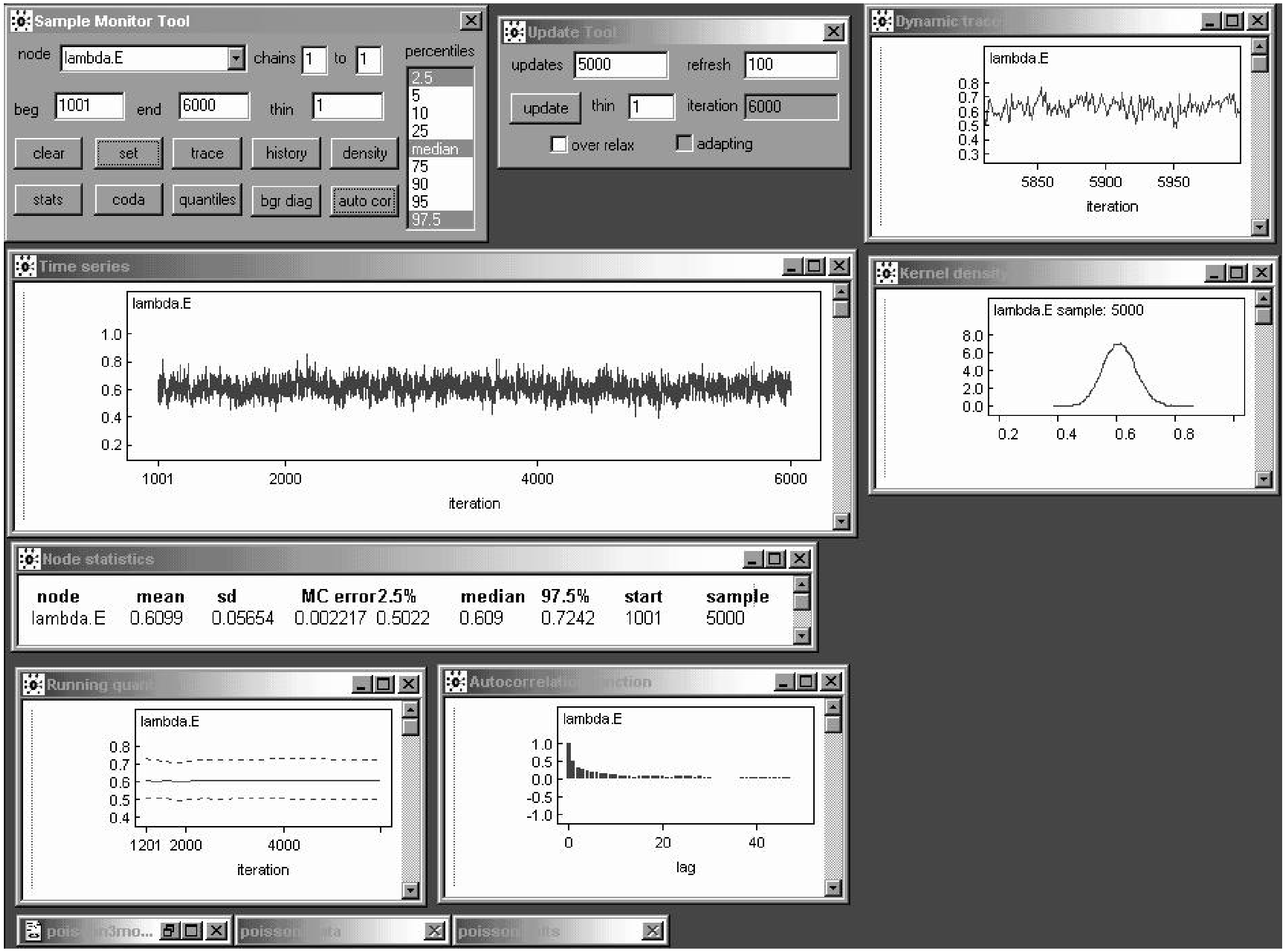
thin 1 iteration 6000

□ over relax □ adapting

Node statistics

```
node      mean   sd      MC error2.5%  median 97.5%  start sample
lambda.C  0.7494 0.06519 0.002959 0.6262 0.7484 0.8809 1001  5000
```

Figure 4.10: *Monitoring λ_C in the REPR model.*



node lambda.E chains 1 to 1

beg 1001 end 6000 thin 1

percentiles
2.5
5
10
25
median
75
90
95
97.5

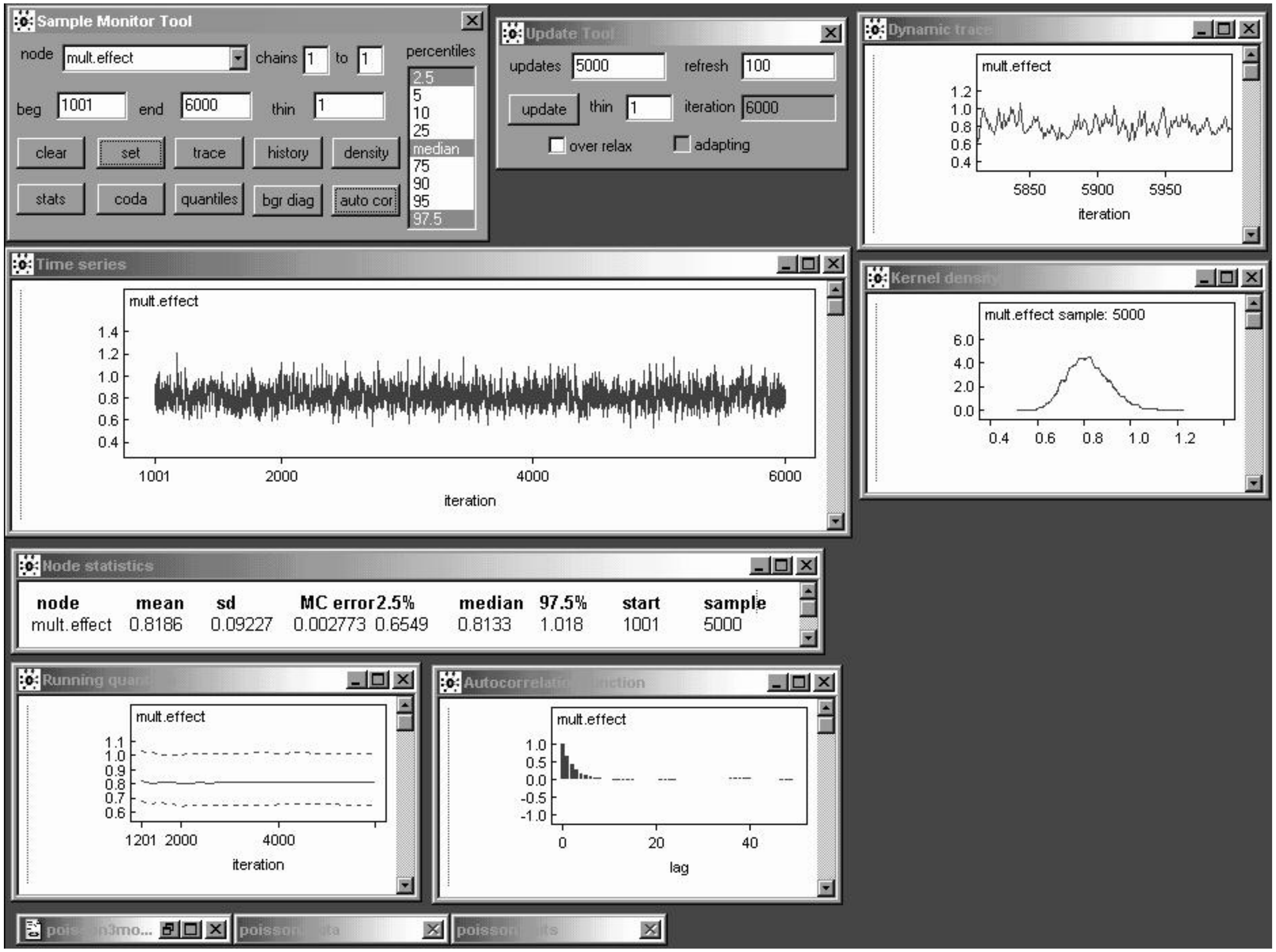
updates 5000 refresh 100

thin 1 iteration 6000

□ over relax □ adapting

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
lambda.E	0.6099	0.05654	0.002217	0.5022	0.609	0.7242	1001	5000

Figure 4.11: *Monitoring λ_E in the REPR model.*



node mult.effect chains 1 to 1

beg 1001 end 6000 thin 1

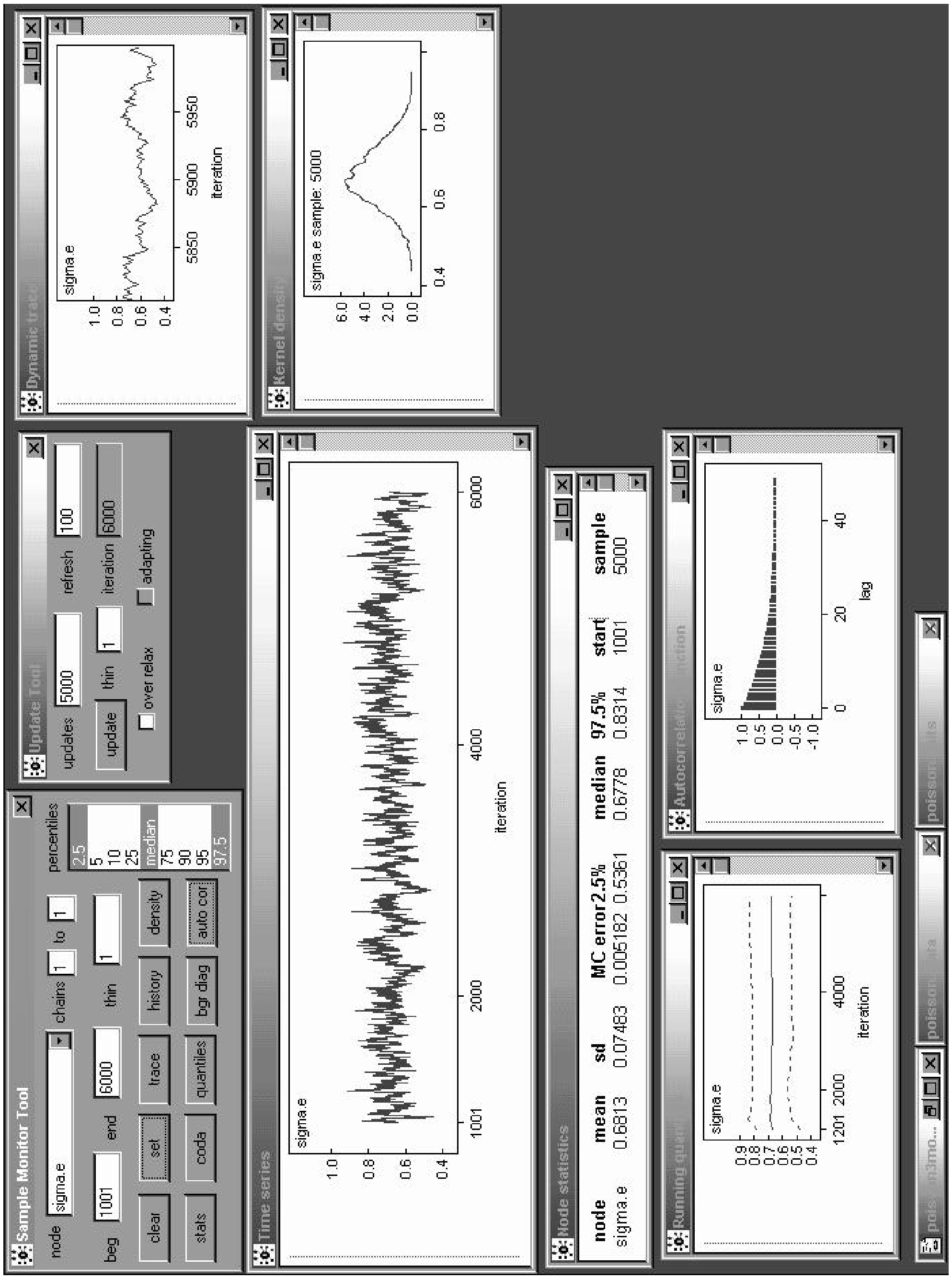
updates 5000 refresh 100

thin 1 iteration 6000

□ over relax □ adapting

```
node         mean    sd       MC error2.5%     median  97.5%  start  sample
mult.effect  0.8186  0.09227  0.002773 0.6549  0.8133  1.018  1001   5000
```

Figure 4.12: *Monitoring the multiplicative effect parameter in the REPR model.*

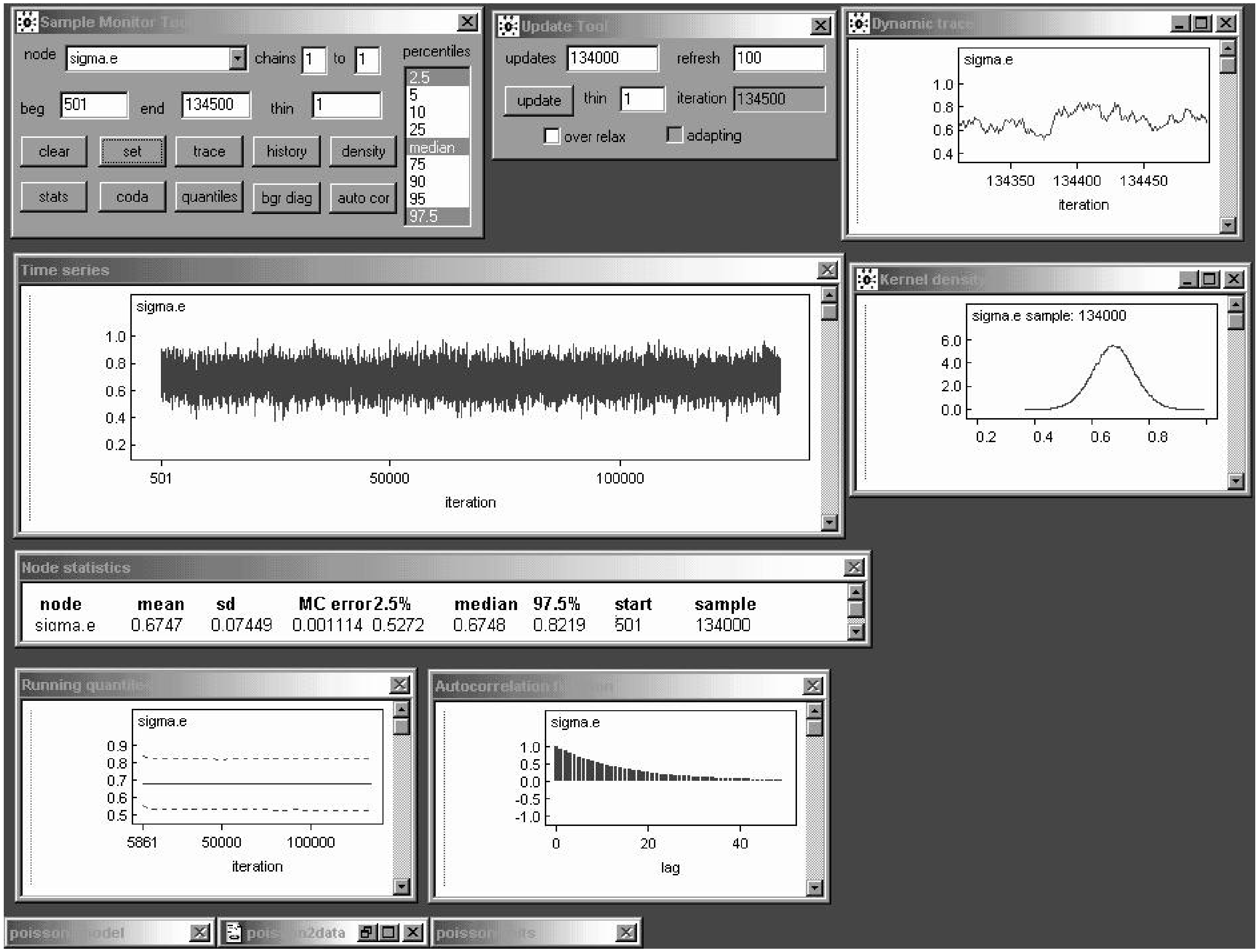


node sigma.e
beg 1001 end 6000
chains 1 to 1
thin 1

updates 5000 refresh 100
thin 1 iteration 6000

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
sigma.e	0.6813	0.07483	0.005182	0.5361	0.6778	0.8314	1001	5000

Figure 4.13: *Monitoring σ_e in the REPR model.*



node sigma.e chains 1 to 1

beg 501 end 134500 thin 1

updates 134000 refresh 100

thin 1 iteration 134500

□ over relax □ adapting

```
node     mean    sd       MC error2.5%      median  97.5%   start  sample
sigma.e  0.6747  0.07449  0.001114 0.5272   0.6748  0.8219  501    134000
```

Figure 4.14: *There is clear evidence that σ_e is far from 0.*

Model	Posterior Mean	Posterior SD	Central 95% Interval
FEPR	0.816	0.0735	$(0.683, 0.969)$
REPR	0.830	0.0921	$(0.665, 1.02)$

Table 4.5 compares the REPR model inferential results with those from model (11), which could also be called a fixed-effects Poisson regression (FEPR) model.

The “error” SD σ_e has posterior mean 0.68, give or take about 0.07 (on the $\log(\lambda)$ scale), corresponding to substantial extra-Poisson variability, which translates into increased uncertainty about the multiplicative effect parameter e^{γ_1}.

I'll argue later that the REPR model fits the data well, so the conclusion I'd publish from these data is that IHGA reduces the average number of hospitalizations per two years by about $100\,(1 - 0.083)\% = 17\%$ give or take about 9% (ironically this conclusion is similar to that from the Gaussian model, but this is coincidence).

4.2 Bayesian model choice

What is a Bayesian model?

I'd like model to arise as much as possible from contextual information (scientific, policy, business, ...).

de Finetti (1970): Bayesian model = joint predictive distribution

$$p(y) = p(y_1, \ldots, y_n) \tag{4.10}$$

for as-yet-unobserved observables $y = (y_1, \ldots, y_n)$.

Example 1: Data = health outcomes for all patients at one hospital with heart attack admission diagnosis.

Simplest possible: $y_i = 1$ if patient i dies within 30 days of admission, 0 otherwise.

de Finetti (1930): in absence of any other information, predictive uncertainty about y_i is exchangeable.

Representation theorem for binary data: if $(y_1, \ldots, y_n)$ part of infinitely exchangeable sequence, all coherent joint predictive distributions $p(y_1, \ldots, y_n)$ must have simple hierarchical form

$$\begin{array}{rcl} \theta & \sim & p(\theta) \\ (y_i|\theta) & \overset{\text{IID}}{\sim} & \text{Bernoulli}(\theta), \end{array} \tag{4.11}$$

where $\theta = P(y_i = 1) =$ limiting value of mean of y_i in infinite sequence.

Mathematically $p(\theta)$ is mixing distribution in

$$p(y_1, \ldots, y_n) = \int_0^1 \theta^s (1-\theta)^{n-s} \, p(\theta) \, d\theta, \tag{4.12}$$

where $s = \sum_{i=1}^n y_i$; statistically, $p(\theta)$ provides opportunity to quantify prior information about θ and combine with information in y.

Thus, in simplest situation, Bayesian model specification = choice of scientifically appropriate prior distribution $p(\theta)$.

Example 2 (elaborating Example 1): Now I want to predict real-valued sickness-at-admission score instead of mortality (still no covariates).

Uncertainty about y_i still exchangeable; de Finetti's (1937) representation theorem for real-valued data: if $(y_1, \ldots, y_n)$ part of infinitely exchangeable sequence, all coherent joint predictive distributions $p(y_1, \ldots, y_n)$ must have (no longer quite so simple) hierarchical form

$$\begin{array}{rcl} F & \sim & p(F) \\ (y_i|F) & \overset{\text{IID}}{\sim} & F, \end{array} \tag{4.13}$$

where F = limiting empirical cumulative distribution function (CDF) of infinite sequence $(y_1, y_2, \ldots)$.

Bayesian nonparametrics. Thus here Bayesian model specification = choosing scientifically appropriate mixing (prior) distribution $p(F)$ for F.

However, F is infinite-dimensional parameter; putting probability distribution on $\mathcal{D} = \{$all possible CDFs$\}$ is harder.

Specifying distributions on function spaces is task of Bayesian nonparametric (BNP) modeling (e.g., Dey et al. 1998).

Example 3 (elaborating Example 2): In practice, in addition to outcomes y_i, covariates x_{ij} will typically be available.

For instance (Hendriksen et al. 1984), 572 elderly people randomized, 287 to control (C) group (standard care) and 285 to treatment (T) group (standard care plus in-home geriatric assessment (IHGA): preventive medicine in which each person's medical/social needs assessed, acted upon individually).

One important outcome was number of hospitalizations (in two years).

y_i^T, y_j^C = numbers of hospitalizations for treatment person i, control person j, respectively.

Suppose treatment/control (T/C) status is only available covariate.

Conditional exchangeability. Unconditional judgment of exchangeability across all 572 outcomes no longer automatically scientifically appropriate.

Instead design of experiment compels (at least initially) judgment of conditional exchangeability given T/C status (e.g., de Finetti 1938, Draper et al. 1993), as in

$$\begin{array}{c} (F_T, F_C) \;\sim\; p(F_T, F_C) \\ (y_i^T | F_T, F_C) \;\stackrel{\text{IID}}{\sim}\; F_T \;\Big|\; (y_j^C | F_T, F_C) \;\stackrel{\text{IID}}{\sim}\; F_C \end{array} \qquad (14)$$

It will later be necessary to decide if F_T, F_C are sufficiently similar that data are consistent with judgment of unconditional exchangeability (to see if treatment has effect or not).

This framework, in which (a) covariates specify conditional exchangeability judgments, (b) de Finetti's representation theorem reduces model specification task to placing appropriate prior distributions on CDFs, covers much of field of statistical inference/prediction; thus BNP/BSP modeling seem crucial to entire Bayesian enterprise over next 10–20 years.

4.2.1 Data-analytic model specification

However, placing prior distributions on CDFs is hard work, we don't have much experience with it yet; in meantime, in parallel with efforts to accumulate experience, people will still do parametric modeling, we need good tools for specifying such models.

Basic problem: Given future observables $y = (y_1, \ldots, y_n)$, I'm uncertain about y (first-order), but I'm also uncertain about how to specify my uncertainty about y (second-order).

Standard (data-analytic) approach to model specification involves initial choice, for structure of model, of standard parametric family, followed by modification of initial choice—once data begin to arrive—if data suggest deficiencies in original specification.

This approach (e.g., Draper 1995) is incoherent: it uses data both to specify prior distribution on structure space and to update using data-determined prior (result will typically be uncalibrated (too narrow) predictive distributions for future data).

Dilemma is example of Cromwell's Rule: initial model choice placed 0 prior probability on large regions of model space; formally all such regions must also have 0 posterior probability even if data indicate different prior on model space would have been better.

Two possible solutions.

• If use prior on F that places non-zero probability on all Kullback-Leibler neighborhoods of all densities (Walker et al. 2003; e.g., Pólya trees, Dirichlet process mixture priors, when chosen well), then BNP/BSP directly avoids Cromwell's Rule dilemma, at least for large n: as $n \rightarrow \infty$ posterior on F will shrug off any incorrect details of prior specification, will fully adapt to actual data-generating F (NB this assumes correct exchangeability judgments).

• Three-way cross-validation (3CV; Draper and Krnjajić 2005): taking usual cross-validation idea one step further,

(1) Divide data at random into *three* (non-overlapping) subsets S_i.

(2) Fit tentative {likelihood + prior} to S_1. Expand initial model in all feasible ways suggested by data exploration using S_1. Iterate until you're happy.

(3) Use final model (fit to S_1) from (2) to create predictive distributions for all data points in S_2. Compare actual outcomes with these distributions, checking for predictive calibration. Go back to (2), change likelihood as necessary, retune prior as necessary, to get good calibration. Iterate until you're happy.

(4) Announce final model (fit to S_1, S_2) from (3), and report predictive calibration of this model on data points in S_3 as indication of how well it would perform with new data.

With large n probably only need to do this once; with small and moderate n probably best to repeat (1–4) several times and combine results in some appropriate way (e.g., model averaging).

4.2.2 Model selection as a decision problem

Given method like 3CV which permits hunting around in model space without forfeiting calibration, how do you know when to stop?

It would seem self-evident that to choose model you have to say to what purpose model will be put, for how else will you know whether model is good enough?

Specifying this purpose demands decision-theoretic basis for model choice (e.g., Draper 1996; Key et al. 1998).

To take two examples,

(1) If you're going to choose which of several ways to behave in future, then model has to be good enough to reliably aid in choosing best behavior (see, e.g., Draper and Fouskakis example below); or

(2) If you wish to make scientific summary of what's known, then—remembering that hallmark of good science is good prediction—the model has to be good enough to make sufficiently accurate predictions of observable outcomes (in which dimensions along which accuracy is to be monitored are driven by what's scientifically relevant; see, e.g., log score results below).

Utility-based variable selection. Example 4: Draper and Fouskakis (2000, 2004) (also see Fouskakis and Draper 2002) give one example of decision-theoretic model choice in action, demonstrating that variable selection in regression models should often be governed by principle that final model should only contain variables that predict well enough given how much they cost to collect (see the figure below, which compares 2^{14} = 16,384 models).

Choosing the utility function. Any reasonable utility function in Example 4 will have two components, one quantifying data collection costs associated with construction of given sickness scale, other rewarding and penalizing scale's predictive successes, failures.

This requires intimate knowledge of real-world consequences of correct choices, mistakes—a level of understanding that's always desirable but is frequently costly in time, money to acquire.

Sometimes the main goal instead is summary of scientific knowledge, which suggests (as noted above) a utility function that rewards predictive accuracy.

On calibration grounds it's mistake, however, to use data twice in measuring this sort of thing (once to make predictions, again with same data to see how good they are).

Out-of-sample predictive validation (e.g., Geisser and Eddy 1979, Gelfand et al. 1992) solves this problem: e.g., successively remove each observation y_j one at a time, construct predictive distribution for y_j based on y_{-j} (data vector with y_j removed), see where y_j falls in this distribution.

Log score as utility. This motivates log scoring rule (e.g., Good 1950; Bernardo and Smith 1994): with n data values y_j, when choosing among k models $M_i, i \in I$, find that model M_i which maximizes

$$\frac{1}{n}\sum_{j=1}^{n} \log p(y_j|M_i, y_{-j}). \tag{4.15}$$

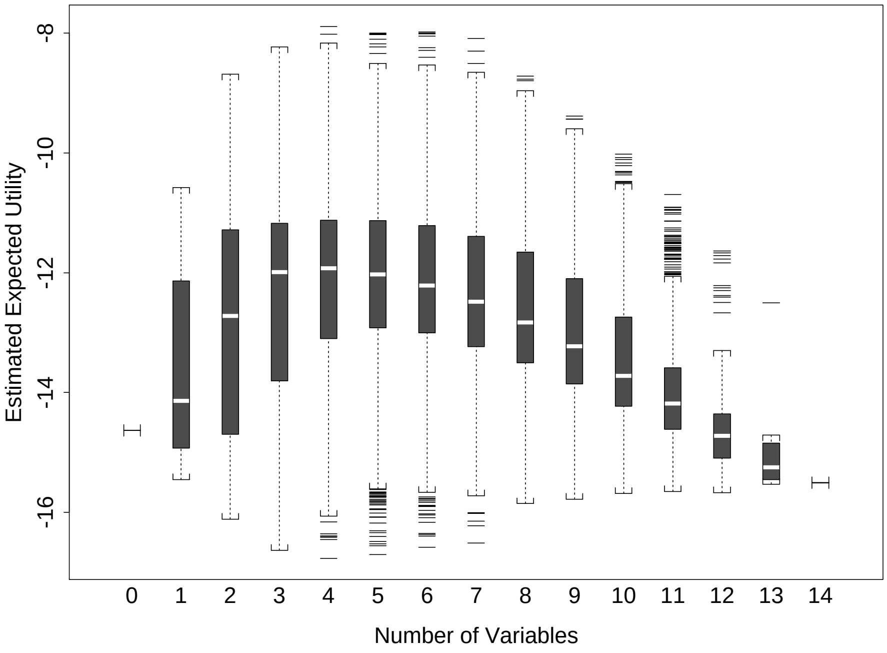


Figure 4.15: *Estimated expected utility as function of number of predictor variables, in problem involving construction of cost-effective scale to measure sickness at hospital admission of elderly pneumonia patients. Best models only have 4–6 sickness indicators out of 14 possible predictors.*

This can be given direct decision-theoretic justification: with utility function for model i

$$U(M_i|y) = \log p(y^*|M_i, y), \tag{4.16}$$

where y^* is future data value, expectation in MEU is over uncertainty about y^*; this expectation can be estimated (assuming exchangeability) by (15).

It can also be revealing to compute predictive z–scores, for observation j under model i:

$$z_{ij} = \frac{y_j - E(y_j|M_i, y_{-j})}{\sqrt{V(y_j|M_i, y_{-j})}}. \tag{4.17}$$

For good predictive calibration of M_i, $\{z_{ij}, j = 1, \ldots, n\}$ should have mean 0, standard deviation (SD) 1; often find instead that SD is larger than 1 (predictive uncertainty bands not wide enough).

Approximating log score utility. With large data sets, in situations in which predictive distribution has to be estimated by MCMC, log score expected utility (15) is computationally expensive; need fast approximation to it.

To see how this might be obtained, examine log score in simplest possible model M_0: for $i = 1, \ldots, n$,

$$\begin{aligned} \mu & \sim N\big(\mu_0, \sigma_\mu^2\big) \\ (Y_i|\mu) & \overset{\text{IID}}{\sim} N(\mu, \sigma^2) \end{aligned} \qquad (4.18)$$

with σ known, take highly diffuse prior on μ so that posterior for μ is approximately

$$(\mu|y) = (\mu|\bar{y}) \;\dot{\sim}\; N\left(\bar{y}, \frac{\sigma^2}{n}\right), \qquad (4.19)$$

where $y = (y_1, \ldots, y_n)$.

Then predictive distribution for next observation is approximately

$$(y_{n+1}|y) = (y_{n+1}|\bar{y}) \;\dot{\sim}\; N\left[\bar{y}, \sigma^2 \left(1 + \frac{1}{n}\right)\right], \qquad (4.20)$$

and log score, ignoring linear scaling constants, is

$$LS(M_0|y) = \sum_{j=1}^{n} \ln p(y_j|y_{-j}), \qquad (4.21)$$

where as before y_{-j} is y with observation j set aside.

But by same reasoning

$$p(y_j|y_{-j}) \doteq N\big(\bar{y}_{-j}, \sigma_n^2\big), \qquad (4.22)$$

where $\bar{y}_{-j}$ is sample mean with observation j omitted, $\sigma_n^2 = \sigma^2 \left(1 + \frac{1}{n-1}\right)$, so that

$$\begin{aligned} \ln p(y_j|y_{-j}) & \doteq c - \frac{1}{2\sigma_n^2}(y_j - \bar{y}_{-j})^2 \quad \text{and} \\ LS(M_0|y) & \doteq c_1 - c_2 \sum_{j=1}^{n} (y_j - \bar{y}_{-j})^2 \end{aligned} \qquad (4.23)$$

for some constants c_1 and c_2 with $c_2 > 0$.

Now it's interesting fact (related to behavior of jackknife), which you can prove by induction, that

$$\sum_{j=1}^{n} (y_j - \bar{y}_{-j})^2 = c \sum_{j=1}^{n} (y_j - \bar{y})^2 \tag{4.24}$$

for some $c > 0$, so finally for $c_2 > 0$ the result is that

$$LS(M_0|y) \doteq c_1 - c_2 \sum_{j=1}^{n} (y_j - \bar{y})^2, \tag{4.25}$$

i.e., in this model log score is almost perfectly negatively correlated with sample variance.

But in this model the deviance (minus twice the log likelihood) is

$$\begin{aligned} D(\mu) &= -2 \ln l(\mu|y) = c_0 - 2 \ln p(y|\mu) \\ &= c_0 + c_3 \sum_{j=1}^{n} (y_j - \mu)^2 \end{aligned} \tag{4.26}$$

for some $c_3 > 0$, encouraging suspicion that log score should be strongly related to deviance.

Deviance Information Criterion (DIC). Given parametric model $p(y|\theta)$, Spiegelhalter et al. (2002) define deviance information criterion (DIC) (by analogy with other information criteria) to be estimate $D(\bar{\theta})$ of model (lack of) fit (as measured by deviance) plus penalty for complexity equal to twice effective number of parameters p_D of model:

$$DIC(M|y) = D(\bar{\theta}) + 2\,\hat{p}_D, \tag{4.27}$$

where $\bar{\theta}$ is posterior mean of θ; they suggest that models with low DIC value are to be preferred over those with higher value.

When p_D is difficult to read directly from model (e.g., in complex hierarchical models, especially those with random effects), they motivate the following estimate, which is easy to compute from standard MCMC output:

$$\hat{p}_D = \overline{D(\theta)} - D(\bar{\theta}), \tag{4.28}$$

i.e., difference between posterior mean of deviance and deviance evaluated at posterior mean of parameters (`WinBUGS` release 1.4 will estimate these quantities).

In model M_0, p_D is of course 1, and $\bar{\theta} = \bar{y}$, so

$$DIC(M_0|y) = c_0 + c_3 \sum_{j=1}^{n} (y_j - \bar{y})^2 + 2 \tag{4.29}$$

and conclusion is that

$$-DIC(M_0|y) \doteq c_1 + c_2 LS(M_0|y) \tag{4.30}$$

for $c_2 > 0$, i.e., (if this generalizes) choosing model by maximizing log score and by minimizing DIC are approximately equivalent behaviors.

(This connection was hinted at in discussion of Spiegelhalter et al. 2002 but never really made explicit.)

4.2.3 The log score and the deviance information criterion

We're now (work in progress) exploring the scope of (30); in several simple models M so far we find for $c_2 > 0$ that

$$-DIC(M|y) \doteq c_1 + c_2 LS(M|y), \tag{4.31}$$

i.e., across repeated data sets generated from given model, even with small n DIC and LS can be fairly strongly negatively correlated.

Above argument generalizes to any situation in which predictive distribution is approximately Gaussian (e.g., Poisson(λ) likelihood with large λ, Beta(α, β) likelihood with large $(\alpha + \beta)$, etc.).

Example 3 continued. With one-sample count data (like number of hospitalizations in the T and C portions of IHGA data), people often choose between fixed- and random-effects Poisson model formulations: for $i = 1, \ldots, n$, and, e.g., with diffuse priors,

$$M_1\text{:} \left\{ \begin{array}{ccc} \lambda & \sim & p(\lambda) \\ (y_i|\lambda) & \stackrel{\text{IID}}{\sim} & \text{Poisson}(\lambda) \end{array} \right\} \quad \text{versus} \tag{4.32}$$

$$M_2\text{:} \left\{ \begin{array}{ccc} (\beta_0, \sigma^2) & \sim & p(\beta_0, \sigma^2) \\ (y_i|\lambda_i) & \stackrel{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) \\ \log(\lambda_i) & = & \beta_0 + e_i \\ e_i & \stackrel{\text{IID}}{\sim} & N(0, \sigma^2) \end{array} \right\} \tag{4.33}$$

M_1 is special case of M_2 with $\left(\sigma^2 = 0, \lambda = e^{\beta_0}\right)$; likelihood in M_2 is Lognormal mixture of Poissons (often similar to fitting negative binomial distribution, which is Gamma mixture of Poissons).

We conducted partial-factorial simulation study with factors $\{n = 18, 32, 42, 56, 100\}$, $\{\beta_0 = 0.0, 1.0, 2.0\}$, $\{\sigma^2 = 0.0, 0.5, 1.0, 1.5, 2.0\}$ in which (data-generating mechanism, assumed model) = $\{(M_1, M_1), (M_1, M_2), (M_2, M_1), (M_2, M_2)\}$; in each cell of this grid we used 100 simulation replications.

When assumed model is M_1 (fixed-effects Poisson), LS and DIC are almost perfectly negatively correlated (we have mathematical explanation of this).

When assumed model is M_2 (random-effects Poisson), LS and DIC are less strongly negatively correlated, but correlation increases with n.

Example 3. As example of correspondence between LS and DIC in real problem, IHGA data were as follows:

Distribution of number of hospitalizations in IHGA study over two-year period:

	Number of Hospitalizations										
Group	0	1	2	3	4	5	6	7	n	Mean	SD
Control	138	77	46	12	8	4	0	2	287	0.944	1.24
Treatment	147	83	37	13	3	1	1	0	285	0.768	1.01

Evidently IHGA lowered mean hospitalization rate (for these elderly Danish people, at least) by $(0.944 - 0.768) = 0.176$, which is about a

$$100 \left(\frac{0.768 - 0.944}{0.944} \right) \% = 19\%$$

reduction from control level, a difference that's large in clinical terms.

Four possible models for these data (not all of them good):

- Two-independent-sample Gaussian (diffuse priors);
- One-sample Poisson (diffuse prior), pretending treatment and control λs are equal;
- Two-independent-sample Poisson (diffuse priors), which is equivalent to fixed-effects Poisson regression (FEPR); and
- Random-effects Poisson regression (REPR), because C and T variance-to-mean ratios (VTMRs) are 1.63 and 1.32, respectively:

$$\begin{aligned} (y_i \,|\, \lambda_i) &\overset{\text{indep}}{\sim} \text{Poisson}(\lambda_i) \\ \log(\lambda_i) &= \beta_0 + \beta_1 x_i + e_i \end{aligned} \tag{4.34}$$

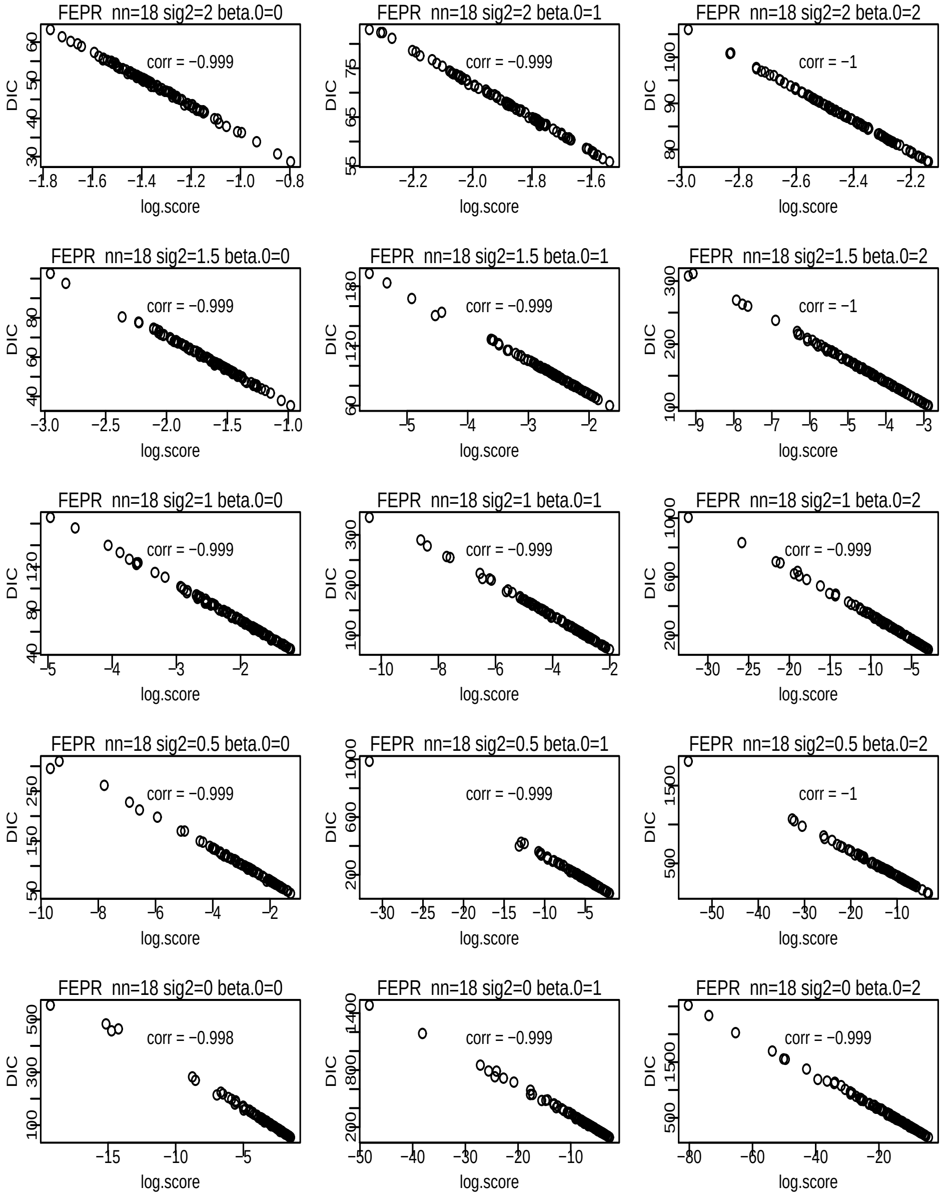


Figure 4.16: *When the assumed model is M_1 (fixed-effects Poisson), LS and DIC are almost perfectly negatively correlated.*

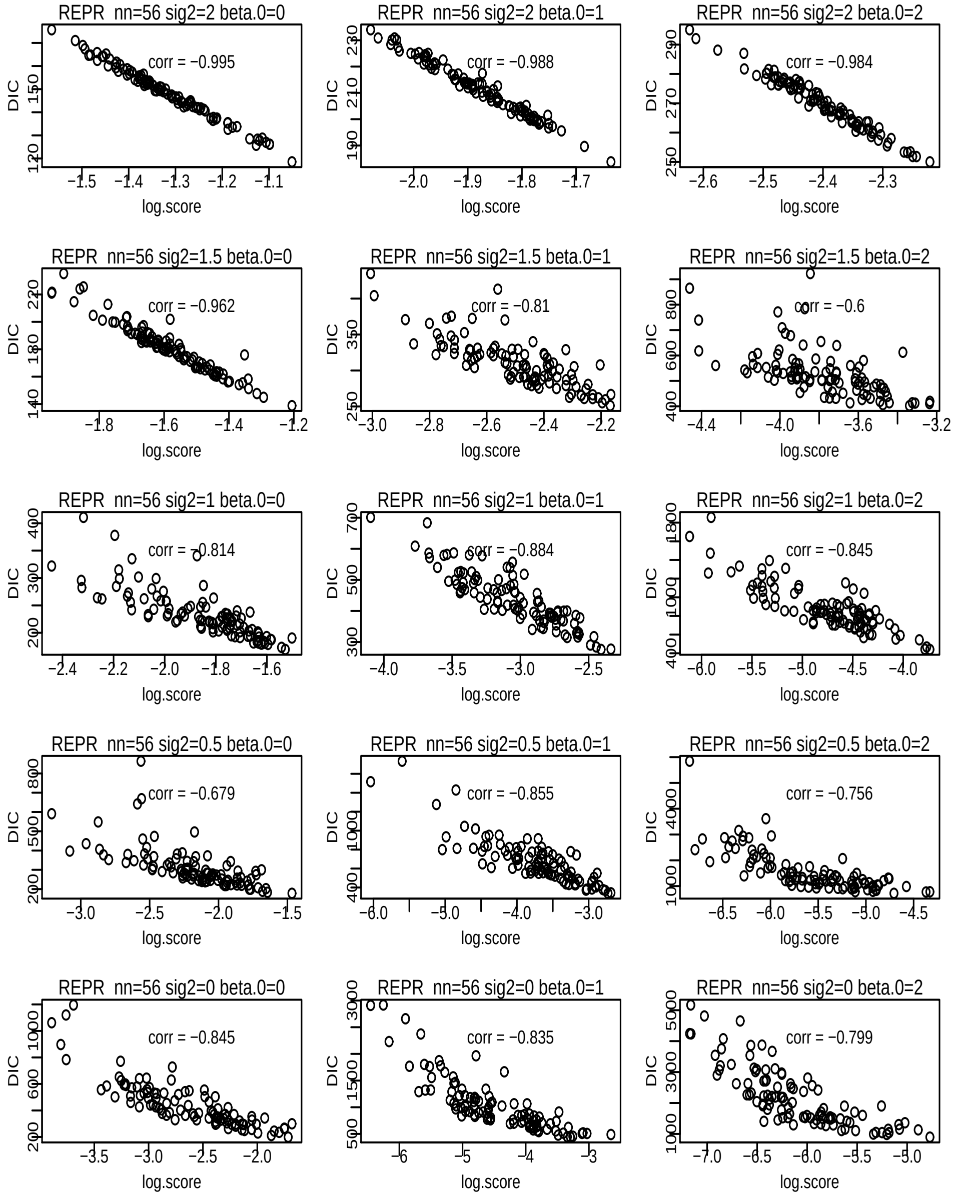


Figure 4.17: *When the assumed model is M_2 (random-effects Poisson), LS and DIC are less strongly negatively correlated, but correlation increases with n.*

$$\begin{aligned} e_i & \overset{\text{IID}}{\sim} N\left(0, \sigma_e^2\right) \\ \left(\beta_0, \beta_1, \sigma_e^2\right) & \sim \text{diffuse} , \end{aligned}$$

where $x_i = 1$ is a binary indicator for T/C status.

DIC and LS results on these four models:

Model	$\overline{D(\theta)}$	$D(\bar{\theta})$	$\hat{p}_D$	DIC	LS
1 (Gaussian)	1749.6	1745.6	3.99	1753.5	−1.552
2 (Poisson, common λ)	1499.9	1498.8	1.02	1500.9	−1.316
3 (FEPR, different λs)	1495.4	1493.4	1.98	1497.4	−1.314
4 (REPR)	1275.7 1274.7 1274.4	1132.0 1131.3 1130.2	143.2 143.5 144.2	1418.3 1418.2 1418.6	−1.180

(3 REPR rows were based on different monitoring runs, all of length 10,000, to give idea of Monte Carlo noise level.)

As $\sigma_e \rightarrow 0$ in REPR model, you get FEPR model, with $p_D = 2$ parameters; as $\sigma_e \rightarrow \infty$, in effect all subjects in study have their own λ and p_D would be 572; in between at $\sigma_e \doteq 0.675$ (posterior mean), `WinBUGS` estimates that there are about 143 effective parameters in REPR model, but its deviance $D(\bar{\theta})$ is so much lower that it wins DIC contest hands down.

To use the DIC feature in `WinBUGS` to produce the screen shot above, I fit the REPR model as usual, did a burn-in of 1,000, selected `DIC` as a pull-down option from the `Inference` menu, clicked the `set` button in the `DIC Tool` window that popped up, changed the 1,000 to 10,000 in the `updates` window of the `Update Tool`, clicked `update`, and then clicked `DIC` in the `DIC Tool` when the monitoring run of 10,000 was finished—the `DIC` results window appears, with the `Dbar` ($\overline{D(\theta)}$), `Dhat` ($D(\bar{\theta})$), `pD` ($\hat{p}_D$), and `DIC` ($DIC(y)$) values.

NB You will usually want to base the estimates in DIC on a monitoring run that's considerably longer than 10,000 iterations (more like 100,000 would be good); this was just to illustrate the basic DIC mouse-click sequence in `WinBUGS`.

DIC can be sensitive to parameterization. $y = (0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5, 6)$ is a data set generated from the negative binomial distribution with parameters $(p, r) = (0.82, 10.8)$ (in `WinBUGS` notation); y has `mean 2.35` and `VTMR 1.22`.

Using standard diffuse priors for p and r as in the `BUGS` examples manuals,

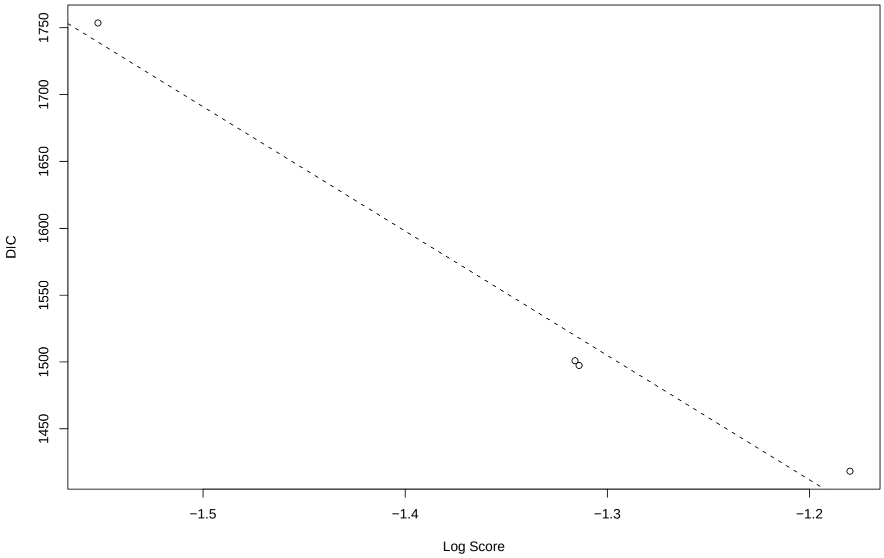


Figure 4.18: *The correlation between LS and DIC across these four models is –0.98.*

the effective number of parameters p_D for the negative binomial model (which fits the data quite well) is estimated at –66.2:

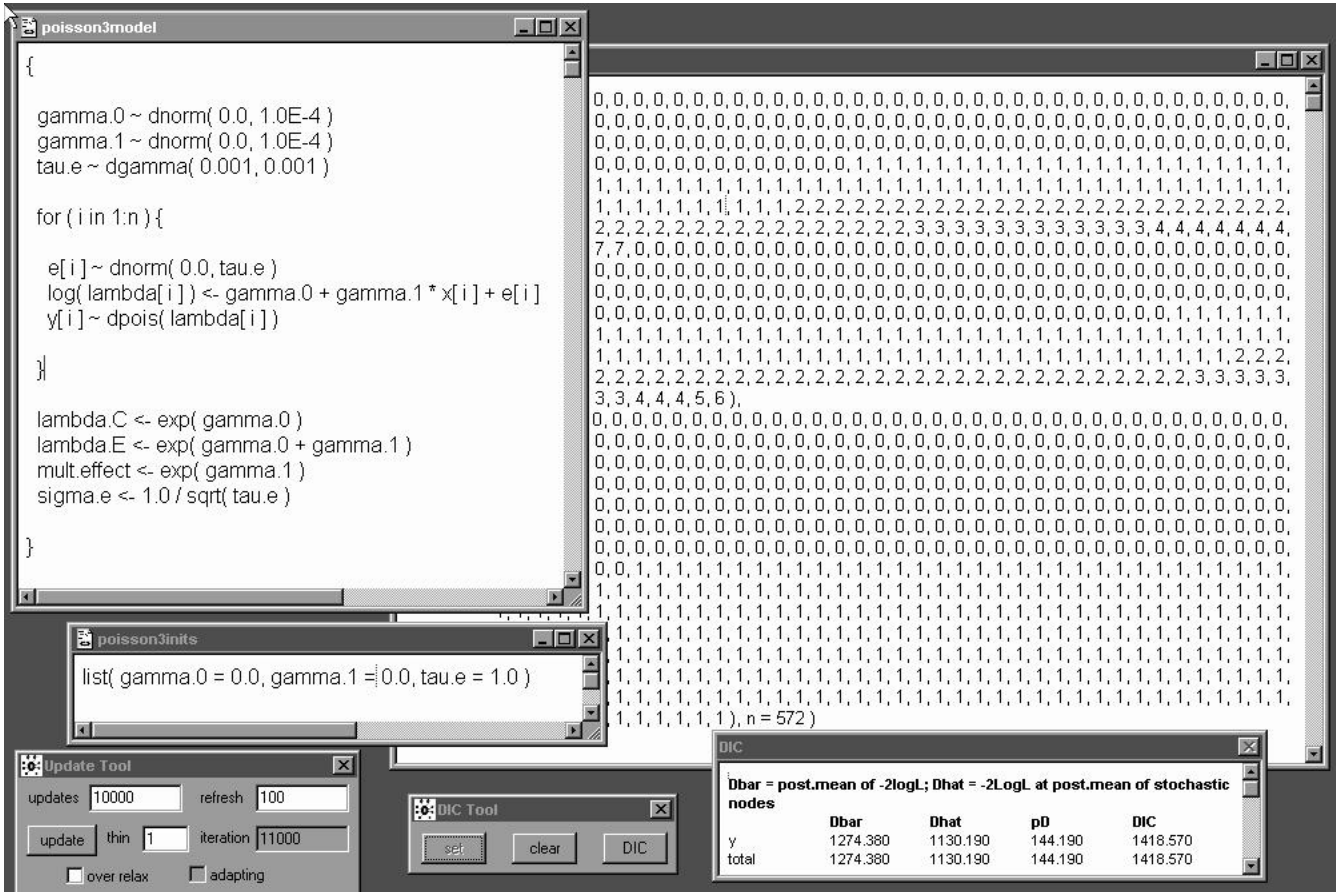
The basic problem here is that the MCMC estimate of p_D can be quite poor if the marginal posteriors for one or more of the parameters given prior distributions in the modeling are far from normal; reparameterization helps but can still lead to poor estimates of p_D.

Reparameterization and DIC. Here the marginal posteriors for both p and r were very heavily skewed; the idea in trying to improve the DIC estimate of the effective number of parameters is to find transformed versions of p and r whose posterior distributions are closer to normal.

Since p and r live on $(0, 1)$ and $(0, \infty)$, respectively, it's natural to try working with $\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$ and $\log(r)$.

You can see with these transformed parameters that the DIC estimate of p_D is about 1.1, which is a great improvement over –66.2 (the right answer is of course 2.0).

To be fair to DIC, it's clear from the plots above that we're trying to



```
{

 gamma.0 ~ dnorm( 0.0, 1.0E-4 )
 gamma.1 ~ dnorm( 0.0, 1.0E-4 )
 tau.e ~ dgamma( 0.001, 0.001 )

 for ( i in 1:n ) {

  e[ i ] ~ dnorm( 0.0, tau.e )
  log( lambda[ i ] ) <- gamma.0 + gamma.1 * x[ i ] + e[ i ]
  y[ i ] ~ dpois( lambda[ i ] )

 }

 lambda.C <- exp( gamma.0 )
 lambda.E <- exp( gamma.0 + gamma.1 )
 mult.effect <- exp( gamma.1 )
 sigma.e <- 1.0 / sqrt( tau.e )

}
```

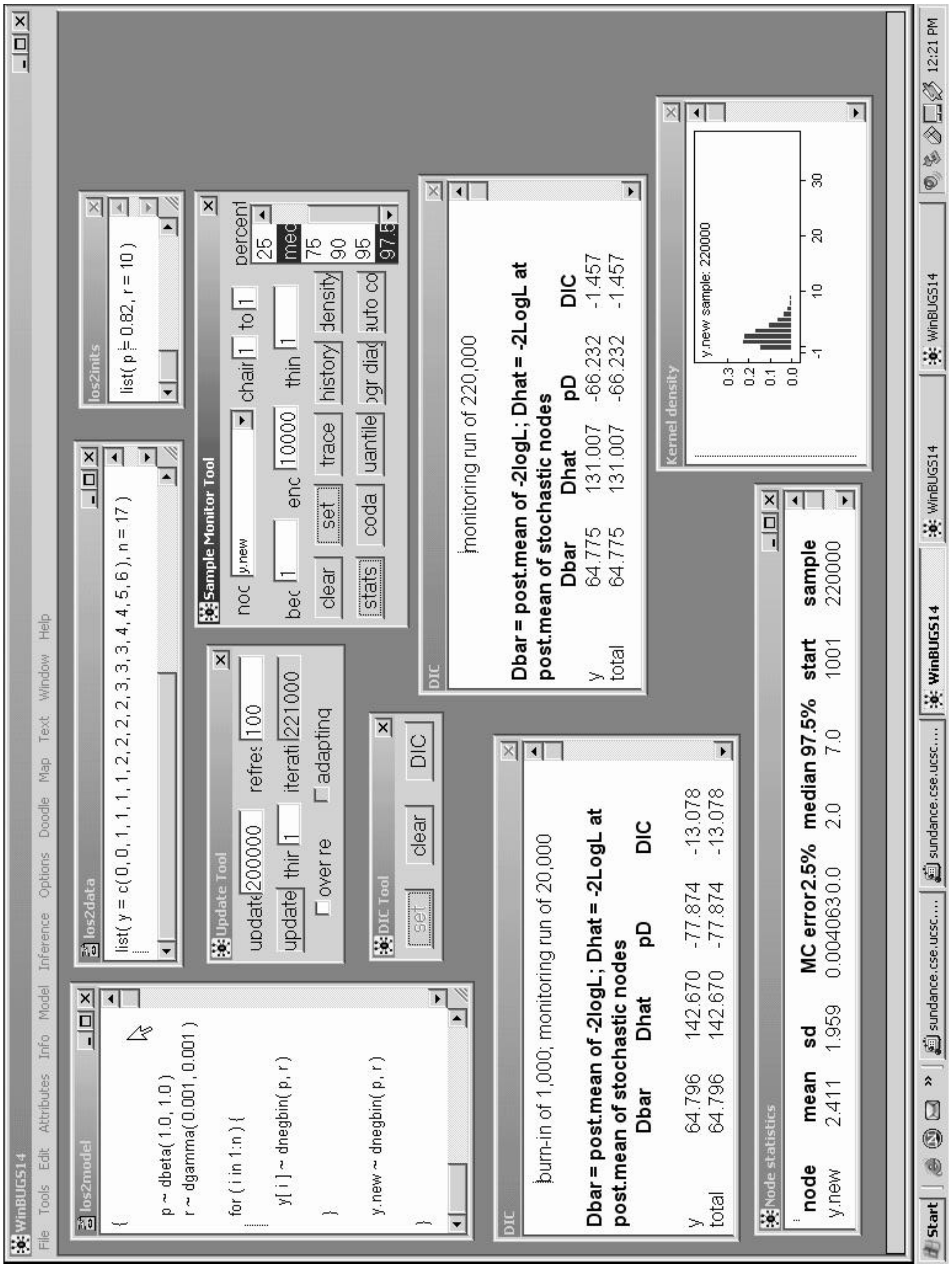
```
list( gamma.0 = 0.0, gamma.1 = 0.0, tau.e = 1.0 )
```

```
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4,
7, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3,
3, 3, 4, 4, 4, 5, 6 ),
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1 ), n = 572 )
```

updates 10000 refresh 100
thin 1 iteration 11000
☐ over relax ☐ adapting

```
Dbar = post.mean of -2logL; Dhat = -2LogL at post.mean of stochastic
nodes
        Dbar        Dhat        pD        DIC
y       1274.380    1130.190    144.190   1418.570
total   1274.380    1130.190    144.190   1418.570
```

Figure 4.19: *Using the DIC feature in* ***WinBUGS***.



```
{

  p ~ dbeta( 1.0, 1.0 )
  r ~ dgamma( 0.001, 0.001 )

  for ( i in 1:n ) {

    y[ i ] ~ dnegbin( p, r )

  }

  y.new ~ dnegbin( p, r )

}
```

```
list( y = c( 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5, 6 ), n = 17 )
```

```
list( p = 0.82, r = 10 )
```

Update Tool: updat 200000 refre 100; update thir 1 iterati 221000; ☐ over re ☐ adapting

Sample Monitor Tool: noc y.new; chair 1 to 1; bec 1 enc 10000 thin 1; percent: 25, mec, 75, 90, 95, 97.5; clear, set, trace, history, density, stats, coda, uantile, bgr diag, auto co

DIC Tool: set, clear, DIC

```
monitoring run of 220,000

Dbar = post.mean of -2logL; Dhat = -2LogL at
post.mean of stochastic nodes
          Dbar      Dhat      pD        DIC
y         64.775    131.007   -66.232   -1.457
total     64.775    131.007   -66.232   -1.457
```

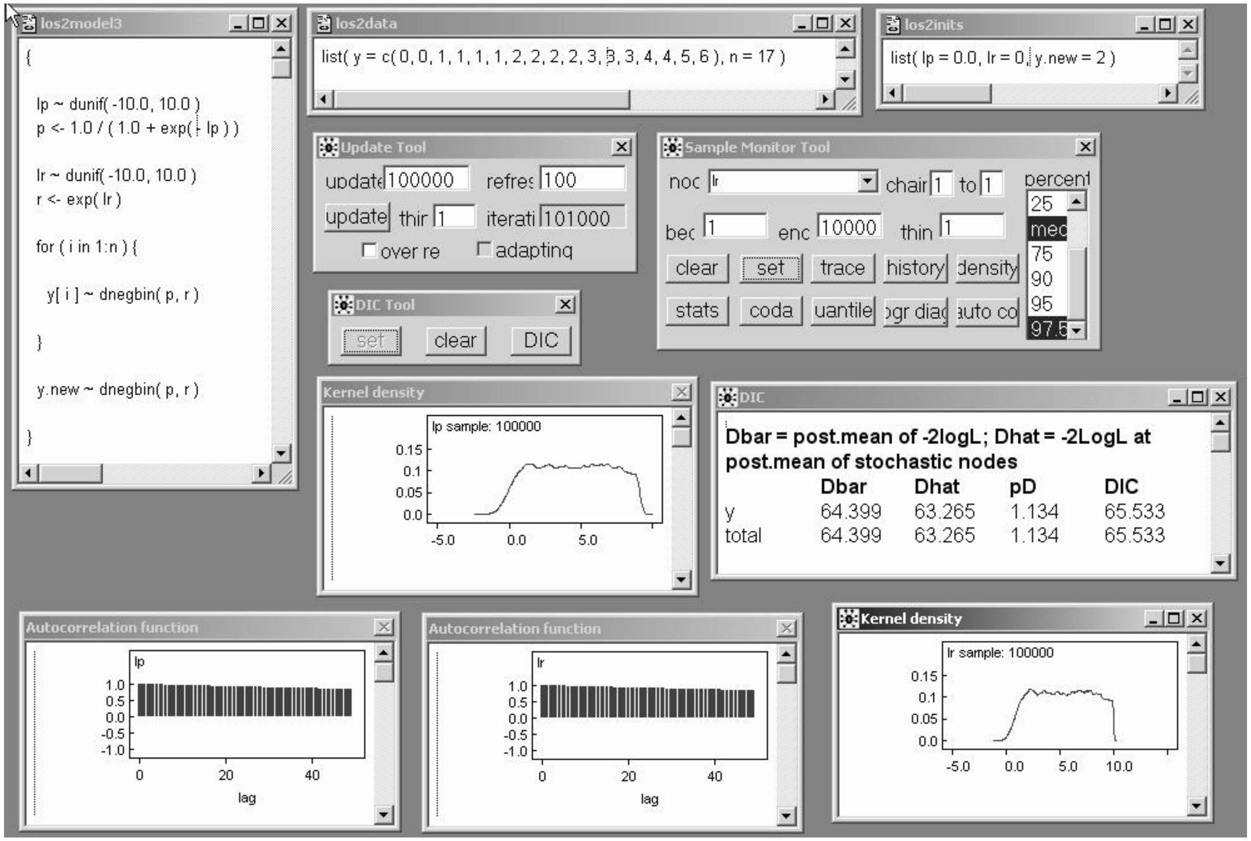
```
burn-in of 1,000; monitoring run of 20,000

Dbar = post.mean of -2logL; Dhat = -2LogL at
post.mean of stochastic nodes
          Dbar      Dhat      pD        DIC

y         64.796    142.670   -77.874   -13.078
total     64.796    142.670   -77.874   -13.078
```

```
node     mean    sd      MC error2.5%   median 97.5%   start   sample
y.new    2.411   1.959   0.00406300.0   2.0    7.0     1001    220000
```

Figure 4.20: *Sometimes close attention must be paid to parameterization when using DIC.*



los2model3

```
{

  lp ~ dunif( -10.0, 10.0 )
  p <- 1.0 / ( 1.0 + exp( - lp ) )

  lr ~ dunif( -10.0, 10.0 )
  r <- exp( lr )

  for ( i in 1:n ) {

    y[ i ] ~ dnegbin( p, r )

  }

  y.new ~ dnegbin( p, r )

}
```

los2data

```
list( y = c( 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5, 6 ), n = 17 )
```

los2inits

```
list( lp = 0.0, lr = 0, y.new = 2 )
```

Update Tool: updates 100000, refresh 100, thin 1, iteration 101000, ☐ over relax, ☐ adapting

Sample Monitor Tool: node lr, chains 1 to 1, beg 1, end 10000, thin 1, percentiles 25, median, 75, 90, 95, 97.5

DIC

```
Dbar = post.mean of -2logL; Dhat = -2LogL at
post.mean of stochastic nodes
           Dbar     Dhat     pD      DIC
y          64.399   63.265   1.134   65.533
total      64.399   63.265   1.134   65.533
```

Figure 4.21: *With transformed parameters the DIC estimate of model complexity is greatly improved.*

ask it to evaluate a model in which there is very little likelihood information about the parameters.

LS model discrimination. On-going work: DIC calculated with single MCMC run; in one-sample setting with n observations and no closed form for predictive distribution, brute-force LS requires n parallel MCMC runs; how exactly is DIC approximating LS with far less computation?

Tentative conclusions: With large n (when LS can be expensive), DIC may provide acceptable approximation to model choice based on LS, but for small and moderate n direct calculation of LS may be safer.

With LS utility function and two models M_1 and M_2 to choose between, MEU says compute

$$\hat{E}\left[U(M_i|y)\right] = \frac{1}{n}\sum_{j=1}^{n} \log p(y_j|M_i, y_{-j}) \tag{4.35}$$

and choose M_1 if $\hat{E}\left[U(M_1|y)\right] \geq \hat{E}\left[U(M_2|y)\right]$, otherwise M_2.

How accurately does this behavioral rule discriminate between M_1 and M_2?

Example: Recall that in earlier simulation study, for $i = 1, \ldots, n$, and with diffuse priors, we considered

$$M_1\text{:} \left\{ \begin{array}{ccc} \lambda & \sim & p(\lambda) \\ (y_i|\lambda) & \overset{\text{IID}}{\sim} & \text{Poisson}(\lambda) \end{array} \right\} \quad \text{versus}$$

$$M_2\text{:} \left\{ \begin{array}{ccc} (\beta_0, \sigma^2) & \sim & p(\beta_0, \sigma^2) \\ (y_i|\lambda_i) & \overset{\text{indep}}{\sim} & \text{Poisson}(\lambda_i) \\ \log(\lambda_i) & = & \beta_0 + e_i \\ e_i & \overset{\text{IID}}{\sim} & N(0, \sigma^2) \end{array} \right\}$$

As extension of previous simulation study, we generated data from M_2 and computed LS for models M_1 and M_2 in full-factorial grid $\{n = 32, 42, 56, 100\}$, $\{\beta_0 = 0.0, 1.0\}$, $\sigma^2 = 0.1, 0.25, 0.5, 1.0, 1.5, 2.0\}$, with 100 simulation replications in each cell, and monitored percentages of correct model choice (here M_2 is always correct).

Examples of results:

$n = 32$

	% Correct Decision			Mean Absolute Difference in LS	
	β_0			β_0	
σ^2	0	1	σ^2	0	1
0.10	31	47	0.10	0.001	0.002
0.25	49	85	0.25	0.002	0.013
0.50	76	95	0.50	0.017	0.221
1.00	97	100	1.00	0.237	4.07
1.50	98	100	1.50	1.44	17.4
2.00	100	100	2.00	12.8	63.9

Even with n only 32, the right model choice is made more than 90% of the time when $\sigma^2 > 0.5$ for $\beta_0 = 1$ and when $\sigma^2 > 1.0$ for $\beta_0 = 0$.

Graphical summary of these simulations (Bayesian decision-theoretic power curves):

NB Food for thought: In previous table mean absolute LS difference between M_1 and M_2 can be tiny even when this approach to model choice is performing well.

E.g., with $(n, \sigma^2, \beta_0) = (32, 0.25, 1)$, M_2 is correctly identified 85% of the time, but its typical LS edge over M_1 is only about 0.013—is this difference large in either practical or statistical terms?

4.2.4 Connections with Bayes factors

Much has been written about use of Bayes factors for model choice (e.g., Jeffreys 1939, Kass and Raftery 1995; excellent recent book by O'Hagan and Forster 2004 devotes almost 40 pages to this topic).

After all, why not use probability scale to choose between M_1 and M_2?

$$\begin{aligned} \left[\frac{p(M_1|y)}{p(M_2|y)} \right] &= \left[\frac{p(M_1)}{p(M_2)} \right] \cdot \left[\frac{p(y|M_1)}{p(y|M_2)} \right] \\ \left(\begin{array}{c} \text{posterior} \\ \text{odds} \end{array} \right) &= \left(\begin{array}{c} \text{prior} \\ \text{odds} \end{array} \right) \cdot \left(\begin{array}{c} \text{Bayes} \\ \text{factor} \end{array} \right) \end{aligned} \tag{4.36}$$

In fact, Kass and Raftery (1995) note that

$$\begin{aligned} \log \left[\frac{p(y|M_1)}{p(y|M_2)} \right] &= \log p(y|M_1) - \log p(y|M_2) \\ &= LS^*(M_1|y) - LS^*(M_2|y), \end{aligned} \tag{4.37}$$

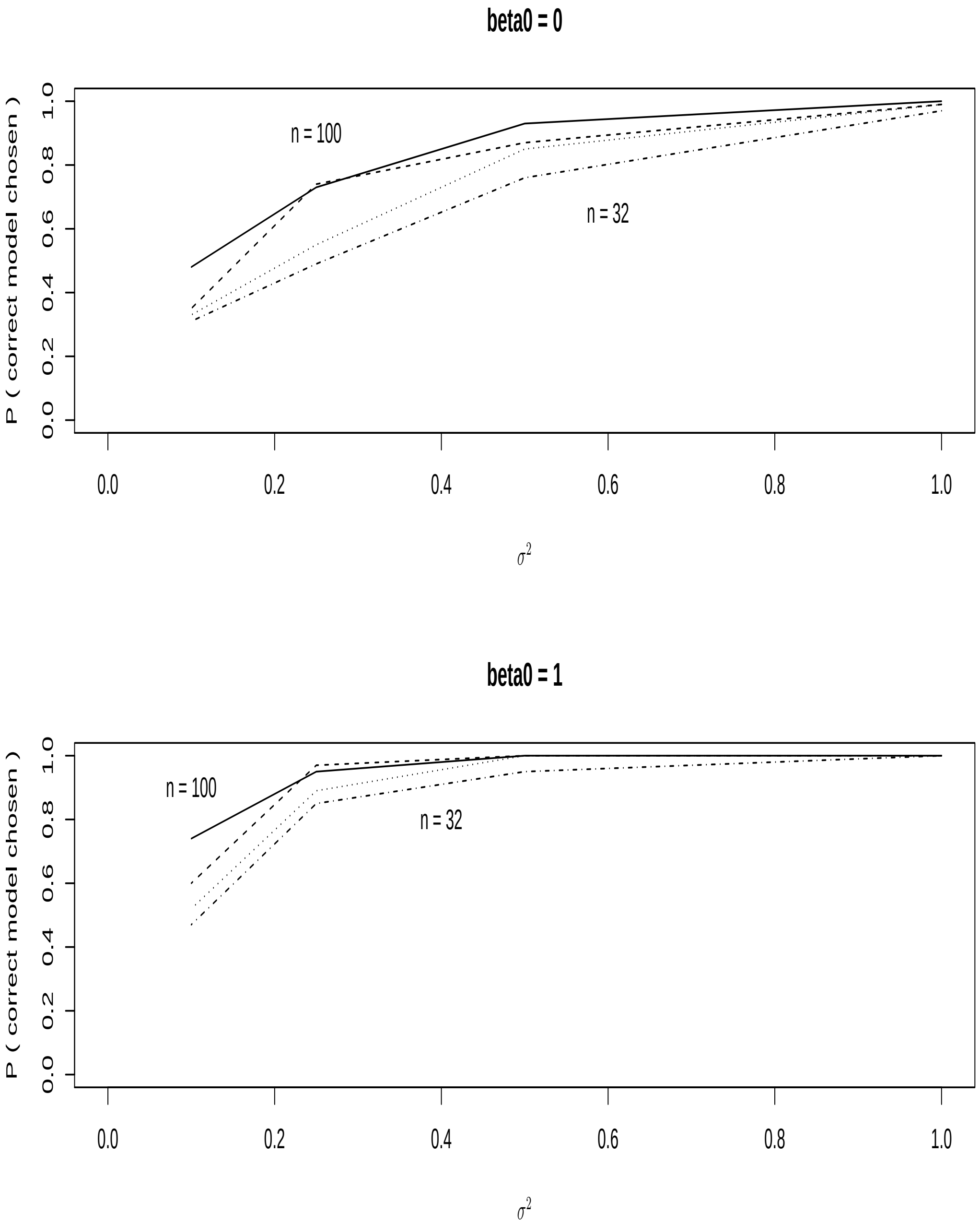


Figure 4.22: *Bayesian decision-theoretic power curves for LS.*

where

$$\begin{aligned} LS^*(M_i|y) &\equiv \log p(y|M_i) \\ &= \log \left[p(y_1|M_i)\, p(y_2|y_1, M_i) \cdots p(y_n|y_1, \ldots, y_{n-1}, M_i) \right] \\ &= \log p(y_1|M) + \sum_{j=2}^{n} \log p(y_j|y_1, \ldots, y_{j-1}, M_i). \end{aligned}$$

Thus log Bayes factor equals difference between models in something that looks like previous log score, i.e., isn't previous procedure equivalent to choosing M_i whenever the Bayes factor in favor of M_i exceeds 1?

Out-of-sample LS $\neq$ BF. No; crucially, LS* is defined via sequential prediction of y_2 from y_1, y_3 from (y_1, y_2), etc., whereas LS is based on averaging over all possible out-of-sample predictions:

$$nLS(M_i|y) = \sum_{j=1}^{n} \log p(y_j|M_i, y_{-j}).$$

This distinction really matters: as is well known, with diffuse priors Bayes factors are hideously sensitive to particular form in which diffuseness is specified, but this defect is entirely absent from LS (and from other properly-defined utility-based model choice criteria).

(Various attempts have been made to fix this defect of Bayes factors, e.g., {partial, intrinsic, fractional} Bayes factors, well calibrated priors, conventional priors, intrinsic priors, expected posterior priors, ... (e.g., Pericchi 2004); all of these methods appear to require an appeal to ad-hockery which is not required by LS.)

Example: Integer-valued data $y = (y_1, \ldots, y_n)$;
M_1 = Geometric(θ_1) likelihood with Beta(α_1, β_1) prior on θ_1;
M_2 = Poisson(θ_2) likelihood with Gamma(α_2, β_2) prior on θ_2.
Bayes factor in favor of M_1 over M_2 is

$$\frac{\Gamma(\alpha_1+\beta_1)\Gamma(n+\alpha_1)\Gamma(n\bar{y}+\beta_1)\Gamma(\alpha_2)(n+\beta_2)^{n\bar{y}+\alpha_2}\left(\prod_{i=1}^n y_i!\right)}{\Gamma(\alpha_1)\Gamma(\beta_1)\Gamma(n+n\bar{y}+\alpha_1+\beta_1)\Gamma(n\bar{y}+\alpha_2)\beta_2^{\alpha_2}}.$$

Diffuse priors: take $(\alpha_1, \beta_1) = (1, 1)$ and $(\alpha_2, \beta_2) = (\epsilon, \epsilon)$ for some $\epsilon > 0$.
Bayes factor reduces to

$$\frac{\Gamma(n+1)\Gamma(n\bar{y}+1)\Gamma(\epsilon)(n+\epsilon)^{n\bar{y}+\epsilon}\left(\prod_{i=1}^n y_i!\right)}{\Gamma(n+n\bar{y}+2)\Gamma(n\bar{y}+\epsilon)\epsilon^{\epsilon}}.$$

This goes to $+\infty$ as $\epsilon \downarrow 0$, i.e., you can make the evidence in favor of the Geometric model over the Poisson as large as you want as a function of a quantity near 0 that scientifically you have no basis to specify.

By contrast

$$
\begin{aligned}
LS(M_1|y) &= \log\left[\frac{(\alpha_1+n-1)\Gamma(\beta_1+s)}{\Gamma(\alpha_1+n+\beta_1+s)}\right] \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{\Gamma(\alpha_1+n-1+\beta_1+s_i)}{\Gamma(\beta_1+s_i)}\right]
\end{aligned}
$$

and

$$
\begin{aligned}
LS(M_2|y) &= \frac{1}{n}\sum_{i=1}^{n}\log\left[\frac{\Gamma(\alpha_2+s)}{\Gamma(y_i+1)\Gamma(\alpha_2+s_i)}\right. \\
&\quad \left.\cdot\left(\frac{\beta_2+n}{\beta_2+n+1}\right)^{\alpha_2+s_i}\left(\frac{1}{\beta_2+n+1}\right)^{y_i}\right]
\end{aligned}
$$

and both of these quantities are entirely stable as a function of (α_1, β_1) and (α_2, β_2) near zero.

4.2.5 When is a model good enough?

LS method described here (not LS* method) can stably and reliably help in choosing between M_1 and M_2; but suppose M_2 has a (substantially) higher LS than M_1.

This doesn't say that M_2 is adequate—it just says that M_2 is better than M_1.

As mentioned above, a full judgment of adequacy requires real-world input (to what purpose will the model be put?), but you can answer a somewhat related question—could the data have arisen from a given model?—by simulating from that model many times, developing a distribution of (e.g.) LS values, and seeing how unusual the actual data set's log score is in this distribution (Draper and Krnjajić 2004).

This is related to the posterior predictive model-checking method of Gelman, Meng and Stern (1996).

However, this sort of thing cannot be done naively, or result will be poor calibration—indeed, Robins et al. (2000) demonstrated that the Gelman et al. procedure may be (sharply) conservative.

Using modification of idea in Robins et al., we have developed method for accurately calibrating the log score scale.

Inputs to our procedure: (1) A data set (e.g., with regression structure); (2) A model (can be parametric, non-parametric, or semi-parametric).

Simple example: data set $y = (1, 2, 2, 3, 3, 3, 4, 6, 7, 11)$, $n = 10$.

Given model $(*)$

$$\begin{aligned} (\lambda) &\sim \text{Gamma}(0.001, 0.001) \\ (y_i|\lambda) &\overset{\text{IID}}{\sim} \text{Poisson}(\lambda) \end{aligned} \tag{4.38}$$

Calibrating the LS scale.

Step 1:

Calculate log score for this data set; say get LS $= -1.1$; call this actual log score (ALS).

Obtain posterior for λ given y based on this data set; call this actual posterior.

Step 2:

```
for ( i in 1:m1 ) {

  make a lambda draw from the actual posterior;
    call it lambda[ i ]

  generate a data set of size n from the second
    line of model (*) above, using
    lambda = lambda[ i ]

  compute the log score for this generated
    data set; call it LS[ i ]

}
```

Output of this loop is a vector of log scores; call this V.LS.

Locate ALS in distribution of LS values by computing percentage of LS values in V.LS that are $\leq$ ALS; call this percentage unadjusted actual tail area (say this is 0.22).

So far this is just Gelman et al. with LS as the discrepancy function.

We know from our own simulations and the literature (Robins et al. 2000) that this tail area (a p-value for a composite null hypothesis, e.g., Poisson(λ)

with λ unspecified) is conservative, i.e., with the 0.22 example above an adjusted version of it that is well calibrated would be smaller.

We've modified and implemented one of the ways suggested by Robins et al., and we've shown that it does indeed work even in rather small-sample situations, although our approach to implementing the basic idea can be computationally intensive.

Step 3:

```
for ( j in 1:m2 ){

   make a lambda draw from the actual posterior;
     call it lambda*.

   generate a data set of size n from the second line
     of model (*) above, using lambda = lambda*;
     call this the simulated data set

   repeat steps 1, 2 above on this
     simulated data set

}
```

The result will be a vector of unadjusted tail areas; call this V.P.

Compute the percentage of tail areas in V.P that are $\leq$ the unadjusted actual tail area; this is the adjusted actual tail area.

The claim is that the 3-step procedure above is well-calibrated, i.e., if the sampling part of model $(*)$ really did generate the observed data, the distribution of adjusted actual tail areas obtained in this way would be uniform, apart from simulation noise.

Step 3 in this procedure solves the calibration problem by applying the old idea that if $X \sim F_X$ then $F_X(X) \sim U(0,1)$.

This claim can be verified by building a big loop around steps 1–3 as follows:

```
Choose a lambda value of interest; call it lambda.sim

for ( k in 1:m3 ) {

   generate a data set of size n from the
```

```
      second line of model (*) above, using
      lambda = lambda.sim; call this the
      validation data set

    repeat steps 1-3 on the validation data set

  }
```

The result will be a vector of adjusted P-values; call this V.Pa.

We have verified (via simulation) in several simple (and some less simple) situations that the values in V.Pa are close to $U(0, 1)$ in distribution.

Two examples—Poisson(λ) and Gaussian(μ, σ^2):

Conclusions

• {Exchangeability judgments plus nonparametric (BNP) modeling} = Bayesian model specification in many problems, but we need to develop a lot more experience in practical problems in putting distributions on functions.

• BNP/BSP is one way to avoid the dilemma posed by Cromwell's Rule in Bayesian model specification; three-way cross-validation (3CV) is another.

• Model choice is really a decision problem and should be approached via MEU, with a utility structure that's sensitive to the real-world context.

• The leave-one-out predictive log score (LS) has a sound utility basis, and can yield stable and accurate model specification decisions.

• DIC is a good and fast approximation to LS in some models, but sometimes it misbehaves.

• Bayes factors are a bad choice for model specification when context suggests diffuse priors.

• The basic Gelman et al. (1996) method of posterior predictive model-checking can be badly calibrated: when it gives you a tail area of, e.g., 0.4, the calibrated equivalent may well be 0.04.

• We have modified an approach suggested by Robins et al. (2000) to help answer the question "Could the data have arisen from model M?" in a well-calibrated way.

4.3 Problems

1. (Bayesian analysis of proportions (based on problems 3.8 and 5.11 from Gelman et al.) In the spring of 1993 a survey was taken of bicycle and other vehicular traffic in the neighborhood of the campus of the

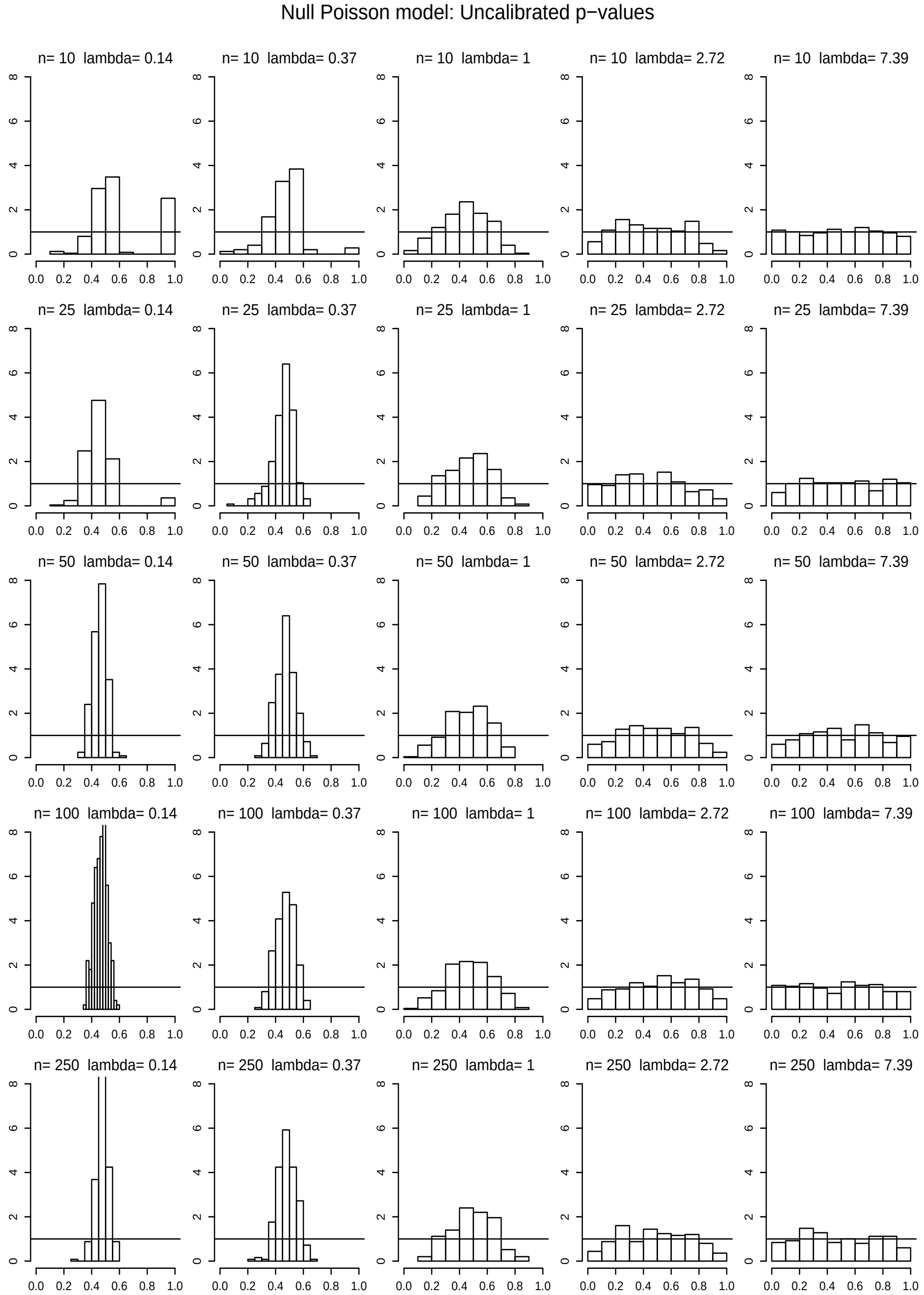


Figure 4.23: *Null Poisson model: uncalibrated p–values.*

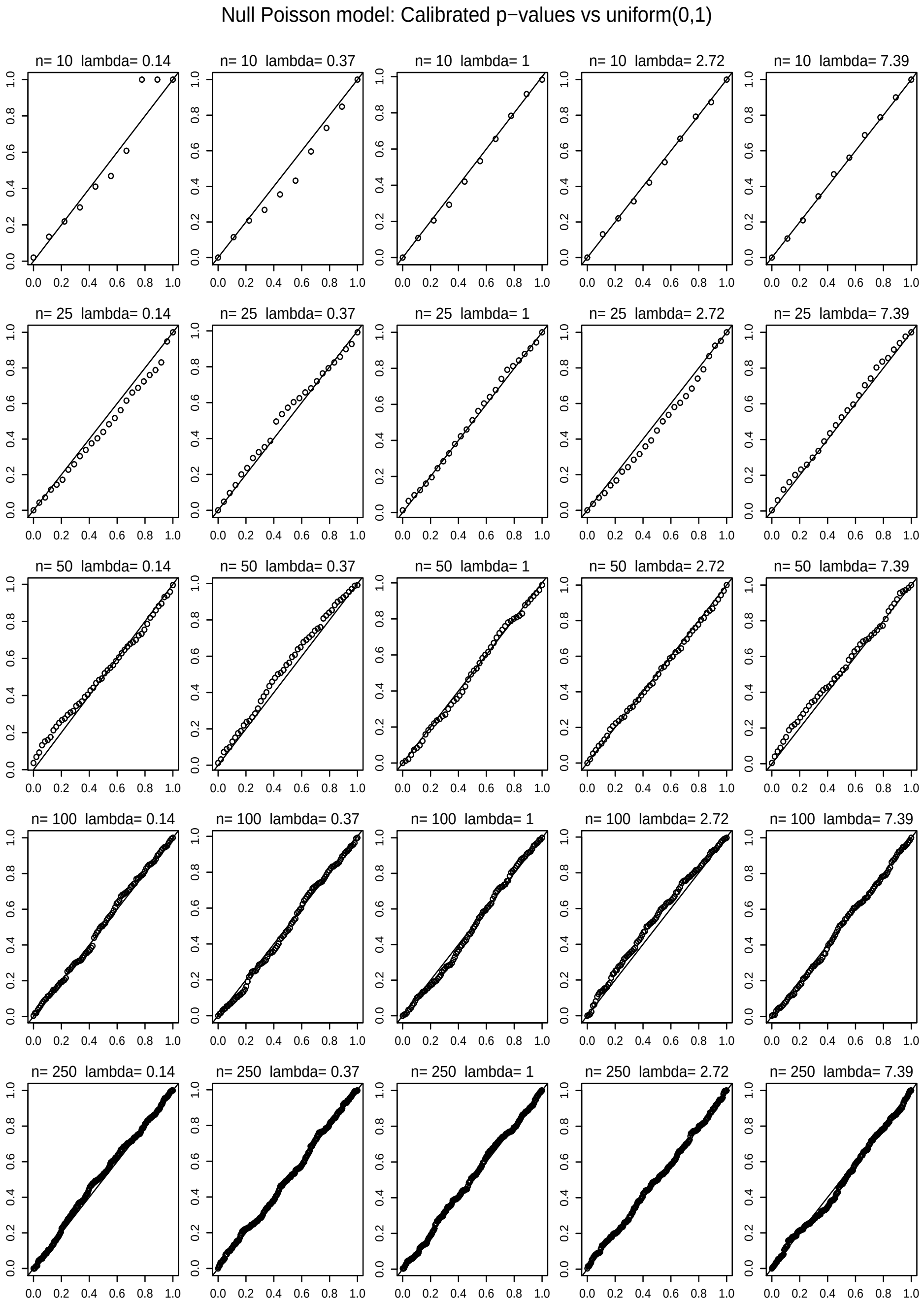


Figure 4.24: *Null Poisson model: calibrated p–values.*

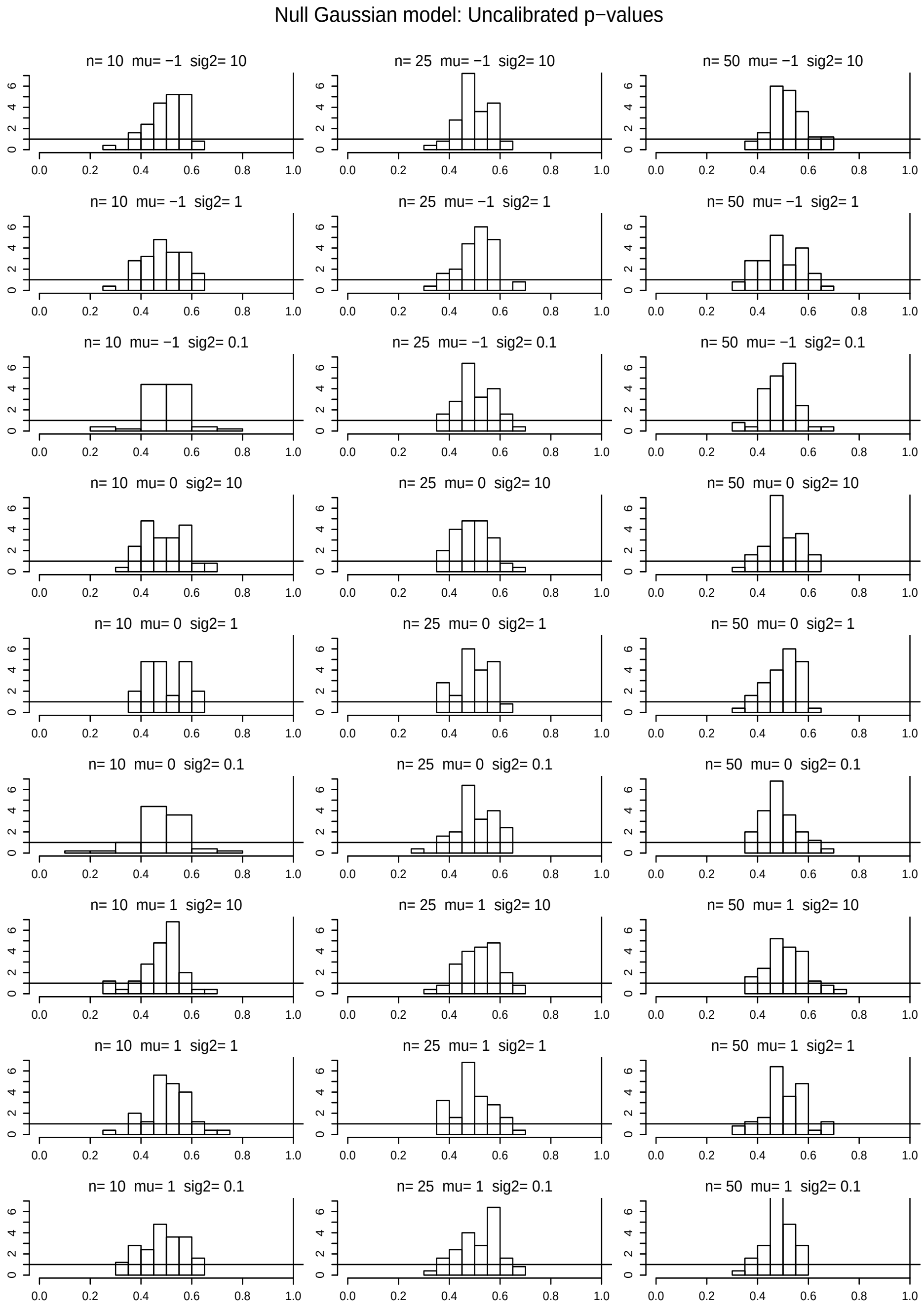


Figure 4.25: *Null Gaussian model: uncalibrated p–values.*

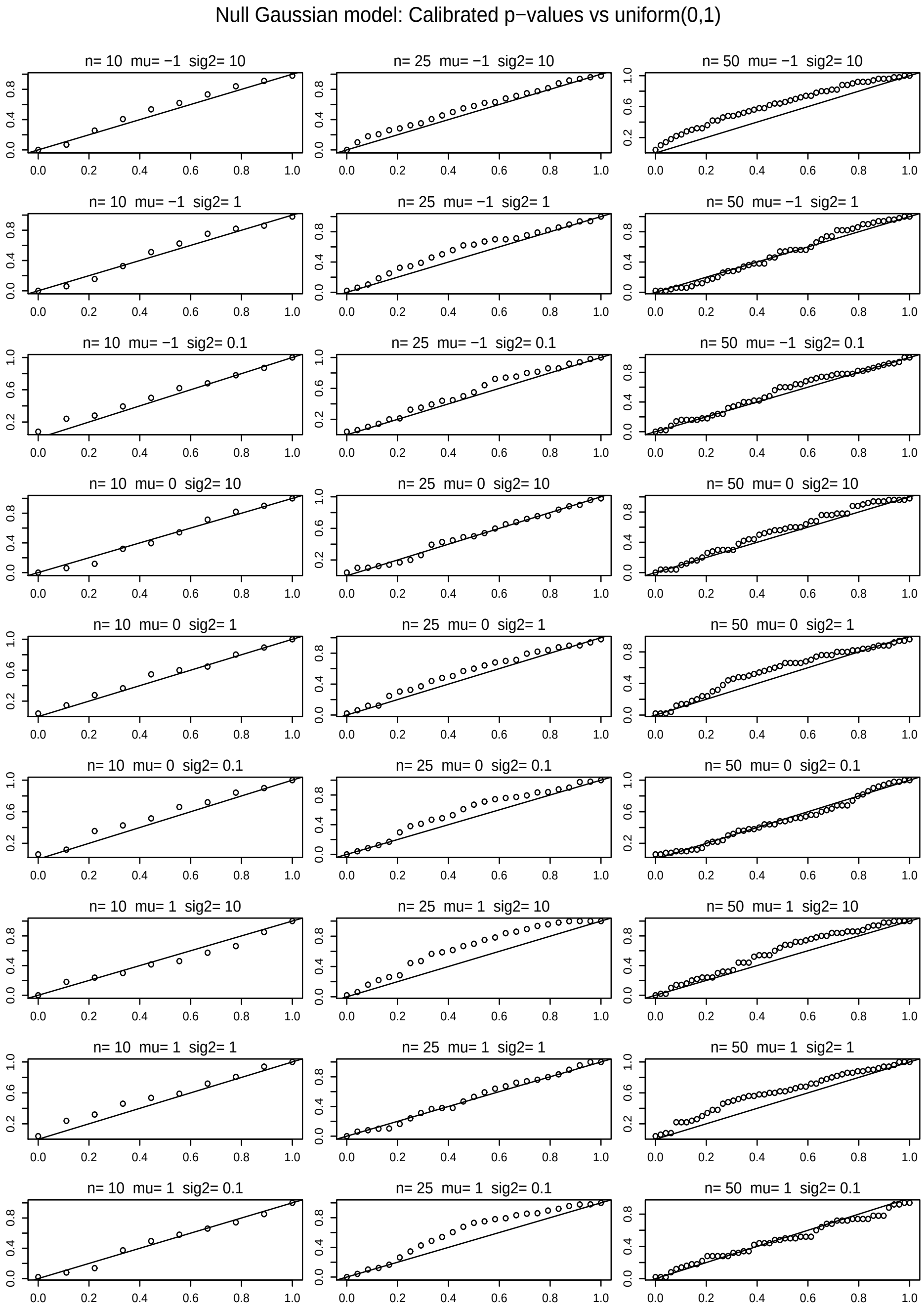
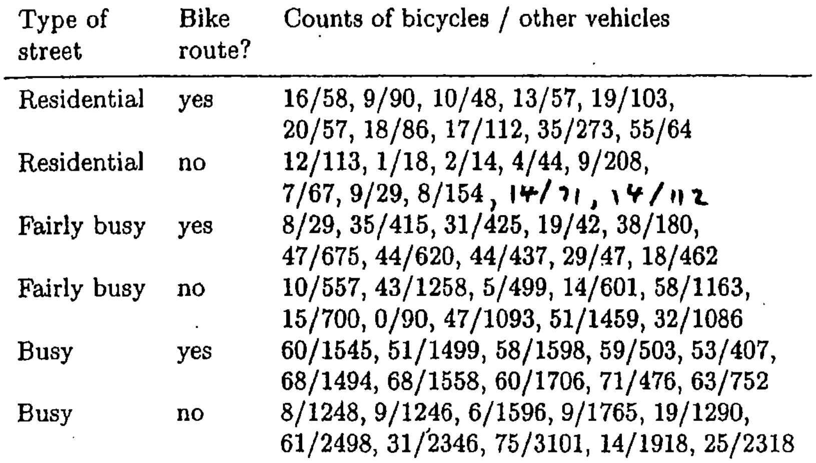


Figure 4.26: *Null Gaussian model: calibrated p–values.*

Table 4.1: *Counts of bicycles and other vehicles in one hour in each of 10 city blocks in each of six categories in the Berkeley traffic study.*



Type of street	Bike route?	Counts of bicycles / other vehicles
Residential	yes	16/58, 9/90, 10/48, 13/57, 19/103, 20/57, 18/86, 17/112, 35/273, 55/64
Residential	no	12/113, 1/18, 2/14, 4/44, 9/208, 7/67, 9/29, 8/154, 14/71, 14/112
Fairly busy	yes	8/29, 35/415, 31/425, 19/42, 38/180, 47/675, 44/620, 44/437, 29/47, 18/462
Fairly busy	no	10/557, 43/1258, 5/499, 14/601, 58/1163, 15/700, 0/90, 47/1093, 51/1459, 32/1086
Busy	yes	60/1545, 51/1499, 58/1598, 59/503, 53/407, 68/1494, 68/1558, 60/1706, 71/476, 63/752
Busy	no	8/1248, 9/1246, 6/1596, 9/1765, 19/1290, 61/2498, 31/2346, 75/3101, 14/1918, 25/2318

University of California, Berkeley. Ten city blocks were selected at random in each of the six categories of a 2×3 table that cross-tabulates {presence or absence of a bike route on a street} against whether the street was {residential, fairly busy, or busy}. Each block was observed for one hour at the same time and day of the week on a randomly chosen day, and the numbers of bicycles and other vehicles traveling along that block were recorded. The data are given in Table 1 below from Gelman et al., except that I have imputed two additional observations (written in pen) in the (residential, no bike route) category to make up for two blocks whose data were missing. The entry 16/58 at the beginning of the table means (for example) that a total of $16 + 58 = 74$ vehicles were observed in the block in question, of which 16 were bicycles. An electronic copy of the dataset is available on the course website `www.soe.ucsc.edu/classes/ams206/Winter05/`.

(a) Convert all 60 observations into proportions of traffic taken up by bicycles (for example, the 16/58 entry above becomes $\frac{16}{16+58} \doteq 0.2162$). Make a 2×3 table with rows for (presence or absence of a bike route on the street) and columns for (residential, fairly

Table 4.2: *Summaries of the proportions of vehicular traffic taken up by bicycles.*

```
     Means, Standard Deviations and Frequencies of PBT

           |             Street Type            |
    Bike   |                                    |
    Route? |Residential  Fairly Busy    Busy    |      Total
-----------+------------------------------------+----------
       Yes | .19614125                          | .13676775
           | .10545459                          | .10636461
           |        10           10         10 |        30
-----------+------------------------------------+----------
        No |                          .01152844 |
           |                          .00727029 |
           |        10           10         10 |        30
-----------+------------------------------------+----------
     Total |              .08780747             | .09233102
           |              .10339536             | .09489362
           |        20           20         20 |        60
```

busy, or busy); as entries in this table put the mean, standard deviation (SD), and sample size of the proportions in each cell of the 2 × 3 grid, and complete the table by computing {row and column means, SDs, and sample sizes and the overall mean, SD, and sample size} and adding these values as row, column, and overall margins to the table. I've given you a headstart by filling in some of what I'm asking for below in Table 2.

Study your completed version of Table 2. What kind of street has the highest proportion of bicycle traffic (PBT) on average? the lowest? Summarize the effect of the presence or absence of a bike route on PBT, and the effect of street type. An *interaction* between the bike route and street type factors is said to be present if the effect of bike route is different for different street types, and vice versa; does this seem to be true here? Do the differences you observe in this table between mean PBT values across the different kinds of streets seem large to you in *practical* terms? Explain briefly.

(b) Create a similar summary for the total number of vehicles. Do the "fairly busy" streets indeed have more traffic than the "residential" blocks, and are the "busy" streets on average even busier than the "fairly busy" ones? Is there a big difference in total traffic on average between streets with and without a bike route? Explain briefly.

(c) For the remainder of the problem let's focus on models for PBT, which will help to settle whether the differences in Table 2 between mean PBT values are large in *statistical* terms. Working with the data in Table 2 in all of its glory involves the *analysis of variance*, which we've not had time to cover (you'll hear about it in AMS 207 if you take it), so we'll make do with a piece of machinery we looked at in the IHGA case study: the comparison of two independent samples. Consider first a comparison of the (residential, bike route) and (residential, no bike route) samples, and let's agree to call the estimated PBT values in these samples $(y_1, \ldots, y_n)$ and $(z_1, \ldots, z_n)$, respectively (here $n = 10$, and [for example] $y_1 = \frac{16}{74} \doteq 0.2162$). As is often the case, it's constructive in getting warmed up to start with Gaussian modeling, even though it can obviously be criticized here (and below I ask you to list some criticisms). Use `WinBUGS` to fit the following model to the PBT data:

$$\begin{aligned} (y_i | \mu_y, \sigma_y^2) & \stackrel{\text{IID}}{\sim} N\big(\mu_y, \sigma_y^2\big) \\ (z_j | \mu_z, \sigma_z^2) & \stackrel{\text{IID}}{\sim} N\big(\mu_z, \sigma_z^2\big) \\ (\mu_y, \sigma_y^2, \mu_z, \sigma_z^2) & \sim \text{diffuse}. \end{aligned} \tag{4.39}$$

Summarize your inferences about the *additive effect* of a bike route on PBT for residential streets in Berkeley in the early 1990s, in the form of an approximate or exact posterior mean, SD, and 95% central interval for $(\mu_y - \mu_z)$. According to this model, is the mean difference addressed by these calculations statistically meaningful (different from 0)? Having drawn this conclusion, give several reasons why the Gaussian model (1) above is not entirely appropriate for these data. Explain briefly. Calculate the DIC value for this model based on a monitoring run of appropriate length, and save it for later use. Does DIC seem to have estimated the number of parameters in the model correctly? Explain briefly.

(d) Looking at $(\mu_y - \mu_z)$ focuses on the additive effect of the bike route, but a model that incorporates a *multiplicative* effect might be more reasonable. Because the sample sizes in the y and z estimated PBT samples are the same, a quantile-quantile plot (qqplot) can be made of the two data sets just by plotting the sorted z values against the sorted y values (e.g., with the sorted y values on the horizontal scale). Use `R` or some other good graphing environment to make a qqplot for the data in (c), and superimpose three lines on it: (i) the line $z = y$, which represents no effect of the bike route; (ii) the line with slope 1 and intercept $(\bar{z} - \bar{y})$ (where $\bar{z}$ as usual is the mean of the z values), which represents the best additive fit; and (iii) the line with intercept 0 and slope $\frac{\bar{z}}{\bar{y}}$, which represents (a decent estimate of) the best multiplicative fit. Is there evidence of an effect of any kind in this plot? Explain why the qqplot supports a multiplicative fit, at least for the data in (c), over an additive fit.

(e) Give several reasons justifying the view that a more appropriate model for these data would be to take the y_i and z_j as conditionally IID draws from *Beta* distributions rather than Gaussians. The improved model I have in mind is

$$\begin{aligned} (y_i|\alpha_y, \beta_y) &\overset{\text{IID}}{\sim} \text{Beta}(\alpha_y, \beta_y) \\ (z_j|\alpha_z, \beta_z) &\overset{\text{IID}}{\sim} \text{Beta}(\alpha_z, \beta_z) \\ (\alpha_y, \beta_y, \alpha_z, \beta_z) &\sim \text{diffuse}. \end{aligned} \tag{4.40}$$

(Notice that this is a different use of the Beta distribution than (for example) in the AMI mortality case study in class: here it's being used as the basis of the likelihood function rather than as a prior.)

Use `WinBUGS` to fit model (2) to the data in (c). For diffuse priors on the Beta hyperparameters you can use uniform $U(0, c)$ distributions, with c chosen just barely large enough so that no truncation of the likelihood occurs (you will need to use a different c for each hyperparameter). You can get some idea of how big the c values should be by using a crude but often roughly effective approach called the *method of moments* (MoM). Recalling that the Beta(α, β) distribution has

$$\text{mean} \quad \frac{\alpha}{\alpha+\beta} \quad \text{and variance} \quad \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}, \tag{4.41}$$

the idea behind the method of moments is to equate the sample mean and variance to the theoretical values in (3) and solve for α and β (either with `Maple` or by hand) to get rough estimates of these quantities. Use this approach on the y data from (c), for which the sample mean and variance are 0.1961 and 0.01112, respectively, to show that the MoM estimates based on y are $\left(\hat{\alpha}_y, \hat{\beta}_y\right) \doteq (2.58, 10.6)$, and repeat this calculation on the z data from (c). The resulting MoM estimates will give you an idea of the order of magnitude of the c values for α and β for each of the y and z samples in your fitting of model (2) via `BUGS` with $U(0, c)$ priors, but even with this hint some experimentation is needed to prevent truncation of the likelihood. Show, for instance, that with $(c_{\alpha_y}, c_{\beta_y}, c_{\alpha_z}, c_{\beta_z}) = (10, 20, 10, 40)$ the posteriors for β_y and β_z are truncated at their upper values, so that larger c values are needed, and show further by a bit of trial and error that $(15, 60, 15, 90)$ are approximately the smallest c values that do not lead to truncation.

To compare models (1) and (2), and also to obtain information about both additive and multiplicative effects, monitor both of the following quantities in your MCMC work: the difference of means in the Beta model, $\left(\frac{\alpha_y}{\alpha_y+\beta_y} - \frac{\alpha_z}{\alpha_z+\beta_z}\right)$, and the ratio of these means, $\frac{\alpha_y(\alpha_z+\beta_z)}{\alpha_z(\alpha_y+\beta_y)}$. Make a full report using `CODA` or the diagnostics built into `WinBUGS` on what you think is an appropriate MCMC strategy to get accurate posterior summaries for all of the α and β parameters and the additive and multiplicative effects. How do the Bayesian posterior mean estimates of $(\alpha_y, \beta_y, \alpha_z, \beta_z)$ compare with their MoM counterparts? How do the additive effect estimate and its uncertainty band in the Beta model compare with the corresponding values from the Gaussian model? Use `WinBUGS` to compute the DIC value for model (2) based on a monitoring run of appropriate length, and compare this with the DIC value you got for the Gaussian model in (c). Did DIC get the number of parameters right this time? According to the DIC approach to

Bayesian model selection, which model is better? Explain briefly. Conclude this part of the problem by summarizing the multiplicative effect of a bike route on PBT for residential streets in Berkeley in the early 1990s and its associated uncertainty band.

(f) Choose any two pairwise comparisons in Table 2 that interest you (other than the one in (e)) and perform an analysis like (e) on each of them, with an eye to learning more about the effect of a bike route and/or the street type on PBT. An example of what I have in mind would be to collect together all 30 PBT values for blocks with a bike route and call that your y vector, and similarly let z be all 30 PBT values for blocks without a bike route; this comparison would summarize the overall effect of a bike route on PBT regardless of street type. Another interesting pairwise comparison would involve using as y the 20 PBT values from residential blocks and using as z the 20 PBT values from (say) busy blocks; this would summarize the effect of (residential versus busy) on PBT regardless of presence or absence of a bike route. (You can see that there are three possible pairwise comparisons that all bear on the overall effect of street type on PBT; the cumbersome nature of an approach based on lots of pairwise comparisons is what prompted the invention of the analysis of variance back in the 1920s.) If your choice of comparisons involves the (fairly busy, no) cell in Table 1, something unpleasant will happen when you try to fit model (2) to the data. What is the cause of this unpleasant behavior, and what is a simple approximate remedy? Explain briefly, and conduct your analysis using that remedy.

Chapter 5

Hierarchical models for combining information

5.1 The role of scientific context in formulating hierarchical models

Case Study: *Meta-analysis of effects of aspirin on heart attacks.* Table 5.1 (Draper et al., 1993a) gives the number of patients and mortality rate from all causes, for six randomized controlled experiments comparing the use of aspirin and placebo by patients following a heart attack.

Table 5.1. Aspirin meta-analysis data.

	Aspirin		Placebo	
	# of	Mortality	# of	Mortality
Study (i)	Patients	Rate (%)	Patients	Rate (%)
UK-1	615	7.97	624	10.74
CDPA	758	5.80	771	8.30
GAMS	317	8.52	309	10.36
UK-2	832	12.26	850	14.82
PARIS	810	10.49	406	12.81
AMIS	2267	10.85	2257	9.70
Total	5599	9.88	5217	10.73

	Comparison			
	y_i = Diff	$\sqrt{V_i}$ = SE		
Study (i)	(%)	of Diff (%)	Z_i[‡]	p_i[§]
UK-1	2.77	1.65	1.68	.047
CDPA	2.50	1.31	1.91	.028
GAMS	1.84	2.34	0.79	.216
UK-2	2.56	1.67	1.54	.062
PARIS	2.31	1.98	1.17	.129
AMIS	–1.15	0.90	–1.27	.898
Total	0.86	0.59	1.47	.072

[‡] Z_i denotes the ratio of the difference in mortality rates over its standard error, assuming a binomial distribution. [§] p_i is the one-sided p value associated with Z_i, using the normal approximation.

A Gaussian meta-analysis model. The first five trials are reasonably consistent in showing a (weighted average) mortality decline for aspirin patients of 2.3 percentage points, a 20% drop from the (weighted average) placebo mortality of 11.5% (this difference is highly clinically significant).

However, the sixth and largest trial, AMIS, went the other way: an increase of 1.2 percentage points in aspirin mortality (a 12% rise from the placebo baseline of 9.7%).

Some relevant questions (Draper, 1995):

Q_1 Why did AMIS get such different results?

Q_2 What should be done next to reduce the uncertainty about Q_1?

Q_3 If you were a doctor treating a patient like those eligible for the trials in Table 5.1, what therapy should you employ while answers to Q_1 and Q_2 are sought?

One possible paraphrase of Q_3: Q_4 How should the information from these six experiments be combined to produce a more informative summary than those obtained from each experiment by itself?

The discipline of meta-analysis is devoted to answering questions like Q_4.

One leading school of frequentist meta-analysis (e.g., Hedges and Olkin, 1985) looks for methods for combining the Z and p values in Table 5.1, an approach that often leads only to an overall p value.

A Gaussian HM. A more satisfying form of meta-analysis (which has both frequentist and Bayesian versions) builds a hierarchical model (HM) that indicates how to combine information from the mortality differences in the table.

A Gaussian meta-analysis model for the aspirin data, for example (Draper et al., 1993a), might look like

$$\begin{array}{rcll} \big(\theta, \sigma^2\big) & \sim & p\big(\theta, \sigma^2\big) & \text{(prior)} \\ (\theta_i | \theta, \sigma^2) & \stackrel{\text{IID}}{\sim} & N\big(\theta, \sigma^2\big) & \text{(underlying effects)} \\ (y_i | \theta_i) & \stackrel{\text{indep}}{\sim} & N(\theta_i, V_i) & \text{(data)} \ . \end{array} \qquad (5.1)$$

The bottom level of (1), the data level of the HM, says that—because of relevant differences in patient cohorts and treatment protocols—each study has its own underlying treatment effect θ_i, and the observed mortality differences y_i are like random draws from a normal distribution with mean θ_i and variance V_i (the normality is reasonable because of the Central Limit Theorem, given the large numbers
of patients).

In meta-analyses of data like those in Table 5.1 the V_i are typically taken to be known (again because the patient sample sizes are so big), $V_i = SE_i^2$, where SE_i is the standard error of the mortality difference for study i in Table 5.1.

The middle level of the HM is where you would bring in the study-level covariates, if you have any, to try to explain why the studies differ in their underlying effects.

Here there are no study-level covariates, so the middle level of (1) is equivalent to a Gaussian regression with no predictor variables.

Why the "error" distribution should be Gaussian at this level of the HM is not clear—it's a conventional option, not a choice that's automatically scientifically reasonable (in fact I'll challenge it later).

σ^2 in this model represents study-level heterogeneity.

The top level of (1) is where the prior distribution on the regression parameters from the middle level is specified.

Here, with only an intercept term in the regression model, a popular conventional choice is the normal/scaled-inverse-χ^2 prior we looked at earlier in our first Gaussian case study.

Fixed effects and random effects. If σ^2 were known somehow to be 0, all of the θ_i would have to be equal deterministically to a common value θ, yielding a simpler model: $(y_i | \theta) \stackrel{\text{indep}}{\sim} N(\theta, V_i)\,, \theta \sim p(\theta)$.

Meta-analysts call this a fixed-effects model, and refer to model (1) as a random-effects model.

When σ^2 is not assumed to be 0, with this terminology the θ_i are called random effects (this parallels the use of this term in the random-effects Poisson regression case study).

Approximate fitting of Gaussian hierarchical models: maximum likelihood and empirical Bayes. Some algebra based on model (1) yields that the conditional distributions of the study-level effects θ_i given the data and the parameters (θ, σ^2), have a simple and revealing form:

$$\left(\theta_i | y_i, \theta, \sigma^2\right) \overset{\text{indep}}{\sim} N[\theta_i^*, V_i(1 - B_i)] \,, \tag{5.2}$$

$$\text{with} \quad \theta_i^* = (1 - B_i)\, y_i + B_i\, \theta \quad \text{and} \quad B_i = \frac{V_i}{V_i + \sigma^2}. \tag{5.3}$$

In other words, the conditional mean of the effect for study i given y_i, θ, and σ^2 is a weighted average of the sample mean for that study, y_i, and the overall mean θ.

The weights are given by the so-called shrinkage factors B_i (e.g., Draper et al., 1993a), which in turn depend on how the variability V_i within study i compares to the between-study variability σ^2: the more accurately y_i estimates θ_i, the more weight the "local" estimate y_i gets in the weighted average.

The term shrinkage refers to the fact that, with this approach, unusually high or low individual studies are drawn back or "shrunken" toward the overall mean θ when making the calculation $(1 - B_i)\, y_i + B_i\, \theta$.

Note that θ_i^* uses data from all the studies to estimate the effect for study i—this is referred to as borrowing strength in the estimation process.

Closed-form expressions for $p(\theta|y)$ and $p(\theta_i|y)$ with $y = (y_1, \ldots, y_k), k = 6$ are not available even with a normal/scaled-inverse-χ^2 prior for (θ, σ^2); MCMC is needed (see below).

5.1.1 Maximum likelihood and empirical Bayes

In the meantime maximum likelihood calculations provide some idea of what to expect: the likelihood function based on model (1) is

$$l\big(\theta, \sigma^2 | y\big) = c \prod_{i=1}^{k} \frac{1}{\sqrt{V_i + \sigma^2}} \exp\left[-\frac{1}{2} \sum_{i=1}^{k} \frac{(y_i - \theta)^2}{V_i + \sigma^2} \right]. \tag{5.4}$$

The maximum likelihood estimates (MLEs) $\left(\hat{\theta}, \hat{\sigma}^2\right)$ then turn out to be the iterative solutions to the following equations:

$$\hat{\theta} = \frac{\sum_{i=1}^k \hat{W}_i \, y_i}{\sum_{i=1}^k \hat{W}_i} \quad \text{and} \quad \hat{\sigma}^2 = \frac{\sum_{i=1}^k \hat{W}_i^2 \left[(y_i - \hat{\theta})^2 - V_i \right]}{\sum_{i=1}^k \hat{W}_i^2}, \tag{5.5}$$

$$\text{where} \quad \hat{W}_i = \frac{1}{V_i + \hat{\sigma}^2}. \tag{5.6}$$

Start with $\hat{\sigma}^2 = 0$ and iterate (5–6) to convergence (if $\hat{\sigma}^2$ converges to a negative value, $\hat{\sigma}^2 = 0$ is the MLE); the MLEs of the θ_i are then given by

$$\hat{\theta}_i = \left(1 - \hat{B}_i\right) \, y_i + \hat{B}_i \, \theta \quad \text{where} \quad \hat{B}_i = \frac{V_i}{V_i + \hat{\sigma}^2}. \tag{5.7}$$

These are called empirical Bayes (EB) estimates of the study-level effects, because it turns out that this analysis approximates a fully Bayesian solution by (in effect) using the data to estimate the prior specifications for θ and σ^2.

Large-sample (mainly meaning large k) approximations to the (frequentist) distributions of the MLEs are given by

$$\hat{\theta} \sim N\left(\theta, \left[\sum_{i=1}^k \frac{1}{V_i + \hat{\sigma}^2} \right]^{-1} \right) \quad \text{and} \quad \hat{\theta}_i \sim N\left[\theta_i, V_i \left(1 - \hat{B}_i\right) \right]. \tag{5.8}$$

NB The variances in (8) don't account fully for the uncertainty in σ^2 and therefore underestimate the actual sampling variances for small k (adjustments are available; see, e.g., Morris (1983, 1988)).

MLEB estimation can be implemented simply in about 15 lines of `R` code (Table 5.2).

Table 5.2. `R` program to perform MLEB calculations.

```
mleb <- function( y, V, m ) {
  sigma2 <- 0.0
  for ( i in 1:m ) {
    W <- 1.0 / ( V + sigma2 )
    theta <- sum( W * y ) / sum( W )
    sigma2 <- sum( W^2 * ( ( y - theta )^2 - V ) ) / sum( W^2 )
    B <- V / ( V + sigma2 )
    effects <- ( 1 - B ) * y + B * theta
    se.theta <- 1.0 / sqrt( sum( 1.0 / ( V + sigma2 ) ) )
```

```
      se.effects <- sqrt( V * ( 1.0 - B ) )
      print( c( i, theta, se.theta, sigma2 ) )
      print( cbind( W, ( W / sum( W ) ), B, y, effects,
        se.effects ) )
    }
  }
```

With the aspirin data it takes 18 iterations (less than 0.1 second on a 400MHz `UltraSPARC` Unix machine) to get convergence to 4-digit accuracy, leading to the summaries in Table 5.3 and the following estimates (standard errors in parentheses):

$$\hat{\theta} = 1.45 \ (0.809), \quad \hat{\sigma}^2 = 1.53.$$

Table 5.3. Maximum likelihood empirical Bayes meta-analysis of the aspirin data.

study(i)	$\hat{W}_i$	normalized $\hat{W}_i$	$\hat{B}_i$	y_i	$\hat{\theta}_i$	$\widehat{SE}\left(\hat{\theta}_i\right)$
1	0.235	0.154	0.640	2.77	1.92	0.990
2	0.308	0.202	0.529	2.50	1.94	0.899
3	0.143	0.0934	0.782	1.84	1.53	1.09
4	0.232	0.151	0.646	2.56	1.84	0.994
5	0.183	0.120	0.719	2.31	1.69	1.05
6	0.427	0.280	0.346	−1.15	−0.252	0.728

Aspirin meta-analysis: conclusions. Note that (1) AMIS gets much less weight (normalized $\hat{W}_i$) than would have been expected given its small V_i; (2) the shrinkage factors ($\hat{B}_i$) are considerable, with AMIS shrunk almost all the way into positive territory (see Figure 5.1); (3) there is considerable study-level heterogeneity ($\hat{\sigma} \doteq 1.24$ percentage points of mortality); and (4) the standard errors of the effects are by and large smaller than the $\sqrt{V_i}$ (from the borrowing of strength) but are still considerable.

The 95% interval estimate of θ, the overall underlying effect of aspirin on mortality, from this approach comes out

$$\hat{\theta} \pm 1.96 \cdot \widehat{SE}\left(\hat{\theta}\right) \doteq (-0.140, 3.03),$$

which if interpreted Bayesianly gives

$$P(\text{aspirin reduces mortality}|\text{data}) \doteq 1 - \Phi\left(\tfrac{0-1.45}{0.809}\right) = 0.96,$$

where Φ is the standard normal CDF.

Thus although the interval includes 0, so that in a frequentist sense the effect is not statistically significant, in fact from a Bayesian point of view the evidence is running strongly in favor of aspirin's usefulness.

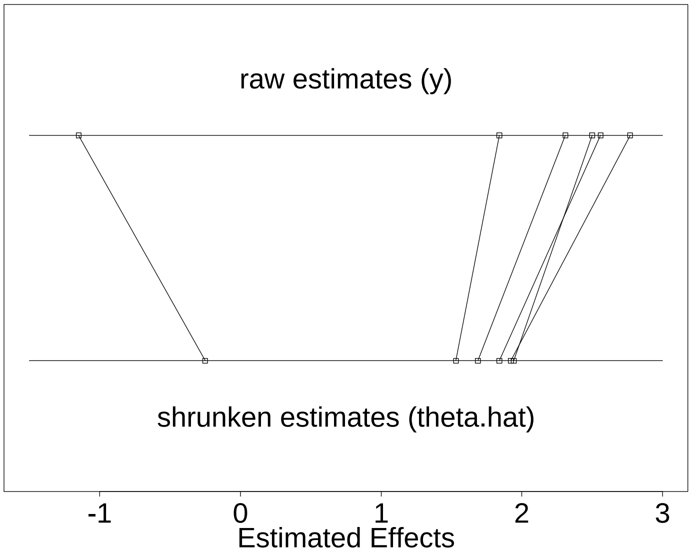


Figure 5.1: *Shrinkage plot for the aspirin MLEB meta-analysis.*

5.1.2 Incorporating study-level covariates

In many cases (as with this example) empirical Bayes methods have the advantage of yielding closed-form solutions, but I view them at best as approximations to fully Bayesian analyses—which can in any case be carried out with MCMC—so I won't have any more to say about EB methods here (see Carlin and Louis, 1996, for more on this topic).

Case Study: *Meta-analysis of the effect of teacher expectancy on student IQ* (Bryk and Raudenbush, 1992). Do teachers' expectations influence students' intellectual development,
as measured by IQ scores?

Table 5.4. Results from 19 experiments estimating the effects of teacher expectancy on pupil IQ.

Study (i)	Weeks of Prior Contact (x_i)	Estimated Effect Size (y_i)	Standard Error of $y_i = \sqrt{V_i}$
1. Rosenthal et al. (1974)	2	0.03	0.125
2. Conn et al. (1968)	3	0.12	0.147
3. Jose & Cody (1971)	3	-0.14	0.167
4. Pellegrini & Hicks (1972)	0	1.18	0.373
5. Pellegrini & Hicks (1972)	0	0.26	0.369
6. Evans & Rosenthal (1969)	3	-0.06	0.103
7. Fielder et al. (1971)	3	-0.02	0.103
8. Claiborn (1969)	3	-0.32	0.220
9. Kester & Letchworth (1972)	0	0.27	0.164
10. Maxwell (1970)	1	0.80	0.251
11. Carter (1970)	0	0.54	0.302
12. Flowers (1966)	0	0.18	0.223
13. Keshock (1970)	1	-0.02	0.289
14. Henrickson (1970)	2	0.23	0.290
15. Fine (1972)	3	-0.18	0.159
16. Greiger (1970)	3	-0.06	0.167
17. Rosenthal & Jacobson (1968)	1	0.30	0.139
18. Fleming & Anttonen (1971)	2	0.07	0.094
19. Ginsburg (1970)	3	-0.07	0.174

Teacher expectancy. Raudenbush (1984) found $k = 19$ experiments, published between 1966 and 1974, estimating the effect of teacher expectancy on student IQ (Table 5.4).

In each case the experimental group was made up of children for whom teachers were (deceptively) encouraged to have high expectations (e.g., experimenters gave treatment teachers lists of students, actually chosen at random, who allegedly displayed dramatic potential for intellectual growth), and the controls were students about whom no particular expectations were encouraged.

The estimated effect sizes $y_i = \frac{\bar{T}_i - \bar{C}_i}{\text{SD}_{i:\text{pooled}}}$ (column 3 in Table 5.4) ranged from $-.32$ to $+1.18$; why?

One good reason: the studies differed in how well the experimental teachers knew their students at the time they were given the deceptive information: this time period x_i (column 2 in Table 5.4) ranged from 0 to 3 weeks.

Figure 5.2 plots y_i against x_i—you can see that the studies with bigger x_i had smaller IQ effects on average.

Conditional exchangeability. Evidently model (1) will not do here—

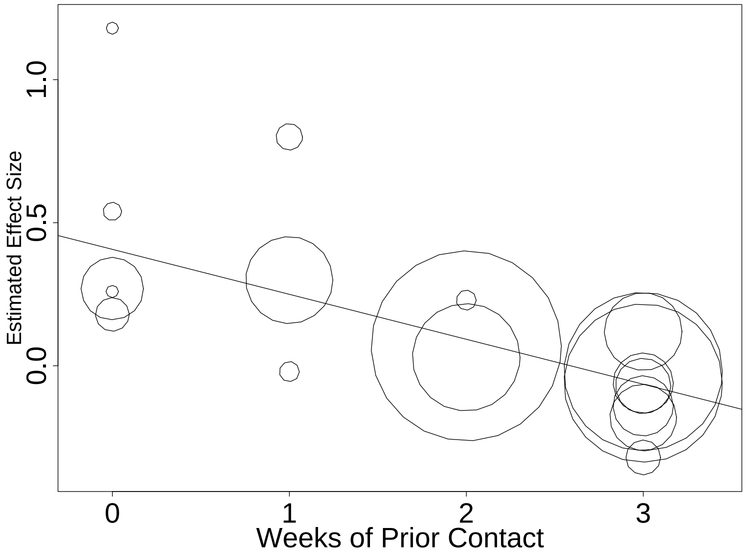


Figure 5.2: *Scatterplot of estimated effect size against weeks of prior contact in the IQ meta-analysis. Radii of circles are proportional to* $w_i = V_i^{-1}$ *(see column 4 in Table 5.4); fitted line is from weighted regression of* y_i *on* x_i *with weights* w_i*.*

it says that your predictive uncertainty about all the studies is exchangeable (similar, i.e., according to (1) the underlying study-level effects θ_i are like IID draws from a normal distribution), whereas Figure 5.2 clearly shows that the x_i are useful in predicting the y_i.

This is another way to say that your uncertainty about the studies is not unconditionally exchangeable but conditionally exchangeable given x (Draper et al., 1993b).

In fact Figure 5.2 suggests that the y_i (and therefore the θ_i) are related linearly to the x_i.

Bryk and Raudenbush, working in the frequentist paradigm, fit the following HM to these data:

$$\begin{aligned} (\theta_i|\alpha,\beta,\sigma_\theta^2) &\overset{\text{indep}}{\sim} N\big(\alpha+\beta\,x_i,\sigma_\theta^2\big) && \text{(underlying effects)} \\ (y_i|\theta_i) &\overset{\text{indep}}{\sim} N(\theta_i,V_i) && \text{(data).} \end{aligned} \tag{5.9}$$

According to this model the estimated effect sizes y_i are like draws from a Gaussian with mean θ_i and variance V_i, the squared standard errors from

column 4 of Table 5.4—here as in model (1) the V_i are taken to be known—and the θ_i themselves are like draws from a Gaussian
with mean $\alpha + \beta x_i$ and variance σ_θ^2.

The top level of this HM in effect assumes, e.g., that the 5 studies with $x = 0$ are sampled representatively from $\{$all possible studies with $x = 0\}$, and similarly for the other values of x.

This (and the Gaussian choice on the top level) are conventional assumptions, not automatically scientifically reasonable—for example, if you know of some way in which (say) two of the studies with $x = 3$ differ from each other that's relevant to the outcome of interest, then you should include this in the model as a study-level covariate along with x.

An MLEB drawback. Bryk and Raudenbush used MLEB methods, based on the EM algorithm, to fit this model.

As in Section 5.2, this estimation method combines the two levels of model (9) to construct a single likelihood for the y_i, and then maximizes this likelihood as usual in the ML approach.

They obtained $(\hat{\alpha}, \hat{\beta}) = (.407 \pm .087, -.157 \pm .036)$ and $\hat{\sigma_\theta}^2 = 0$, naively indicating that all of the study-level variability has been "explained" by the covariate x.

However, from a Bayesian point of view, this model is missing a third layer:

$$\begin{array}{rcl} (\alpha, \beta, \sigma_\theta^2) & \sim & p(\alpha, \beta, \sigma_\theta^2) \\ (\theta_i | \alpha, \beta, \sigma_\theta^2) & \stackrel{\text{IID}}{\sim} & N\big(\alpha + \beta(x_i - \bar{x}), \sigma_\theta^2\big) \\ (y_i | \theta_i) & \stackrel{\text{indep}}{\sim} & N(\theta_i, V_i) \,. \end{array} \tag{5.10}$$

(it will help convergence of the sampling-based MCMC methods to make α and β uncorrelated by centering the x_i at 0 rather than at $\bar{x}$).

As will subsequently become clear, the trouble with MLEB is that in Bayesian language it assumes in effect that the posterior for σ_θ^2 is point-mass on the MLE. This is bad
(e.g., Morris, 1983) for two reasons:

• If the posterior for σ_θ^2 is highly skewed, the mode will be a poor summary; and

• Whatever point-summary you use, pretending the posterior SD for σ^2 is zero fails to propagate uncertainty about σ_θ^2 through to uncertainty about α, β, and the θ_i.

The best way to carry out a fully Bayesian analysis of model (10) is with MCMC methods.

For $p(\alpha, \beta, \sigma_\theta^2)$ in model (10) I've chosen the usual `WinBUGS` diffuse prior $p(\alpha)p(\beta)p(\sigma_\theta^2)$: since α and β live on the whole real line I've taken marginal Gaussian priors for them with mean 0 and precision 10^{-6}, and since $\tau_\theta = \frac{1}{\sigma^2}$ is positive I use a $\Gamma(0.001, 0.001)$ prior for it.

Model (10) treats the variances V_i of the y_i as known (and equal to the squares of column 4 in Table 5.4); I've converted these into precisions in the data file (e.g., $\tau_1 = \frac{1}{0.125^2} = 64.0$).

A burn-in of 1,000 (certainly longer than necessary) from default initial values $(\alpha, \beta, \tau_\theta) = (0.0, 0.0, 1.0)$ and a monitoring run of 10,000 yield the following preliminary MCMC results.

Because this is a random-effects model we don't expect anything like IID mixing: the output for α behaves like an AR_1 time series with $\hat{\rho}_1 \doteq 0.86$.

The posterior mean for α, 0.135 (with an MCSE of 0.002), shows that α in model (10) and α in model (9) are not comparable because of the recentering of the predictor x in model (10): the MLE of α in (9) was 0.41 ± 0.09.

But β means the same thing in both models (9) and (10): its posterior mean in (10) is -0.161 ± 0.002, which is not far from the MLE -0.157.

Note, however, that the posterior SD for β, 0.0396, is 10% larger than the standard error of the maximum likelihood estimate of β (0.036).

This is a reflection of the underpropagation of uncertainty about σ_θ in maximum likelihood mentioned on page 15.

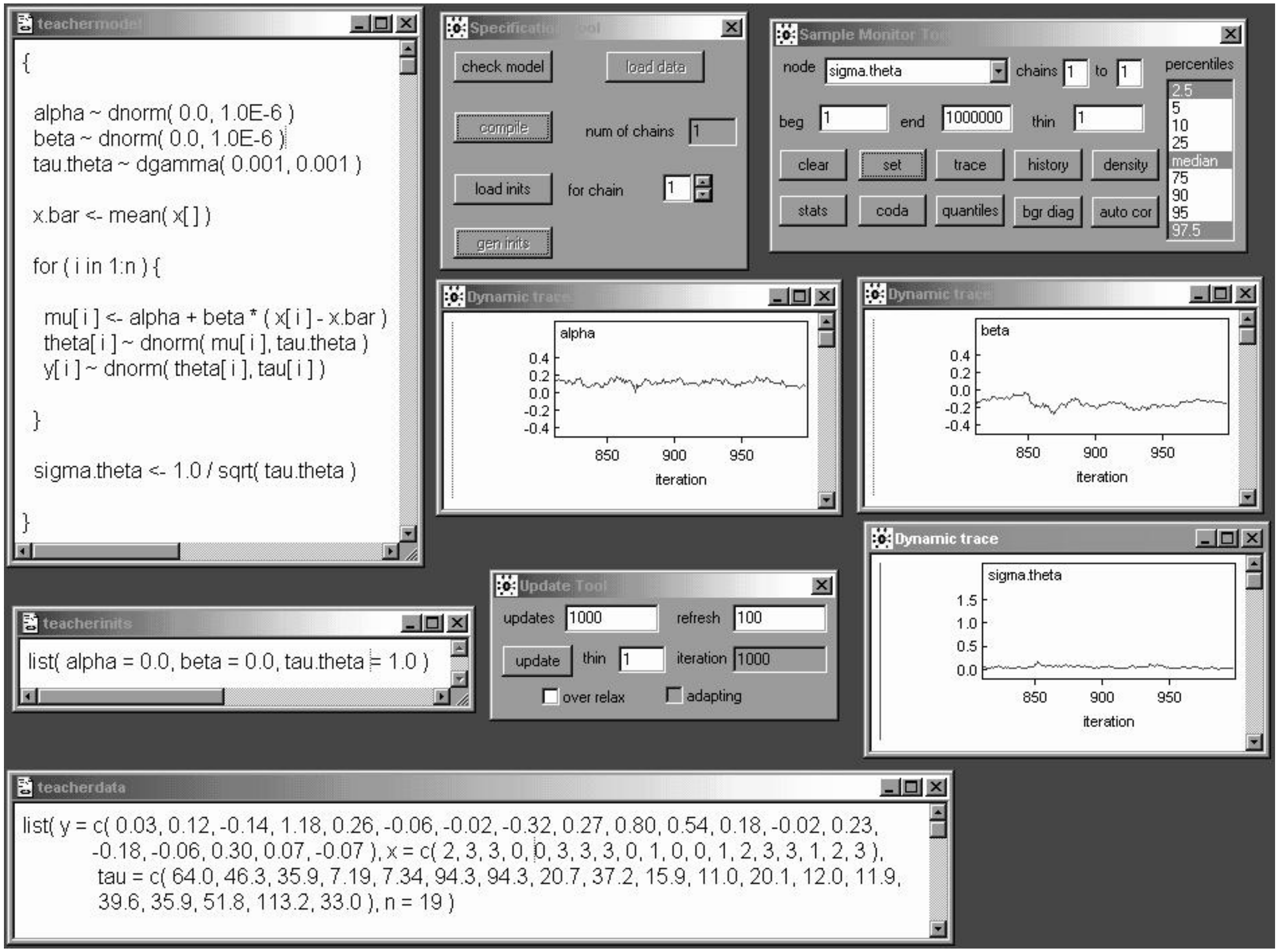
In these preliminary results σ_θ has posterior mean 0.064 ± 0.002 and SD 0.036, providing clear evidence that the MLE $\hat{\sigma}_\theta = 0$ is a poor summary.

Note, however, that the likelihood for σ_θ may be appreciable in the vicinity of 0 in this case, meaning that some sensitivity analysis with diffuse priors other than $\Gamma(0.001, 0.001)$—such as $U(0, c)$ for c around 0.5—would be in order.

When you specify node `theta` in the `Sample Monitor Tool` and then look at the results, you see that `WinBUGS` presents parallel findings with a single click for all elements of the vector θ.

Some of the θ_i are evidently mixing better than others.

Shrinkage estimation. In a manner parallel to the situation with the simpler model (1), the posterior means of the underlying study effects θ_i should be at least approximately related to the raw effect sizes y_i and the μ_i



teachermodel

```
{

 alpha ~ dnorm( 0.0, 1.0E-6 )
 beta ~ dnorm( 0.0, 1.0E-6 )
 tau.theta ~ dgamma( 0.001, 0.001 )

 x.bar <- mean( x[ ] )

 for ( i in 1:n ) {

  mu[ i ] <- alpha + beta * ( x[ i ] - x.bar )
  theta[ i ] ~ dnorm( mu[ i ], tau.theta )
  y[ i ] ~ dnorm( theta[ i ], tau[ i ] )

 }

 sigma.theta <- 1.0 / sqrt( tau.theta )

}
```

teacherinits

```
list( alpha = 0.0, beta = 0.0, tau.theta = 1.0 )
```

teacherdata

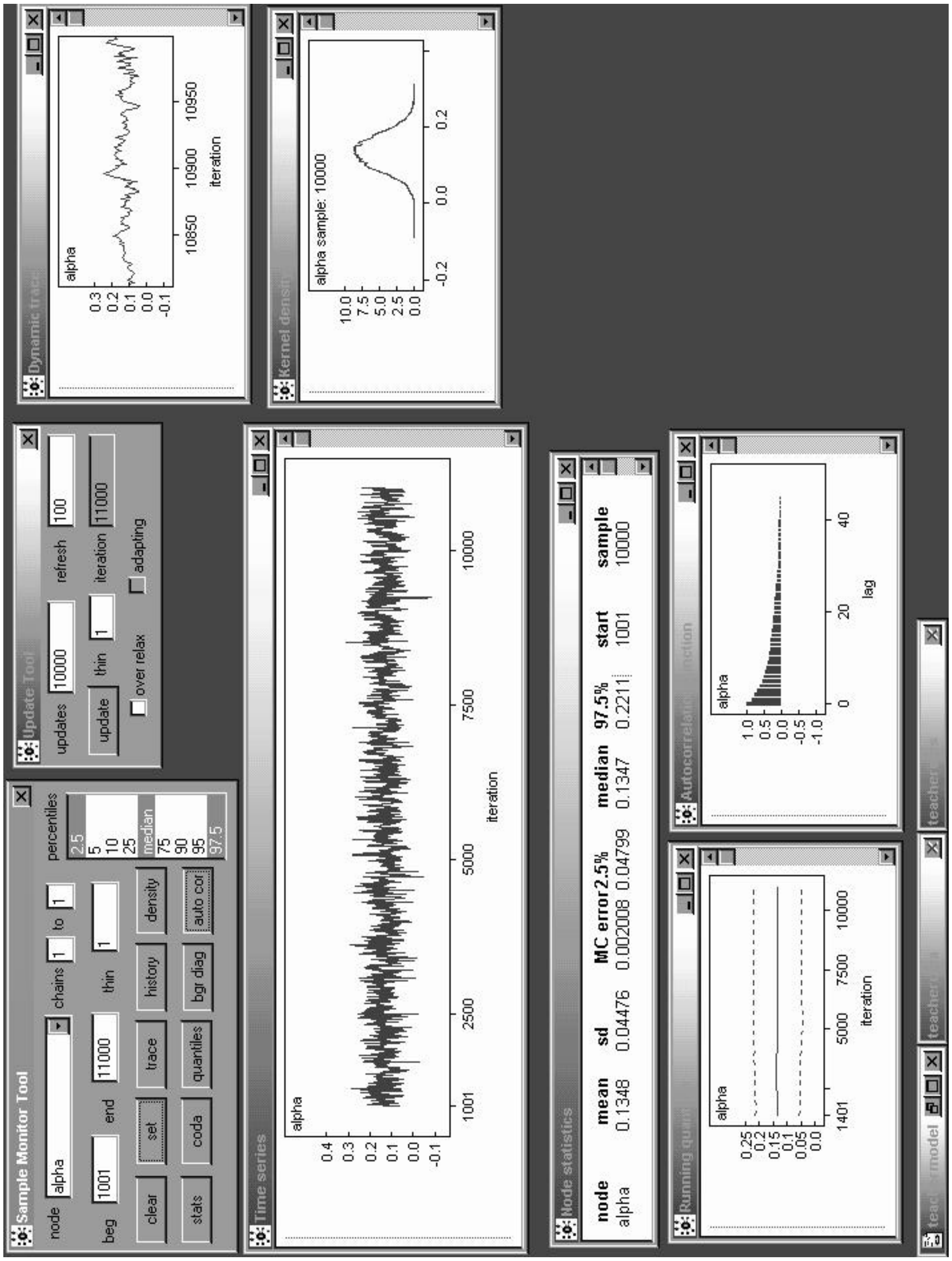
```
list( y = c( 0.03, 0.12, -0.14, 1.18, 0.26, -0.06, -0.02, -0.32, 0.27, 0.80, 0.54, 0.18, -0.02, 0.23,
        -0.18, -0.06, 0.30, 0.07, -0.07 ), x = c( 2, 3, 3, 0, 0, 3, 3, 3, 0, 1, 0, 0, 1, 2, 3, 3, 1, 2, 3 ),
        tau = c( 64.0, 46.3, 35.9, 7.19, 7.34, 94.3, 94.3, 20.7, 37.2, 15.9, 11.0, 20.1, 12.0, 11.9,
        39.6, 35.9, 51.8, 113.2, 33.0 ), n = 19 )
```

Specification Tool: num of chains 1; for chain 1

Sample Monitor Tool: node sigma.theta; chains 1 to 1; beg 1; end 1000000; thin 1

Update Tool: updates 1000; refresh 100; thin 1; iteration 1000; □ over relax; □ adapting

Figure 5.3: ***WinBUGS*** *implementation of model (5.10) applied to the teacher expectancy data.*



Sample Monitor Tool

node alpha chains 1 to 1 percentiles

beg 1001 end 11000 thin 1

Update Tool

updates 10000 refresh 100

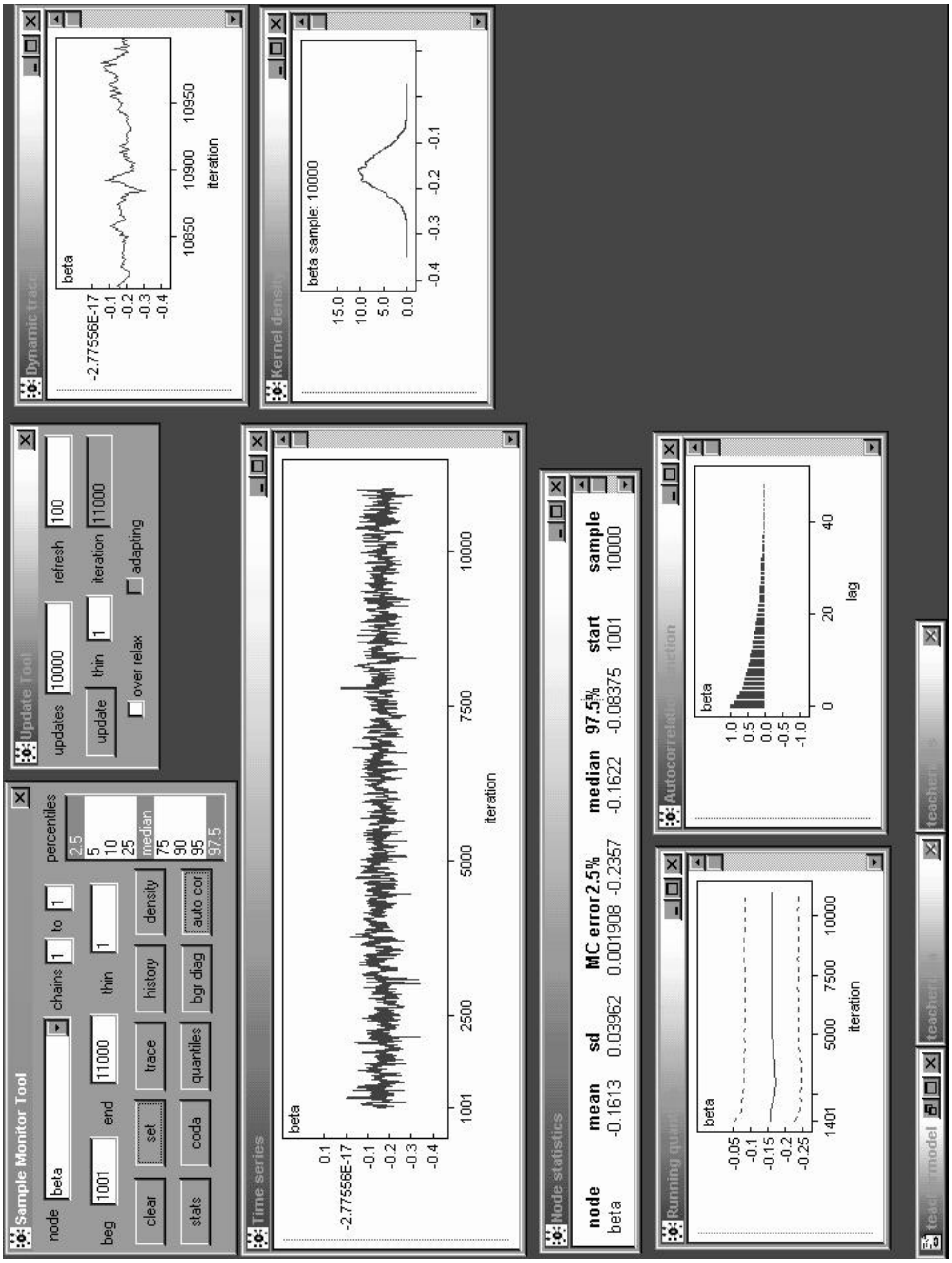
thin 1 iteration 11000

□ over relax □ adapting

Node statistics

```
node   mean    sd       MC error 2.5%    median  97.5%   start  sample
alpha  0.1348  0.04476  0.002008 0.04799 0.1347  0.2211  1001   10000
```

Figure 5.4: *Posterior inference for α in model (5.10).*



Sample Monitor Tool

node beta

beg 1001 end 11000

chains 1 to 1

thin 1

Update Tool

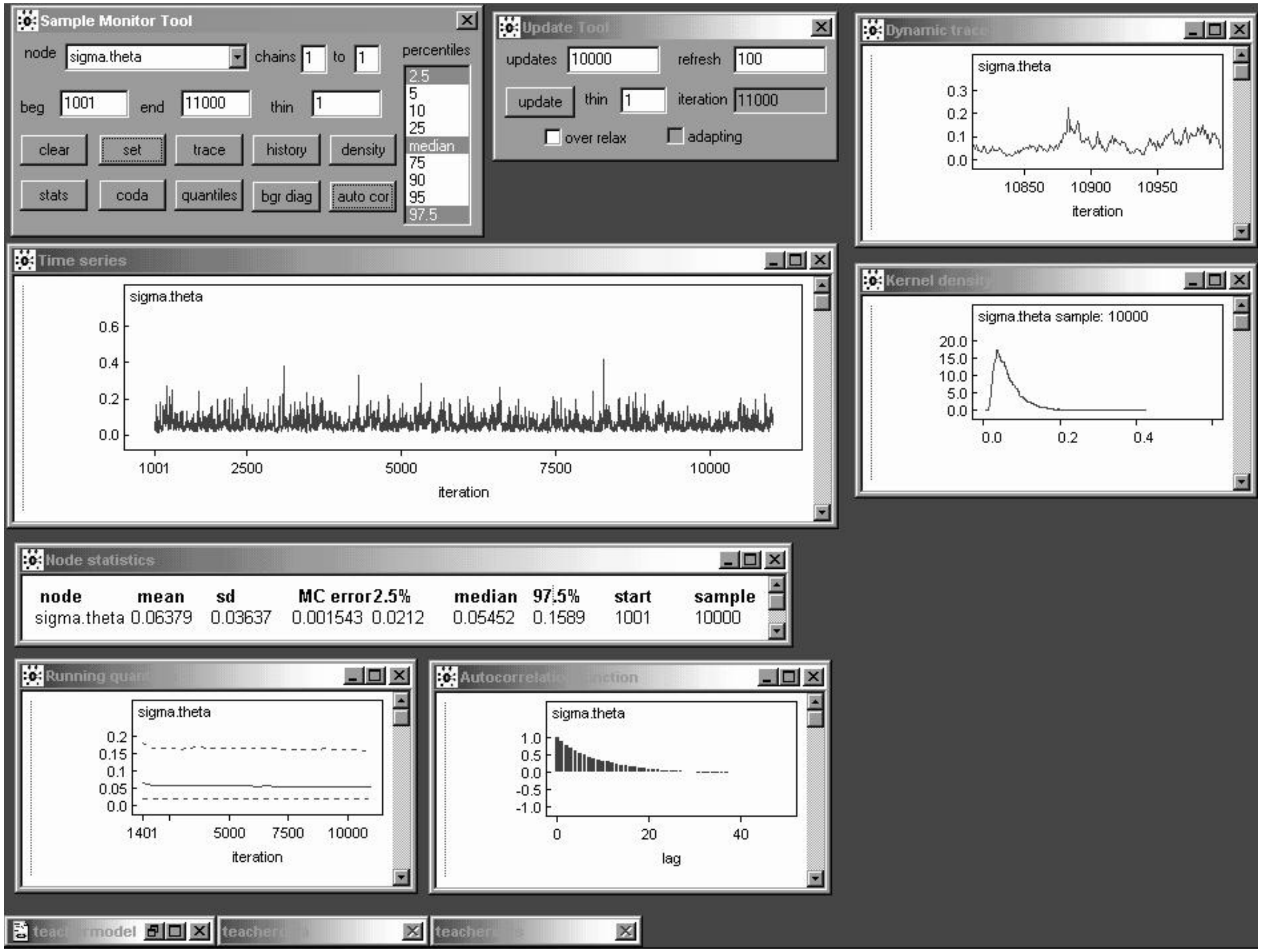
updates 10000 refresh 100

thin 1 iteration 11000

□ over relax □ adapting

```
node   mean     sd       MC error 2.5%     median   97.5%     start  sample
beta   -0.1613  0.03962  0.001908 -0.2357  -0.1622  -0.08375  1001   10000
```

Figure 5.5: *Posterior inference for β in model (5.10).*



Sample Monitor Tool

node sigma.theta chains 1 to 1

beg 1001 end 11000 thin 1

Update Tool

updates 10000 refresh 100

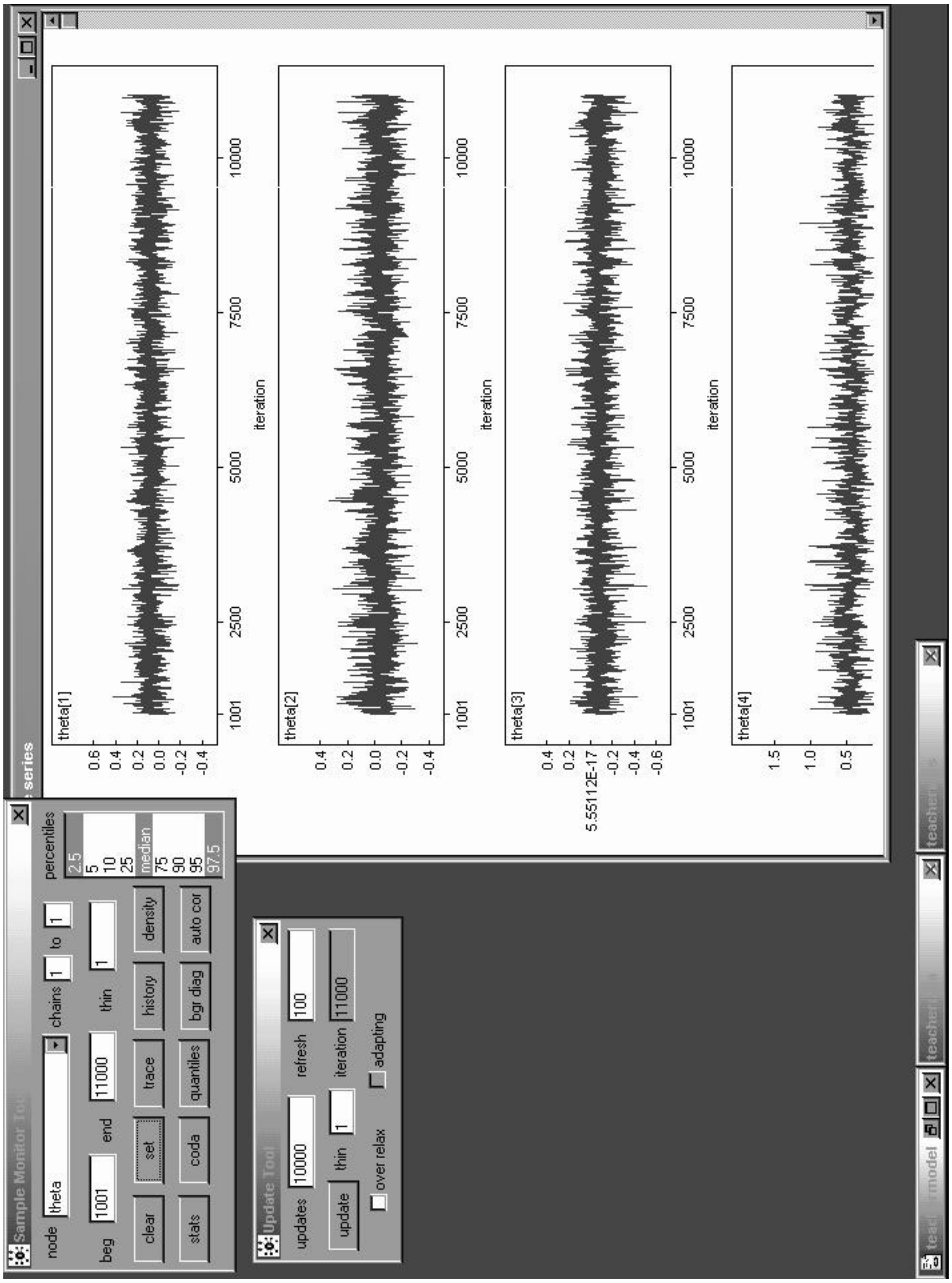
thin 1 iteration 11000

□ over relax □ adapting

Node statistics

```
node        mean    sd      MC error2.5%    median  97.5%   start   sample
sigma.theta 0.06379 0.03637 0.001543 0.0212 0.05452 0.1589  1001    10000
```

Figure 5.6: *Posterior inference for σ_θ in model (5.10).*



node theta

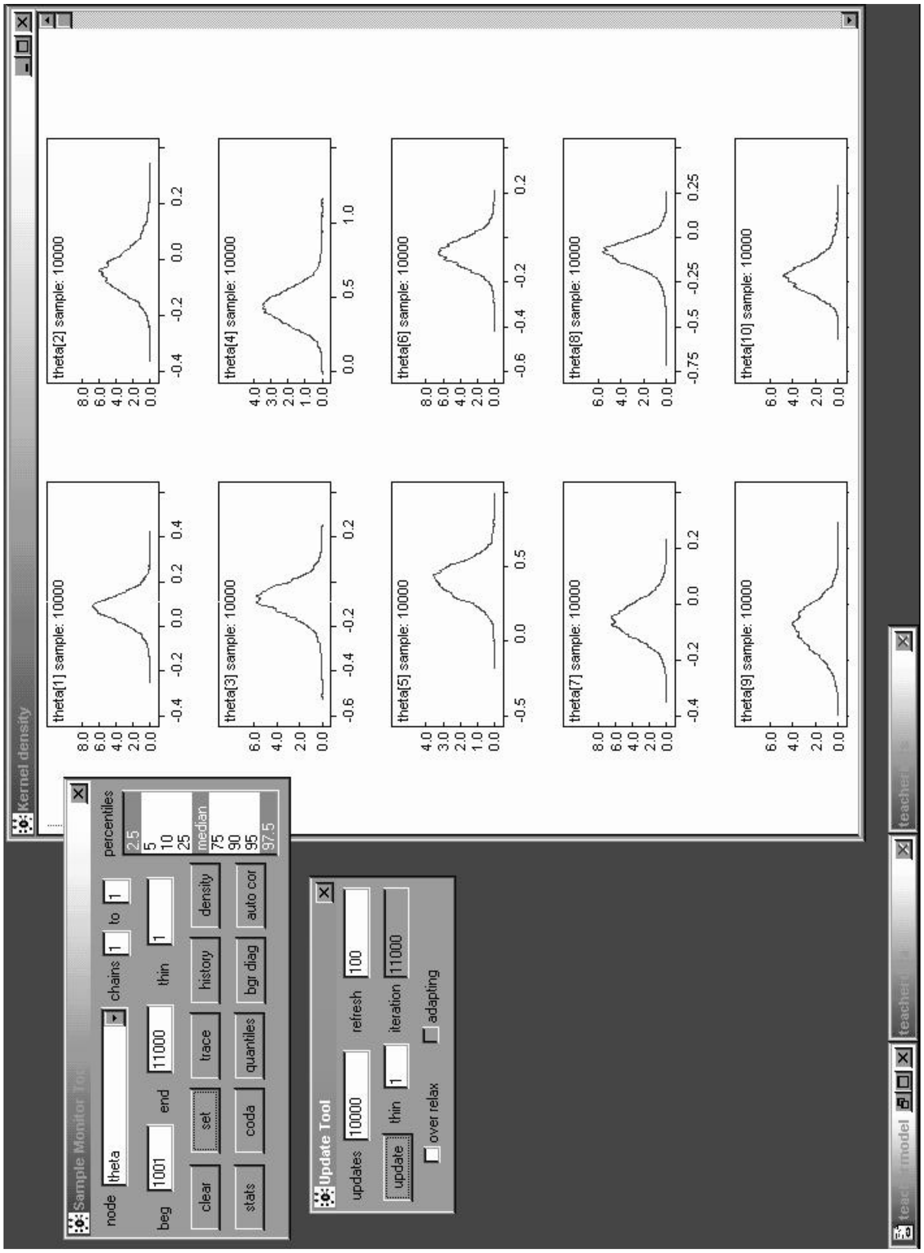
beg 1001 end 11000 thin 1

chains 1 to 1

updates 10000 refresh 100

thin 1 iteration 11000

Figure 5.7: *Looking at the study-level underlying effects θ_i.*



Sample Monitor Tool

node theta chains 1 to 1

beg 1001 end 11000 thin 1

percentiles: 2.5, 5, 10, 25, median, 75, 90, 95, 97.5

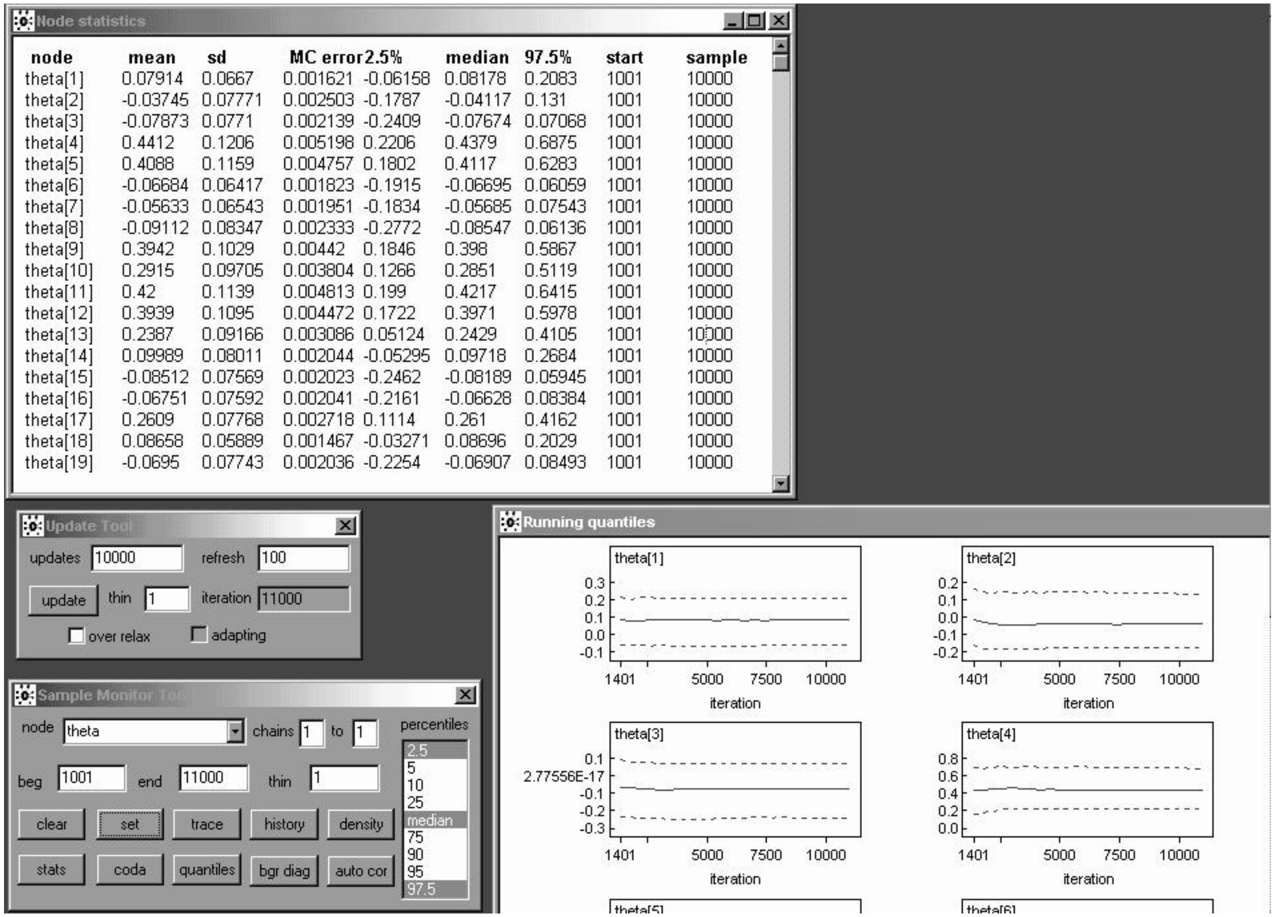
Update Tool

updates 10000 refresh 100

thin 1 iteration 11000

□ over relax □ adapting

Figure 5.8: *The marginal density traces of the θ_i look rather like t distributions with fairly low degrees of freedom (fairly heavy tails), which makes sense since the number of studies is small.*



Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
theta[1]	0.07914	0.0667	0.001621	-0.06158	0.08178	0.2083	1001	10000
theta[2]	-0.03745	0.07771	0.002503	-0.1787	-0.04117	0.131	1001	10000
theta[3]	-0.07873	0.0771	0.002139	-0.2409	-0.07674	0.07068	1001	10000
theta[4]	0.4412	0.1206	0.005198	0.2206	0.4379	0.6875	1001	10000
theta[5]	0.4088	0.1159	0.004757	0.1802	0.4117	0.6283	1001	10000
theta[6]	-0.06684	0.06417	0.001823	-0.1915	-0.06695	0.06059	1001	10000
theta[7]	-0.05633	0.06543	0.001951	-0.1834	-0.05685	0.07543	1001	10000
theta[8]	-0.09112	0.08347	0.002333	-0.2772	-0.08547	0.06136	1001	10000
theta[9]	0.3942	0.1029	0.00442	0.1846	0.398	0.5867	1001	10000
theta[10]	0.2915	0.09705	0.003804	0.1266	0.2851	0.5119	1001	10000
theta[11]	0.42	0.1139	0.004813	0.199	0.4217	0.6415	1001	10000
theta[12]	0.3939	0.1095	0.004472	0.1722	0.3971	0.5978	1001	10000
theta[13]	0.2387	0.09166	0.003086	0.05124	0.2429	0.4105	1001	10000
theta[14]	0.09989	0.08011	0.002044	-0.05295	0.09718	0.2684	1001	10000
theta[15]	-0.08512	0.07569	0.002023	-0.2462	-0.08189	0.05945	1001	10000
theta[16]	-0.06751	0.07592	0.002041	-0.2161	-0.06628	0.08384	1001	10000
theta[17]	0.2609	0.07768	0.002718	0.1114	0.261	0.4162	1001	10000
theta[18]	0.08658	0.05889	0.001467	-0.03271	0.08696	0.2029	1001	10000
theta[19]	-0.0695	0.07743	0.002036	-0.2254	-0.06907	0.08493	1001	10000

Update Tool

updates 10000 refresh 100

thin 1 iteration 11000

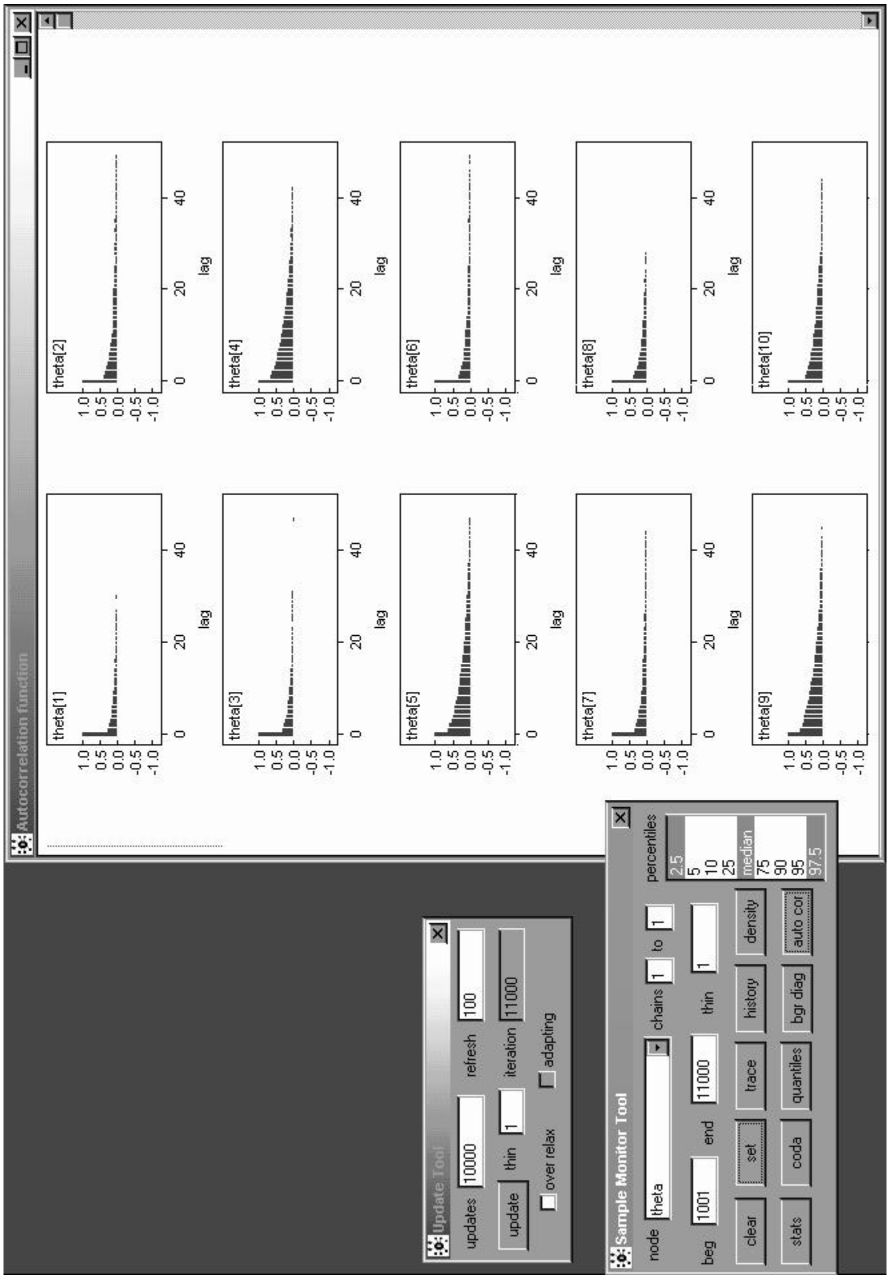
□ over relax □ adapting

Sample Monitor Tool

node theta chains 1 to 1

beg 1001 end 11000 thin 1

Figure 5.9: *Many of the θ_i have posterior probability concentrated near 0, but not all; $\theta_4, \theta_5, \theta_9, \theta_{11}$, and θ_{12} are particularly large (why?).*



updates 10000 refresh 100

thin 1 iteration 11000

□ over relax □ adapting

node theta

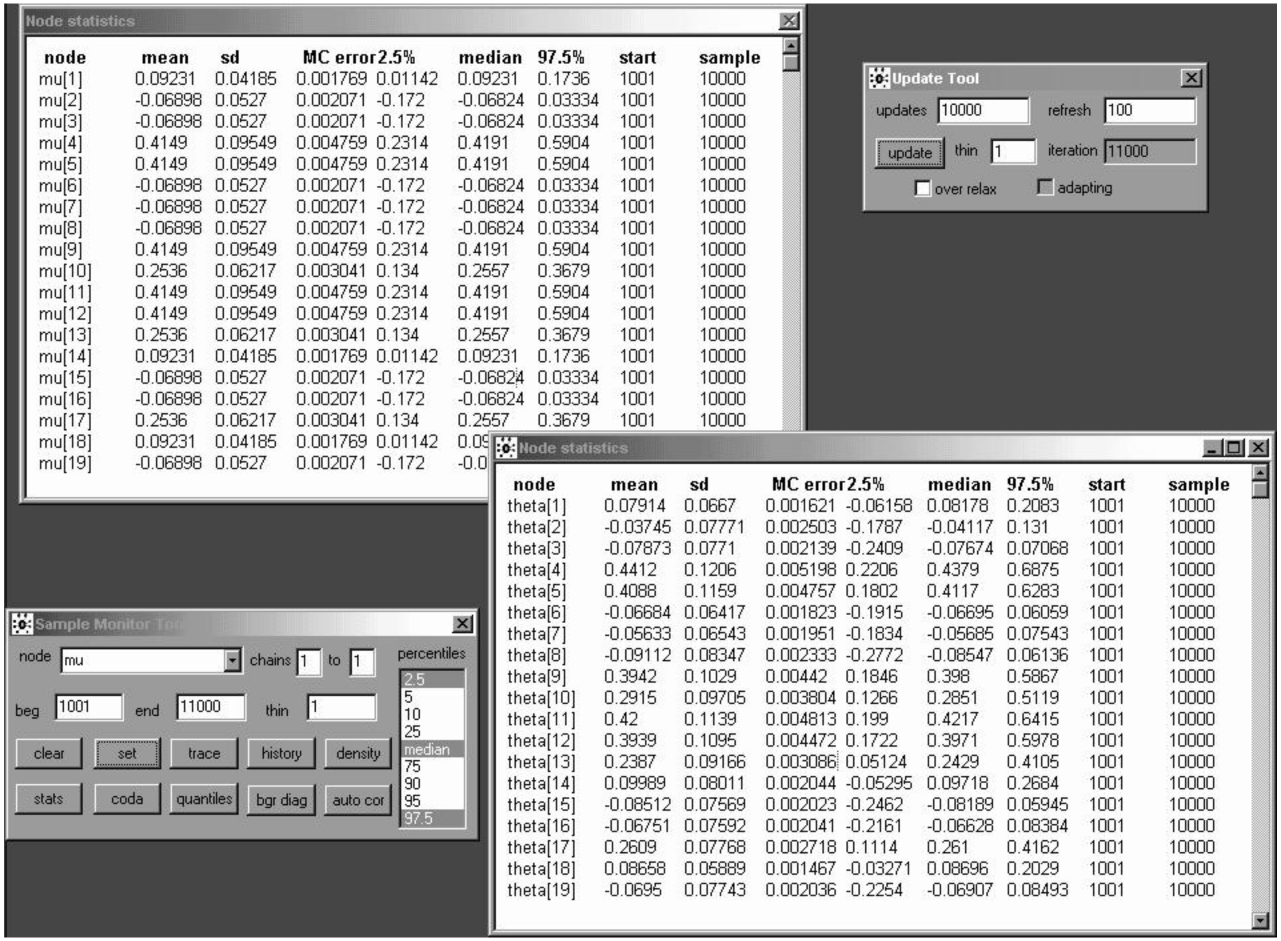
chains 1 to 1

beg 1001 end 11000

thin 1

percentiles: 2.5, 5, 10, 25, median, 75, 90, 95, 97.5

Figure 5.10: *Some of the θ_i are not far from white noise; others are mixing quite slowly.*



Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu[1]	0.09231	0.04185	0.001769	0.01142	0.09231	0.1736	1001	10000
mu[2]	-0.06898	0.0527	0.002071	-0.172	-0.06824	0.03334	1001	10000
mu[3]	-0.06898	0.0527	0.002071	-0.172	-0.06824	0.03334	1001	10000
mu[4]	0.4149	0.09549	0.004759	0.2314	0.4191	0.5904	1001	10000
mu[5]	0.4149	0.09549	0.004759	0.2314	0.4191	0.5904	1001	10000
mu[6]	-0.06898	0.0527	0.002071	-0.172	-0.06824	0.03334	1001	10000
mu[7]	-0.06898	0.0527	0.002071	-0.172	-0.06824	0.03334	1001	10000
mu[8]	-0.06898	0.0527	0.002071	-0.172	-0.06824	0.03334	1001	10000
mu[9]	0.4149	0.09549	0.004759	0.2314	0.4191	0.5904	1001	10000
mu[10]	0.2536	0.06217	0.003041	0.134	0.2557	0.3679	1001	10000
mu[11]	0.4149	0.09549	0.004759	0.2314	0.4191	0.5904	1001	10000
mu[12]	0.4149	0.09549	0.004759	0.2314	0.4191	0.5904	1001	10000
mu[13]	0.2536	0.06217	0.003041	0.134	0.2557	0.3679	1001	10000
mu[14]	0.09231	0.04185	0.001769	0.01142	0.09231	0.1736	1001	10000
mu[15]	-0.06898	0.0527	0.002071	-0.172	-0.06824	0.03334	1001	10000
mu[16]	-0.06898	0.0527	0.002071	-0.172	-0.06824	0.03334	1001	10000
mu[17]	0.2536	0.06217	0.003041	0.134	0.2557	0.3679	1001	10000
mu[18]	0.09231	0.04185	0.001769	0.01142	0.09			
mu[19]	-0.06898	0.0527	0.002071	-0.172	-0.0			

Update Tool

updates 10000 refresh 100

thin 1 iteration 11000

□ over relax □ adapting

Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
theta[1]	0.07914	0.0667	0.001621	-0.06158	0.08178	0.2083	1001	10000
theta[2]	-0.03745	0.07771	0.002503	-0.1787	-0.04117	0.131	1001	10000
theta[3]	-0.07873	0.0771	0.002139	-0.2409	-0.07674	0.07068	1001	10000
theta[4]	0.4412	0.1206	0.005198	0.2206	0.4379	0.6875	1001	10000
theta[5]	0.4088	0.1159	0.004757	0.1802	0.4117	0.6283	1001	10000
theta[6]	-0.06684	0.06417	0.001823	-0.1915	-0.06695	0.06059	1001	10000
theta[7]	-0.05633	0.06543	0.001951	-0.1834	-0.05685	0.07543	1001	10000
theta[8]	-0.09112	0.08347	0.002333	-0.2772	-0.08547	0.06136	1001	10000
theta[9]	0.3942	0.1029	0.00442	0.1846	0.398	0.5867	1001	10000
theta[10]	0.2915	0.09705	0.003804	0.1266	0.2851	0.5119	1001	10000
theta[11]	0.42	0.1139	0.004813	0.199	0.4217	0.6415	1001	10000
theta[12]	0.3939	0.1095	0.004472	0.1722	0.3971	0.5978	1001	10000
theta[13]	0.2387	0.09166	0.003086	0.05124	0.2429	0.4105	1001	10000
theta[14]	0.09989	0.08011	0.002044	-0.05295	0.09718	0.2684	1001	10000
theta[15]	-0.08512	0.07569	0.002023	-0.2462	-0.08189	0.05945	1001	10000
theta[16]	-0.06751	0.07592	0.002041	-0.2161	-0.06628	0.08384	1001	10000
theta[17]	0.2609	0.07768	0.002718	0.1114	0.261	0.4162	1001	10000
theta[18]	0.08658	0.05889	0.001467	-0.03271	0.08696	0.2029	1001	10000
theta[19]	-0.0695	0.07743	0.002036	-0.2254	-0.06907	0.08493	1001	10000

Sample Monitor Tool

node mu chains 1 to 1

beg 1001 end 11000 thin 1

Figure 5.11: *It's also useful to monitor the* $\mu_i = \alpha + \beta(x_i - \bar{x})$*, because they represent an important part of the shrinkage story with model (5.10).*

via the shrinkage equation

$$E(\theta_i|y) \doteq \left(1 - \hat{B}_i\right) y_i + \hat{B}_i E(\mu_i|y)\,; \tag{5.11}$$

here $\hat{B}_i = \frac{V_i}{V_i + \hat{\sigma}_\theta^2}$ and $\hat{\sigma}_\theta^2$ is the posterior mean of σ_θ^2.

This is easy to check in R:

```
> mu <- c( 0.09231, -0.06898, -0.06898, 0.4149, 0.4149, -0.06898,
    -0.06898, -0.06898, 0.4149, 0.2536, 0.4149, 0.4149, 0.2536,
    0.09231, -0.06898, -0.06898, 0.2536, 0.09231, -0.06898 )

> y <- c( 0.03, 0.12, -0.14, 1.18, 0.26, -0.06, -0.02, -0.32, 0.27,
    0.80, 0.54, 0.18, -0.02, 0.23, -0.18, -0.06, 0.30, 0.07,
    -0.07 )

> theta <- c( 0.08144, -0.03455, -0.07456, 0.4377, 0.4076,
    -0.0628, -0.05262, -0.08468, 0.3934, 0.289, 0.4196, 0.3938,
    0.2393, 0.1014, -0.08049, -0.06335, 0.2608, 0.08756,
    -0.06477 )

> V <- 1 / tau

> B.hat <- V / ( V + 0.064^2 )

> theta.approx <- ( 1 - B.hat ) * y + B.hat * mu

> cbind( y, theta, mu, sigma.2, B.hat, theta.approx )

          y    theta       mu        V     B.hat theta.approx
 [1,]  0.03  0.08144  0.09231 0.015625 0.7923026   0.07936838
 [2,]  0.12 -0.03455 -0.06898 0.021609 0.8406536  -0.03886671
 [3,] -0.14 -0.07456 -0.06898 0.027889 0.8719400  -0.07807482
 [4,]  1.18  0.43770  0.41490 0.139129 0.9714016   0.43678060
 [5,]  0.26  0.40760  0.41490 0.136161 0.9707965   0.41037637
 [6,] -0.06 -0.06280 -0.06898 0.010609 0.7214553  -0.06647867
 [7,] -0.02 -0.05262 -0.06898 0.010609 0.7214553  -0.05533688
 [8,] -0.32 -0.08468 -0.06898 0.048400 0.9219750  -0.08856583
 [9,]  0.27  0.39340  0.41490 0.026896 0.8678369   0.39574956
[10,]  0.80  0.28900  0.25360 0.063001 0.9389541   0.28695551
[11,]  0.54  0.41960  0.41490 0.091204 0.9570199   0.42027681
```

```
[12,]  0.18  0.39380  0.41490 0.049729 0.9239015   0.39702447
[13,] -0.02  0.23930  0.25360 0.083521 0.9532511   0.24080950
[14,]  0.23  0.10140  0.09231 0.084100 0.9535580   0.09870460
[15,] -0.18 -0.08049 -0.06898 0.025281 0.8605712  -0.08445939
[16,] -0.06 -0.06335 -0.06898 0.027889 0.8719400  -0.06783002
[17,]  0.30  0.26080  0.25360 0.019321 0.8250843   0.26171609
[18,]  0.07  0.08756  0.09231 0.008836 0.6832663   0.08524367
[19,] -0.07 -0.06477 -0.06898 0.030276 0.8808332  -0.06910155
```

You can see that equation (11) is indeed a good approximation to what's going on: the posterior means of the θ_i (column 3 of this table, counting the leftmost column of study indices) all fall between the y_i (column 2) and the posterior means of the μ_i (column 4), with the closeness to y_i or $E(\mu_i|y)$ expressed through the shrinkage factor $\hat{B}_i$.

Since $\hat{\sigma}_\theta^2$ is small (i.e., most—but not quite all—of the between-study variation has been explained by the covariate x), the raw y_i values are shrunken almost all of the way toward the regression line $\alpha + \beta(x_i - \bar{x})$.

5.2 Problems

1. (hierarchical random-effects modeling) Continuing problem 1 from Chapter 4, consider just the data from the residential streets with a bike route, and let w_i be the number of bicycles observed out of n_i total vehicles in block $i = 1, \ldots, n_w = 10$ (e.g., $w_1 = 16$ and $n_1 = 74$; we will regard the n_i as fixed known constants in this problem). Then by the nature of the sampling w_i should be binomially distributed with sample size n_i and success probability θ_i, the true underlying PBT in block i and other blocks similar to it. The 10 blocks were themselves chosen randomly (exchangeably) from an underlying population of blocks on residential streets with a bike route, so it's natural to model the θ_i themselves as like draws from a population distribution, which I argued above should plausibly be Beta(α, β) for some values of the hyperparameters (α, β). Placing uniform $U(0, c_\alpha)$ and $U(0, c_\beta)$ priors on α and β as in part (e) yields the following hierarchical random-effects model for the w_i:

$$\begin{array}{rcl} (w_i|\theta_i) & \overset{\text{IID}}{\sim} & \text{Binomial}\,(n_i, \theta_i) \\ (\theta_i|\alpha, \beta) & \overset{\text{IID}}{\sim} & \text{Beta}(\alpha, \beta) \end{array} \qquad (5.12)$$

$$\alpha \sim U(0, c_\alpha) \quad \text{and} \quad \beta \sim U(0, c_\beta).$$

Use WinBUGS to fit this model to the data from the residential streets with a bike route (as in (e) I got good results with $(c_\alpha, c_\beta) = (15, 60)$). Monitor α, β, the underlying population mean $\frac{\alpha}{\alpha+\beta}$, and all 10 of the θ_i values in your Gibbs sampling. Make a full report using CODA or the diagnostics built into WinBUGS on what you think is an appropriate MCMC strategy to get accurate posterior summaries for all of these quantities, and report posterior means, SDs, and 95% central intervals for all of them. Make a table with 10 rows (one for each block) and the following columns: the raw PBT estimates $\hat{p}_i = \frac{w_i}{n_i}$ (these were earlier called the y_i values in parts (a) and (c)); the *shrunken* estimates $\hat{\theta}_i$ (the posterior means of the θ_i); the *shrinkage factors* $\hat{B}_i$ (obtained by solving the linear equations $(1 - \hat{B}_i)\hat{p}_i + \hat{B}_i\hat{\theta} = \hat{\theta}_i$, where $\hat{\theta}$ is the posterior mean of $\frac{\alpha}{\alpha+\beta}$); and the sample sizes n_i. What relationship do you notice between the shrinkage factors and the sample sizes, and does this relationship make good intuitive sense? Explain briefly, and summarize what all of this says about the proportion of bicycle traffic on residential streets with a bike route in Berkeley in the early 1990s.

Chapter 6

Bayesian nonparametric modeling

6.1 de Finetti's representation theorem revisited

6.2 The Dirichlet process

6.3 Dirichlet process mixture modeling

6.4 Pólya trees

6.5 Problems

References

Aitkin M (1991). Posterior Bayes factors (with discussion). *Journal of the Royal Statistical Society, Series B*, **53**, 111–142.

Bayarri MJ, Berger JO (1998). Quantifying surprise in the data and model verification (with discussion). In *Bayesian Statistics 6*, Bernardo JM, Berger JO, Dawid AP, Smith AFM (editors). Oxford University Press, 53–82.

Bayes T (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, **53**, 370–418.

Berger JO, Pericchi LR (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, **91**, 109–122.

Bernardo JM, Smith AFM (1994). *Bayesian Theory.* New York: Wiley.

Best NG, Cowles MK, Vines SK (1995). *CODA Manual version 0.30.* MRC Biostatistics Unit, Cambridge, UK.

Brooks SP, Draper D (2005). Comparing the efficiency of MCMC samplers. In preparation.

Bryk AS, Raudenbush SW (1992). *Hierarchical Linear Models: Applications and Data Analysis Methods.* London: Sage.

Carlin BP, Louis TA (1996). *Bayes and Empirical Bayes Methods for Data Analysis.* London: Chapman & Hall.

Cowles MK, Carlin BP (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, **91**, 883–904.

Craig PS, Goldstein M, Seheult AH, Smith JA (1997). Constructing partial prior specifications for models of complex physical systems. *The Statistician*, **46**.

Dey D, Mueller P, Sinha D (1998). *Practical Nonparametric and Semiparametric Bayesian Statistics*. New York: Springer Verlag (Lecture Notes in Statistics, Volume 133).

Draper D (1995a). Assessment and propagation of model uncertainty (with discussion). *Journal of the Royal Statistical Society Series B*, **57**, 45–97.

Draper D (1995b). Inference and hierarchical modeling in the social sciences (with discussion). *Journal of Educational and Behavioral Statistics*, **20**, 115–147, 233–239.

Draper D (1996). Utility, sensitivity analysis, and cross-validation in Bayesian model-checking. Comment on "Posterior predictive assessment of model fitness via realized discrepancies," by Gelman A, Meng XL, Stern H, *Statistica Sinica*, **6**, 760–767.

Draper D (2005). On the relationship between model uncertainty and inferential/predictive uncertainty. Submitted.

Draper D, Fouskakis D (2000). A case study of stochastic optimization in health policy: problem formulation and preliminary results. *Journal of Global Optimization*, **18**, 399–416.

Draper D, Fouskakis D (2005). Stochastic optimization methods for cost-effective quality assessment in health. Submitted.

Draper D, Gaver D, Goel P, Greenhouse J, Hedges L, Morris C, Tucker J, Waternaux C (1993a). *Combining Information: Statistical Issues and Opportunities for Research*. Contemporary Statistics Series, No. 1. American Statistical Association, Alexandria VA.

Draper D, Hodges JS, Mallows CL, Pregibon D (1993). Exchangeability and data analysis (with discussion). *Journal of the Royal Statistical Society, Series A*, **156**, 9–37.

Draper D, Krnjajić M (2005). Three-way cross-validation for well-calibrated model exploration. In preparation.

de Finetti B (1930). Funzione caratteristica de un fenomeno aleatorio. *Mem. Acad. Naz. Lincei*, **4**, 86–133.

de Finetti B (1937). La prévision: ses lois logiques, ses sources subjectives. *Ann. Inst. H. Poincaré*, **7**, 1–68.

de Finetti B (1938). Sur la condition d'equivalence partielle. *Actualités Scientifiques et Industrielles*, **739**.

de Finetti B (1964). Foresight: its logical laws, its subjective sources. In *Studies in Subjective Probability*, HE Kyburg, Jr., HE Smokler, eds., New York: Wiley (1980), 93–158.

de Finetti B (1974/5). *Theory of Probability*, **1–2**. New York: Wiley.

Fisher RA (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London A*, **222**, 309–368.

Fisher RA (1935). *The Design of Experiments*. London: Oliver and Boyd.

Fisher RA (1956). *Statistical Methods and Scientific Inference.* London: Oliver and Boyd.

Fouskakis D, Draper D (2002). Stochastic optimization: a review. *International Statistical Review*, **70**, 315–349.

Freedman D, Pisani R, Purves R (1998). *Statistics*, third edition. New York: Norton.

Geisser S, Eddy WF (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, **74**, 153–160.

Gelfand AE, Dey DK, Chang H (1992). Model determination using predictive distributions, with implementation via sampling-based methods (with discussion). In *Bayesian Statistics 4* (Bernardo JM, Berger JO, Dawid AP, Smith AFM, editors), Oxford: Oxford University Press, 147–167.

Gelfand AE, Smith AFM (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85**, 398–409.

Gelman A, Carlin JB, Stern HS, Rubin DB (2003). *Bayesian Data Analysis*, second edition. London: Chapman & Hall.

Gelman A, Meng X-L, Stern H (1996). Posterior predictive assessment of model fitness via realized discrepancies (with discussion). *Statistica Sinica*, **6**, 733–760.

Gelman A, Rubin DB (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, **7**, 457–472.

Geweke J (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In *Bayesian Statistics 4*, JM Bernardo, JO Berger, AP Dawid, AFM Smith (eds.). Oxford: Clarendon Press.

Gilks WR, Clayton DG, Spiegelhalter DJ, Best NG, McNeil AJ, Sharples LD, Kirby AJ (1993). Modeling complexity: Applications of Gibbs sampling in medicine. *Journal of the Royal Statistical Society, Series B*, **55**, 39–52.

Gilks WR, Richardson S, Spiegelhalter DJ (eds.) (1995). *Markov Chain Monte Carlo in Practice.* London: Chapman & Hall.

Gilks WR, Wild P (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, **41**, 337–348.

Good IJ (1950). *Probability and the Weighing of Evidence.* London: Griffin.

Hacking I (1975). *The Emergence of Probability.* Cambridge: Cambridge University Press.

Hedges LV, Olkin I (1985). *Statistical Methods for Meta-Analysis.* New York: Academic Press.

Heidelberger P, Welch P (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, **31**, 1109–1144.

Hendriksen C, Lund E, Stromgard E (1984). Consequences of assessment and intervention among elderly people: a three year randomised controlled trial. *British Medical Journal*, **289**, 1522–1524.

Jeffreys H (1961). *Theory of Probability* (Third Edition). Oxford: Clarendon Press.

Johnson NL, Kotz S (1970). *Distributions in statistics: Continuous univariate distributions*, **1**. New York: Wiley.

Kadane JB, Dickey JM, Winkler RL, Smith WS, Peters SC (1980). Interactive elicitation of opinion for a normal linear model. *Journal of the American Statistical Association*, **75**, 845–854.

Kadane JB, Wolfson LJ (1997). Experiences in elicitation. *The Statistician*, **46**, forthcoming.

Kahn K, Rubenstein L, Draper D, Kosecoff J, Rogers W, Keeler E, Brook R (1990). The effects of the DRG-based Prospective Payment System on quality of care for hospitalized Medicare patients: An introduction to the series. *Journal of the American Medical Association*, **264**, 1953–1955 (with editorial comment, 1995–1997).

Kass RE, Raftery AE (1995). Bayes factors. *Journal of the American Statistical Association*, **90**, 773–795.

Key JT, Pericchi LR, Smith AFM (1998). Bayesian model choice: what and why? (with discussion). In *Bayesian Statistics 6*, Bernardo JM, Berger JO, Dawid AP, Smith AFM (editors). Oxford University Press, 343–370.

Laplace PS (1774). Mémoire sur la probabilité des causes par les évenements. *Mémoires de l'Academie de Science de Paris*, **6**, 621–656. English translation in 1986 as "Memoir on the probability of the causes of events," with an introduction by SM Stigler, *Statistical Science*, **1**, 359–378.

Leamer EE (1978). *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. New York: Wiley.

Lindley DV (1982). The Bayesian approach to statistics. In *Some Recent Advances in Statistics*, Tiago de Olivera J (editor). London: Academic Press, 65–87.

McCulloch RE, Rossi PE (1992). Bayes factors for nonlinear hypotheses and likelihood distributions. *Biometrika*, **79**, 663–676.

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1092.

Morris CN (1983). Parametric empirical Bayes inference: Theory and applications. *Journal of the American Statistical Association*, 78, 47–59.

Morris CN (1988). Determining the accuracy of Bayesian empirical Bayes estimators in the familiar exponential families. In *Proceedings of the Fourth Purdue Symposium on Statistical Decision Theory and Related Topics IV, part 1.*, SS Gupta, JO Berger, eds. New York: Springer-Verlag, 251–263.

von Neumann J, Morgenstern O (1944). *Theory of Games and Economic Behavior*. Princeton, NJ: Princeton University Press.

Neyman J (1923). [confidence intervals first proposed]

Oakes M (1990). *Statistical Inference*. Chestnut Hill, MA: Epidemiology Resources.

O'Hagan A (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society, Series B*, **57**, 99–138.

O'Hagan A (1997). Eliciting expert beliefs in substantial practical applications. *The Statistician*, **46**, forthcoming.

O'Hagan A, Forster J (2004). *Bayesian Inference*, second edition. London: Arnold.

Pericchi L (2004). Model selection and hypothesis testing based on objective probabilities and Bayes factors. Manuscript.

Raftery AL, Lewis S (1992). How many iterations in the Gibbs sampler? In *Bayesian Statistics 4*, JM Bernardo, JO Berger, AP Dawid, AFM Smith (eds.). Oxford: Clarendon Press, 763–774.

Ramsay FP (1926). Truth and probability. In *The Foundations of Mathematics and Other Logical Essays*, RB Braithwaite, ed., London: Kegan Paul, 156–198.

Raudenbush SW (1984). Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 19 experiments. *Journal of Educational Psychology*, 76, 85–97.

Robins JM, van der Vaart A, Ventura V (2000). Asymptotic distribution of P values in composite null models. *Journal of the American Statistical Association*, **95**, 1143–1156.

Samaniego FJ, Reneau DM (1994). Toward a reconciliation of the Bayesian and frequentist approaches to point estimation. *Journal of the American Statistical Association*, **89**, 947–957.

Savage LJ (1954). *The Foundations of Statistics.* New York: Wiley.

Spiegelhalter DJ, Best NG, Carlin BR, van der Linde A (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society Series B*, **64**, 583–616.

Spiegelhalter DJ, Smith AFM (1982). Bayes factors for linear and log-linear models with vague prior information. *Journal of the Royal Statistical Society, Series B*, **44**, 377–387.

Stigler SM (1986). Laplace's 1774 memoir on inverse probability. *Statistical Science*, **1**, 359–378 (contains an English translation of Laplace, 1774).

Tierney L, Kadane JB (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, **81**, 82–86.

Walker S, Damien P, Lenk P (2004). On priors with a Kullback-Leibler property. *Journal of the American Statistical Association*, **99**, 404–408.

Index