A Student's Guide to Bayesian Statistics: problems and answers

Ben Lambert

March 6, 2019

Chapter 1

How best to use this book

There are no problems for Chapter 1.

Chapter 2

The subjective worlds of Frequentist and Bayesian statistics

2.1 The deterministic nature of random coin throwing

Suppose that, in an idealised world, the ultimate fate of a thrown coin heads or tails is deterministically given by the angle at which you throw the coin and its height above a table. Also in this ideal world, the heights and angles are discrete. However, the system is chaotic (highly sensitive to initial conditions), and the results of throwing a coin at a given angle and height are shown in Table 2.1.

	Height above table (m)				
Angle (degrees)	**0.2**	**0.4**	**0.6**	**0.8**	**1**
0	T	H	T	T	H
45	H	T	T	T	T
90	H	H	T	T	H
135	H	H	T	H	T
180	H	H	T	H	H
225	H	T	H	T	T
270	H	T	T	T	H
315	T	H	H	T	T

Table 2.1: The results of a coin throw from a given angle and height above a table.

Problem 2.1.1. Suppose that all combinations of angles and heights are equally likely to be chosen. What is the probability that the coin lands heads up?

To do this we count the occurrence of heads and tails in Table 2.1, and find that $Pr(H) = \frac{19}{40}$.

Problem 2.1.2. Now suppose that some combinations of angles and heights are more likely to be chosen than others, with the probabilities shown in Table 2.2. What are the new probabilities that the coin lands heads up?

Angle	0.2	0.4	0.6	0.8	1
0	0.05	0.03	0.02	0.04	0.04
45	0.03	0.02	0.01	0.05	0.02
90	0.05	0.03	0.01	0.03	0.02
135	0.02	0.03	0.04	0.00	0.04
180	0.03	0.02	0.02	0.00	0.03
225	0.00	0.01	0.04	0.03	0.02
270	0.03	0.00	0.03	0.01	0.04
315	0.02	0.03	0.03	0.02	0.01

Table 2.2: The probability that a given person throws a coin at a particular angle, and at a certain height above a table.

We must now find a weighted average of the coin flip outcomes where the weights are provided by the values in Table 2.2. If we do so we find that $Pr(H) = 0.5$.

Problem 2.1.3. We force the coin-thrower to throw the coin at an angle of 45 degrees. What is the probability that the coin lands heads up?

We must now find a weighted average of the coin flip outcomes *given* that we are constrained to be in the row corresponding to 45 degrees. If we do so we find that the $Pr(H) \approx 0.23$.

Problem 2.1.4. We force the coin-thrower to throw the coin at a height of 0.2m. What is the probability that the coin lands heads up?

Similarly to the previous question we now constrain ourselves to be in the relevant column. Now we obtain $Pr(H) \approx 0.70$.

Problem 2.1.5. If we constrained the angle and height to be fixed, what would happen in repetitions of the same experiment?

The coin would always land the same way up.

Problem 2.1.6. In light of the previous question, comment on the Frequentist assumption of exact repetitions of a given experiment.

We cannot have *exact* repetition because if we did so we would always get the same result! We need enough variation in the throwing method to allow different outcomes but not too much variation. Where do we draw the line?

2.2 Objections to Bayesianism

The following criticisms of Bayesian statistics are raised in an article by Gelman [6]. Provide a response to each of these.

Problem 2.2.1. 'As scientists we should be concerned with objective knowledge rather than subjective belief.'

As we argue in this chapter *all* analyses are associated with a degree of subjective knowledge. At least with Bayesian inference we are required to explicitly state one aspect of the analysis - the priors - that represent our pre-data experimental beliefs in a particular parameter set. This transparency is desirable and it is the job of the analyst to report if inferences are sensitive to a particular choice of prior. Also, (as Gelman himself indicates) priors can be highly informed from previous data not *only* from inherent subjective beliefs.

Problem 2.2.2. 'Subjective prior distributions don't transfer well from person to person.'

This depends on two things: the degree to which priors are informed by previous data; and the variation in beliefs between people. If there is a paucity of data and little consensus over the state of nature then, yes, there will be significant variation in priors between people. However even though there may be variance in priors this does not necessarily imply variation in the posteriors. In fact, the more data we collect (in general) the less sensitive our inferences become to prior choices.

Problem 2.2.3. 'There's no good objective principle for choosing a noninformative prior Where do prior distributions come from, anyway?'

I like Gelman's response here: there is no objective method for choosing a likelihood!

Problem 2.2.4. A student in a class of mine: 'If we have prior expectations of a donkey and our dataset is a horse then Bayesians estimate a mule.'

This presupposes that a mule is undesirable because of our lack of belief in it. If we were that certain that a mule was impossible then we could address this by setting it a zero prior beforehand. Also this is really a question about the validity of point estimates versus those that contain a measure of uncertainty. If we simply give a point estimate (perhaps the posterior mean) then it may be the case that we get a mule. However our uncertainty interval will no doubt contain both a horse and a donkey. If we were predisposed to want a horse or a donkey then we could choose an estimator (or likelihood) that reflects this predisposition.

Problem 2.2.5. 'Bayesian methods seem to quickly move to elaborate computation.'

All modern statistical methods make extensive use of computation. Perhaps peoples' complaint with this method is that the time required for an applied analysis with Bayesian statistics is non-deterministic.

2.3 Model choice

Suppose that you have been given the data contained in `subjective_overfitShort.csv` and are asked to find a 'good' statistical model to fit the (x, y) data.

Problem 2.3.1. Fit a linear regression model using least squares. How reasonable is the fit?

See Figure 2.1. Not a great fit but given the paucity of data we probably can't do much better.

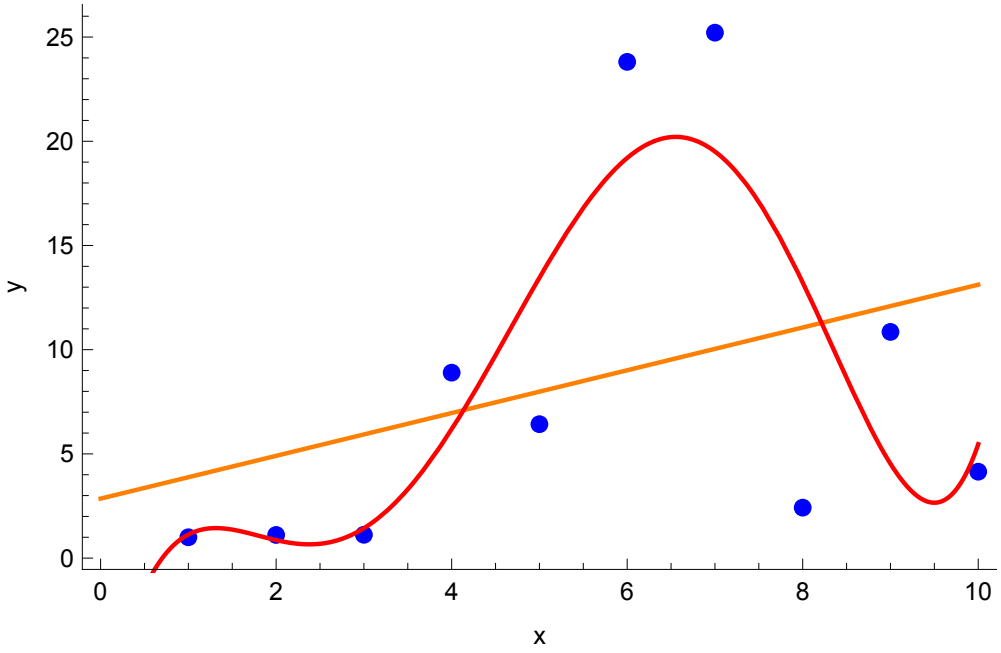

Figure 2.1: The data (blue) versus a linear regression line (orange) and a quintic regression line (red).

Problem 2.3.2. Fit a quintic (powers up to the fifth) model to the data. How does its fit compare to that of the linear model?

See Figure 2.1. Fitting the data more closely, but is almost certainly overfitting (see next parts of the question).

Problem 2.3.3. You are now given new data contained within `subjective_overfitLong.csv`. This contains data on 1000 replications of the same experiment, where the x values are held fixed. Using the least squares fits from the first part of this question, compare the performance of the linear regression model with that of the quintic model.

Computing the mean squared residual for each case we have that for each of the fits,

1. Simple: $\overline{RMSE} \approx 11.8$.
2. Complex: $\overline{RMSE} \approx 13.5$.

And so the simple model has a greater predictive accuracy for out of sample data.

Problem 2.3.4. Which of the two models do you prefer, and why?

The simple one! The complex model fits the noise not the signal. It is overfit.

Chapter 3

Probability - the nuts and bolts of Bayesian inference

3.1 Messy probability density

Suppose that a probability density is given by the following function:

$$f(X) = \begin{cases} 1, & \text{if } 0 \leq X < 0.5 \\ 0.2, & \text{if } 0.5 \leq X < 1 \\ 0.8(X-1), & \text{if } 1 \leq X < 2 \\ 0, \text{otherwise} & \end{cases}$$

Problem 3.1.1. Demonstrate that the above density is a valid probability distribution.

This can be done by summing the areas from each of the three parts,

$$\text{area} = 1 \times 0.5 + 0.2 \times 0.5 + 0.5 \times 1 \times 0.8 = 1 \tag{3.1}$$

Problem 3.1.2. What is the probability that $0.2 \leq X \leq 0.5$?

The relevant area is,

$$Pr(0.2 \leq X \leq 0.5) = 0.3 \tag{3.2}$$

Problem 3.1.3. Find the mean of the distribution.

To do this we need to integrate,

$$\mathbb{E}(X) = \int_0^2 x f(x) \mathrm{d}x \tag{3.3}$$
$$= \int_0^{0.5} x \mathrm{d}x + \int_{0.5}^1 0.2x \mathrm{d}x + \int_1^2 (x-1)0.8x \mathrm{d}x \tag{3.4}$$
$$= [0.5x^2]_0^{0.5} + [0.1x^2]_{0.5}^1 + 0.8[\frac{1}{3}x^3 - \frac{1}{2}x^2]_1^2 \tag{3.5}$$
$$= 0.5 \times 0.5^2 + 0.1(1 - 0.5^2) + 0.8(\frac{8}{3} - 2 - \frac{1}{3} + \frac{1}{2}) \tag{3.6}$$
$$\approx 0.867 \tag{3.7}$$

Problem 3.1.4. What is the median of the distribution?

This is the point at which $Pr(X < a) = 0.5$, which happens to occur at $a = 0.5$.

3.2 Keeping it discrete

Suppose that the number of heads obtained X in a series of N coin flips is described by a binomial distribution:

$$Pr(X = K|\theta) = \binom{N}{K} \theta^K (1-\theta)^{N-K}, \tag{3.8}$$

where $\binom{N}{K} = \frac{N!}{K!(N-K)!}$ is the binomial coefficient and θ is the probability of obtaining a heads on any particular throw.

Problem 3.2.1. Suppose that $\theta = 0.5$ (that is, the coin is fair). Calculate the probability of obtaining 5 heads in 10 throws.

This is given by,

$$Pr(X = 5|\theta = 0.5) = \binom{10}{5} 0.5^5 0.5^5 \approx 0.246 \tag{3.9}$$

Problem 3.2.2. Calculate the probability of obtaining fewer than 3 heads.

This is given by,

$$Pr(X < 3|\theta = 0.5) = Pr(X = 0|\theta = 0.5) + Pr(X = 1|\theta = 0.5) + Pr(X = 2|\theta = 0.5) \tag{3.10}$$
$$= 0.5^{10} + 10 \times 0.5^{10} + 45 \times 0.5^{10} \tag{3.11}$$
$$\approx 0.055 \tag{3.12}$$

Problem 3.2.3. Find the mean of this distribution. (You can either derive the mean of this distribution or take it as given that $\mathbb{E}(X) = N\theta$.)

The mean here is given by $\mathbb{E}(X) = N\theta = 10 \times 0.5 = 5$.

Problem 3.2.4. Suppose I flip another coin with $\theta = 0.2$. What is the probability that I get more than 8 heads?

$$\begin{aligned} Pr(X_2 > 8|\theta = 0.2) &= Pr(X = 9|\theta = 0.2) + Pr(X = 10|\theta = 0.2) && (3.13)\\ &= 10 \times 0.2^9 0.8^1 + 0.2^{10} && (3.14)\\ &\approx 4 \times 10^{-6} && (3.15) \end{aligned}$$

Problem 3.2.5. What is the probability that I obtain fewer than 3 heads in 10 flips of the first coin, and more than 8 heads with the second?

The two events are independent and so,

$$\begin{aligned} Pr(X_1 < 3, X_2 > 8|\theta_1 = 0.5, \theta_2 = 0.2) &= Pr(X_1 < 3|\theta_1 = 0.5) \times Pr(X_2 > 8|\theta_2 = 0.2) && (3.16)\\ &\approx 0.055 \times 4 \times 10^{-6} && (3.17)\\ &\approx 2 \times 10^{-7} && (3.18) \end{aligned}$$

3.3 Continuously confusing

Suppose that the time that elapses before a particular component on the Space Shuttle fails can be modelled as being exponentially distributed:

$$p(t|\lambda) = \lambda e^{-\lambda t}, \tag{3.19}$$

where $\lambda > 0$ is a rate parameter.

Problem 3.3.1. Show that the above distribution is a valid probability density.

To do this we integrate,

$$\begin{aligned} \int_0^\infty \lambda e^{-\lambda t} \mathrm{d}t &= [-e^{-\lambda t}]_0^\infty && (3.20)\\ &= 1 && (3.21) \end{aligned}$$

Problem 3.3.2. Find the mean of this distribution.

To do this we integrate (using integration by parts),

$$\int_0^\infty \lambda t e^{-\lambda t} \mathrm{d}t = \frac{1}{\lambda} \tag{3.22}$$

Problem 3.3.3. Suppose that $\lambda = 0.2$ per hour. Find the probability that the component fails in the first hour of flight.

To do this we integrate,

$$\begin{aligned} Pr(0 \leq t \leq 1 | \lambda = 0.2) &= [-e^{-0.2t}]_0^1 && (3.23)\\ &= 1 - e^{-0.2} && (3.24)\\ &\approx 0.18 && (3.25) \end{aligned}$$

Problem 3.3.4. What is the probability that the component survives for the first hour but fails during the second?

To do this we integrate,

$$\begin{aligned} Pr(1 \leq t \leq 2 | \lambda = 0.2) &= [-e^{-0.2t}]_1^2 && (3.26)\\ &= e^{-0.2} - e^{-0.4} && (3.27)\\ &\approx 0.148 && (3.28) \end{aligned}$$

Problem 3.3.5. What is the probability that the component fails during the second hour given that it has survived the first?

This is a conditional probability,

$$\begin{aligned} Pr(1 \leq t \leq 2 | t \geq 1, \lambda = 0.2) &= \frac{Pr(1 \leq t \leq 2 | \lambda = 0.2)}{Pr(t \geq 1 | \lambda = 0.2)} && (3.29)\\ &= \frac{0.148}{0.82} && (3.30)\\ &\approx 0.18 && (3.31) \end{aligned}$$

We could have obtained this from the memoryless property of the exponential distribution (see next question).

Problem 3.3.6. Show that the probability of the component failing during the $(n+1)$th hour given that it has survived n hours is always 0.18.

This is a conditional probability,

$$Pr(n \leq t \leq n+1 | t \geq n, \lambda = 0.2) = \frac{Pr(n \leq t \leq n+1 | \lambda = 0.2)}{Pr(t \geq n | \lambda = 0.2)} \tag{3.32}$$

$$= \frac{[-e^{-0.2t}]_n^{n+1}}{1 - [-e^{-0.2t}]_0^n} \tag{3.33}$$

$$= \frac{e^{-0.2n} - e^{-0.2(n+1)}}{e^{-0.2n}} \tag{3.34}$$

$$= 1 - e^{-0.2} \tag{3.35}$$

$$\approx 0.18, \tag{3.36}$$

and so we have demonstrated the memoryless property of the exponential distribution.

3.4 The boy or girl paradox

The boy or girl paradox was first introduced by Martin Gardner in 1959. Suppose we are told the following information:

Problem 3.4.1. Mr Bayes has two children. The older child is a girl. What is the probability that both children are girls?

There are two potentialities that include the older child being a girl: boy-girl or girl-girl. Hence the probability that both children are girls is $\frac{1}{2}$.

Problem 3.4.2. Mr Laplace has two children. At least one of the children is a girl. What is the probability that both children are girls?

Here there are three potentialities: boy-girl, girl-boy or girl-girl. This means that there is a $\frac{1}{3}$ probability that both children are girls. Note that there has been some controversy over the asking of this question. See `https://en.wikipedia.org/wiki/Boy_or_Girl_paradox`.

3.5 Planet Scrabble

On a far-away planet suppose that people's names are always two letters long, with each of these letters coming from the 26 letters of the Latin alphabet. Suppose that there are no constraints on individuals' names, so they can be composed of two identical letters, and there is no need to include a consonant or a vowel.

Problem 3.5.1. How many people would need to be gathered in one place for there to be a 50% probability that at least two of them share the same name?

There are $26 \times 26 = 676$ possible names out there. Let's start with four people, and work out the probability that two of them do *not* share the same name, i.e. $X = 0$, where X is the number of occurrences of people with shared names in the group.

$$Pr(X = 0) = \frac{676}{676} \times \frac{675}{676} \times \frac{674}{676} \times \frac{673}{676} \tag{3.37}$$

$$= \left(\frac{1}{676}\right)^4 \times (676 \times 675 \times 674 \times 673) \tag{3.38}$$

$$= \left(\frac{1}{676}\right)^4 \times \frac{676!}{672!} \tag{3.39}$$

Or more generally we have that for n individuals,

$$Pr(X = 0) = \frac{{}_{676}\mathcal{P}_n}{676^n}, \tag{3.40}$$

where ${}_{676}\mathcal{P}_n = \frac{676!}{(676-n)!}$ is the permutation coefficient. The graph of this function is shown in Figure 3.1, where we see that if 31 or more people are gathered in a room there is a probability that exceeds 50% that two will share the same name.

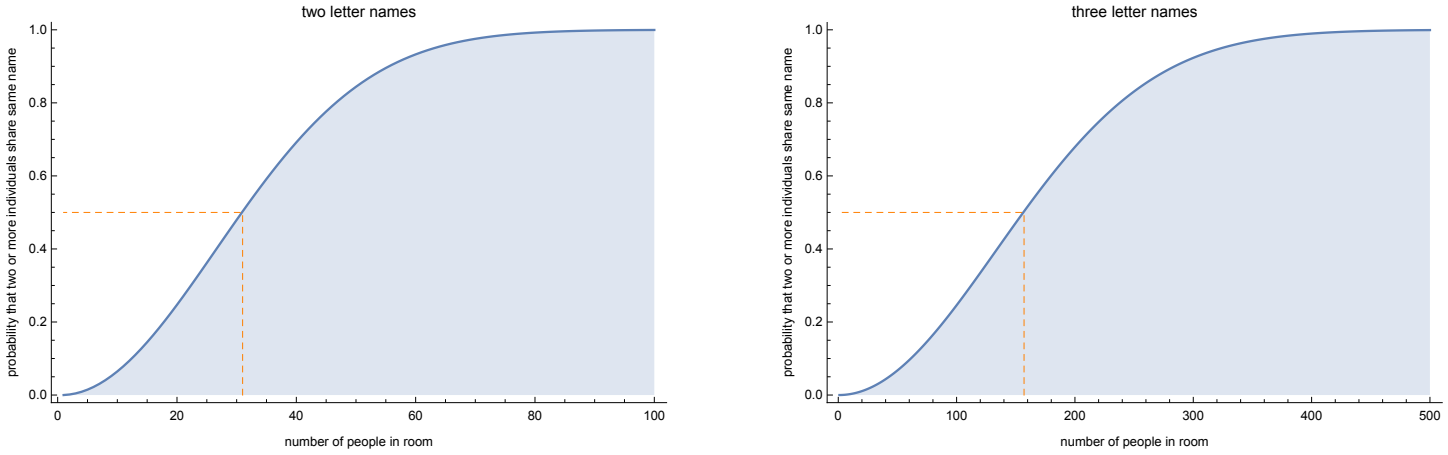

Figure 3.1: The probability that two or more people share the same name in a two- (left) and three- (right) letter name world.

Problem 3.5.2. Suppose instead that the names are composed of three letters. Now how many people would need to be gathered in one place for there to be a 50% probability that at least two of them share the same name?

The same analysis as above applies although now with $26 \times 26 \times 26 = 17,576$ possible names out there. In this case we find 157 is the minimum number of people required for there to be at least a 50% probability of two sharing the same name (see Figure 3.1).

3.6 Game theory

A game show presents contestants with four doors: behind one of the doors is a car worth \$1000; behind another is a forfeit whereby the contestant must pay \$1000 out of their winnings thus far on the show. Behind the other two doors there is nothing. The game is played as follows:

1. The contestant chooses one of four doors.
2. The game show host opens another door, always to reveal that there is nothing behind it.
3. The contestant is given the option of changing their choice to one of the two remaining unopened doors.
4. The contestant's final choice of door is opened, to their delight (a car!), dismay (a penalty), or indifference (nothing).

Assuming that:

- the contestant wants to maximise their expected wealth, and
- the contestant is risk-averse,

what is the optimal strategy for the contestant?

This question hinges on deriving the distribution of outcomes under either remaining, or changing, after the game show host has opened the empty door. This can be answered through application of Bayes' rule, but I prefer here to describe more intuitively what is happening.

Imagine we are considering repeating the show a number of times. There are three possibilities for the initial choice of door:

- $\frac{1}{4}$ of the time, the door hides the car.
- $\frac{1}{2}$ of the time, the door hides a null.
- $\frac{1}{4}$ of the time, the door hides the penalty.

Considering now each of these in turn:

If the door contains the car, then the other three doors are two nulls, and the penalty. The game show host opens one of the nulls, meaning that only one null and the penalty remain. In this circumstance, if you stay put, you definitely obtain the car. If you change, you get a null with probability $\frac{1}{2}$ and similarly for the penalty.

If the door contains a null, then the other three doors are one null, one penalty and one car. When the host opens the remaining null, then the other two doors are the car, and the penalty. This is the key step. By remaining, you gain/lose nothing, whereas if you change you face risk; you get the car with probability $\frac{1}{2}$ and similarly for the penalty. Both of these choices have the same expected payoff, but the latter increases risk.

Finally, if the door contains the penalty, then the other three doors are two nulls, and the car. The game show host opens one of the nulls, meaning that only one null and the car remain. In this circumstance, if you stay put, you definitely obtain the penalty. If you change, you get a null with probability $\frac{1}{2}$ and similarly for the car.

We can now write down probability distributions for the outcomes given each possible action. For remaining, the probabilities are what they were if you hadn't received any new information, in other words $(p(car), p(null), p(penalty)) = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$. Whereas if you change:

$$p(car) = \frac{1}{4} \times 0 + \frac{1}{2} \times \frac{1}{2} + \frac{1}{4} \times \frac{1}{2} = \frac{3}{8} \tag{3.41}$$

$$p(null) = \frac{1}{4}\frac{1}{2} + \frac{1}{2} \times 0 + \frac{1}{4} \times \frac{1}{2} = \frac{2}{8} \tag{3.42}$$

$$p(penalty) = \frac{1}{4} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2} + \frac{1}{4} \times 0 = \frac{3}{8} \tag{3.43}$$

So in summary, we get face $(p(car), p(null), p(penalty)) = (\frac{3}{8}, \frac{2}{8}, \frac{3}{8})$ by changing.

Both of these two outcomes face the same expected return of \$0:

$$\mathbb{E}[return|stay] = \frac{1}{4} \times \$1,000 + \frac{1}{2} \times \$0 + \frac{1}{4} \times -\$1,000 \tag{3.44}$$

$$= \$0 \tag{3.45}$$

$$\mathbb{E}[return|change] = \frac{3}{8} \times \$1,000 + \frac{2}{8} \times \$0 + \frac{3}{8} \times -\$1,000 \tag{3.46}$$

$$= \$0 \tag{3.47}$$

However, what is different is the variance in return from each decision. The variance in return is greater by changing than it is by staying, since the latter places more weight on the risky outcomes. Obviously, these can be calculated explicitly using the same methodology as above.

If the individual is risk-averse, then he prefers the less risky outcome of *remaining*, given that they both have the same return.

3.7 Blood doping in cyclists

Suppose, as a benign omniscient observer, we tally up the historical cases where professional cyclists either used or did not use blood doping, and either won or lost a particular race. This results in the probability distribution shown in Table 3.1.

Problem 3.7.1. What is the probability that a professional cyclist wins a race?

This is the marginal given by: $p(won) = p(won, dope) + p(won, clean) = 0.05 + 0.1 = 0.15$.

	Lost	**Won**
Clean	0.70	0.05
Doping	0.15	0.10

Table 3.1: The historical probabilities of behaviour and outcome for professional cyclists.

Problem 3.7.2. What is the probability that a cyclist wins a race, given that they have cheated?

$$p(won|doped) = \frac{p(doped, won)}{p(doped)} \tag{3.48}$$
$$= \frac{0.1}{0.25} = 0.4 \tag{3.49}$$

Problem 3.7.3. What is the probability that a cyclist is cheating, given that they win?

$$p(doped|won) = \frac{p(won|doped)p(doped)}{p(won)} \tag{3.50}$$
$$= \frac{0.4 \times 0.25}{0.15} \tag{3.51}$$
$$= \frac{2}{3} \tag{3.52}$$

Now suppose that drug testing officials have a test that can accurately identify a blood-doper 90% of the time. However, it incorrectly indicates a positive for clean athletes 5% of the time.

Problem 3.7.4. (Rephrased from book) If the officials want to maximize the proportion of people correctly identified as dopers (ie keep down the proportion of false positives), should they test all the athletes or only the winners?

Here we want to compare $p(doped|positive, group)$ across $group \in \{everyone, winners\}$. For everyone, this is simple and given by:

$$p(doped|positive) = \frac{p(positive|doped)p(doped)}{p(positive)} \tag{3.53}$$
$$= \frac{p(positive|doped)p(doped)}{p(positive, doped) + p(positive, clean)} \tag{3.54}$$
$$= \frac{p(positive|doped)p(doped)}{p(positive|doped)p(doped) + p(positive|clean)p(clean)} \tag{3.55}$$
$$= \frac{0.9 \times 0.25}{0.9 \times 0.25 + 0.05 \times 0.75} \tag{3.56}$$
$$\approx 0.86 \tag{3.57}$$

Whereas for the winners:

$$p(doped|positive, won) = \frac{p(doped, positive|won)}{p(positive|won)} \tag{3.58}$$

$$\tag{3.59}$$

We will proceed to calculate each of these bits in turn. Via Bayes' rule:

$$\begin{aligned} p(doped, positive|won) &= \frac{p(won|doped, positive)p(doped, positive)}{p(won)} && (3.60)\\ &= \frac{p(won|doped)p(doped, positive)}{p(won)} && (3.61)\\ &= \frac{p(won|doped)p(positive|doped)p(doped)}{p(won)} && (3.62)\\ &= \frac{0.4 \times 0.9 \times 0.25}{0.15} && (3.63)\\ &= 0.6 && (3.64) \end{aligned}$$

We have got the second line from the first by assuming that there is a conditional independence between winning and testing positive, once we account for their drug status. This is a fairly safe assumption, unless of course winners are more effective at hiding their drug use!

Now for the last bit:

$$\begin{aligned} p(positive|won) &= p(positive, doped|won) + p(positive, clean|won) && (3.65)\\ &= p(positive|doped, won)p(doped|won) + p(positive|clean, won)p(clean|won) && (3.66)\\ &= p(positive|doped)p(doped|won) + p(positive|clean)p(clean|won) && (3.67)\\ &= 0.9 \times \frac{2}{3} + 0.05 \times \frac{1}{3} && (3.68)\\ &\approx 0.62 && (3.69) \end{aligned}$$

Combining these two, we have $p(doped|positive, won) = \frac{0.6}{0.62} \approx 0.97$. Hence we should only test the winners. This makes intuitive sense, since they are a group which have a higher than average percentage of dopers.

As suggested by Martin Oldfield and Sean Jackson, in answering this question and the next parts, it's perhaps easiest to calculate the probabilities if all the outcomes are listed (see Table 3.2).

Problem 3.7.5. If the officials care five times as much about the number of people who are falsely identified as they do about the number of people who are correctly identified as dopers, should they test all the athletes or only the winners?

Race result	Drug status	Test	Probability
L	C	+	$\frac{14}{400}$
L	C	-	$\frac{266}{400}$
L	D	+	$\frac{54}{400}$
L	D	-	$\frac{6}{400}$
W	C	+	$\frac{1}{400}$
W	C	-	$\frac{19}{400}$
W	D	+	$\frac{36}{400}$
W	D	-	$\frac{4}{400}$

Table 3.2: The probabilities of the various outcomes in the cycling testing. Here 'L' and 'W' denote losers and winners; 'C' and 'D' denote clean athletes and dopers; '+' and '-' denote positive and negative test results respectively.

Note: this answer previously had an error in it which was pointed out in an email by Martin Oldfield and Sean Jackson.

Now we need to specify a utility function. The previous answer used $n(group)$ whereas what we want is $n(group, positive)$ since the questions states that we only care about the true positives ($p(D|+)$) and the false positives ($p(C|+)$).

$$\begin{aligned}
U(group) &= n(doped|positive, group) - 5n(clean|positive, group) && (3.70)\\
&= n(group, positive)\left[p(doped|positive, group) - 5p(clean|positive, group)\right] && (3.71)\\
&= n(group, positive)\left[p(doped|positive, group) - 5(1 - p(doped|positive, group))\right] && (3.72)\\
&= n(group, positive)\left[6p(doped|positive, group) - 5\right] && (3.73)
\end{aligned}$$

Calculating this for everyone, we have:

$$\begin{aligned}
U(total) &= n(total)p(positive)\left[6p(doped|positive) - 5\right] && (3.74)\\
&\approx n(total)p(positive)\left[6 \times 0.86 - 5\right] && (3.75)\\
&= 0.14 \times n(total) \times p(positive) && (3.76)\\
&= 0.14 \times 0.2625n(total) && (3.77)\\
&= 0.0375n(total) && (3.78)
\end{aligned}$$

For only the winners' group, we have:

$$\begin{aligned}
U(won) &= n(total) \times p(won, positive) \times \left[6p(doped|positive, won) - 5\right] && (3.79)\\
&\approx n(total) \times p(won, positive) \times \left[6 \times 0.97 - 5\right] && (3.80)\\
&\approx \frac{37}{400} \times 0.84n(total) && (3.81)\\
&= 0.0775n(total) && (3.82)
\end{aligned}$$

So in this case they should test just the winners.

Problem 3.7.6. What factor would make the officials choose the other group? (By factor, we mean the number 5 in the previous problem.)

Note: this answer previously had an error in it which was pointed out in an email by Martin Oldfield and Sean Jackson.

We can calculate $U(g)$ of a group g as a function of α; the factor:

$$U(g) = n(g)\left[(1+\alpha)p(D|+,g) - \alpha\right] \tag{3.83}$$

This means we can calculate:

$$\begin{aligned} U(total) &= n(total,+)\left[(1+\alpha)p(D|+) - \alpha\right] \\ &= n(total,+)\left[(1+\alpha)0.86 - \alpha\right] \\ &= n(total,+)\left[0.86 - 0.14\alpha\right] \\ &= n(total) \times p(+)\left[0.86 - 0.14\alpha\right] \\ &= n(total) \times \frac{105}{400}\left[0.86 - 0.14\alpha\right] \end{aligned}$$

And:

$$\begin{aligned} U(win) &= n(win,+)\left[(1+\alpha)p(D|+,W) - \alpha\right] \\ &= n(win,+)\left[(1+\alpha)0.97 - \alpha\right] \\ &= \frac{37}{400}n(total)\left[0.97 - 0.03\alpha\right] \end{aligned}$$

Now finding α such that $U(total) > U(win)$ we find, $\alpha < 4.00$.

3.8 Breast cancer revisited

Suppose that the prevalence of breast cancer for a randomly chosen 40-year-old woman in the UK population is about 1%. Further suppose that mammography has a relatively high sensitivity to breast cancer, where in 90% of cases the test shows a positive result if the individual has the disease. However, the test also has a rate of false positives of 8%.

Problem 3.8.1. Show that the probability that a woman tests positive is about 9%.

Here we need to determine $p(+)$, which is the marginal probability of testing positive. We get this by marginalising the joint probability, $p(+,C)$,

$$p(+) = \underbrace{p(+|C) \times p(C)}_{\text{positive and cancer}} + \underbrace{p(+|NC) \times p(NC)}_{\text{false positive}} \tag{3.84}$$

$$= 0.9 \times 0.01 + 0.08 \times 0.99 \tag{3.85}$$

$$= 8.8\% \tag{3.86}$$

Problem 3.8.2. A woman tests positive for breast cancer. What is the probability she has the disease?

Use Bayes' rule to find this quantity,

$$p(C|+) = \frac{p(+|C) \times p(C)}{p(+)} \tag{3.87}$$

$$= \frac{0.9 \times 0.01}{0.088} \tag{3.88}$$

$$\approx 10\% \tag{3.89}$$

Problem 3.8.3. Draw a graph of the probability of having a disease, given a positive test, as a function of (a) the test sensitivity (true positive rate) (b) the false positive rate, and (c) the disease prevalence. Draw graphs (a) and (b) for a rare (1% prevalence) and a common (10% prevalence) disease. What do these graphs imply about the relative importance of the various characteristics of medical tests?

We can write down the probability as we did before using Bayes' rule,

$$p(C|+) = \frac{p(+|C) \times p(C)}{p(+|C) \times p(C) + p(+|NC) \times (1 - p(C))} \tag{3.90}$$

which we can then graph across the three variables in this expression (fig. 3.2). For rare diseases this illustrates the weak dependence of the test on the true positive rate, and the strong dependence on the false positive rate. So for diseases that are rare, the most important thing is ensuring that the test has a low false positive rate. This makes intuitive sense, because for rare diseases the number of false positives quickly dwarfs the true positives. Whereas for more common diseases this factor is not so important; ensuring that the test sensitivity is high is a more pressing concern.

Of course the previous analysis does not give a utility to each outcome, so it is hard to make conclusions about the optimality of the test; Bayesian decision theory would be needed here to determine optimal testing parameters!

The plot of probability of testing positive for a disease versus disease prevalences illustrates that this medical test conveys most information for common diseases (15-30% prevalence). This is because there is the biggest gain in information between not having the test (black line) and post-test information (blue line).

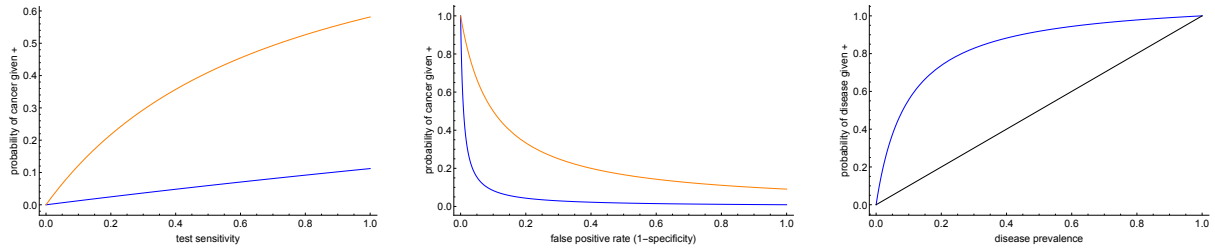

Figure 3.2: A plot of the probability of disease given a positive test result as a function of a. test sensitivity, b. the false positive rate for rare (blue) and common (orange) diseases, and c. the disease prevalence. For rare diseases we assume prevalence is 1%, and for common ones we assume 10% prevalence. For the right hand plot we assume a sensitivity of 90% and a specificity of 92%.

Problem 3.8.4. Assume the result of a mammography is independent when retesting an individual (probably a terrible assumption!). How many tests (assume a positive result in each) would need to be undertaken to ensure that the individual has a 99% probability that they have cancer?

Let's start by working out the probability of cancer for two positive test results,

$$p(C|++) = \frac{p(++|C) \times p(C)}{p(++)} \tag{3.91}$$

$$= \frac{p(+|C) \times p(+|C) \times p(C)}{p(++)} \tag{3.92}$$

where the marginal probability of two positive test results is similar to before,

$$p(++) = p(+|C) \times p(+|C) \times p(C) + p(+|NC) \times p(+|NC) \times p(NC) \tag{3.93}$$

$$= 0.9^2 \times 0.01 + 0.08^2 \times 0.99 \tag{3.94}$$

$$\approx 0.0144 \tag{3.95}$$

and so $p(C|++) \approx \frac{0.9^2 \times 0.01}{0.0144} = 56\%$. Using the above formulae we note that the probability for the case of n tests is given by,

$$p(C|+^n) = \frac{p(+^n|C) \times p(C)}{p(+^n)} \tag{3.96}$$

$$= \frac{p(+|C)^n \times p(C)}{p(+|C)^n \times p(C) + p(+|NC)^n \times p(NC)} \tag{3.97}$$

$$= \frac{0.9^n \times 0.01}{0.9^n \times 0.01 + 0.08^n \times 0.99} \tag{3.98}$$

If we graph this function we find that it is logistic-sigmoid shaped (Figure 3.3), and that after four tests we have reached the required threshold.

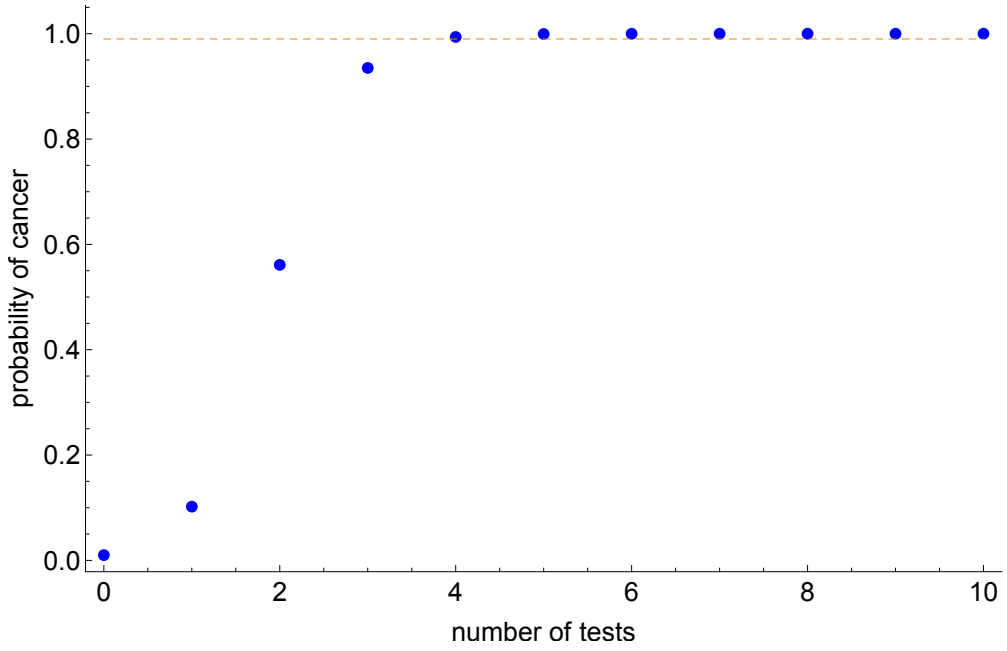

Figure 3.3: A plot of the probability of cancer as a function of the number of tests.

Problem 3.8.5. Now we make the more realistic assumption that the probability of testing positive in the nth trial depends on whether positive tests were achieved in the $(n+1)$th trials, for both individuals with cancer and those without. For a cancer status $\kappa \in \{C, NC\}$:

$$p(n+|(n-1)+,\kappa) = 1-(1-p(+|\kappa))e^{-(n-1)\epsilon} \tag{3.99}$$

where $n+$ denotes testing positive in the nth trial, $p(+|\kappa)$ and $\epsilon \geq 0$ determine the persistence in test results. Assume that $p(+|C) = 0.9$ and $p(+|NC) = 0.08$. For $\epsilon = 0.15$ show that we now need at least 17 positive test results to conclude with 99% probability that a patient has cancer.

We can determine the probability of n positive tests for a cancer status κ by multiplying together the individual conditional probabilities,

$$p(+^n|\kappa) = \prod_{m=1}^{n} p(m+|(m-1)+,\kappa) \tag{3.100}$$

$$= \prod_{m=1}^{n} \left[1-(1-p(+|\kappa))e^{-(m-1)\epsilon}\right] \tag{3.101}$$

We can then determine the probability that an individual has cancer given n positive test results,

$$p(C|+^n) = \frac{p(+^n|C) \times p(C)}{p(+^n|C) \times p(C) + p(+^n|NC) \times p(NC)} \tag{3.102}$$

$$= \frac{p(C) \prod_{m=1}^{n} \left[1 - (1 - p(+|C))e^{-(m-1)\epsilon}\right]}{p(C) \prod_{m=1}^{n} \left[1 - (1 - p(+|C))e^{-(m-1)\epsilon}\right] + p(NC) \prod_{m=1}^{n} \left[1 - (1 - p(+|NC))e^{-(m-1)\epsilon}\right]} \tag{3.103}$$

If we graph this function we see that it takes many more tests to reach the required threshold (Figure 3.4), which is obtained after 17 tests.

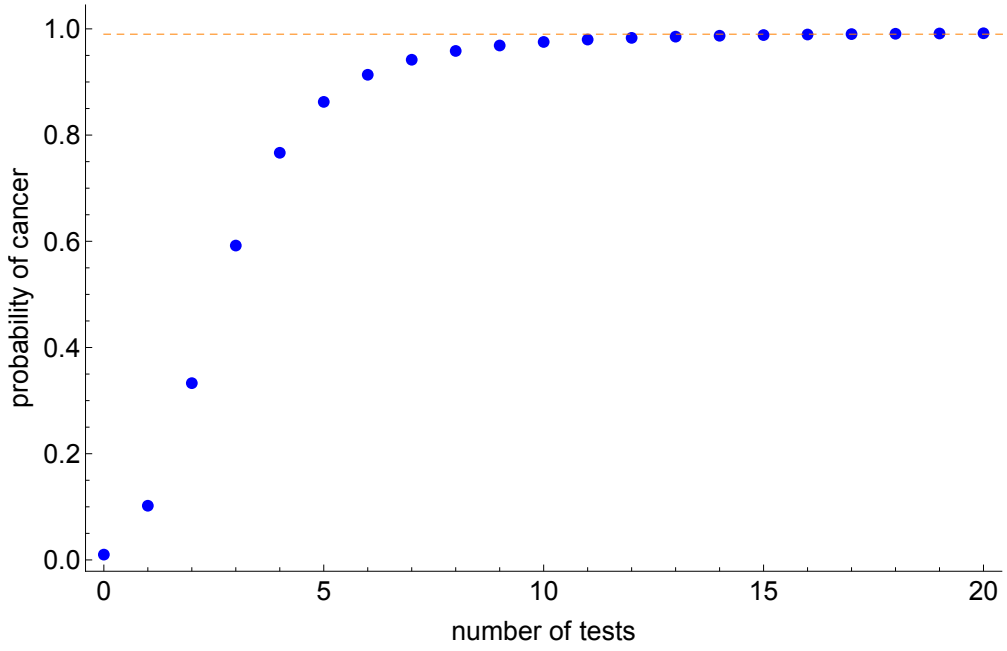

Figure 3.4: A plot of the probability of cancer as a function of the number of tests when we allow for persistence in the test results.

Chapter 4

Likelihoods

4.1 Blog blues

Suppose that visits to your newly launched blog occur sporadically. Imagine you are interested in the length of time between consecutive first-time visits to your homepage. You collect the time data for a random sample of 50 visits to your blog for a particular time period and day, and you decide to build a statistical model to fit the data.

Problem 4.1.1. What assumptions might you make about the first-time visits?

Assume visits occur continuously at a constant rate and independently of one another.

Problem 4.1.2. What might be an appropriate probability model for the time between visits?

If the number of visits is Poisson distributed (which it is if we have the above assumptions) then the time between the events is exponentially distributed.

Problem 4.1.3. Using your chosen probability distribution from the previous part, algebraically derive the maximum likelihood estimate (MLE) of the mean.

Assuming independence between time intervals we can write the overall likelihood as,

$$p(t_1, t_2, ..., t_T|\lambda) = \prod_{i=1}^{T} \lambda e^{-\lambda t_i} \tag{4.1}$$

We then take the log of this,

$$\log p(t_1, t_2, ..., t_T|\lambda) = \sum_{i=1}^{T} (\log \lambda - \lambda t_i) \tag{4.2}$$

$$= T\log \lambda - \lambda T\bar{t} \tag{4.3}$$

Now differentiating the above with respect to λ and solving for the maximum,

$$\frac{\partial \log p}{\partial \lambda} = \frac{T}{\hat{\lambda}} - T\bar{t} = 0, \tag{4.4}$$

which has a solution $\hat{\lambda} = \frac{1}{\bar{t}}$, which makes sense intuitively: the mean event rate we estimate $\hat{\lambda}$ is the inverse of the average time interval between events.

Problem 4.1.4. You collect data from Google Analytics that contains the time (in minutes) between each visit for a sample of 50 randomly chosen visits to your blog. The data set is called `likelihood_blogVisits.csv`. Derive an estimate for the mean number of visits per minute.

Using the above estimator we estimate that $\hat{\lambda} \approx 1.63$ visits per minute.

Problem 4.1.5. Graph the log-likelihood near the MLE. Why do we not plot the likelihood?

Figure 4.1 shows this plot. We don't plot the likelihood as it is far too small for moderately-sized datasets.

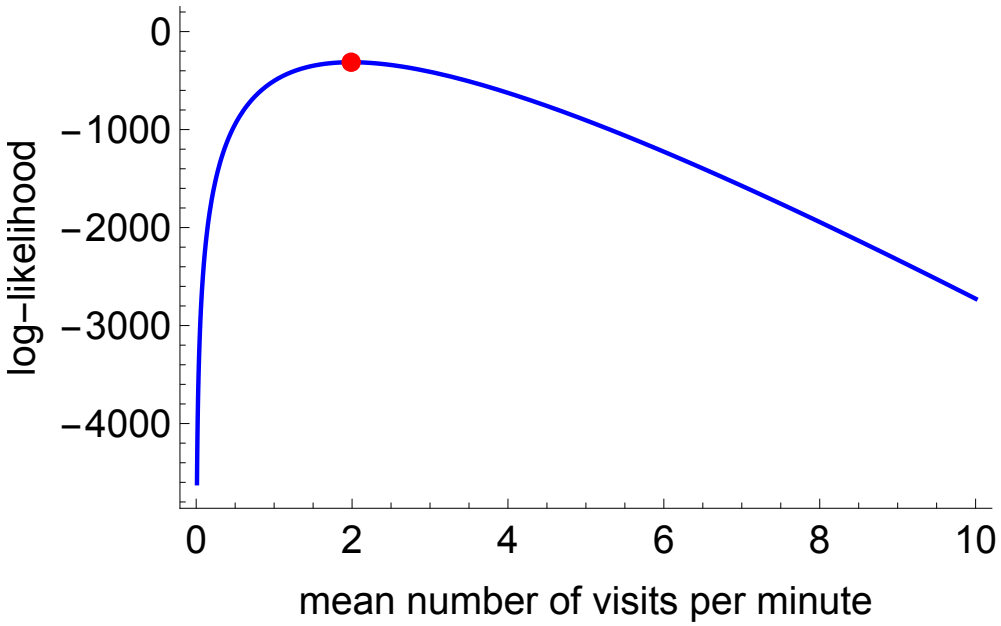

Figure 4.1: A plot of the log likelihood for the exponential model for blog visits. The maximum likelihood estimate is shown in red.

Problem 4.1.6. Estimate 95% confidence intervals around your estimate of the mean visit rate.

The correct way to do this is to calculate the Information matrix (here the observed and expected information matrices are the same),

$$\mathcal{I}(\lambda) = -\mathbb{E}\left(\frac{\partial^2 \log p}{\partial \lambda^2}\right) \tag{4.5}$$

$$= \frac{T}{\lambda^2} \tag{4.6}$$

To convert this to a confidence interval we note that asymptotically (because of the Cramer-Rao lower bound),

$$\sqrt{T}\left(\hat{\lambda} - \lambda\right) \xrightarrow{\mathcal{D}} \mathcal{N}\left(0, \mathcal{I}(\hat{\lambda})^{-\frac{1}{2}}\right), \tag{4.7}$$

which means that for large samples we can approximate,

$$\hat{\lambda} \approx \mathcal{N}\left(\lambda, \frac{1}{\sqrt{T}}\mathcal{I}(\hat{\lambda})^{-\frac{1}{2}}\right), \tag{4.8}$$

where we have that $\frac{1}{\sqrt{T}}\mathcal{I}(\hat{\lambda})^{-\frac{1}{2}} = \frac{\hat{\lambda}}{T} = \frac{1.63}{50} \approx 0.0325$. We therefore construct 95% intervals using the $z = 1.96$ critical value from a standard normal,

$$1.56247 \leq \lambda \leq 1.68996, \tag{4.9}$$

which is a pretty tight interval!

Problem 4.1.7. What does this interval mean?

If we repeatedly sampled from the population (an infinite number of times) and for each sample constructed the 95% confidence interval, then 95% of those hypothetical samples would contain the true parameter value.

Problem 4.1.8. Using your maximum likelihood estimate, what is the probability you will wait: (a) 1 minute or more, (b) 5 minutes or more, (c) half an hour or more before your next visit?

We answer this question using the survival function of the exponential distribution,

$$Pr(t > a|\lambda) = e^{-\lambda a}. \tag{4.10}$$

So answering each part in turn,

1. $Pr(t > 1|\hat{\lambda}) \approx 0.20$.

2. $Pr(t > 5|\hat{\lambda}) \approx 0.0003$.
3. $Pr(t > 30|\hat{\lambda}) \approx 0.000$.

Problem 4.1.9. Evaluate your model.

One way to evaluate your model is to compare real data with that which you simulate from the exponential model at the maximum likelihood estimate of the rate parameter (see Figure 4.2). When we do this we see that the exponential model is clearly unable to generate the upper extremes that we see in the real data.

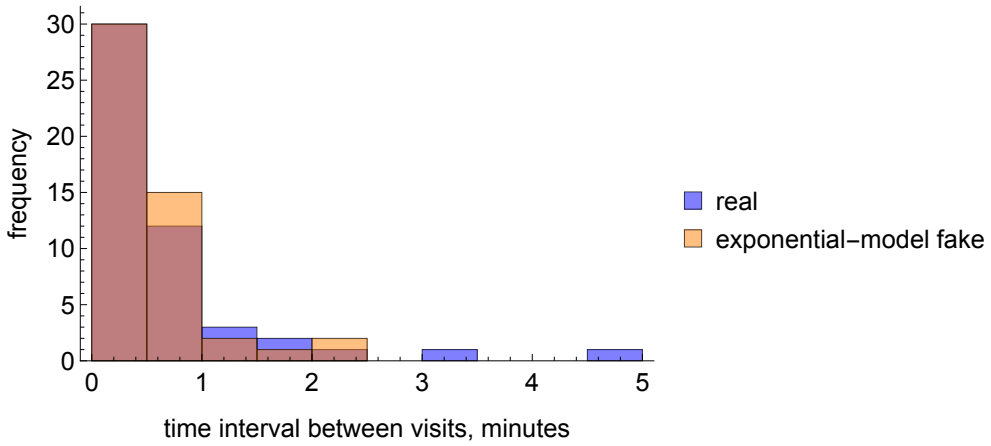

Figure 4.2: Real and fake data simulated from the exponential model using the maximum likelihood estimates of the parameter.

Problem 4.1.10. Can you think of a better model to use? What assumptions are relaxed in this model?

The Poisson is to the exponential what the negative binomial is to the generalised Pareto type 2 distribution, see:

http://stats.stackexchange.com/questions/37814/poisson-is-to-exponential-as-gamma-poisson-is-to-what

The events for the negative binomial distribution are no longer constrained to be independent.

Problem 4.1.11. Estimate the parameters of your new model, and hence estimate the mean number of website visits per minute.

The PDF for this model is of the form,

$$p(t|\alpha, \beta) = \frac{\alpha \left(\frac{\beta+t}{\beta}\right)^{-\alpha-1}}{\beta} \tag{4.11}$$

Using maximum likelihood estimators of the parameters we find that $(\hat{\alpha}, \hat{\beta}) = (0.984052, 2.52871)$. The mean of this distribution is given by,

$$\mathbb{E}(t|\alpha, \beta) = \frac{\beta}{\alpha - 1} \approx 0.64, \tag{4.12}$$

and so the mean number of website visits per minute is about 1.55.

Problem 4.1.12. Use your new model to estimate the probability that you will wait: (a) 1 minute or more, (b) 5 minutes or more, (c) half an hour or more before your next visit.

Using the survival function,

$$Pr(t > a|\alpha, \beta) = \left(\frac{a}{\beta} + 1\right)^{-\alpha}, \tag{4.13}$$

We now see (slightly) less extreme probabilities for waiting times,

1. $Pr(t > 1|\hat{\lambda}) \approx 0.17$.
2. $Pr(t > 5|\hat{\lambda}) \approx 0.01$.
3. $Pr(t > 30|\hat{\lambda}) \approx 0.0002$.

Whilst the question doesn't ask it, as before we can generate data from the model assuming the MLEs (see Figure 4.3), which now look in much better accordance with the real data.

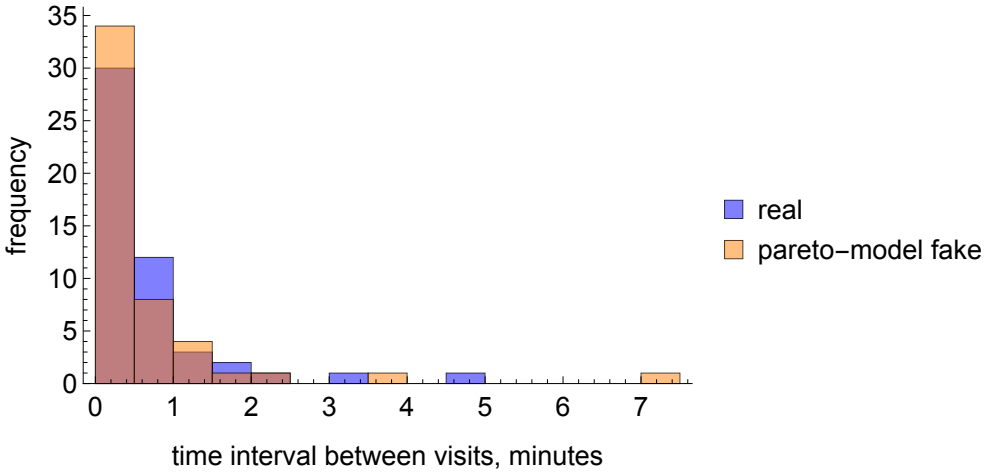

Figure 4.3: Real and fake data simulated from the pareto model of website visits using the maximum likelihood estimates of the parameter.

4.2 Violent crime counts in New York counties

In data file `likelihood_NewYorkCrimeUnemployment.csv` is a data set of the population, violent crime count and unemployment across New York counties in 2014 (openly available from the New York Criminal Justice website).

Problem 4.2.1. Graph the violent crime count against population size across all the counties. What type of relationship does this suggest?

A strong linear relationship (see Figure 4.4).

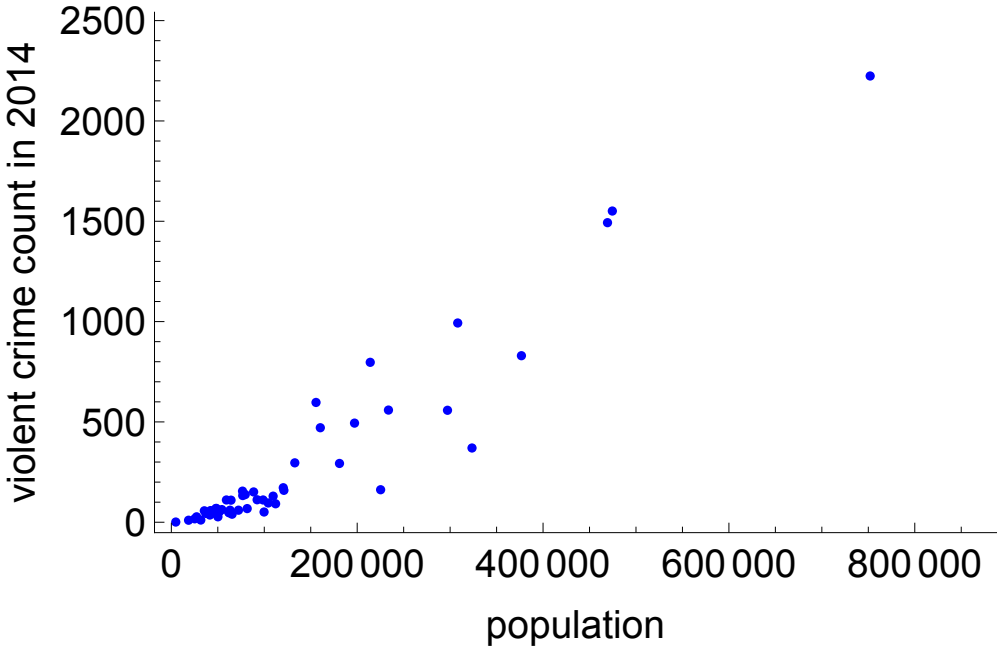

Figure 4.4: Population versus violent crime counts for New York counties in 2014.

Problem 4.2.2. A simple model here might be to assume that the crime count in a particular county is related to the population size by a Poisson model:

$$crime_i \sim Poisson(n_i\theta), \tag{4.14}$$

where $crime_i$ and n_i are the crime count and population in county i. Write down an expression for the likelihood.

Assuming conditional independence between the pairs of observations $(crime_i, n_i)$,

$$p\left(\{crime_1, n_1\}, ..., \{crime_N, n_N\}|\theta\right) = \prod_{i=1}^{N} \frac{(n_i\theta)^{crime_i} e^{-n_i\theta}}{crime_i!}. \tag{4.15}$$

Problem 4.2.3. Find the maximum likelihood estimators of the parameters.

It's easiest to start by working with the log-likelihood,

$$\log p = \sum_{i=1}^{N} [crime_i \log(n_i\theta) - n_i\theta - \log crime_i!] \tag{4.16}$$

$$= -N\bar{n}\theta + \sum_{i=1}^{N} crime_i \log(n_i\theta) + const. \tag{4.17}$$

We then differentiate this expression,

$$\frac{\partial \log p}{\partial \theta} = -N\bar{n} + \frac{1}{\hat{\theta}} N\overline{crime} = 0, \tag{4.18}$$

which we then rearrange for $\hat{\theta} = \frac{\overline{crime}}{\bar{n}}$ which is the sample average violent crime count per capita. So in our case we have that,

$$\hat{\theta} = \frac{318417}{1165.69} \approx 0.004, \tag{4.19}$$

that is, a prevalence of about 4/1000.

Problem 4.2.4. By generating fake data, assess this model.

One way to compare the real and fake data across all counties is to look at the per capita crime rates (see Figure 4.5). We see that the variation in the real data is much greater than that we have in the fake, and hence our model is quite weak.

Problem 4.2.5. What are the assumptions of this model? And do you think that these hold in this case?

This model assumes that one instance of violent crime occurs independently of another, and that the underlying rate of violent crime is the same across all counties. Given that we found our model is deficient we suspect that one or both of the above are probably violated. This makes intuitive sense since violent crimes may often be linked to one another (violating independence) and the counties probably differ in societal ways meaning that some are more/less predisposed to crime.

Problem 4.2.6. Suggest an alternative model and estimate its parameters by maximum likelihood.

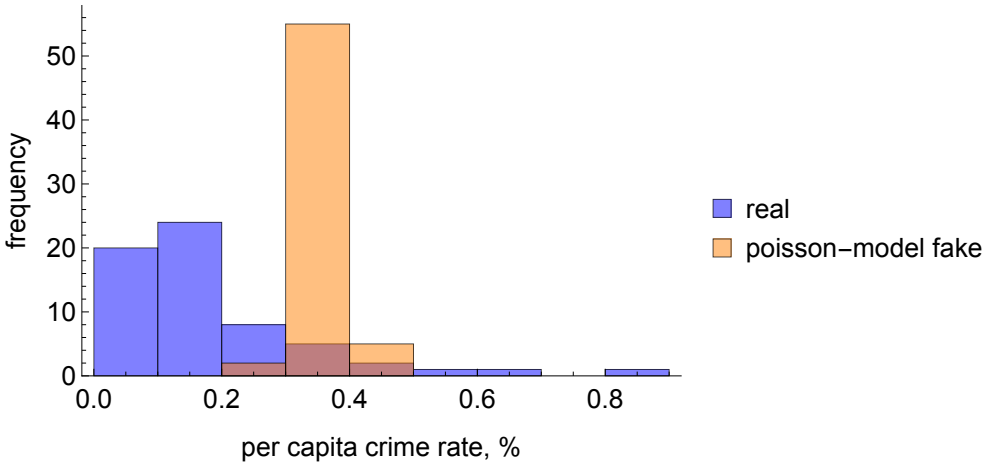

Figure 4.5: The per capita violent crime rate for New York counties in 2014 for the real and Poisson model generated datasets.

A better model that allows a degree of overdispersion in the data is the *negative binomial*, which has a pdf,

$$Pr(crime = x|\lambda,\kappa) = \left(\frac{k}{k+\lambda}\right)^{\frac{k\lambda}{(k+\lambda)\left(1-\frac{k}{k+\lambda}\right)}} \left(1-\frac{k}{k+\lambda}\right)^{x} \binom{x+\frac{k\lambda}{(k+\lambda)\left(1-\frac{k}{k+\lambda}\right)}-1}{\frac{k\lambda}{(k+\lambda)\left(1-\frac{k}{k+\lambda}\right)}-1}, \tag{4.20}$$

where $\lambda = \theta n_i$ is the mean and $\kappa \geq 0$ is the overdispersion parameter. Assuming conditional independence for each of the data points we can again estimate the model via maximum likelihood and obtain $(\hat{\theta}, \hat{\kappa}) = (0.00185, 2.2222)$.

Problem 4.2.7. Evaluate this new model.

Again we can compare model-simulated data with the actual and this time we find that there is much better correspondence (see Figure 4.6)

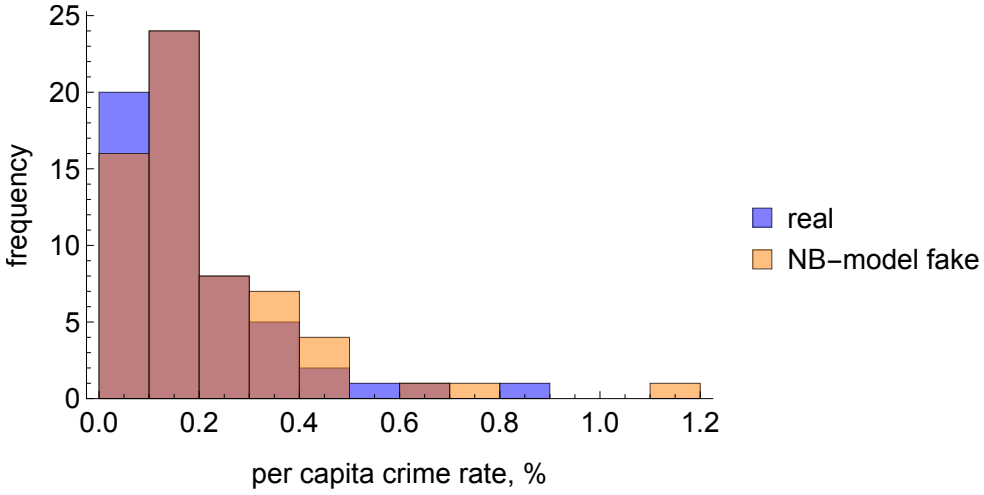

Figure 4.6: The per capita violent crime rate for New York counties in 2014 for the real and negative binomial model generated datasets.

Chapter 5

Priors

5.1 Dodgy coins

Suppose there are three coins in a bag. The first coin is biased towards heads, with a 75% probability of a heads occurring if the coin is flipped. The second is fair, so a 50% chance of heads occurring. The third coin is biased towards tails, and has a 25% probability of coming up heads. Assume that it is impossible to identify which coin is which from looking at or touching them.

Problem 5.1.1. Suppose we put our hand into the bag and pull out a coin. We then flip the coin and find it comes up heads. Let the random variable C=1,2,3 denote the identity of the coin, where the probability of heads is (0.75,0.50,0.25), respectively. Obtain the likelihood by using the equivalence relation (that a likelihood of a parameter value given data is equal to the probability of data given a parameter value), and show that the sum of the likelihood over all parameter values is 1.5.

Using the equivalence relation we obtain a likelihood of the form,

$$L(C|X=H) = \begin{cases} Pr(X=H|C=1) = 0.75, & \text{if } C=1 \\ Pr(X=H|C=2) = 0.50, & \text{if } C=2 \\ Pr(X=H|C=3) = 0.25, & \text{if } C=3 \end{cases}$$

So we obtain 1.50 by summing across all likelihoods.

Problem 5.1.2. What is the maximum likelihood estimate of the coin's identity?

From the likelihood we can see that $C=1$ maximises the likelihood.

Problem 5.1.3. Use Bayes' rule to prove that:

$$Pr(C=c|X=H) \propto Pr(X=H|C=c) \times Pr(C=c) \tag{5.1}$$

where $c=1,2,3$.

parameter C	likelihood $Pr(X = H\|C = c)$	prior $Pr(C = c)$	likelihood × prior $Pr(X = H\|C = c) \times Pr(C = c)$	posterior $Pr(C = c\|X = H)$
1				
2				
3				
			$Pr(X = H) =$	

Table 5.1: A Bayes box for the coins example.

parameter C	likelihood $Pr(X = H\|C = c)$	prior $Pr(C = c)$	likelihood × prior $Pr(X = H\|C = c) \times Pr(C = c)$	posterior $Pr(C = c\|X = H)$
1	3/4	1/3	3/12	1/2
2	2/4	1/3	2/12	1/3
3	1/4	1/3	1/12	1/6
			$Pr(X = H) = 6/12 = 1/2$	

Table 5.2: A Bayes box for the coins example.

Problem 5.1.4. Assume that since we cannot visually detect the coin's identity we use a uniform prior $Pr(C = c) = 1$ for $c = 1, 2, 3$. Use this to complete Table 5.1 (known as a Bayes' box) and determine the (marginal) probability of the data.

The completed Bayes box is shown in Table 5.2, where since the posterior probabilities are non-negative and their sum is 1, we can conclude that we have a valid probability distribution. The probability of the data is 0.5.

Problem 5.1.5. Confirm that the posterior is a valid probability distribution.

Since the individual probabilities are non-negative and their sum is 1, this is a valid probability distribution.

Problem 5.1.6. Now assume that we flip the same coin twice, and find that it lands heads up on both occasions. By using a Table similar in form to Table 5.1, or otherwise, determine the new posterior distribution.

Table 5.3 shows how we calculate the new posterior distribution. There is now an increased weighting towards $C = 1$, reflecting the increased chance of observing two heads given this state of the world.

Problem 5.1.7. Now assume that you believe that the tails-biased coin is much more likely to be drawn from the bag, and thus specify a prior: $Pr(C = 1) = 1/20$, $Pr(C = 2) = 5/20$ and $Pr(C = 3) = 14/20$. What is the posterior probability that $C = 1$ now?

The Bayes box for this example is shown in Table 5.4.

parameter C	likelihood $Pr(X_1 = H, X_2 = H \mid C = c)$	prior $Pr(C = c)$	likelihood × prior $Pr(X_1 = H, X_2 = H \mid C = c) \times Pr(C = c)$	posterior $Pr(C = c \mid X_1 = H, X_2 = H)$
1	9/16	1/3	9/48	9/14
2	4/16	1/3	4/48	4/14
3	1/16	1/3	1/48	1/14
			$Pr(X = H) = 14/48 = 7/24$	

Table 5.3: A Bayes box for the coins example where the coin is flipped twice, landing heads up both times.

parameter C	likelihood $Pr(X_1 = H, X_2 = H \mid C = c)$	prior $Pr(C = c)$	likelihood × prior $Pr(X_1 = H, X_2 = H \mid C = c) \times Pr(C = c)$	posterior $Pr(C = c \mid X_1 = H, X_2 = H)$
1	9/16	1/20	9/320	9/43
2	4/16	5/20	20/320	20/43
3	1/16	14/20	14/320	14/43
			$Pr(X = H) = 43/320$	

Table 5.4: A Bayes box for the coins example where the coin is flipped twice, landing heads up both times.

Problem 5.1.8. Continuing on from the previous example, calculate the posterior mean, maximum a posteriori (MAP) and maximum likelihood estimates. Does the posterior mean indicate much here?

Using the Bayes Box in Table 5.4, we calculate a posterior mean of,

$$\mathbb{E}(C|X_1 = H, X_2 = H) = \sum_{C=1}^{3} 1 \times 9/43 + 2 \times 20/43 + 3 \times 14/43 \tag{5.2}$$

$$= 91/43 \approx 2.12 \tag{5.3}$$

The MAP estimate is $C = 2$ since this is the posterior mode. The maximum likelihood estimate is $C = 1$.

Problem 5.1.9. For the case when we flip the coin once and obtain $X = H$, using the uniform prior on C, determine the posterior predictive distribution for a new coin flip with result $\tilde{X}$, using the below expression,

$$Pr(\tilde{X}|X = H) = \sum_{C=1}^{3} Pr(\tilde{X}|C) \times Pr(C|X = H) \tag{5.4}$$

Using Table 5.2 we first determine the probability of a heads,

$$Pr(\tilde{X} = H|X = H) = 3/4 \times 1/2 + 2/4 \times 1/3 + 1/4 \times 1/6 \tag{5.5}$$

$$= 7/12 \tag{5.6}$$

Then we calculate the probability of a tails,

$$Pr(\tilde{X} = H|X = H) = 1/4 \times 1/2 + 2/4 \times 1/3 + 3/4 \times 1/6 \tag{5.7}$$
$$= 5/12 \tag{5.8}$$

These two results taken together form a valid probability distribution since the probabilities sum to 1.

Problem 5.1.10. (Optional) Justify the use of the expression in the previous question.

To do this we marginalise out C from the joint probability $Pr(\tilde{X}, C|X = H)$,

$$Pr(\tilde{X}|X = H) = \sum_{C=1}^{3} Pr(\tilde{X}, C|X = H) \tag{5.9}$$
$$= \sum_{C=1}^{3} Pr(\tilde{X}|C, X = H) \times Pr(C|X = H) \tag{5.10}$$
$$= \sum_{C=1}^{3} \underbrace{Pr(\tilde{X}|C)}_{likelihood} \times \underbrace{Pr(C|X = H)}_{posterior} \tag{5.11}$$

where we got from the first line to the second using the law of conditional probability, and from the second to the third by realising that once we know C the result $\tilde{X}$ is independent of $X = H$.

5.2 Left-handedness

Suppose that we are interested in the prevalence of left-handedness in a particular population.

Problem 5.2.1. We begin with a sample of one individual whose dexterity we record as $X = 1$ for left-handed, $X = 0$ otherwise. Explain why the following probability distribution makes sense here:

$$Pr(X|\theta) = \theta^X (1 - \theta)^{1-X}, \tag{5.12}$$

where θ is the probability that a randomly chosen individual is left-handed.

Under the two circumstances it yields the relevant probabilities, $Pr(X = 1|\theta) = \theta$ and $Pr(X = 0|\theta) = 1 - \theta$.

Problem 5.2.2. Suppose we hold θ constant. Demonstrate that under these circumstances the above distribution is a valid probability distribution. What sort of distribution is this?

Summing together the two possibilities here,

$$Pr(X = 1|\theta) + Pr(X = 0|\theta) = \theta + 1 - \theta = 1. \tag{5.13}$$

In this case we are dealing with a discrete distribution (a Bernoulli).

Problem 5.2.3. Now suppose we randomly sample a person who happens to be left-handed. Using the above function calculate the probability of this occurring.

$$Pr(X = 1|\theta) = \theta \tag{5.14}$$

Problem 5.2.4. Show that when we vary θ the above distribution does not behave as a valid probability distribution. Also, what sort of distribution is this?

The way to demonstrate this is by integrating the above over a range of θ,

$$\int_0^1 \theta \mathrm{d}\theta = \frac{1}{2} \tag{5.15}$$

So it is not a valid continuous probability distribution and hence we call it a 'likelihood'.

Problem 5.2.5. What is the maximum likelihood estimator for θ?

$\hat{\theta} = 1$ maximises the probability of obtaining one individual who is left-handed.

Chapter 6

The devil is in the denominator

6.1 Too many coin flips

Suppose we flip two coins. Each coin i is either fair ($Pr(H) = \theta_i = 0.5$) or biased towards heads ($Pr(H) = \theta_i = 0.9$) however, we cannot visibly detect the coin's nature. Suppose we flip both coins twice and record each result.

Problem 6.1.1. Suppose that we specify a discrete uniform prior on both θ_1 and θ_2. Find the joint distribution of the data and the coins' identity.

Denote the result of coin 1's flips by X_1 and X_2, where $X_i = 1$ is heads. Similarly for coin 2 except we use Y_i to denote the result on the ith flip. We can then write down the joint distribution as follows,

$$\begin{aligned} Pr(X_1, X_2, Y_1, Y_2, \theta_1, \theta_2) &= Pr(X_1, X_2|\theta_1)Pr(Y_1, Y_2|\theta_2)Pr(\theta_1)Pr(\theta_2) && (6.1)\\ &= \theta_1^{X_1+X_2}(1-\theta_1)^{2-X_1-X_2}\theta_2^{Y_1+Y_2}(1-\theta_2)^{2-Y_1-Y_2}0.5^2 && (6.2)\end{aligned}$$

Problem 6.1.2. Show that the above distribution is a valid probability distribution.

It trivially satisfies $Pr(.) \geq 0$ and so we only need to show that the sum of all values is 1 by summing over all the $2^4 = 16$ possible parameter combinations. This can be done by exhaustively checking every term but the symmetry of the problem means there are probably easier ways to proceed here.

Problem 6.1.3. We flip each coin twice and obtain for coin 1 $\{HH\}$ and coin 2 $\{HT\}$. Assuming that the result of each coin flip is independent of the previous result write down a likelihood function.

Since the coin flips are independent we can write down the overall likelihood by multiplying together the individual ones,

$$Pr(\{HH\},\{HT\}|\theta_1,\theta_2) = \theta_1^2\theta_2(1-\theta_2) \tag{6.3}$$

Problem 6.1.4. What are the maximum likelihood estimators of each parameter?

We can consider each parameter separately due to the independence of the problem. For coin 1, clearly $\hat{\theta}_1 = 0.9$ since this maximises θ_1^2; for coin 2 $\hat{\theta}_2 = 0.5$ since this maximises $\theta_2(1-\theta_2)$.

Problem 6.1.5. Calculate the marginal likelihood of the data (that is, the denominator of Bayes' rule).

This can be obtained by marginalising out all θ dependence in the joint distribution,

$$\begin{aligned} Pr(HH,HT) &= \sum_{\theta_1}\sum_{\theta_2} Pr(HH,HT|\theta_1,\theta_2)Pr(\theta_1)Pr(\theta_2) && (6.4)\\ &= \sum_{\theta_1} Pr(HH|\theta_1)Pr(\theta_1)\sum_{\theta_2} Pr(HT|\theta_2)Pr(\theta_2) && (6.5)\\ &= \frac{1}{4}(\frac{1}{4}+\frac{81}{100})(\frac{1}{4}+\frac{9}{10}\frac{1}{10}) && (6.6)\\ &= \frac{901}{10000} && (6.7)\end{aligned}$$

Problem 6.1.6. Hence calculate the posterior distribution, and demonstrate that this is a valid probability distribution.

Using Bayes' rule we have that,

$$\begin{aligned} Pr(\theta_1,\theta_2|HH,HT) &= \frac{\theta_1^2\theta_2(1-\theta_2)\frac{1}{4}}{\frac{901}{10000}} && (6.8)\\ &= \frac{2500}{901}\theta_1^2\theta_2(1-\theta_2). && (6.9)\end{aligned}$$

Summing the above over both θ_1 and θ_2,

$$\begin{aligned} \sum_{\theta_1}\sum_{\theta_2}\frac{2500}{2809}\theta_1^2\theta_2(1-\theta_2) &= \frac{2500}{901}\sum_{\theta_1}\theta_1^2\sum_{\theta_2}\theta_2(1-\theta_2) && (6.10)\\ &= \frac{2500}{901}\left(0.5^2+0.9^2\right)\left(0.5\times 0.5+0.9\times 0.1\right) && (6.11)\\ &= 1. && (6.12)\end{aligned}$$

Problem 6.1.7. Find the posterior mean of θ_1. What does this signify?

θ_1	θ_2	Z	$Pr(HT, HH, \theta_1, \theta_2 \mid Z)$
0.5	0.5	0	0.000625
0.5	0.5	1	0.04
0.5	0.9	0	0.002025
0.5	0.9	1	0.0036
0.9	0.5	0	0.018225
0.9	0.5	1	0.0324
0.9	0.9	0	0.059049
0.9	0.9	1	0.002916

Table 6.1: The likelihood function for the dependent coin flip example.

$$\begin{aligned}
\mathbb{E}(\theta_1 | HH, HT) &= \mathbb{E}(\theta_1 | HH) && (6.13)\\
&= \sum_{\theta_1} \theta_1 \frac{\theta_1^2}{0.5^2 + 0.9^2} && (6.14)\\
&= \frac{1}{0.5^2 + 0.9^2}(0.5^3 + 0.9^3) && (6.15)\\
&\approx 0.81 && (6.16)
\end{aligned}$$

This is a bit tricky to interpret since $\theta_1 \in \{0.5, 0.9\}$. However it basically means that there is greater weight towards $\theta_1 = 0.9$.

Problem 6.1.8. Now suppose that away from our view a third coin is flipped, and denote $Z = 1$ for a heads. The result of this coin affects the bias of the other two coins that are flipped subsequently so that,

$$Pr(\theta_i = 0.5 | Z) = 0.8^Z 0.1^{1-Z} \tag{6.17}$$

Suppose we again obtain for coin 1 $\{HH\}$ and coin 2 $\{HT\}$. Find the maximum likelihood estimators (θ_1, θ_2, Z). How do the inferred biases of coin 1 and coin 2 compare to the previous estimates?

To do this we enumerate over the 8 possible combinations of θ_1, θ_2 and Z (see Table 6.1). From this it is evident that the maximum likelihood estimators are $(\hat{\theta}_1, \hat{\theta}_2, \hat{Z}) = (0.9, 0.9, 0)$. So here we have that coin 2's ML bias has changed from 0.5 to 0.9. Intuitively this is because there is a strong penalty for the coin 2 being fair if coin 1 is not, because of the dependence structure.

Problem 6.1.9. Calculate the marginal likelihood for the coin if we suppose that we specify a discrete uniform prior on Z, i.e. $Pr(Z = 1) = 0.5$.

To do this we simply multiply all the calculated likelihoods from Table 6.1 by 0.5 and sum them, to obtain, $Pr(HH, HT) = 0.0794$.

Problem 6.1.10. Suppose we believe that the independent coin flip model (where there is no third coin) and the dependent coin flip model (where the outcome of the third coin affects the biases of the two coins) are equally likely *a priori*. Which of the two models do we prefer?

Basically we want to compare,

$$\frac{Pr(\text{independent M}|HH,HT)}{Pr(\text{dependent M}|HH,HT)} = \frac{Pr(HH,HT|\text{independent M})}{Pr(HH,HT|\text{dependent M})} \times \underbrace{\frac{Pr(\text{independent M})}{Pr(\text{dependent M})}}_{1} \tag{6.18}$$

$$= \frac{0.0901}{0.0794} \tag{6.19}$$

$$\approx 1.13 \tag{6.20}$$

so using this basic test we prefer the independent flips model.

6.2 Coins combined

Suppose that we flip two coins, each of which has $Pr(H) = \theta_i$ where $i \in \{1, 2\}$, which is unknown. If their outcome is both the same then we regard this as a success; otherwise a failure. We repeatedly flip both coins (a single trial) and record whether the outcome is a success or failure. We do not record the result of flipping each coin. Suppose we model the number of failures, X, we have to undergo to attain n successes.

Problem 6.2.1. Stating any assumptions that you make specify a suitable probability model here.

The negative binomial fits this description perfectly. However we need to modify it to allow the probability of success to be a function of both coins' biases $p = \theta_1\theta_2 + (1-\theta_1)(1-\theta_2)$. This means we can write down the pmf,

$$Pr(X|n,\theta_1,\theta_2) = \begin{cases} \binom{n+X-1}{n-1}((1-\theta_1)(1-\theta_2)+\theta_1\theta_2)^n(1-(1-\theta_1)(1-\theta_2)-\theta_1-\theta_2)^X & X \geq 0 \\ 0 & \text{True} \end{cases} \tag{6.21}$$

This assumes that the flips of each coin are independent.

Problem 6.2.2. We obtain the data in `denominator_NBCoins.csv` for the number of failures to wait before 5 successes occur. Suppose that we specify the following priors $\theta_1 \sim U(0,1)$ and $\theta_2 \sim U(0,1)$. Calculate the denominator of Bayes' rule. (Hint: use a numerical integration routine.)

This requires us to do the following integral,

$$\int_0^1 \int_0^1 NB(X|n,\theta_1,\theta_2)\mathrm{d}\theta_1\mathrm{d}\theta_2 \approx 2.48731 \times 10^{-170}. \tag{6.22}$$

I carried out the above using Mathematica's 'NIntegrate' function which took around three seconds.

Problem 6.2.3. Draw a contour plot of the posterior. Why does the posterior have this shape?

The posterior is shown in Figure 6.1. There is a thin band of probability mass associated with the lines $\theta_1\theta_2 = const$ or $(1-\theta_1)(1-\theta_2) = const$. The mass is mostly associated in the upper left and bottom right because the data has relatively long runs before 5 successes occur, meaning that the same values for the parameters are not likely.

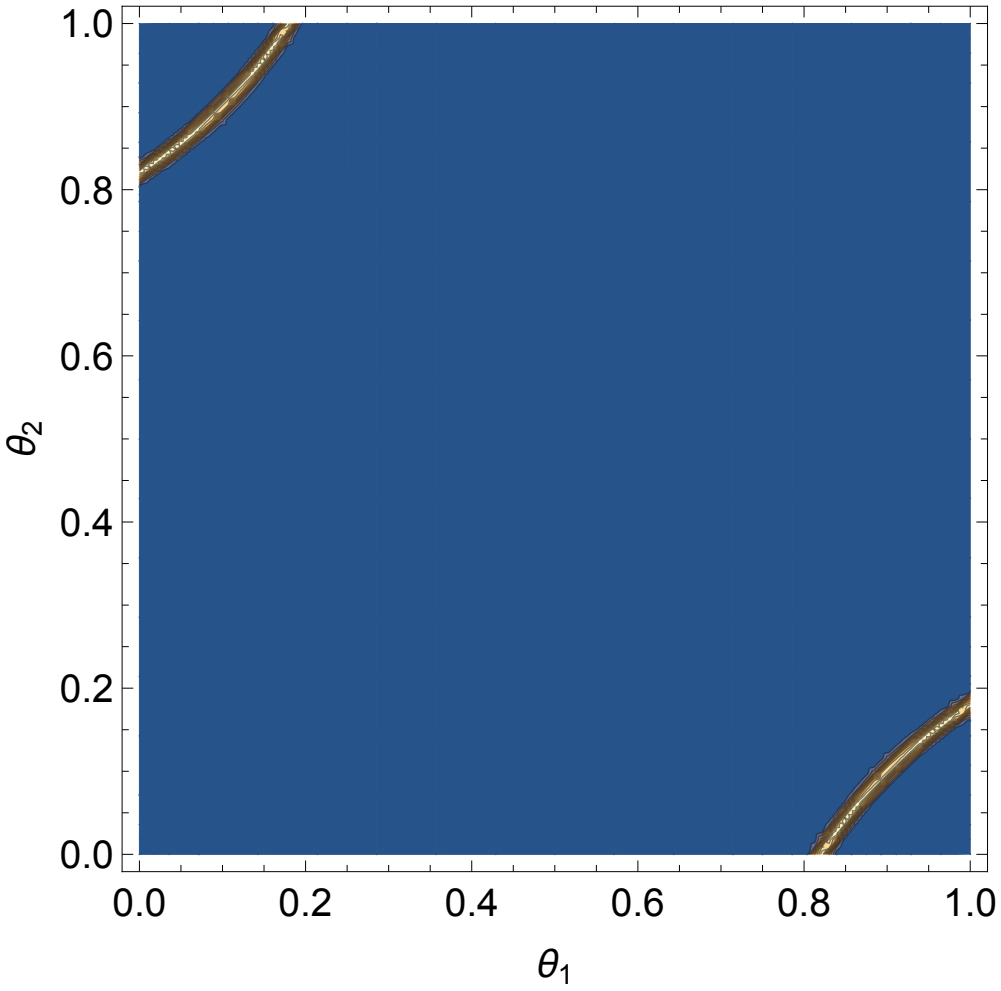

Figure 6.1: The posterior for the negative binomial coins example.

Problem 6.2.4. Comment on any issues with parameter identification for this model and how this might be rectified.

Clearly the model cannot differentiate between θ_1 and θ_2 because we have provided no further information on these. A solution would be to use a prior that assigns a strong weight to high/low values of one of the parameters. This isn't an issue that can be solved by collecting more data, unless we could see the identities of each coin.

Problem 6.2.5. Now suppose that we have three coins instead of two. Here we regard a success as all three coins showing the same result. Using the same data as before attempt to calculate the denominator term. Why is there a problem?

Even with just three dimensions even sophisticated deterministic routines struggle. After a few minutes Mathematica gave me this result,

$$\int_0^1 \int_0^1 \int_0^1 NB(X|5, \theta_1, \theta_2, \theta_3) \mathrm{d}\theta_1 \mathrm{d}\theta_2 \theta_3 \approx 3.64959 \times 10^{-169} \tag{6.23}$$

Clearly with higher dimensions evaluating these integrals is going to be just too hard to attempt!

Problem 6.2.6. Assuming a denominator term equal to 3.64959×10^{-169} estimate the posterior mean of θ_1.

Again we run into problems but now with a different integral,

$$\mathbb{E}(\theta_1) = \int_0^1 \int_0^1 \int_0^1 \theta_1 NB(X|5, \theta_1, \theta_2, \theta_3) \frac{1}{3.64959 \times 10^{-169}} \mathrm{d}\theta_1 \mathrm{d}\theta_2 \theta_3 \approx 0.500. \tag{6.24}$$

The above took around three minutes on Mathematica 11 on my laptop. The moral of the story is that even if we have the denominator there are still issues with integrating!

Chapter 7

The posterior - the goal of Bayesian inference

7.1 Googling

Suppose you are chosen, for your knowledge of Bayesian statistics, to work at Google as a search traffic analyst. Based on historical data you have the data shown in Table 7.1 for the actual word searched, and the starting string (the first three letters typed in a search). It is your job to help make the search engines faster, by reducing the search-space for the machines to lookup each time a person types.

	Barack Obama	**Baby clothes**	**Bayes**
Bar	50%	30%	30%
Bab	30%	60%	30%
Bay	20%	10%	40%

Table 7.1: The columns give the historic breakdown of the search traffic for three topics: Barack Obama, Baby clothes, and Bayes; by the first three letters of the user's search.

Problem 7.1.1. Find the minimum-coverage confidence intervals of topics that are at least at 70%.

In both cases (here and the next question) we are looking for sets of the actual words. Frequentists assume that the data we receive (each set of three letters) is a sample from an infinity of such experiments. As such they design their intervals such that at least 70% of such intervals contain the true word searched across all the potential data samples we could receive. This means that here we want to choose sets such that regardless of the three letters typed we get a coverage of at least 70% for each of the columns. These are shown in Table 7.2.

Problem 7.1.2. Find most narrow credible intervals for topics that are at least at 70%.

Bayesians condition of the data we *actually* receive, and derive intervals based on this information.

	Barack Obama	Baby clothes	Bayes	Credibility
Bar	[50%	30%	30%	45%
Bab	30%	[60%	30%]	75%
Bay	[20%	10%	40%]	100%
Coverage	70%	70%	70%	

Table 7.2: ≥70% confidence intervals.

This means we need to consider the individual row sums; each time making an interval that exceeds at least 70% of that row. The answer for this question is shown in Table 7.3.

	Barack Obama	Baby clothes	Bayes	Credibility
Bar	[50%	30%]	30%	73%
Bab	30%	[60%	30%]	75%
Bay	20%	[10%	40%]	71%
Coverage	50%	100%	70%	

Table 7.3: ≥70% credible intervals.

Now we suppose that your boss gives you the historic search information shown in Table 7.4. Further, you are told that it is most important to correctly suggest the actual topic as one of the first auto-complete options, *irrespective* of the topic searched.

Problem 7.1.3. Do you prefer confidence intervals or credible intervals in this circumstance?

Here all we need to do is work out the total losses under the confidence and credible intervals. For both cases this means we need to work out the expected loss for each of the actual words being searched, using the volumes given in Table 7.4. This is easily done using the coverages at the bottom of each of tables 7.2 and 7.3.

For the confidence intervals we thus get an expected loss:

$$loss = 0.6 \times (1 - 0.7) + 0.3 \times (1 - 0.7) + 0.1 \times (1 - 0.7) = 0.3 \tag{7.1}$$

And for the credible intervals:

$$loss = 0.6 \times (1 - 0.5) + 0.3 \times (1 - 1) + 0.1 \times (1 - 0.7) = 0.33 \tag{7.2}$$

So in this circumstance we prefer the confidence intervals.

	Barack Obama	Baby clothes	Bayes
Search volume	60%	30%	10%

Table 7.4: The historic search traffic broken down by topic.

Problem 7.1.4. Now assume that it is most important to pick the correct actual word across all potential sets of three letters. Which interval do you prefer now?

Now we need to find the loss for each possible three letter search. This requires that we first of all calculate the historic search volumes for these letters using Tables 7.1 and 7.4. Specifically you take the matrix product of the two, yielding a percentage of historical searches of (42%, 39%, 19%) for (bar, bab bay). You then use the credible levels for each of the rows from the confidence and credible interval tables respectively to weight the losses.

For confidence intervals:

$$loss = 0.42 \times (1 - 0.45) + 0.39 \times (1 - 0.75) + 0.19 \times (1 - 1) = 0.33 \tag{7.3}$$

And for credible intervals:

$$loss = 0.42 \times (1 - 0.73) + 0.39 \times (1 - 0.75) + 0.19 \times (1 - 0.71) = 0.27 \tag{7.4}$$

So in this case we prefer the credible intervals.

7.2 GDP versus infant mortality

The data in `posterior_gdpInfantMortality.csv` contains the GDP per capita (in real terms) and infant mortality across a large sample of countries in 1998.

Problem 7.2.1. A simple model is fit to the data of the form:

$$M_i \sim \mathcal{N}(\alpha + \beta GDP_i, \sigma) \tag{7.5}$$

Fit this model to the data using a Frequentist approach. How well does the model fit the data?

Graph the data first! I have perhaps been a bit misleading here asking the student to fit the model *before* graphing the data. However, this is sort of the point. You should never - blindly - fit a model to data. If you graph the data, you see that a linear model is not well suited to the data at all; a power law is better-suited. You can also use AIC/BIC/adjusted-R^2 etc. to compare between these models, but really the graphical explanation is the gold standard.

Problem 7.2.2. An alternative model is:

$$log(M_i) \sim \mathcal{N}(\alpha + \beta log(GDP)_i, \sigma) \tag{7.6}$$

Fit this model to the data using a Frequentist approach. Which model do you prefer, and why?

See above - this model is much better suited!

Problem 7.2.3. Construct 80% confidence intervals for (α, β) for the log-log model.

Take the point estimates of the parameters and add the relevant critical values of a *standardised* Student T distribution with $n-2$ degrees of freedom (here the population standard deviation is unknown so we need to use a T rather than a normal) multiplied by the parameter's standard error. The 10% critical value (we need 10% values because we are using a two-sided test) of a T with the relevant degrees of freedom is 1.28. We therefore obtain the following 80% confidence intervals:

$$\begin{aligned} 6.8 \leq \alpha \leq 7.3 \\ -0.53 \leq \beta \leq -0.46 \end{aligned}$$

Problem 7.2.4. We have fit the log-log model to the data using MCMC. Samples from the posterior for (α, β, σ) are contained within the file `posterior_posteriorsGdpInfantMortality.csv`. Using this data find the 80% credible intervals for all parameters (assuming these intervals to be symmetric about the median). How do these compare with the confidence intervals calculated above for (α, β)? How does the point estimate of σ from the Frequentist approach above compare?

Using the "quantile" function these can be estimated:

$$\begin{aligned} 6.8 \leq \alpha \leq 7.3 \\ -0.53 \leq \beta \leq -0.46 \\ 0.56 \leq \sigma \leq 0.64 \end{aligned}$$

The first two are, to the accuracy shown, indistinguishable from the Frequentist estimates. The point estimate for σ for the two approaches is:

$$\begin{aligned} \hat{\sigma}_F &= 0.59 \\ \hat{\sigma}_B &= 0.60 \end{aligned}$$

where I have used the posterior mean for the Bayesian estimate.

Problem 7.2.5. The following priors were used for the three parameters:

$$\begin{aligned} \alpha &\sim \mathcal{N}(0, 10) \\ \beta &\sim \mathcal{N}(0, 10) \\ \sigma &\sim \mathcal{N}(0, 5), \text{where } \sigma \geq 0 \end{aligned}$$

Explain any similarity between the confidence and credible intervals in this case.

Here the priors are very diffuse over the range of possible range of the parameters. To a (rough) approximation this is equivalent to a flat prior on the parameters. This means from Bayes' rule we have (approximately):

$$p(\theta|X) \propto p(X|\theta) \tag{7.7}$$

Therefore the confidence and credible intervals are going to be largely similar here.

Problem 7.2.6. How are the estimates of parameters (α, β, σ) correlated? Why?

α and β are negatively correlated. This is because we want a line that goes through the centre of the data: if the y intercept increases then the slope must decrease.

Problem 7.2.7. Generate samples from the prior predictive distribution. How does the min and max of the prior predictive distribution compare with the actual data?

The prior predictive distributions show about two orders of magnitude greater variation in data compared to the actual data.

Problem 7.2.8. Generate samples from the posterior predictive distribution, and compare these with the actual data. How well does the model fit the data?

There are a number of ways to compare the model vs the data here. I have just used the min and max as a point of comparison. What we see with these is that the minimum is captured well by the model, but the max isn't. In particular the variation seen in fitted model is *greater* than that in the data. This is because at low values of GDP there could be a deviation from the log-log model (or it's just due to sampling variation, of course).

7.3 Bayesian neurosurgery

Suppose that you are a neurosurgeon and have been given the unenviable task of finding the position of a tumour within a patient's brain, and cutting it out. Along two dimensions - vertical height and left-right axis - the tumour's position is known to a high degree of confidence. However, along the remaining axis (front-back) the position is uncertain, and cannot be ascertained without surgery. However, a team of brilliant statisticians has already done most of the job for you, and has generated samples from the posterior for the tumour's location along this axis, and is given by the data contained within the data file `posterior_brainData.csv`.

Suppose that the more brain that is cut, the more the patient is at risk of losing cognitive functions. Additionally, suppose that there is uncertainty over the amount of damage done to the patient during surgery. As such, three different surgeons have differing views on the damage caused:

1. *Surgeon 1:* Damage varies quadratically with the distance the surgery starts away from the tumour.
2. *Surgeon 2:* There is no damage if tissue cut is within 0.0001mm of the tumour; for cuts further away there is a fixed damage.

3. *Surgeon 3:* Damage varies linearly with the absolute distance the surgery starts away from the tumour. (Hard - use fundamental theorem of Calculus for this part of the question.)

Problem 7.3.1. Under each of the three regimes above, find the best position along this axis to cut.

Surgeon 1:

A **quadratic** loss function has the form: $L(\hat{\theta}, \theta) = (\hat{\theta} - \theta)^2$, where $\hat{\theta}$ is the point estimate of the parameter, and θ is the actual value. We can find the expected loss:

$$\begin{aligned} \mathrm{E(L)} &= \int (\hat{\theta} - \theta)^2 p(\theta|x) \mathrm{d}\theta \\ &= \hat{\theta}^2 \int p(\theta|x) \mathrm{d}\theta - 2\hat{\theta} \int \theta p(\theta|x) \mathrm{d}\theta + \int \theta^2 p(\theta|x) \mathrm{d}\theta \\ &= \hat{\theta}^2 - 2\hat{\theta} < \theta|x > + < \theta|x >^2 \end{aligned}$$

which is minimised if $\hat{\theta} =< \theta|x >$. From the data the mean is 6.1.

Surgeon 2:

A **binary** loss function has the form: $L(\hat{\theta}, \theta) = 1 - \delta_{\theta=\hat{\theta}}$, where the δ is a Dirac delta at $\theta = \hat{\theta}$. Finding the expected loss:

$$\begin{aligned} \mathbb{E}(L) &= \int (1 - \delta_{\theta=\hat{\theta}}) p(\theta|x) \mathrm{d}\theta \\ &= 1 - p(\theta = \hat{\theta}|x) \end{aligned}$$

which is minimised when $\hat{\theta} = \arg\max_\theta p(\theta|x)$; in other words the MAP estimator, which from a histogram of the data is at around 4.5.

Surgeon 3:

A **linear** loss is of the form: $L(\hat{\theta}, \theta) = |\hat{\theta} - \theta|$, resulting in an expected loss of:

$$\begin{aligned} \mathbb{E}(L) &= \int |\hat{\theta} - \theta| p(\theta|x) \mathrm{d}\theta \\ &= \int_{-\infty}^{\hat{\theta}} (\hat{\theta} - \theta) p(\theta|x) \mathrm{d}\theta + \int_{\hat{\theta}}^{\infty} (\theta - \hat{\theta}) p(\theta|x) \mathrm{d}\theta \\ &= \hat{\theta} \left(\int_{-\infty}^{\hat{\theta}} p(\theta|x) \mathrm{d}\theta - \int_{\hat{\theta}}^{\infty} p(\theta|x) \mathrm{d}\theta \right) - \int_{-\infty}^{\hat{\theta}} \theta p(\theta|x) \mathrm{d}\theta + \int_{\hat{\theta}}^{\infty} \theta p(\theta|x) \mathrm{d}\theta \end{aligned}$$

Differentiating the above we get (using Feynman's differentiation under the equals sign):

$$\begin{aligned}\frac{\mathrm{d}L}{\mathrm{d}\hat{\theta}} &= 2\hat{\theta}p(\hat{\theta}|x) + \int_{-\infty}^{\hat{\theta}} p(\theta|x)\mathrm{d}\theta - \int_{\hat{\theta}}^{\infty} p(\theta|x)\mathrm{d}\theta - 2\hat{\theta}p(\hat{\theta}|x) \\ &= \int_{-\infty}^{\hat{\theta}} p(\theta|x)\mathrm{d}\theta - \int_{\hat{\theta}}^{\infty} p(\theta|x)\mathrm{d}\theta = 0\end{aligned}$$

which is true only when $\int_{-\infty}^{\hat{\theta}} p(\theta|x)\mathrm{d}\theta = 0.5$; in other words $\hat{\theta}$ is the median. For the data the median is 5.19.

Problem 7.3.2. Which of the above loss functions do you think is most appropriate, and why?

I would say that either the quadratic or linear loss functions are more appropriate than the binary loss. One could argue that losses to brain function are likely the proportional to the *volume* of tissue lost. Since a quadratic loss is closer to this (a cubic loss), then we might suppose that this is most appropriate.

Problem 7.3.3. Which loss function might you choose to be most robust to any situation?

Either the quadratic or linear losses since most problems exhibit a loss that increases in the distance away from the true value.

Problem 7.3.4. Following from the previous point, which type of posterior point measure might be most widely applicable?

Either the posterior mean or median.

Problem 7.3.5. Using the data estimate the loss under the three different regimes assuming that the true loss $L(\hat{\theta}, \theta) = (\hat{\theta} - \theta)^3$.

The mean loss from the mean is 151; from the mode it is 245; from the median it is 175.

Chapter 8

An introduction to distributions for the mathematically uninclined

8.1 Drug trials

We suppose that we are testing the efficacy of a certain drug which aims to cure depression, across two groups, each of size 10, with varying levels of the underlying condition: *mild* and *severe*. We suppose that the success rate of the drug varies across each of the groups, with $\theta_{mild} > \theta_{severe}$. We are comparing this with another group of 10 individuals, which has a success rate equal to the mean of the other two groups $\theta_{homogeneous} = \frac{\theta_{mild}+\theta_{severe}}{2}$.

Problem 8.1.1. Calculate the mean number of successful trials in each of the three groups.

Across each of the three groups:

- $\mathbb{E}[X_{mild}] = 10\theta_{mild}$
- $\mathbb{E}[X_{severe}] = 10\theta_{severe}$
- $\mathbb{E}[X_{homogeneous}] = 10\theta_{homogeneous}$

Problem 8.1.2. Compare the mean across the two heterogeneous groups with that of the single group of 10 homogeneous people.

Taking the mean of the means for each group,

$$\mathbb{E}[X_{combined}] = \frac{1}{2} \times 10(\theta_{mild} + \theta_{severe}) \tag{8.1}$$
$$= 10\theta_{homogeneous} \tag{8.2}$$
$$\tag{8.3}$$

In words, the mean outcome across the two groups is the same.

Problem 8.1.3. Calculate the variance of outcomes across each of the three groups.

The variance across each of the three groups is given by:

- $var(X_{mild}) = 10\theta_{mild}(1 - \theta_{mild})$
- $var(X_{severe}) = 10\theta_{severe}(1 - \theta_{severe})$
- $var(X_{homogeneous}) = 10\theta_{homogeneous}(1 - \theta_{homogeneous})$

Problem 8.1.4. How does the variance across both heterogeneous studies compare with that of a homogeneous group of the same sample size and same mean?

Here we need to use the law of total variance. This is because there are two sources of variance: that which is intra-group, and another which is between group,

$$var(X_{combined}) = \mathbb{E}[var(X|D)] + var(\mathbb{E}[X|D]), \tag{8.4}$$

where D means the depressive status of the particular subgroup.

Using this we have:

$$\begin{aligned} var(X_{combined}) &= \mathbb{E}[var(X|D)] + \mathbb{E}\left(\mathbb{E}[X|D]^2\right) - \left(\mathbb{E}\left(\mathbb{E}[X|D]\right)\right)^2 && (8.5)\\ &= \frac{1}{2} \times 10 \times \theta_{mild}(1 - \theta_{mild}) + \frac{1}{2}10 \times \theta_{severe}(1 - \theta_{severe}) && (8.6)\\ &+ \frac{1}{2} \times 10^2 \times \theta^2_{mild} + \frac{1}{2} \times 10^2 \times \theta^2_{severe} - 10^2 \times \theta^2_{homogeneous} && (8.7) \end{aligned}$$

Now supposing that we can write $\theta_{mild} = \theta_{homogeneous} - \epsilon$ and $\theta_{severe} = \theta_{homogeneous} + \epsilon$. We can then substitute this into the above yielding:

$$var(X_{combined}) = n\theta_{homogeneous}(1 - \theta_{homogeneous}) + \epsilon^2 n(n - 1) \tag{8.8}$$

Here $n = 10$, so the variance is greater than that of the homogeneous group. Note, the latter term disappears if $n = 1$ since there is no between-group variance!

Problem 8.1.5. Now consider the extension to a large number of trials, with the depressive status of each group unknown to the experimenter, but follows $\theta \sim beta(\alpha, \beta)$. Calculate the mean value of the beta distribution.

This can be calculated straightforwardly, and found to be $\frac{\alpha}{\alpha+\beta}$.

Problem 8.1.6. Which combinations of α and β would make the mean the same as that of a single study with success probability θ?

Setting these equal:

$$\frac{\alpha}{\alpha+\beta} = \theta \tag{8.9}$$

Rearranging this we get the following relationship:

$$\alpha = \frac{\beta\theta}{1-\theta} \tag{8.10}$$

Problem 8.1.7. How does the variance change, as the parameters of the beta distribution are changed, so as to keep the same mean of θ?

The variance of a beta distribution can be calculated as:

$$var(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{8.11}$$

This can be shown to be equal to:

$$var(\theta) = \frac{\theta(1-\theta)^2}{\beta+1-\theta} \tag{8.12}$$

Therefore, as $\beta \to \infty \implies var(\theta) \to 0$.

Problem 8.1.8. How does the variance of the number of disease cases compare to that of the a single study with success probability θ?

It is possible to work out the variance of the beta-binomial distribution, and one finds it equal to:

$$var(X|n,\alpha,\beta) = \frac{n\alpha\beta(\alpha+\beta+n)}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{8.13}$$

By recognising that $\theta = \frac{\alpha}{\alpha+\beta}$, and $1-\theta = \frac{\beta}{\alpha+\beta}$, the above expression can be written as:

$$var(X|n,\alpha,\beta) = n\theta(1-\theta)\frac{\alpha+\beta+n}{\alpha+\beta+1} \tag{8.14}$$

This can then be rearranged to yield:

$$\begin{aligned} var(X|n,\alpha,\beta) &= n\theta(1-\theta)\left[1+\frac{n-1}{\alpha+\beta+1}\right] && (8.15)\\ &= n\theta(1-\theta)+\epsilon && (8.16)\\ &\geq var(X|n,\theta) = n\theta(1-\theta) && (8.17) \end{aligned}$$

Therefore the variance of this distribution exceeds that of an equivalent binomial distribution. Hence, why it is called an over-dispersed distribution.

Problem 8.1.9. Under what conditions does the variance in disease cases tend to that from a binomial distribution?

Substituting in $\alpha = \frac{\beta\theta}{1-\theta}$,

$$var(X|n, \alpha, \beta) = n\theta(1-\theta) + \frac{(n-1)(1-\theta)}{1+\beta-\theta}, \tag{8.18}$$

meaning that as $\beta \to \infty$, the variance in disease cases tends to that of a binomial distribution.

8.2 Political partying

Suppose that in polls for an upcoming election there are three political parties that individuals for which can vote denoted by $\{A, B, C\}$ respectively.

Problem 8.2.1. If we assume independence amongst those individuals that are polled then what might likelihood might be choose?

A multinomial likelihood with probabilities of voting for each party given by (p_A, p_B, p_C).

Problem 8.2.2. In a sample of 10 individuals we find that the numbers of individuals who intend to vote for each party are $(n_A, n_B, n_C) = (6, 3, 1)$. Derive and calculate the maximum likelihood estimators of the proportions voting for each party.

The likelihood for this case is of the form,

$$L(p_A, p_B, p_C|n_A, n_B, n_C) = \frac{(n_A+n_B+n_C)!}{n_A!n_B!n_C!} p_A^{n_A} p_B^{n_B} p_C^{n_C}, \tag{8.19}$$

which on taking the log becomes,

$$l(p_A, p_B, p_C|n_A, n_B, n_C) = const + n_A \log p_A + n_B \log p_B + (n - n_A - n_B)\log(1 - p_A - p_B). \tag{8.20}$$

Differentiating with respect to p_A and finding the MLE,

$$\frac{\partial l}{\partial p_A} = \frac{n_A}{p_A} - \frac{n - n_A - n_B}{1 - p_A - p_B} = 0. \tag{8.21}$$

Then solving for the MLE we find $\hat{p}_i = \frac{n_i}{n}$ where $i \in \{A, B, C\}$. So in this case we have that $(\hat{p}_A, \hat{p}_B, \hat{p}_C) = (\frac{6}{10}, \frac{3}{10}, \frac{1}{10})$.

Problem 8.2.3. Graph the likelihood in (p_A, p_B) space.

See Figure 8.1.

Problem 8.2.4. If we specify a $Dirichlet(a, b, c)$ prior on the probability vector $\boldsymbol{p} = (p_A, p_B, p_C)$ the posterior distribution for a suitable likelihood is given by a $Dirichlet(a + n_A, b + n_B, c + n_C)$. Assuming a $Dirichlet(1, 1, 1)$ prior, and for the data given find the posterior distribution and graph it in (p_A, p_B) space.

The posterior in this case is given by a $Dirichlet(7, 4, 2)$ distribution (see Figure 8.1).

Problem 8.2.5. How do the posterior means compare with the maximum likelihood estimates?

The posterior means are given by, $\hat{p}_A^{post} = \frac{7}{7+4+2} = \frac{7}{13} < \frac{6}{10} = \hat{p}_A^{MLE}$. This is because the posterior reflects both the likelihood and the prior. The latter has most weight towards equal proportions of each category.

Problem 8.2.6. How does the posterior shape change if we use a $Dirichlet(10, 10, 10)$ prior?

The probability mass shifts over towards the peak of the prior (see middle row panel of Figure 8.1).

Problem 8.2.7. How does the posterior shape change if we use a $Dirichlet(10, 10, 10)$ prior but have data $(n_A, n_B, n_C) = (60, 30, 10)$?

The probability mass shifts over towards the peak of the likelihood (see bottom row panel of Figure 8.1).

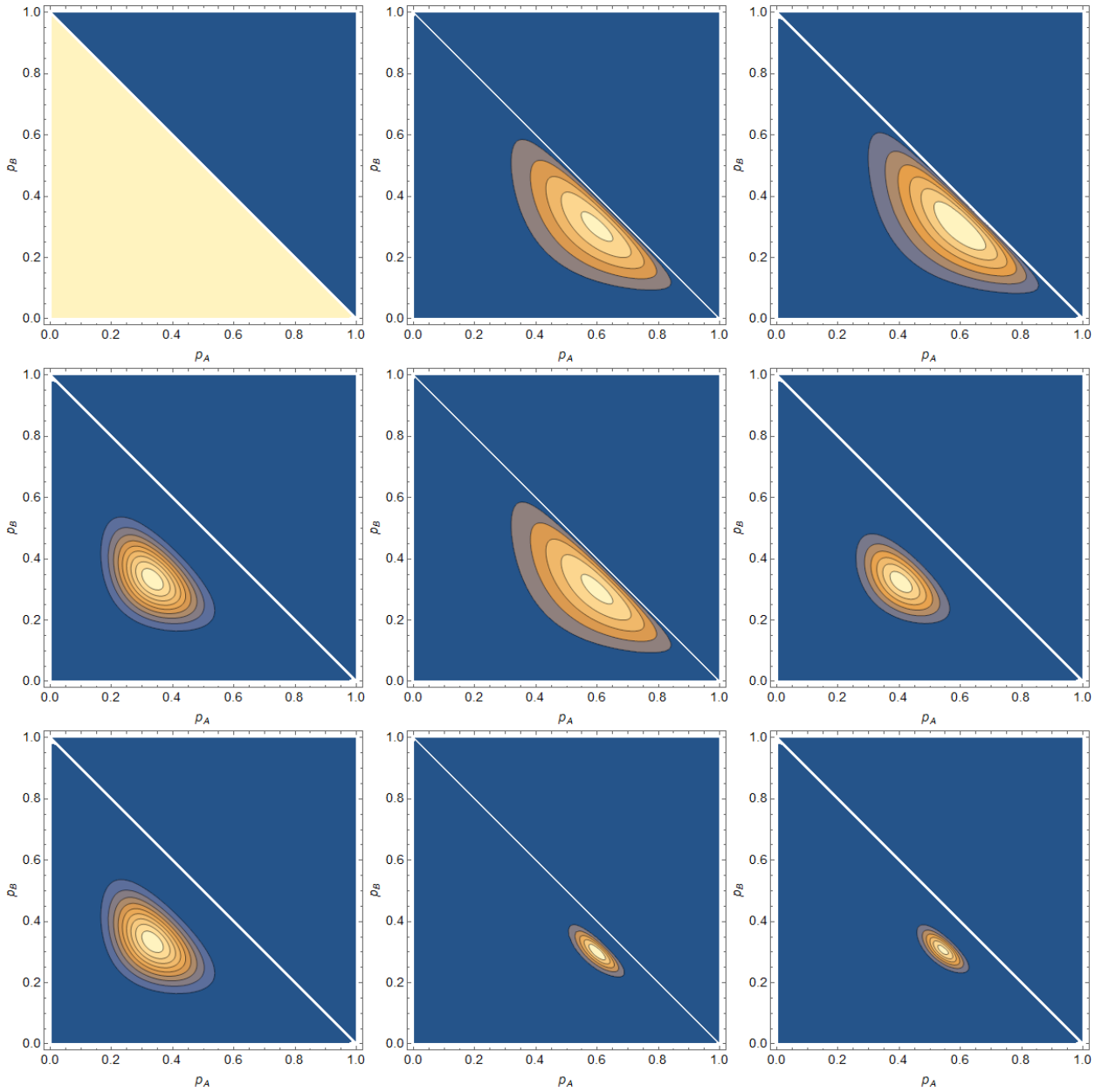

Figure 8.1: The priors (left), likelihoods (middle) and posteriors (right) for: Top, a $Dirichlet(1, 1, 1)$ prior and $(n_A, n_B, n_C) = (6, 3, 1)$; Middle: a $Dirichlet(10, 10, 10)$ prior and $(n_A, n_B, n_C) = (6, 3, 1)$; Bottom: a $Dirichlet(10, 10, 10)$ prior and $(n_A, n_B, n_C) = (60, 30, 10)$.

Chapter 9

Conjugate priors

9.1 The epidemiology of Lyme disease

Lyme disease is a tick-borne infectious disease spread by bacteria of species *Borrelia*, which are transmitted to ticks when they feed on animal hosts. Whilst fairly common in the US, this disease has recently begun to spread throughout Europe.

Imagine you are researching the occurrence of Lyme disease in the UK. As such, you begin by collecting samples of 10 ticks from fields and grasslands around Oxford, and counting the occurrence of the *Borrelia* bacteria.

Problem 9.1.1. You start by assuming that the occurrence of *Borrelia* bacteria in one tick is independent of that in other ticks. In this case, why is it reasonable to assume a binomial likelihood?

If we assume independence in disease between ticks (as well as assuming the underlying prevalence is the same across all surveyed terrains; i.e. identically-distributed), then because the data is discrete, and the sample size fixed $\implies$ **binomial** likelihood.

Problem 9.1.2. Suppose the number of *Borrelia*-positive ticks within each sample i is given by the random variable X_i, and that the underlying prevalence (amongst ticks) of this disease is θ. Write down the likelihood for sample i.

The likelihood is given by the binomial probability (through the equivalence principle):

$$\begin{aligned} L(\theta|X_i) &= Pr(X_i|\theta) \\ &= \binom{10}{X_i}\theta^{X_i}(1-\theta)^{10-X_i} \end{aligned}$$

Problem 9.1.3. Suppose that in your first sample of size 10 you find $X_1 = 1$ case of Borrelia. Graph the likelihood here and hence (by eye) determine the maximum likelihood estimate of θ.

In R this likelihood can be graphed using the following,

```
curve(dbinom(1, 10, x), 0, 1)
```

The likelihood is shown in Figure 9.1. The maximum likelihood estimate is at $\theta = 0.1$.

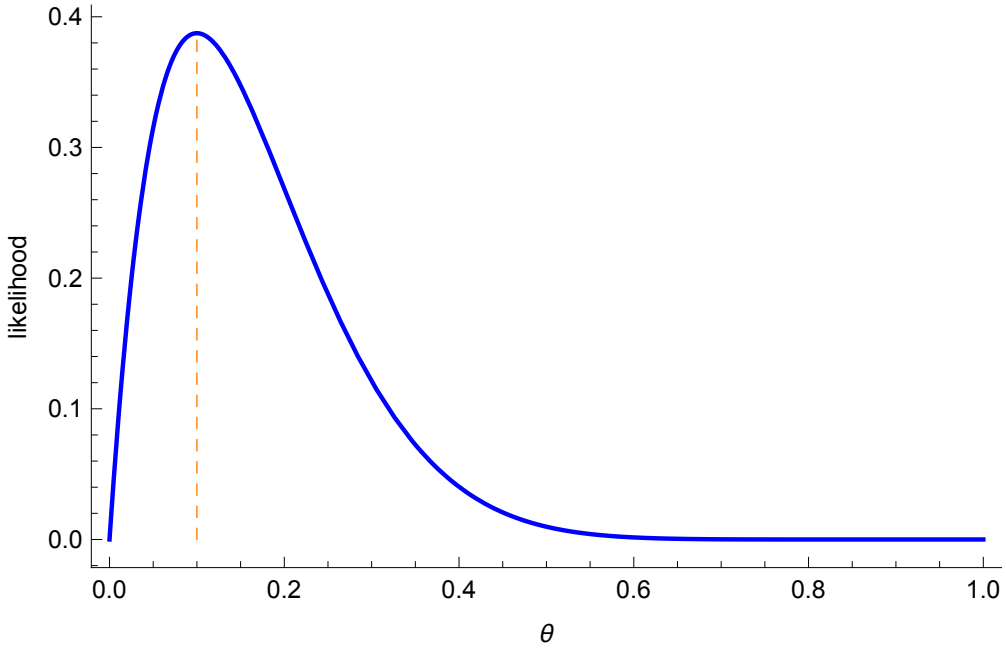

Figure 9.1: The likelihood for $X_1 = 1$ in the ticks example.

Problem 9.1.4. By numerical integration show that the area under the likelihood curve is about 0.09. Comment on this result.

In R this numerical integration can be carried out by the following,

```
integrate(function(x) dbinom(1, 10, x), 0, 1)
```

This is approximately $\frac{1}{11} \approx 0.09$. Therefore not a valid probability distribution!

Problem 9.1.5. Assuming that $\theta = 10\%$, graph the probability distribution (also known as the sampling distribution). Show that, in contrast to the likelihood, this distribution is a valid probability distribution.

This distribution can be graphed in R using,

```
lX <- seq(0, 10, 1)
plot(lX,sapply(lX, function(x) dbinom(x, 10, 0.1)),
     xlab="number of cases of bacteria out of 10", ylab="probability")
```

This is a discrete *probability* distribution shown in Figure 9.2. Since it is a discrete probability distribution we can check its validity by summing over all the probability masses,

```
lX <- seq(0, 10, 1)
sum(sapply(lX, function(x) dbinom(x, 10, 0.1))) == 1
```

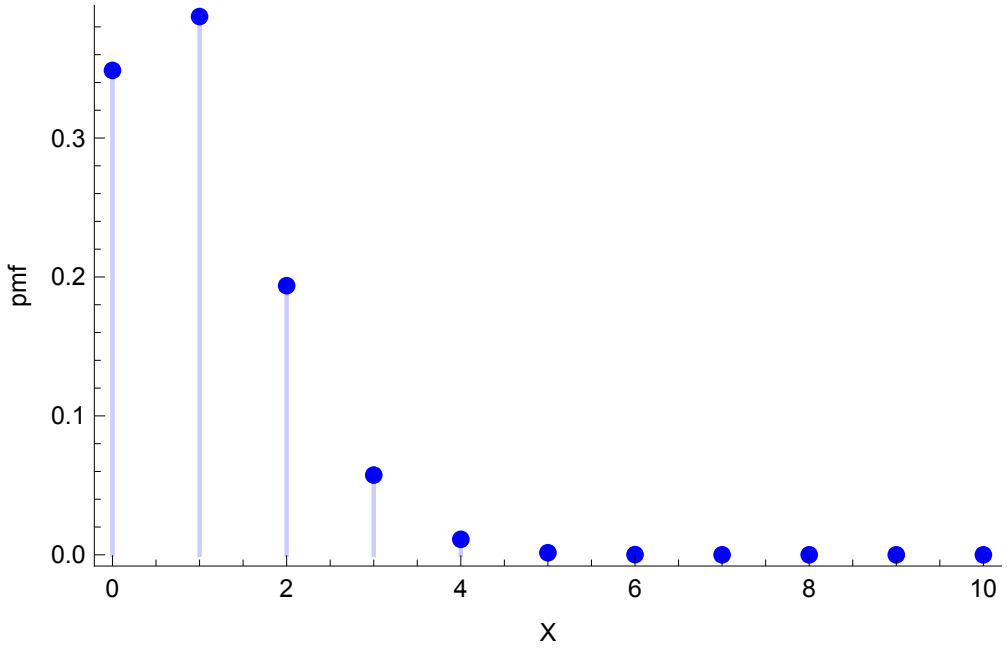

Figure 9.2: The sampling distribution for $\theta = 0.1$.

Problem 9.1.6. (Optional) Now assume that you do not know θ. Use calculus to show that the maximum likelihood estimator of the parameter, for a single sample of size 10 where we found X ticks with the disease is given by:

$$\hat{\theta} = \frac{X}{10} \tag{9.1}$$

(Hint: maximise the log-likelihood rather than the likelihood.)

We can write down the likelihood,

$$L(\theta|X) = \binom{10}{X}\theta^X(1-\theta)^{10-X}$$

Since the log is a monotonic transformation we can take the log of the likelihood, and maximise this instead. Taking the log we obtain,

$$l = log\ L(\theta|X) = \text{constants} + Xlog(\theta) + (10-X)log(1-\theta)$$

which we then differentiate to find the maximum,

$$\frac{\partial l}{\partial \theta} = \frac{X}{\hat{\theta}} - \frac{10 - X}{1 - \hat{\theta}} = 0 \tag{9.2}$$

which is obtained when $\hat{\theta} = \frac{X}{10}$.

Problem 9.1.7. A colleague mentions that a reasonable prior to use for θ is a $beta(a, b)$ distribution. Graph this for $a = 1$ and $b = 1$.

This is a continuous uniform distribution across (0,1), which can be obtained from the following R code,

```
curve(dbeta(x, 1, 1), 0, 1, xlab="theta", ylab="probability")
```

Problem 9.1.8. How does this distribution change as you vary a and b?

The mean is $\frac{a}{a+b}$. This can be obtained from R by doing,

```
?dbeta
```

and looking at the resultant help file. Therefore as $a \uparrow$ the mass of the distribution shifts to the right.

Problem 9.1.9. Prove that a $beta(a, b)$ prior is conjugate to the binomial likelihood, showing that the posterior distribution is given by a $beta(X + a, 10 - X + b)$ distribution.

- Likelihood:
$$X \sim \mathcal{B}(10, \theta) \implies p(X|\theta) \propto \theta^X (1 - \theta)^{10-X} \tag{9.3}$$
- For the prior assume a beta distribution (a reasonable choice if $\theta \in (0, 1)$):
$$\theta \sim beta(a, b) \implies p(\theta) \propto \theta^{a-1} (1 - \theta)^{b-1} \tag{9.4}$$
- Posterior:
$$\begin{aligned} p(\theta|X) &\propto p(X|\theta) \times p(\theta) \\ &\propto \theta^X (1 - \theta)^{10-X} \times \theta^{a-1} (1 - \theta)^{b-1} \\ &= \theta^{X+a-1} (1 - \theta)^{10-X+b-1} \end{aligned}$$

This has same θ-dependence as a $beta(X + a, 10 - X + b)$ density $\implies$ must be this distribution!

Problem 9.1.10. Graph the posterior for $a = 1$ and $b = 1$. How does the posterior distribution vary as you change the mean of the beta prior? (In both cases assume that $X = 1$.)

For $a = 1$ and $b = 1 \implies$ mean is $\frac{1+1}{10-1+1} = \frac{1}{5}$.

Problem 9.1.11. You now collect a larger dataset (encompassing the previous one) that has a sample size of 100 ticks in total; of which you find 7 carry *Borrelia*. Find and graph the new posterior using the conjugate prior rules for a $beta(1,1)$ prior and binomial likelihood.

For $a = 1$ and $b = 1 \implies beta(1 + 7, 100 - 7 + 1)$ posterior, whose mean is $\frac{1+7}{100+2} = \frac{8}{102} \approx 0.078$. The posterior is shown in Figure 9.3, of which a similar curve can be obtained in R by doing the following,

```
curve(dbeta(x, 1 + 7, 100 - 7 + 1), 0, 1, xlab="theta", ylab="probability")
```

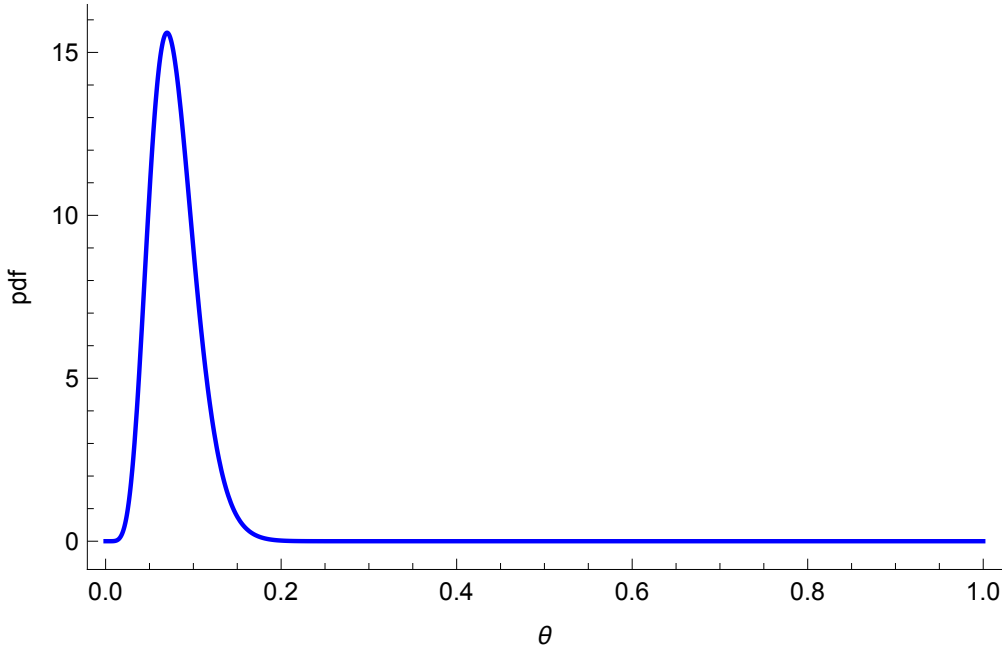

Figure 9.3: The posterior distribution for $X = 7$ out of a sample of 100 ticks.

Problem 9.1.12. You collect a second dataset of 100 ticks; this time finding that 4 carry the disease. Find and graph the new posterior (across both datasets) using the conjugate prior rules for a $beta(1,1)$ prior and binomial likelihood. How does it compare to the previous one?

The new likelihood is the product of the two samples' likelihoods, and so we find a $beta(1 + 11, 200 - 11 + 1)$ posterior. This results in a narrower posterior (see Figure 9.4), which can similarly be produced in R using,

```
curve(dbeta(x, 1 + 11, 200 - 11 + 1), 0, 1, xlab="theta", ylab="probability")
```

Problem 9.1.13. Now we will use sampling to estimate the posterior predictive distribution for a sample size of 100, using the posterior distribution obtained from the entire sample of 200 ticks (11 of which were disease-positive). To do this we will first sample a random value of θ from the posterior: so $\theta_i \sim p(\theta|X)$. We then sample a random value of the data X by sampling from the

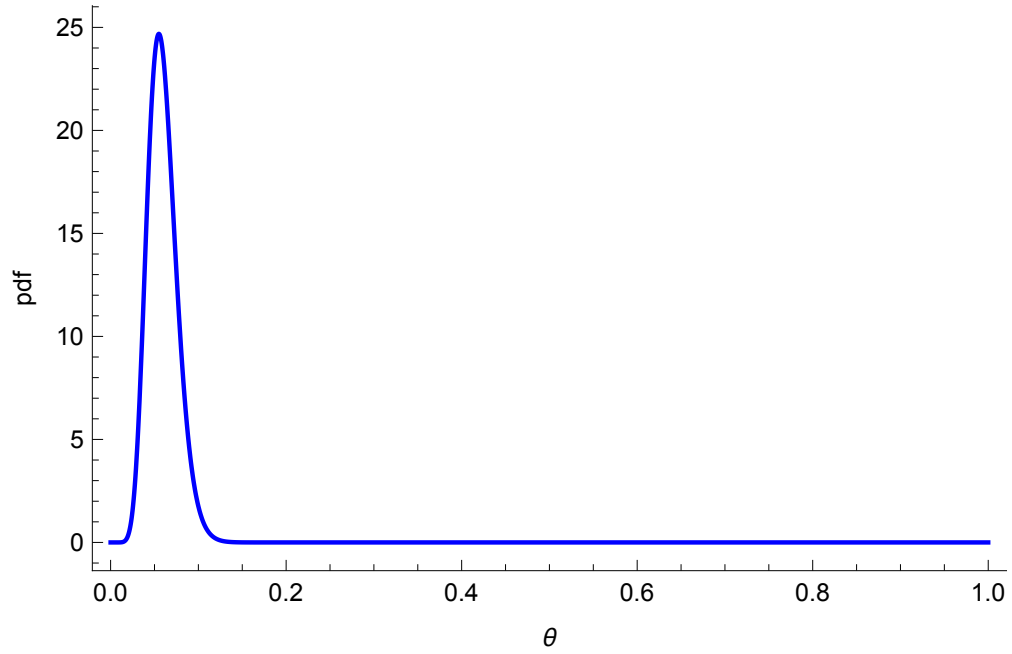

Figure 9.4: The posterior distribution for $X_1 = 7$ and $X_2 = 4$; each out of a sample of 100 ticks.

binomial sampling distribution $X_i \sim \mathcal{B}(100, \theta_i)$. We repeat this process a large number of times to obtain samples from this distribution. Follow the previous rules to produce 10,000 samples from the posterior predictive distribution, which we then graph using a histogram.

The posterior predictive distribution for a sample of 100 ticks is shown in Figure 9.5. I find the best way to do this is to create a function in R that does the above iteration,

```
fPosteriorPredictive <- function(aNumSamples){
  lX <- vector(length=aNumSamples)
  for(i in 1:aNumSamples){
    theta <- rbeta(1, 1 + 11, 200 - 11 + 1)
    X <- rbinom(1, 100, theta)
    lX[i] <- X
  }
  return(lX)
}
```

which we can then use to generate 10,000 posterior samples, then graph these using,

```
X <- fPosteriorPredictive(10000)
hist(X, breaks=seq(0, 100, 1), xlim = c(0, 20),
     xlab="number of disease-positive ticks")
```

Problem 9.1.14. Does our model fit the data?

Both the original data points are well contained within the posterior predictive distribution. Thus the model looks like a reasonable fit.

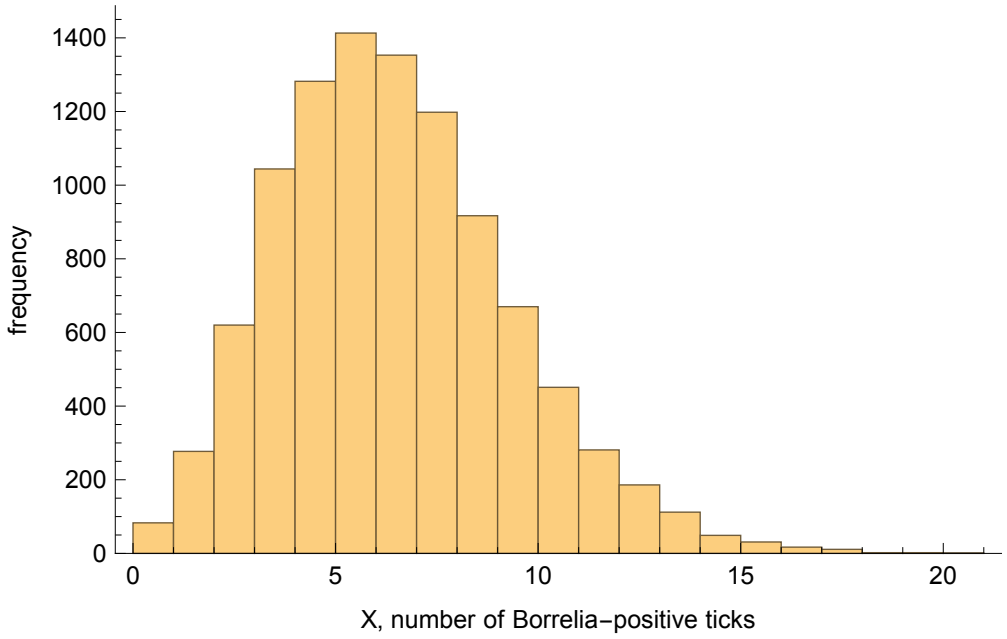

Figure 9.5: Samples from the posterior distribution predictive distribution for $X_1 = 7$ and $X_2 = 4$; for a sample size of 100 ticks.

Problem 9.1.15. Indicate whether you expect this model to hold across future sampling efforts.

Whilst it is a bit imprudent to comment on this, I would argue in this case that the assumption of **independence** of Borrelia amongst ticks is a bit suspect. In particular, the presence of one disease-positive tick makes it more likely that another - nearby - tick will catch the disease whilst blood-feeding. A more robust model might be preferable, for example the beta-binomial.

Problem 9.1.16. If we assume a uniform prior on θ, the probability that a randomly sampled tick carries Lyme disease, what is the shape of the prior for θ^2? (This is the probability that 2/2 ticks carry Lyme disease.)

Hint: do this either using Jacobians (hard-ish), or by sampling (easy-ish).

Assume a change of variables $y = g(x)$, how does the density change? We need the Jacobian of the transformation:

$$f_Y(y) = f_X(g^{-1}(y))g'^{-1}(y) \tag{9.5}$$

$$= f_X(g^{-1}(y))|\frac{dx}{dy}| \tag{9.6}$$

In this case, $\phi = \theta^2$:

$$f_\Phi(\phi) = f_\theta(\sqrt{\phi})\frac{1}{2}\phi^{-\frac{1}{2}} \tag{9.7}$$

$$= 1 \times \frac{1}{2}\phi^{-\frac{1}{2}} \tag{9.8}$$

$$= \frac{1}{2}\phi^{-\frac{1}{2}} \tag{9.9}$$

Alternatively do this by sampling from a uniform prior for θ in R, then squaring each result,

```
fThetaSquared <- function(aNumSamples){
  lThetaSquared <- vector(length=aNumSamples)
  for(i in 1:aNumSamples){
    theta <- rbeta(1, 1, 1)
    lThetaSquared[i] <- theta ^ 2
  }
  return(lThetaSquared)
}
# Draw samples and graph result
theta <- fThetaSquared(100000)
hist(theta, 100, xlab="theta-squared")
```

9.2 Epilepsy

In the data file `conjugate_epil.csv` there is a count of seizures for 112 patients with epilepsy who took part in a study [11]. Assume a) the underlying rate of seizures is the same across all patients, and b) the event of a seizure occurring is independent of any other seizures occurring.

Problem 9.2.1. Under these assumptions what model might be appropriate for this data?

A Poisson distribution.

Problem 9.2.2. Write down the likelihood for the data.

The likelihood for a single observation x is given by:

$$L(\theta|x) = \frac{\theta^x e^{-\theta}}{x!} \tag{9.10}$$

For a data vector $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ if we assume independence between our observations we have:

$$L(\theta|\boldsymbol{x}) = \prod_{i=1}^{n} \frac{\theta^{x_i} e^{-\theta}}{x_i!} \tag{9.11}$$

Problem 9.2.3. Show that a gamma prior is conjugate to this likelihood.

The gamma distribution has the functional form:

$$p(\theta) \propto \theta^{\alpha-1} e^{-\beta\theta} \tag{9.12}$$

The posterior then has the functional form:

$$\begin{aligned} p(\theta|\boldsymbol{x}) &\propto \theta^{\alpha-1} e^{-\beta\theta} \times \prod_{i=1}^{n} \frac{\theta^{x_i} e^{-\theta}}{x_i!} \\ &\propto \theta^{\alpha-1+\sum_{i=1}^{n} x_i} \times e^{-(\beta+n)\theta} \end{aligned}$$

Which is the same θ dependence as a $\Gamma(\alpha+\sum_{i=1}^{n} x_i, \beta+n)$ distribution $\implies$ this must be the posterior distribution! Therefore the posterior is a gamma distribution as well as the prior $\therefore$ conjugate.

Problem 9.2.4. Assuming a $\Gamma(4, 0.25)$ (with a parameterisation such that it has mean of 16) prior. Find the posterior distribution, and graph it.

See above problem for derivation of the posterior density. The graph of the posterior should look like Figure 9.6.

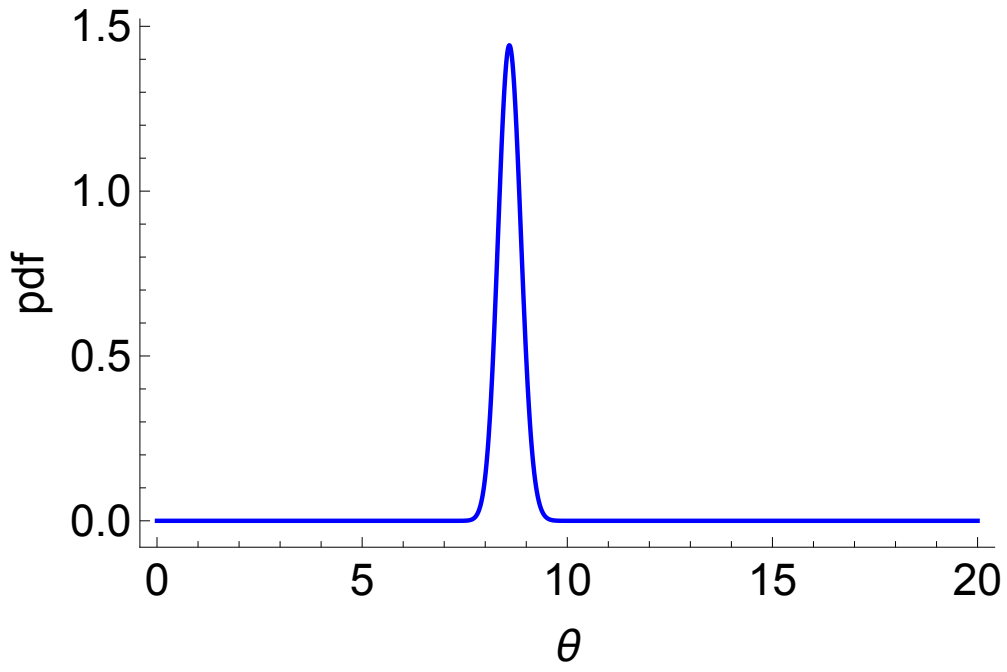

Figure 9.6: The posterior for the epilepsy example.

Problem 9.2.5. Find or look-up the posterior predictive distribution, and graph it.

The posterior predictive distribution is a negative binomial - this can be derived by:

$$\begin{aligned} p(\tilde{x}|x) &= \int p(\tilde{x}|\theta, \boldsymbol{x}) \times p(\theta|\boldsymbol{x}) \mathrm{d}\theta \\ &= \int p(\tilde{x}|\theta) \times p(\theta|\boldsymbol{x}) \mathrm{d}\theta \\ &\ldots \end{aligned}$$

where ... can be found via Googling. The posterior predictive distribution turns out to be $NB(\sum_{i=1}^{n} x_i + \alpha, \beta + n)$, where (α, β) are the parameters of the gamma prior distribution. The graph is shown in Figure 9.7.

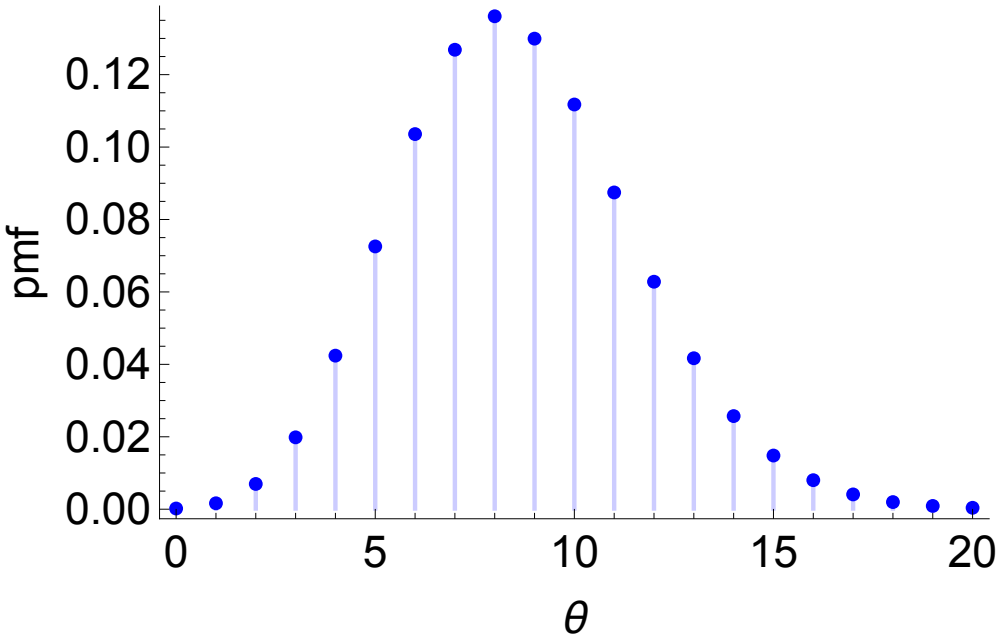

Figure 9.7: Posterior predictive distribution for seizure data.

Problem 9.2.6. Comment on the suitability of the model to the data.

In Figure 9.8 we see that the real data is much more dispersed than the simulated. This is likely for a number of reasons: for example, the event of a seizure is not likely independent of others (they come in clusters); also the rate of seizures varies between subjects (in other words the data are not exchangeable). Amongst other reasons these suggest that a Poisson model is not well suited here, and we would be better off using a more robust distribution for the likelihood, for example the negative binomial.

Figure 9.8: Comparing actual vs simulated seizures.

9.3 Light speed

The data file `conjugate_newcomb.csv` provides Simon Newcombs (1882) measurements of the passage time (in millionths of a second) it took light to travel from his lab to a mirror on the Washington Monument, and back again. The distance of the path travelled is about 7.4km. The primary goal of this experiment is to determine the speed of light, and to quantify the uncertainty of the measurement. We assume there are a multitude of factors that additively result in measurement error for the passage time.

Problem 9.3.1. Why might a normal distribution be appropriate here?

There are a range of factors that influence the measurement of the passage time. If these factors are roughly independent, and they affect the measurement additively, then the (Lindberg-Lévy) central limit theorem applies.

Problem 9.3.2. Write down the likelihood for all the data.

The likelihood of a single data point x is given by:

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{9.13}$$

If we assume measurements are independent, and identically-distributed then we just need to multiply together the individual likelihoods:

$$L(\mu, \sigma|\boldsymbol{x}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \tag{9.14}$$

Problem 9.3.3. Derive the maximum likelihood estimators of all parameters.

It's easiest to first take the log:

$$l(\mu, \sigma|\boldsymbol{x}) = -\frac{n}{2} log(2\pi) - \frac{n}{2} log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2 \tag{9.15}$$

Then maximising this function over (μ, σ^2), we find that:

$$\hat{\mu} = \bar{x}$$
$$\hat{\sigma^2} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

Problem 9.3.4. Based on the likelihood function what functional form for the prior $p(\mu, \sigma^2)$ would make it conjugate?

We want a prior that when multiplied by a normal gives a distribution of the same family. There are a few choices here, but the only one that is a valid probability distribution is a normal-inverse-gamma or normal-inverse-chi-squared (they are both the same thing).

Problem 9.3.5. Assuming a decomposition of the prior $p(\mu, \sigma^2) = p(\sigma^2) \times p(\mu|\sigma^2)$, what priors might we use?

Again a normal inverse gamma. You could use an improper $p(\sigma^2) \propto \frac{1}{\sigma^2}$ but it's better to use fully-valid probability distributions.

Problem 9.3.6. (Difficult) Using these priors, find the parameters of the posterior distribution.

Look it up in Gelman [5].

Problem 9.3.7. Comment on the suitability of the model to the data. (You can use the ML estimates here, or if you're feeling ambitious, the full posterior predictive distribution.)

Using the posterior predictive simulate data and compare with the actual we see that the normal distribution is not sufficiently robust. We would be better using a Student t distribution.

Chapter 10

Evaluation of model fit and hypothesis testing

10.1 WHO's reported novel disease outbreaks

Suppose that you are interested in modelling the number of outbreaks of novel diseases that the WHO reports each year. Since these outbreaks are of new diseases, you assume that you can model the outbreaks as **independent** events, and hence decide to use a Poisson likelihood; $X_t \sim \text{Poisson}(\lambda)$, where X_t is the number of outbreaks in year t, and λ is the mean number of outbreaks.

Problem 10.1.1. You decide to use a $\Gamma(3, 0.5)$ prior for the mean parameter (λ) of your Poisson likelihood (where a $\Gamma(\alpha, \beta)$ is defined to have a mean of $\frac{\alpha}{\beta}$). Graph this prior.

This can be done in R using the following command,

```
curve(dgamma(x, 3, 0.5), 0, 20, xlab='lambda', ylab='pdf')
```

Problem 10.1.2. Suppose that the number of new outbreaks over the past 5 years is $X = (3, 7, 4, 10, 11)$. Using the conjugate prior rules for a Poisson distribution with a gamma prior, find the posterior and graph it.

Hint: look at Table 9.1 in the main text.

The posterior distribution is given by a $\Gamma(3 + \sum_{t=1}^{5} X_t, 0.5 + 5)$ distribution. This can be graphed in R using,

```
X <- c(3, 7, 4, 10, 11)
curve(dgamma(x, 3 + sum(X), 0.5 + length(X)),0 , 20, xlab='lambda', ylab='pdf')
```

It has a peak at $<\sim 7$, near to the mean of the data.

Problem 10.1.3. Generate 10,000 samples from the posterior predictive distribution, and graph the distribution. To do this we first independently sample a value λ_i from the posterior distribution, then sample a value of X from a $Poisson(\lambda_i)$ distribution. We carry out this process 10,000 times.

Hint: use R's `rgamma` and `rpois` functions to draw (pseudo-)independent samples from the gamma and Poisson distributions respectively.

I prefer to do this by creating a function in R that implements the above then plots the result,

```
fPosteriorPredictive <- function(numSamples, alpha, beta){
  X <- vector(length=numSamples)
  for(i in 1:numSamples){
    aLambda <- rgamma(1, alpha, beta)
    X[i] <-rpois(1, aLambda)
  }
  return(X)
}

PPC.X <- fPosteriorPredictive(10000, 3 + sum(X), 0.5 + length(X))
hist(PPC.X, xlab='X', main='10,000 posterior predictive samples')
```

Problem 10.1.4. Compare the actual data with your 10,000 posterior predictive samples. Does your model fit the data?

The most extreme points of the data are the years with 3 and 11 outbreaks respectively. We can compare our posterior predictive samples with these extrema in R,

```
mean(PPC.X >= 11)
mean(PPC.X <= 3)
```

and find that roughly 10% of samples are greater than or equal to 11, and approximately the same proportion are less than or equal to 3. These Bayesian p values aren't too close to 0, and so our data appears to fit the data reasonably well.

Problem 10.1.5. (Optional) Can you think of a better posterior predictive check to carry out on the data?

A better posterior predictive check would generate 10,000 samples of 5 observations, and count the number where the minimum point is 3 **and** the maximum is 11 (or more extreme). To do this I implemented a new function,

```
fPosteriorPredictiveGeneral <- function(numObsPerSample, numSamples,
                                        alpha, beta){
  X <- matrix(nrow=numSamples, ncol=numObsPerSample)
  for(i in 1:numSamples){
    aLambda <- rgamma(1, alpha, beta)
    X[i, ] <-rpois(numObsPerSample, aLambda)
  }
  return(X)
}

aNumSamples <- 10000
```

```
PPC.better <- fPosteriorPredictiveGeneral(5, aNumSamples,
                                          3 + sum(X), 0.5 + length(X))
lIndicator <- vector(length=aNumSamples)
for(i in 1:aNumSamples)
  lIndicator[i] <- ifelse(min(PPC.better[i, ]) <= 3 &
                          max(PPC.better[i, ]) >= 11,
                          1, 0)
mean(lIndicator)
```

and you should get about 10% here. So it still looks like our model fits the data ok.

Problem 10.1.6. The WHO issues a press release where they state that the number of novel disease outbreaks for this year was 20. Use your posterior predictive samples to test whether your model is a good fit to the data.

Since we are just looking at a single data point we can use our simpler posterior predictive function to generate samples (or just reuse the previously-generated sample),

```
fPosteriorPredictive <- function(numSamples, alpha, beta){
  X <- vector(length=numSamples)
  for(i in 1:numSamples){
    aLambda <- rgamma(1, alpha, beta)
    X[i] <-rpois(1, aLambda)
  }
  return(X)
}

PPC.X <- fPosteriorPredictive(10000, 3 + sum(X), 0.5 + length(X))
mean(PPC.X >= 20)
```

where you should obtain a p value of less than 1%, indicating model misfit. This is a test of out-of-sample predictive capability, and so we would expect this p value to be more extreme than the within-sample one that we calculate below.

Problem 10.1.7. By using your previously determined posterior as a prior, update your posterior to reflect the new datum. Graph the PDF for this new distribution.

The new posterior here is a $\Gamma(3 + 35 + 20, 0.5 + 5 + 1)$ distribution,

```
curve(dgamma(x, 3 + sum(X) + 20, 0.5 + 5 + 1),
      0, 20, xlab='lambda', ylab='pdf')
```

Problem 10.1.8. Generate posterior predictive samples from your new posterior and use it to test the validity of your model.

Here I would generate 10,000 samples of 6 observations and count the number of times that you generate 20 or more cases in a particular year.

```
PPC.better <- fPosteriorPredictiveGeneral(6, aNumSamples,
                                          3 + sum(X) + 20,
                                          0.5 + 5 + 1)
lIndicator <- vector(length=aNumSamples)
for(i in 1:aNumSamples)
  lIndicator[i] <- ifelse(max(PPC.better[i, ]) >= 20, 1, 0)
mean(lIndicator)
```

where again the p value is less than 5% and hints at model misfit. This a within-sample measure of predictive capability of the model.

Problem 10.1.9. Would you feel comfortable using this model to predict the number of disease outbreaks next year?

No! Even the within-sample prediction is poor. It's probably that some of these outbreaks are related to one another – either they are different strains from a common disease, or they are the result of a common exogenous shock (e.g. civil war).

10.2 Sleep-deprived reactions

These data are from a study described in Belenky et al. (2003) [2] that measured the effect of sleep deprivation on cognitive performance. Eighteen subjects were chosen from a population of interest (lorry drivers) who were restricted to 3 hours of sleep during the trial. On each day of the experiment their reaction time to a visual stimulus was measured. The data for this example is contained in `evaluation_sleepstudy.csv` and consists of three variables, *Reaction*, *Days* and *Subject ID*, which measure the reaction time of a given subject on a particular day.

A simple model that explains the variation in reaction times is a linear regression model of the form:

$$R(t) \sim \mathcal{N}(\alpha + \beta t, \sigma) \tag{10.1}$$

where $R(t)$ is the reaction time on day t of the experiment across all observations.

Problem 10.2.1. By graphing all the data, critically assess the validity of the model to the data.

A simple graph of the time against reaction time is a first starter here. From this it looks like there may be some heteroscedasticity (higher variance) at later times. This can be done in R using,

```
library(ggplot2)
df <- read.csv('evaluation_sleepstudy.csv')
ggplot(data=df, aes(x=Days, y=Reaction)) + geom_point() +
  geom_smooth(method='lm')
```

Problem 10.2.2. Graph the data at the individual subject data using R's "lattice" package, or otherwise. What does this suggest about assuming a common β across all participants?

Using a lattice plot (see Figure 10.1),

```
xyplot(Reaction ~ Days | Subject, df, type=c("g", "p", "r"),
       index=function(x, y) coef(lm(y ~ x))[1],
       xlab="Days of sleep deprivation",
       ylab="Average reaction time (ms)",
       aspect="xy")
```

From an examination of the data at this level it is clear that there is considerable variability in the performance of the participants. As such, any attempts to lump the data together and apply a single analysis to it are going to suffer from considerable participant-level biases.

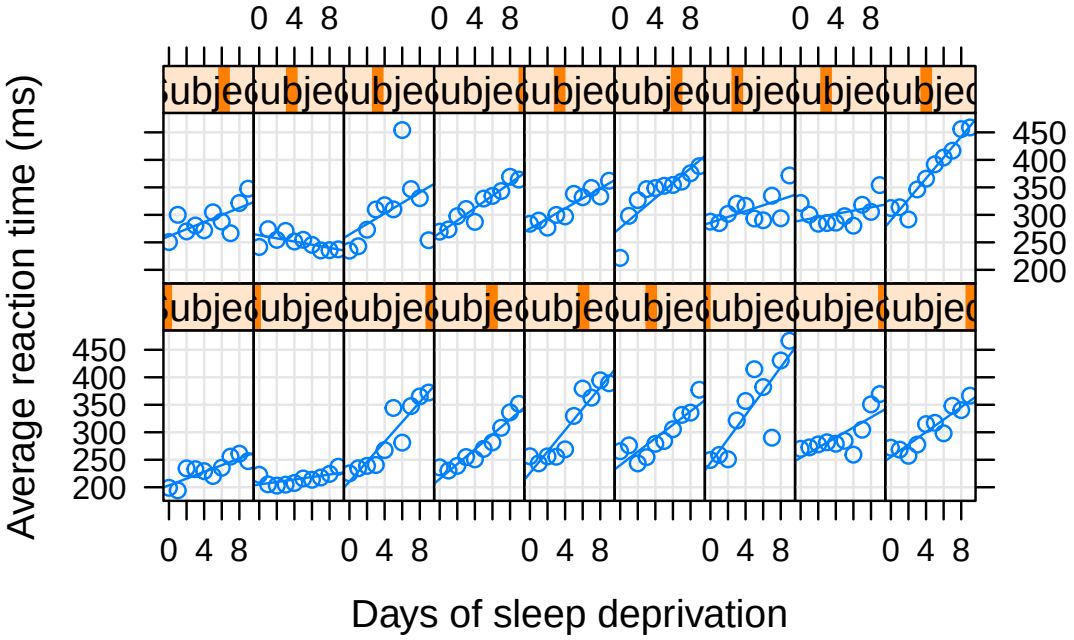

Figure 10.1: Reaction times versus days of sleep deprivation at the participant level.

Problem 10.2.3. The above model has been fit to the data using MCMC, with 2000 samples from the posterior distribution for (α, β, σ) contained in the file `evaluation_sleepPosteriors.csv`. Generate samples from the posterior predictive distribution, and visualise them in an appropriate way.

These are shown in Figure 10.2. It is important here to show the time aspect of the data; just lumping it all together in a histogram misses the point.

Problem 10.2.4. How does the posterior predictive data compare to the actual data?

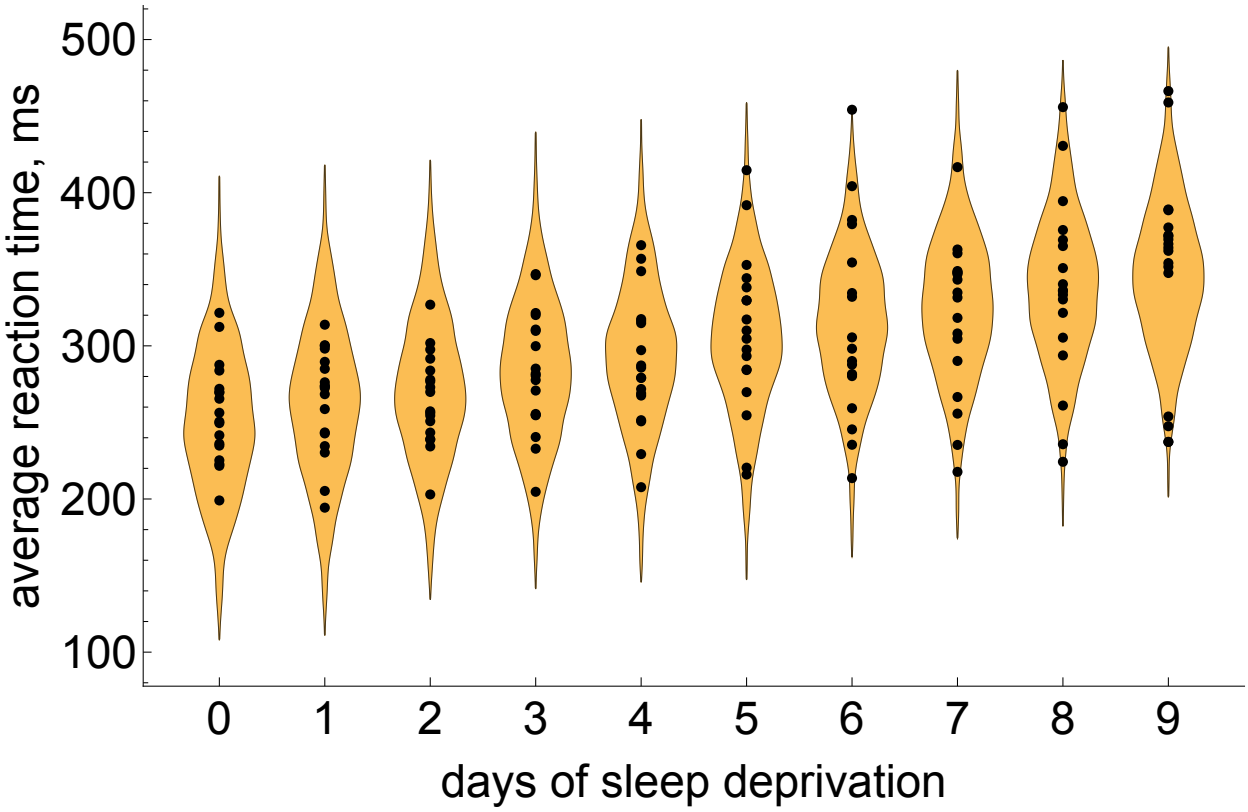

Figure 10.2: Posterior predictive distributions (orange) vs data (black).

The key here is to look at the data at the subject level. Averaging over all subjects makes it look like our model is doing ok, but this masks the (sometimes) very poor performance at the individual subject level (see Figure 10.3 for one example of this for subject 310).

Problem 10.2.5. How (if at all) do the posterior predictive checks suggest we need to change our model?

Hierarchical model where we allow there to be inter-subject variability in the effect of sleep deprivation on reaction time (β).

10.3 Discoveries data

The file `evaluation_discoveries.csv` contains data on the numbers of "great" inventions and scientific discoveries in each year from 1860 to 1959 [1]. The aim of this problem is for you to build a statistical model that provides a reasonable approximation to this series. As such, you will need to choose a likelihood, specify a prior on any parameters, and go through and calculate a posterior. Once you have a posterior, you will want to carry out posterior predictive checks to see that your model behaves as desired.

Answer: first plot the data! Both a time series and histogram are useful here (see Figure 10.4). To me the left hand plot suggests that there is some temporal autocorrelation in the data (perhaps

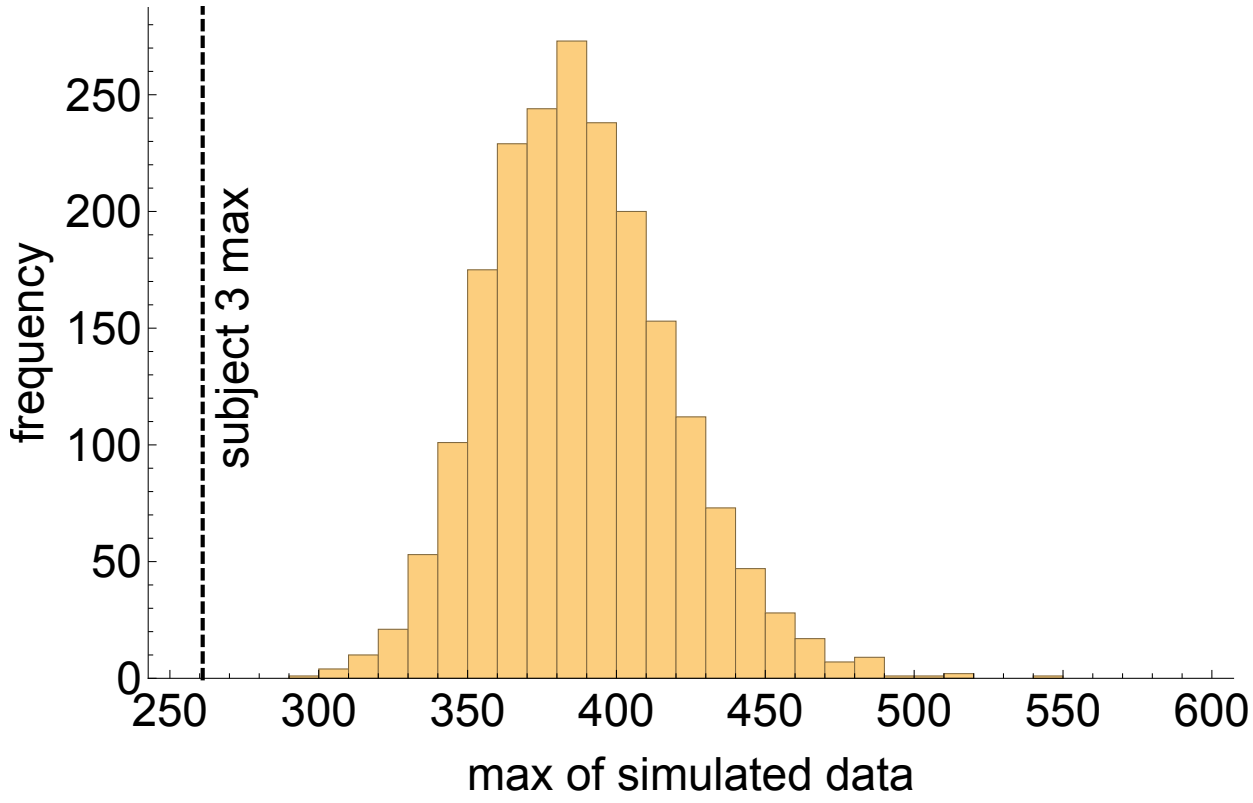

Figure 10.3: Posterior predictive simulated max (orange) versus the maximum of subject 310 (dashed line).

invalidating an assumption of independence, and/or identical distribution). The histogram would seem to support this claim, since the variance is fairly obviously greater than the mean. I also plot an autocorrelogram of the data which suggests that there is autocorrelation in the series.

Now make some assumptions about the occurrence of discoveries; namely that they are independent and identically-distributed over time. Both of these assumptions may be suspect: independence may be violated (as I indicate above) if one discovery leads to another; identical distribution may be invalidated if technological progress leads to an increased rate of discoveries at some points in time.

However, it is not a bad idea to start with making these assumptions, under the supposition that they may be suspect. Our aim is to make the simplest model that explains the data, and so we don't want to jump straight to a more complex model unless we know for sure that the simple one fails.

If we do make the above assumptions then a Poisson model is a reasonable starting point. If we use a Poisson model, then we may as well use the conjugate prior; a gamma distribution. The results of assuming this framework are shown in Figure 10.5; where we see a tight posterior centred around a mean of 3 discoveries per year.

Carrying out some PPCs here using the posterior predictive distribution from the Poisson likelihood model we find that our model is *unable* to generate the same amount of variation seen in the data (Figure 10.6). This suggests that one or more of the assumptions on which our data are based are

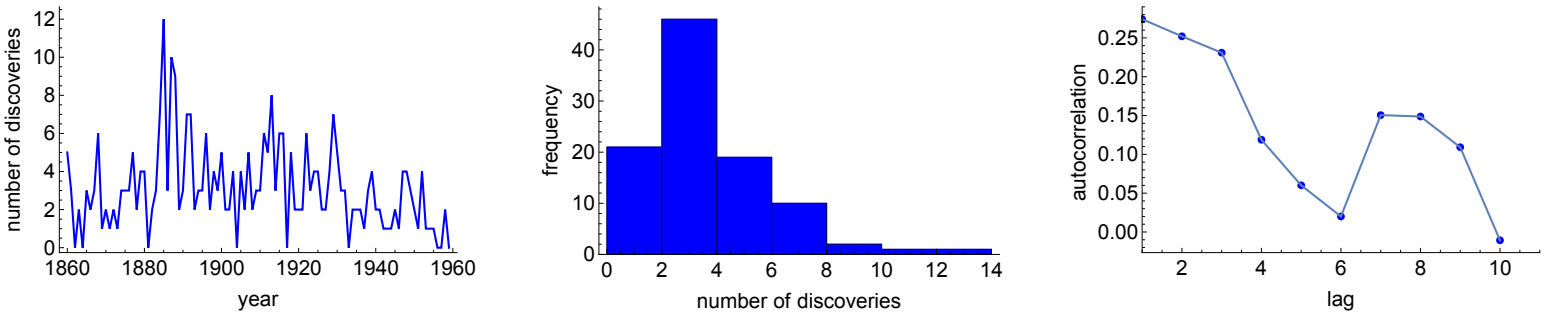

Figure 10.4: Characteristics of the "discoveries" data set.

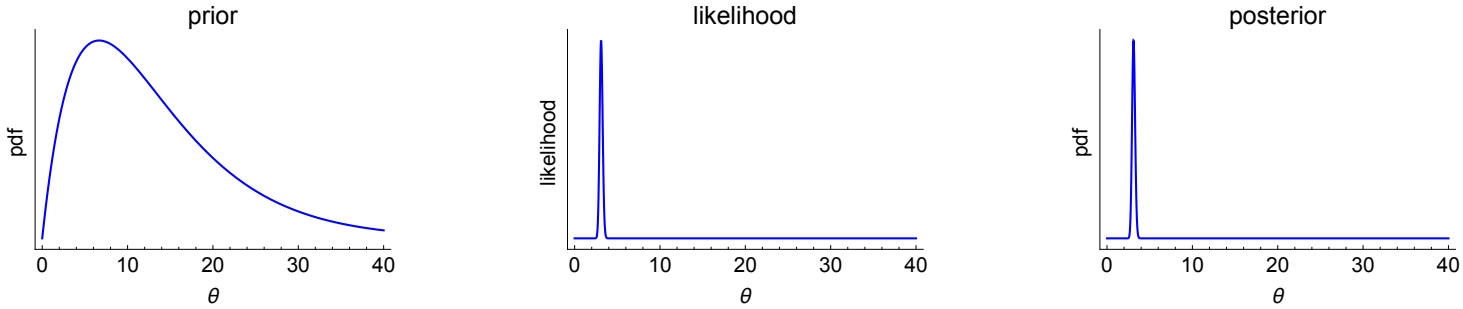

Figure 10.5: Prior, likelihood and posterior for a Poisson likelihood and $\Gamma(2, 0.15)$ prior for the discoveries dataset.

invalid.

There are multiple ways forward from here. To me there are two approaches that "jump out": **a.** use a sampling distribution that allows for non-independent events, but does not explicitly model the cycles of discovery; **b.** explicitly model the latent rate of discovery rate. Approach **a.** would suggests a negative binomial likelihood, and would certainly allow for the range in the data to be replicated well. However, I fear that such an approach - by ignoring the fact that the rate of discoveries changes through time - would fail to capture the intervals of high discovery rate that we see in the data. In other words those times (for example, between 1880 and 1890) where there is a persistently high rate of discovery. Approach **b.** would be more comprehensive and would perhaps use a negative binomial sampling model for each year, but allow its mean to vary over time. So if we imagine that the mean of the process at time t is θ_t, then we might assume:

$$\theta_t = \rho\theta_{t-1} + \epsilon_t \tag{10.2}$$

So an AR(1) process explicitly. both of these approaches favour a MCMC approach (particularly the AR(1) process one). As such, I have not tried either these investigations myself, as I wouldn't necessarily expect a student to be able to do these at this stage.

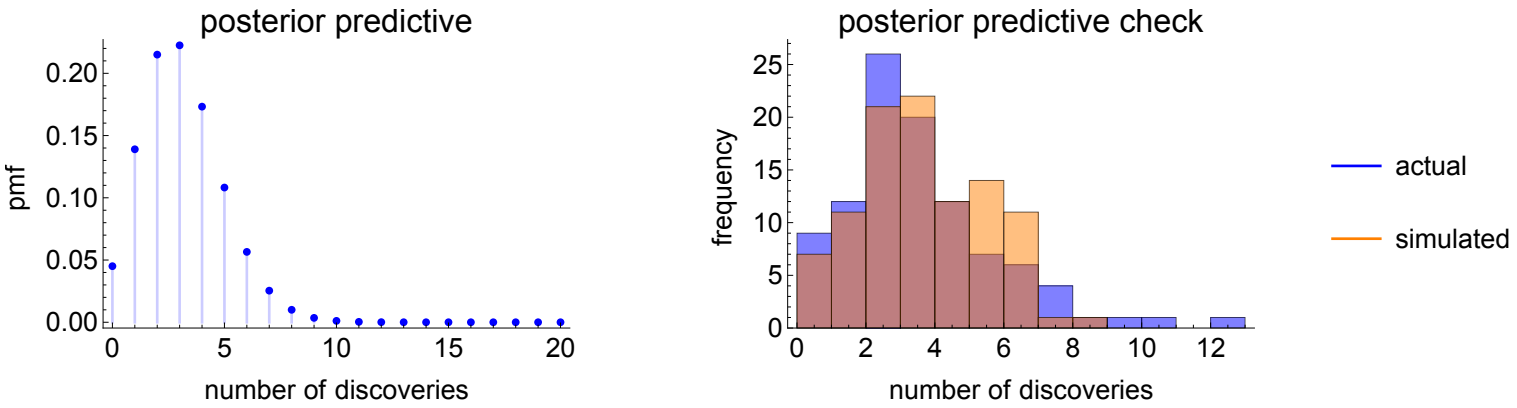

Figure 10.6: The posterior predictive distribution (left) and a posterior predictive comparison of the actual data with a simulated set (right).

10.4 Marginal likelihood of voting

Suppose that we collect survey data where respondents are asked to indicate for whom they will vote in an upcoming election. Each poll consists of a sample size of 10 and we collect the following data for 20 such polls: $\{2, 7, 4, 5, 4, 5, 6, 4, 4, 4, 5, 6, 5, 7, 6, 2, 4, 6, 6, 6\}$. We model each outcome as having been obtained from a $X_i \sim \mathcal{B}(10, \theta)$ distribution.

Problem 10.4.1. Find the posterior distribution where we specify $\theta \sim beta(a, 1)$ as a prior. Graph how the posterior changes as $a \in [1, 10]$.

The posterior distribution is given by (because of conjugacy): $\theta \sim beta(a + \sum X_i, 1 + \sum N_i - \sum X_i)$. The graph of the posterior as a function of a is shown in Figure 10.7, where we note the relative insensitivity to the prior.

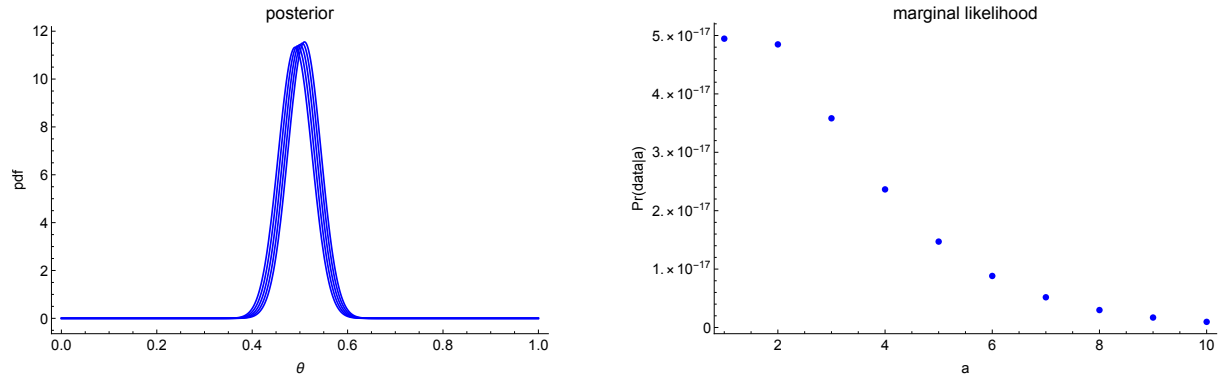

Figure 10.7: The posterior distribution (left) and the marginal likelihood (right) as a function of prior parameter a. Here the prior specified is $\theta \sim beta(a, 1)$. In the left hand graph the different lines correspond to different choices of a.

Problem 10.4.2. Graph the marginal likelihood as a is increased between 1 and 10 (just use integer values).

See Figure 10.7.

Problem 10.4.3. Calculate the Bayes factor where we compare the model where $a = 1$ to that when $a = 10$? Hence comment on the use of Bayes factors as a method for choosing between competing models.

This is approximately,

$$BF = \frac{4.94 \times 10^{-17}}{9.64 \times 10^{-19}} \approx 51. \tag{10.3}$$

So we see that there is a strong sensitivity of Bayes factors to choice of priors, even if the posterior is relatively insensitive.

Chapter 11

Making Bayesian analysis objective?

11.1 Jeffreys prior for a normal likelihood

Suppose that we are modelling the result of a medical test, which to a suitable approximation can be regarded as being continuous and unbounded. We suppose that a normal probability model is a reasonable sampling model to use here, $X_i \sim \mathcal{N}(\mu, \sigma)$, where μ is unknown but σ is known (perhaps based on the results of many previous tests).

Problem 11.1.1. Write down the likelihood for a single observation.

$$L(\mu|X_i, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}} \tag{11.1}$$

Problem 11.1.2. Find the information matrix (here a scalar).

First calculate the log-likelihood for a single observation,

$$l(\mu|X_i, \sigma) = const - \frac{(X_i - \mu)^2}{2\sigma^2}. \tag{11.2}$$

Differentiating this with respect to μ we obtain,

$$\frac{\partial l}{\partial \mu} = const - \frac{\mu}{\sigma^2}, \tag{11.3}$$

which when we differentiate again yields,

$$\frac{\partial^2 l}{\partial \mu^2} = -\frac{1}{\sigma^2}. \tag{11.4}$$

Therefore the information matrix is given by,

$$I(\mu) = \frac{1}{\sigma^2}. \tag{11.5}$$

Problem 11.1.3. Hence calculate the information matrix for a sample of N observations.

Since the only difference is the summation we obtain,

$$I(\mu) = \frac{N}{\sigma^2}. \tag{11.6}$$

Problem 11.1.4. State Jeffreys prior for μ.

The Jeffreys prior is simply the square root of the information matrix,

$$p(\mu) \propto \sqrt{\frac{N}{\sigma^2}} \tag{11.7}$$

Problem 11.1.5. Is Jeffreys prior proper here?

No! It is a uniform prior between $-\infty < \mu < +\infty$.

11.2 The illusion of uninformative priors revisited

Suppose that θ represents the probability that one randomly-chosen individual has a particular disease.

Problem 11.2.1. Suppose that we start by assigning a uniform prior on θ. Use sampling to estimate the prior distribution that in a sample of two one person has the disease and another doesn't. Hence comment on the assumption that a uniform prior is uninformative.

The probability of one of each type in a sample of two is $\phi = \theta(1-\theta)$. So if we first sample θ then transform it as according to this probability we find that there is a maximum probability of 0.25 with this decreasing to 0 as $\phi \to 0$. This is not uninformative. So a uninformative prior in one frame of reference is not uninformative in another.

Problem 11.2.2. Assume instead that we ascribe a uniform prior to the probability that 2/2 individuals have the disease. What is the implicit prior distribution for the probability that one individual has the disease?

Use sampling again here, but instead transform according to $\theta^{0.5}$. This gives us a linearly increasing line (the exact form is $p(\phi) = 2\phi$).

Chapter 12

Leaving conjugates behind: Markov Chain Monte Carlo

12.1 A fairground game

At a fairground a man advertises a gambling game that allows participants the chance to win a monetary prize, if they pay an entrance fee. The game sequence goes like this:

- You pay £X.
- The man flips a fair coin (i.e. with an equal chance of the coin landing heads or tails up),
- If the coin lands tails up, the game ends and you walk away with nothing.
- If the coin lands heads up, he flips the coin a further two times and you receive the total number of heads across these latter two flips, H. So if the coin lands heads up twice, you receive £2; if once, you receive £1; if zero, you receive £0.
- Your winnings are given by £(H - X).

Problem 12.1.1. Calculate the expected value of your winnings W if you participate, and hence determine the fair price of the game.

The expected value of winnings W is given by,

$$\mathbb{E}(W) = (1/2) \times 0 + (1/2) \times 2 \times (1/2) - X = (1/2) - X \tag{12.1}$$

So the fair price is just £0.50.

Problem 12.1.2. Create an R function that simulates a single run of the game, and use this to estimate the expected value of your winnings.

Hint: use R's `rbinom` and `ifelse` functions.

A function that does this is shown below,

```
fGame <- function(){
  Y <- rbinom(1, 1, 0.5)
  Z <- ifelse(Y == 0, 0, rbinom(1, 2, 0.5))
  return(Z)
}
```

which can then be run over 10,000 iterations, and its mean calculated

```
mean(sapply(seq(1, 10000, 1), function(i) fGame()))
```

which should roughly be 0.5.

Problem 12.1.3. Suppose that you pay £1 for each game, and start with £10 in your pocket. By using your previously-created function, or otherwise, determine the expected number of games you can play before going broke.

A function that implements this is,

```
fGamblersRuin <- function(Wealth, Price){
  numGames <- 0
  while(Wealth > 0){
    Win <- fGame() - Price
    Wealth <- Wealth + Win
    numGames <- numGames + 1
  }
  return(numGames)
}
```

Using the above function to simulate 10,000 runs,

```
mean(sapply(seq(1, 10000, 1), function(i) fGamblersRuin(10, 1)))
```

which should be close to 20. This makes sense intuitively, since the expected loss per game is £0.50, and so it takes 20 games for our initial wealth to erode!

Problem 12.1.4. Suppose you start with £10, and play the game 100 times (stopping only if your wealth is below the price of entry), each time paying £0.49. You want to insure against the risk of losing all your wealth. What is the fair price to pay for such an insurance scheme?

The fair price to pay is the probability that you go bust times the loss that causes you, i.e. £10. To determine this risk I use sampling, creating a function that determines the number of plays before you go broke.

```
fGamblersRuinN <- function(N, Wealth, Price){
  numGames <- 0
  for(i in 1:N){
    if(Wealth >= Price){
      Win <- fGame() - Price
      Wealth <- Wealth + Win
      numGames <- numGames + 1
    } else{
      break
    }
  }
  return(numGames)
}
```

which we then use to simulate 10,000 runs, and calculate the proportion of times that you lose all your wealth,

```
lData <- sapply(seq(1, 10000, 1), function(i) fGamblersRuinN(100, 10, 0.49))
mean(lData < 100)
```

which should be around 12-13%. So the fair amount to pay for such a scheme is about £1.20-1.30.

12.2 Independent sampling

An analysis results in a posterior with the following probability density function:

$$f(x) = \begin{cases} \frac{1}{1.33485}\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}, & \text{if } x < 0.9735. \\ 0.186056, & \text{if } 0.9735 \leq x \leq 5. \\ 0, & \text{otherwise.} \end{cases} \tag{12.2}$$

Problem 12.2.1. Verify that this is a valid PDF (hint: see R's numerical integration function).

This is straightforwardly done by integrating the function over the range,

```
fPDF <- function(X){
  y <- ifelse(X < 0.9735, (1 / 1.33485) * (1 / sqrt(2 * pi)) * exp(-X ^ 2 / 2),
                          ifelse(X <= 5, 0.186056, 0))
  return(y)
}
integrate(fPDF,0,8)
```

Problem 12.2.2. Using *independent* sampling estimate the mean and variance of this distribution.

There are a large number of methods here, of which I describe two (importance sampling is discussed below). The first is **rejection sampling** where we generate two (pseudo-)random continuous points: $x \in (0, 5)$ and $y \in (0, \frac{1}{1.335\sqrt{2\pi}}$. We *accept* the point as a sample from our distribution iff $y \leq f(x)$.

```
fReject <- function(N){
  count <- 1
  lSamples <- vector(length=N)
  while(count <= N){
    X <- runif(1, 0, 5)
    Y <- runif(1, 0, (1 / 1.335) * (1 / sqrt(2 * pi)))
    if(Y < fPDF(X)){
      lSamples[count] <- X
      count <- count + 1
    }
  }
  return(lSamples)
}
mean(fReject(10000), 100)
var(fReject(10000))
```

yields a mean of around 2.35, and a variance of about 2.24.

The second is **inverse transform sampling**. To do this we need first to find an approximate CDF function, by creating a function that integrates from 0 to a given point x. We can then generate $(CDF[x], x)$ for $x \in (0, 5)$, which we use to create an interpolating function that represents the inverse CDF. Then use R's uniform random number generator on (0,1) to generate a series of 'CDF' samples, to which the inverse CDF is applied to give actual samples,

```
fIntegrator <- function(X){
  return(integrate(fPDF, 0, X)[[1]])
}

lCDF <- sapply(seq(0, 5, 0.1), fIntegrator)
fICDF <- approxfun(lCDF, seq(0, 5, 0.1))

fInverseTransform <- function(N){
  lCDF <- runif(N, 0, 1)
  return(sapply(lCDF, fICDF))
}
mean(fInverseTransform(100000))
var(fInverseTransform(100000))
```

Problem 12.2.3. Construct uncertainty intervals around your estimates of the mean.

This needn't be done with too much precision. Essentially all you need to do is repeat the above process a reasonable number of times, and examine the quantiles of this distribution.

Problem 12.2.4. Verify your previous answer by calculating the mean and variance of this distribution.

This can be done using R's numerical integration functions,

```
aMean <- integrate(function(x) x * fPDF(x), 0, 5)[[1]]
aVar <- integrate(function(x) x ^ 2 * fPDF(x), 0, 5)[[1]] - aMean ^ 2
```

which yields a mean of about 2.35 and a variance of 2.24.

Problem 12.2.5. On the basis of the equation:

$$\begin{aligned} \mathrm{E(X)} &= \int x f(x) \mathrm{d}x \\ &= \int x \frac{f(x)}{g(x)} g(x) \mathrm{d}x \end{aligned}$$

provide another way to estimate the mean.

If we can generate independent samples $x \sim g(x)$, then we can estimate the mean using:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i \frac{f(x_i)}{g(x_i)} \tag{12.3}$$

Since we know that:

$$\begin{aligned} \mathbb{E}_g(\hat{\mu}) &= \mathrm{E_g}\left(\mathrm{x}\frac{\mathrm{f(x)}}{\mathrm{g(x)}}\right) \\ &= \int x \frac{f(x)}{g(x)} g(x) \mathrm{d}x \\ &= \int x f(x) \mathrm{d}x \\ &= \mathrm{E_f(X)} = \mu \end{aligned}$$

In this case choosing $g(x)$ to be a continuous uniform distribution over (0,5) is a reasonable first choice.

Problem 12.2.6. Using the above method where g is the continuous uniform distribution between 0 and 5 find an estimate of the mean.

This can be done with the following code,

```
fImportance <- function(N){
  lX <- runif(N, 0, 5)
  lF <- sapply(lX, fPDF)
  lG <- rep(1 / 5, N)
  lRatio <- lX * lF / lG
  mean(lRatio)
}

fImportance(10000)
```

Problem 12.2.7. How should we choose $g(x)$ to yield estimators with the lowest variance? (Difficult.)

Essentially we want an estimator with a low variance:

$$\begin{aligned} Var_g(\hat{\mu}) &= \frac{1}{n^2}\sum_{i=1}^{n} Var_g\left[x_i\frac{f(x_i)}{g(x_i)}\right] \\ &= \frac{1}{n}\left[\int \frac{x^2 f(x)^2}{g(x)^2} g(x)\mathrm{d}x - \mu^2\right] \\ &= \frac{1}{n}\left[\int \frac{x^2 f(x)^2}{g(x)}\mathrm{d}x - \mu^2\right] \\ &= \frac{1}{n}\int \frac{(xf(x) - g(x)\mu)^2}{g(x)}\mathrm{d}x \end{aligned}$$

To minimise the above expression we should choose $g(x) = \frac{xf(x)}{\mu}$, or more generally $g(x) \propto xf(x)$. (For clarity we obtained the bottom line of the above because of the following trick:)

$$\begin{aligned} \int \frac{(xf(x) - g(x)\mu)^2}{g(x)}\mathrm{d}x &= \int \frac{x^2 f(x)^2 - 2xf(x)g(x)\mu + g(x)^2\mu^2}{g(x)}\mathrm{d}x \\ &= \int \frac{x^2 f(x)^2}{g(x)}\mathrm{d}x - 2\mu\int xf(x)\mathrm{d}x + \mu^2\int g(x)\mathrm{d}x \\ &= \int \frac{x^2 f(x)^2}{g(x)}\mathrm{d}x - \mu^2 \end{aligned}$$

In this example this means that perhaps using a triangular distribution would yield the lowest variance, although this isn't easy to sample from.

12.3 Integration by sampling

Calculate the following integrals by sampling.

Problem 12.3.1.

$$\int_{-\infty}^{\infty} \frac{x^6}{\sqrt{2\pi}} \times exp(-\frac{x^2}{2})\mathrm{d}x \tag{12.4}$$

This is equivalent to calculating $\mathbb{E}(X^6)$ from a standard normal. So all you do is sample from a standard normal and raise each sample to the power 6. If you then take the mean of these you get an estimator for the above. The actual answer here is 15 (Figure 12.1a).

Problem 12.3.2.

$$\int_{1}^{\infty} \frac{x^3}{\sqrt{2\pi}} \times exp(-\frac{x^2}{2})\mathrm{d}x \tag{12.5}$$

It is possible to rewrite this equation as,

$$\mathrm{E}[x^3(1 - 0.5\Phi(-\frac{1}{\sqrt{2}}))|x \sim \text{truncated-}\mathcal{N}(0, 1, 1)] \tag{12.6}$$

where the last '1' corresponds to the point of truncation and $\Phi()$ is the standard normal cumulative distribution function. Then we can just generate samples from a truncated normal by rejection sampling (i.e. sample from a standard normal and reject any samples where $x < 1$) and calculate the sample mean of $x^3(1 - 0.5\Phi(-\frac{1}{\sqrt{2}}))$, which is approximately 0.726 (Figure 12.1b).

Problem 12.3.3.

$$\int_{1}^{\infty} \frac{x^6}{\sqrt{2\pi}} \times exp(-\frac{x^2 + 4x}{2})\mathrm{d}x \tag{12.7}$$

Note that there was a mistake in the question in the first edition of the book with $-4x$ opposed to the intended $+4x$.

The trick here is to notice that we can rewrite the above as:

$$\frac{x^6}{\sqrt{2\pi}}exp(-\frac{x^2 + 4x}{2}) = x^6exp(-2x) \times \frac{1}{\sqrt{2\pi}}exp(-\frac{x^2}{2}) \tag{12.8}$$

So we are trying to determine $\mathbb{E}\left(x^6exp(-2x)(1 - 0.5\Phi(-\frac{1}{\sqrt{2}}))\right)$ for a truncated normal. The method then proceeds as above, with an answer of about 0.0979 (Figure 12.1c).

Problem 12.3.4.

$$\int_{1}^{10} x^6 \frac{e^{-\frac{x^4}{2}}}{\sqrt{2\pi}}\mathrm{d}x \tag{12.9}$$

This is a bit of a trick. It looks like we could use the normal distribution here, but you can't easily due to the x^4 in the exponent. A better way to approach this is to sample from a continuous uniform distribution on (1,10), and then take each sample x and evaluate $9\frac{e^{-\frac{x^4}{2}}x^6}{\sqrt{2\pi}}$. **Note:** we need a 9 there to account for the fact that the continuous uniform density is equal to $\frac{1}{9}$! If we then take the mean of the transformed samples we get the result of about 0.2665 (Figure 12.1d).

Problem 12.3.5. What is the approximate sampling distribution in using independent sampling to evaluate integrals?

Using the (Lindberg-Lévy) central limit theorem (that applies for independent samples) we get:

$$\bar{X} \approx \mathcal{N}(\mu, \frac{\sigma}{\sqrt{n}}) \tag{12.10}$$

where n is the sample size, and $\sigma = \mathrm{E}[(\mathrm{i} - \mathrm{I})^2]$, where i is a sample estimate of the integral and I is the true value.

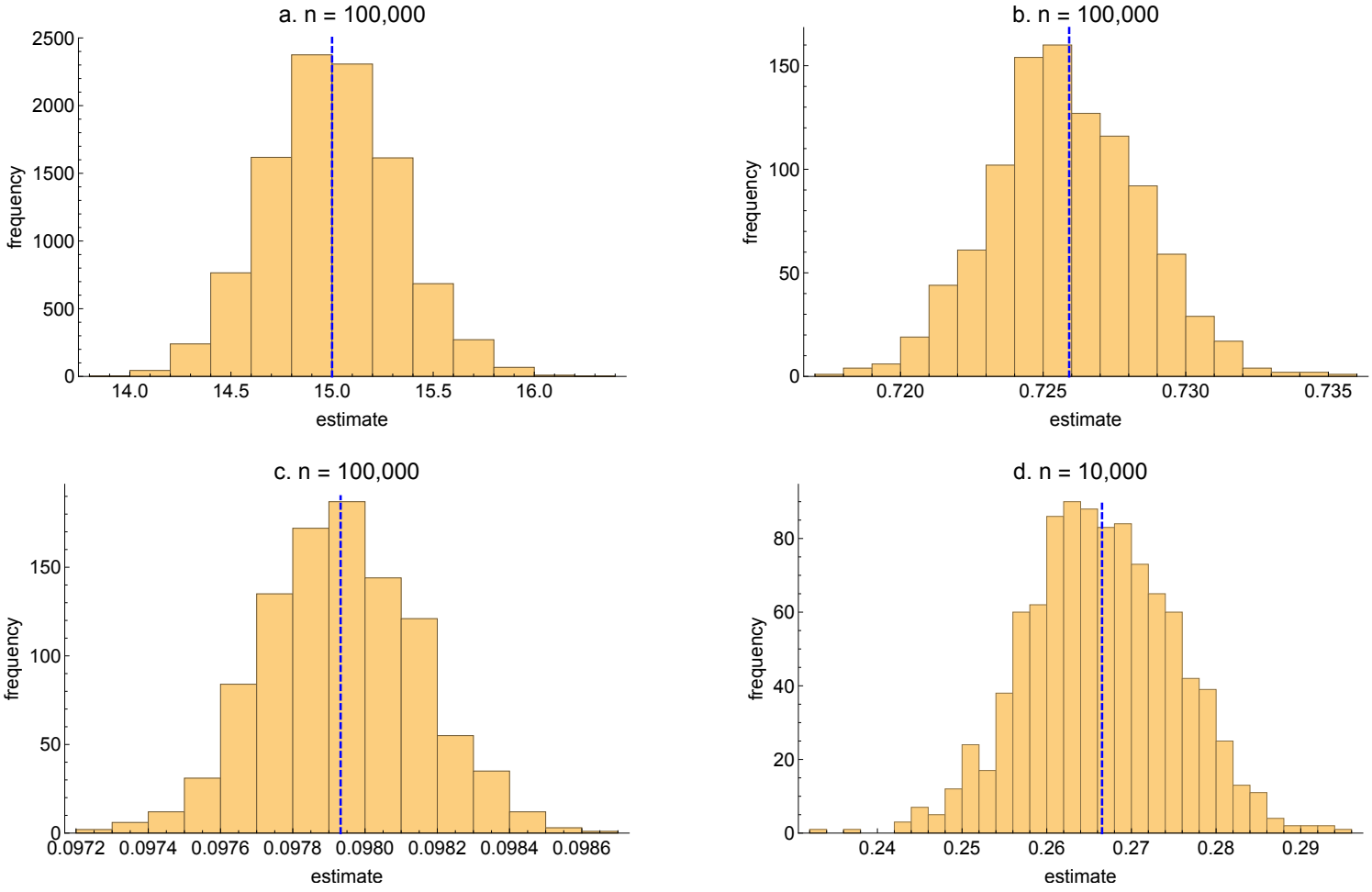

Figure 12.1: Sampling distributions for the estimators of the integrals in the text.

12.4 Markovian coin

Consider a type of coin for which the result of the next throw (heads or tails) can depend on the result of the current throw. In particular if a "heads" is thrown then the probability of obtaining a "heads" on the next throw is $(\frac{1}{2}+\epsilon)$; if instead a tails is thrown then the probability of obtaining a "tails" on the next throw is $(\frac{1}{2}+\epsilon)$. To start, we assume $0 \leq \epsilon \leq \frac{1}{2}$. The random variable $X = 0, 1$ if the coin lands "tails-up" or "heads-up" on a given throw.

Problem 12.4.1. Find the mean of the coin supposing it starts with probability $\frac{1}{2}$ on each side.

Considering the first throw after the initial one:

$$\begin{aligned}\mathbb{E}(X_1|X_0) &= Pr(X_1 = 1|X_0) \times 1 + Pr(X_1 = 0|X_0) \times 0 \\ &= Pr(X_1 = 1|X_0)\end{aligned}$$

By the law of iterated expectations:

$$\begin{aligned}\mathbb{E}(X_1) &= \mathbb{E}\left[\mathbb{E}(X_1|X_0)\right] \\ &= Pr(X_0 = 0) \times Pr(X_1 = 1|X_0 = 0) + Pr(X_0 = 1) \times Pr(X_1 = 1|X_0 = 1) \\ &= \frac{1}{2} \times (\frac{1}{2} - \epsilon) + \frac{1}{2} \times (\frac{1}{2} + \epsilon) \\ &= \frac{1}{2}\end{aligned}$$

Therefore we know that $\mathbb{E}(X_2) = \frac{1}{2}$, and ... $\mathbb{E}(X_k) = \frac{1}{2}$.

So this property is exactly the same as a fair coin.

Problem 12.4.2. Computationally estimate the mean of the coin by simulating 10, 20, and 100 throws for $\epsilon = 0$.

The results of using sampling to estimate the mean of the coin are shown in Figure 12.2. The error (via the Monte Carlo Central Limit Theorem) decreases as $\frac{1}{\sqrt{n}}$.

Problem 12.4.3. As $\epsilon \uparrow$ how does the error in estimating the mean change, and why?

As $\epsilon \uparrow$ there is increased *dependence*, meaning that the information garnered from each incremental sample is less than for the purely *independent* case (Figure 12.3).

Problem 12.4.4. When $\epsilon = \frac{9}{20}$ calculate the effective sample size of an actual sample size of 100. How does the effective sample size depend on ϵ?

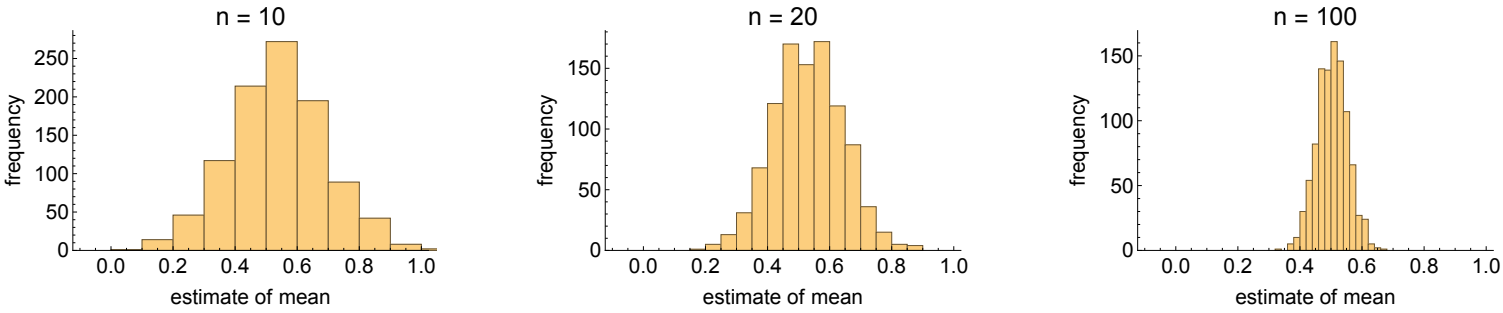

Figure 12.2: Estimating the mean of a Markovian coin (where $\epsilon = 0$, i.e. it is fair), using sampling across three sample sizes. In all cases 1,000 iterates were used to estimate the sampling distribution.

The equivalent sample size for an *independent* coin, for another with $\epsilon = \frac{9}{20}$ and 100 samples is about 6 (Figure 12.4)! This effect is due to dependence.

Problem 12.4.5. Now assume that $\epsilon = -\frac{9}{20}$. what is the effective sample size of an actual sample size of 100? Explain your result.

This is antithetic sampling, where there is a negative correlation between the values of the sampler at each time step. Consider a sample $\frac{X_1+X_2}{2}$. Calculating its variance:

$$var\left(\frac{X_1+X_2}{2}\right) = \frac{1}{4}\left(var(X_1) + var(X_2) - 2Cov(X_1, X_2)\right) \tag{12.11}$$

If $Cov(X_1, X_2) = 0$ as in independent sampling, then we get $var(\bar{X}) = \frac{1}{n}\sigma^2$, where σ^2 is the variance of one sample. However, if $Cov(X_1, X_2) < 0$, then we can achieve a variance $var(\bar{X}) < \frac{1}{n}\sigma^2$. So antithetic sampling beats independent! Intuitively this is because an antithetic sampler can visit the state space much more efficiently than an independent sampler.

By sampling we have an effective sample size of about 1600 for an antithetic sample size of 100 (Figure 12.5).

I am sure this can be proved semi-analytically, since the series of autocovariances follow a geometric series $(\epsilon/2, \epsilon^2, 2\epsilon^3, ...)$ (see Mathematica file). However, finding the sums is a bit tricky, and I'm not convinced it will simplify greatly!

12.5 Markovian die

Consider a type of die whose next value thrown can depend on the current value. The degree of dependence is specified by a parameter $0 \leq \epsilon \leq 1$ (see Figure 12.6). If $\epsilon = 0$ then each separate throw of the die can be considered independent of the previous value. Another way of saying this is that each number has an equal probability of being thrown irrespective of the current value. If $\epsilon = 1$ then there is strong dependence from one throw to the next, where from a given number on a throw only neighbouring numbers are possible on the next. So $1 \to (6, 2)$, $2 \to (1, 3)$ etc. If

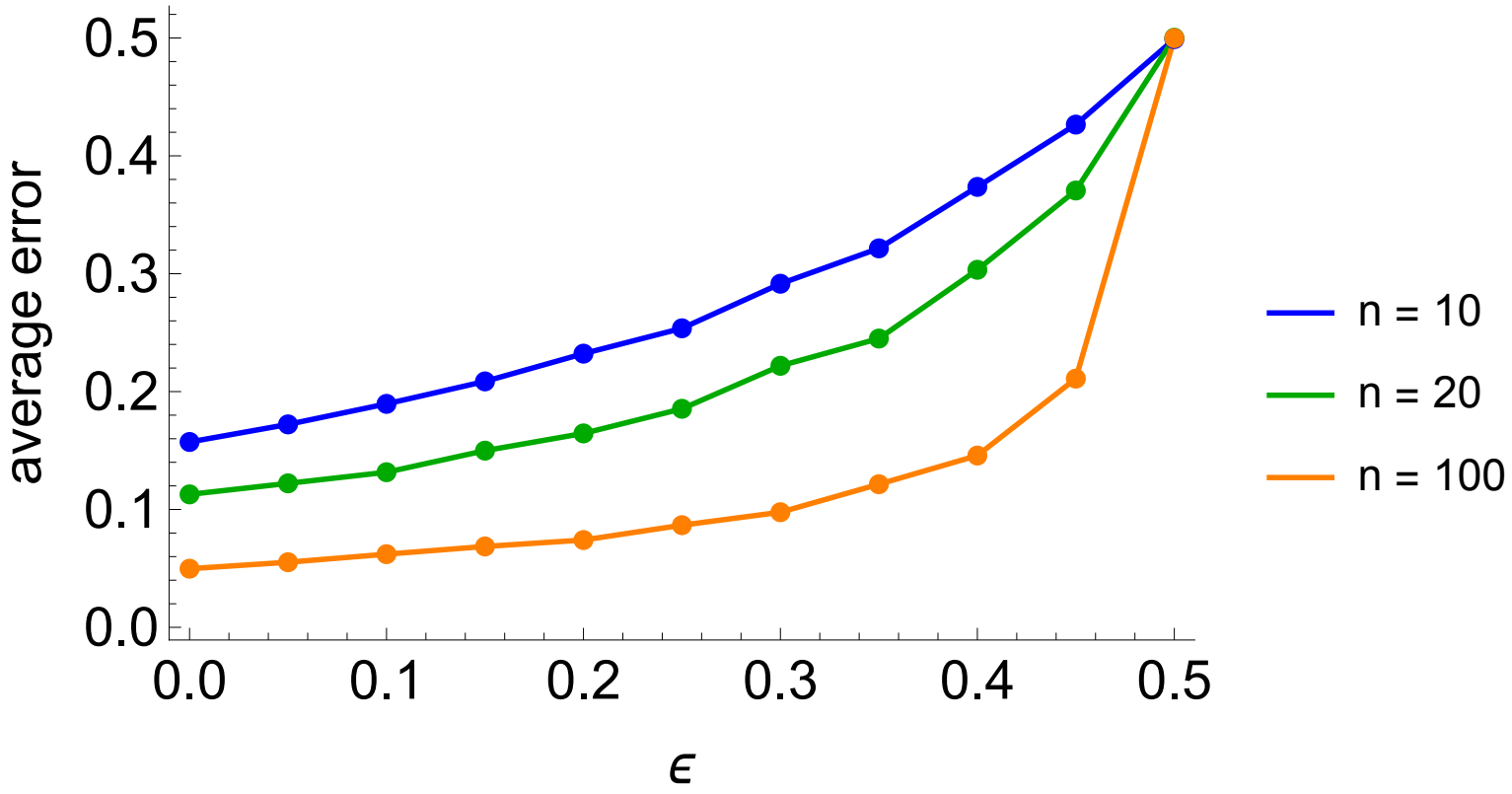

Figure 12.3: Standard error in estimating the mean of the Markovian coin, across different levels of ϵ.

$0 < \epsilon < 1$ we suppose that there is preference towards consecutive numbers, with the preference increasing in ϵ.

For all values of ϵ we assume that both the forward and backward steps are equally likely, so $1 \to 2$ and $1 \to 6$ are of the same probability. If $0 < \epsilon < 1$, we suppose that those transitions that are not neighbours are all of the same probability (which is less than the probability of consecutive numbers).

Specifically, we define ϵ in the following way:

$$Pr(X_{n+1}|X_n) = \frac{1}{6}(1-\epsilon) + \frac{\epsilon}{2}1_{X_{n+1}\in\mathcal{C}(X_n)}$$

Where $1_{X_{n+1}\in\mathcal{C}(X_n)}$ is an indicator function which is equal to 1 if the next value of the die, X_{n+1} is in the neighbour set $\mathcal{C}(X_n)$ of the current value, X_n. (The above is just a fancy way of saying that we increase the probability of neighbours by an amount $\frac{\epsilon}{2}$ relative to the non-neighbours.)

Problem 12.5.1. Find the mean of the die across all values of ϵ assuming it starts on a randomly-selected side.

Since it starts on a random side this means all numbers of equally likely (on the first throw), so

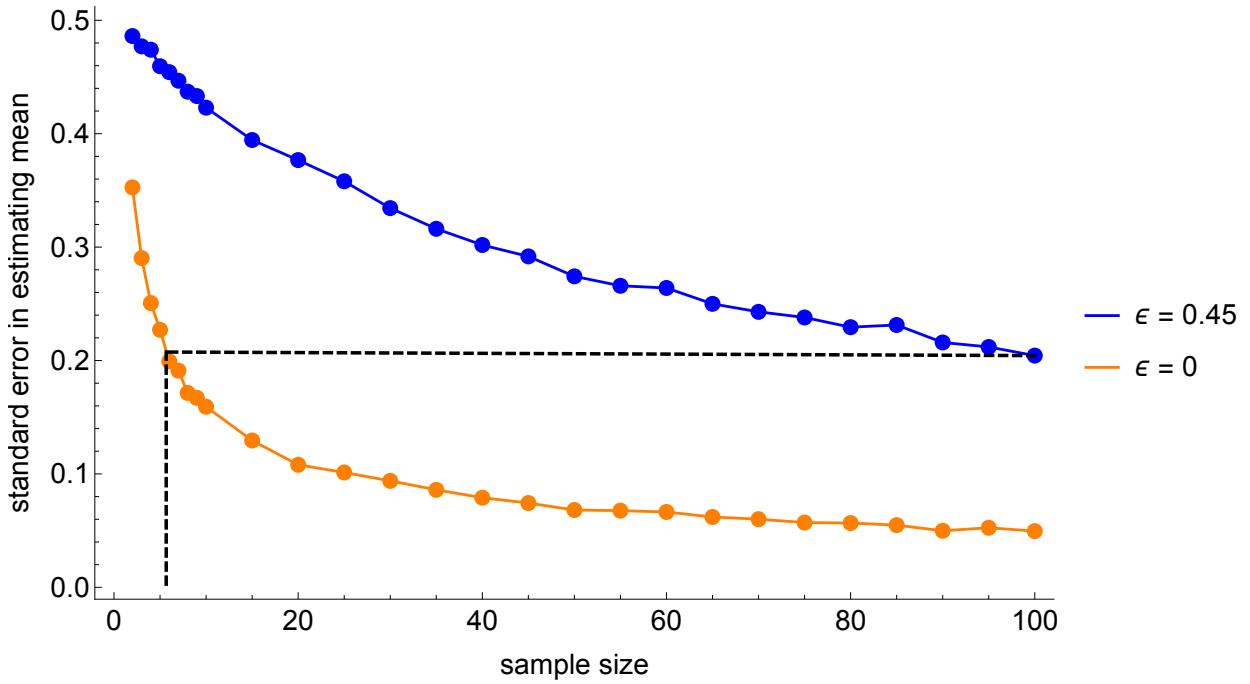

Figure 12.4: Calculating the effective sample size for $\epsilon = \frac{9}{20}$ for an actual sample size of 100.

its mean is just that of a typical die $\mu = \frac{7}{2}$. From then onwards we have the following conditional expectation:

$$\begin{aligned}\mathbb{E}(X_{n+1}|X_n) &= \sum_{i=1}^{6} Y_i \frac{1}{6}(1-\epsilon) + \sum_{i \in \mathcal{C}(X_n)} Y_i \\ &= \frac{7}{2}(1-\epsilon) + \sum_{i \in \mathcal{C}(X_n)} Y_i\end{aligned}$$

The only part of the above expression that depends on X_n is the second bit concerning the neighbours. So all we need to do is calculate this term for all possible neighbour combinations, and average over them yielding $\frac{7}{2}\epsilon$. Using this we can then find the *unconditional* expectation:

$$\mathbb{E}(X_{n+1}) = \frac{7}{2} \tag{12.12}$$

So the dependent die has the same mean as a typical fair die.

Problem 12.5.2. By simulating throws of the die, find an estimator of its mean.

The following Mathematica code does this simulation.

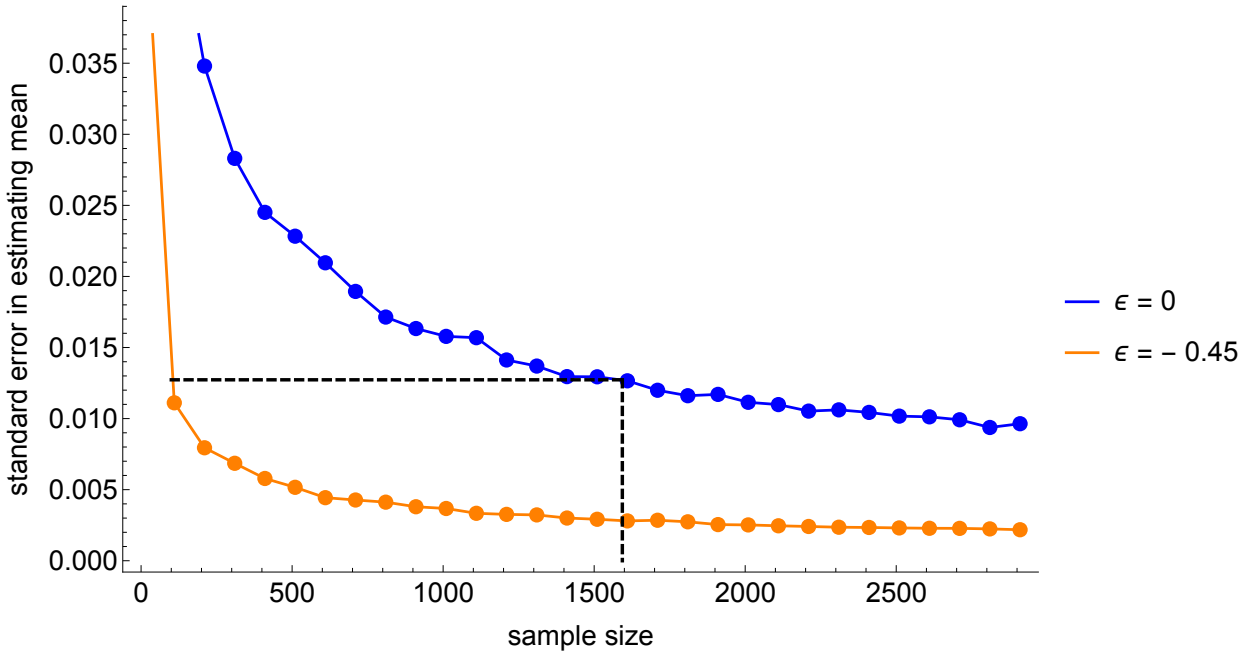

Figure 12.5: Calculating the effective sample size for $\epsilon = -0.45$ for an actual sample size of 100.

```
fPreferredSelector[aNumber_, n_Integer] :=
  If[aNumber == 1, {n, 2},
  If[aNumber == n, {n - 1, 1}, {aNumber - 1, aNumber + 1}]]

fPreferredBinary[aNumber_, n_Integer] :=
  If[# == 1, 1, 0] & /@ Mod[Abs[Range@n - aNumber], n - 2]

fProposalProbabilities[aNumber_, n_Integer, epsilon_] :=
  Module[{lPreferredSelector =
    fPreferredBinary[aNumber, n]},
        (0.5 - (epsilon/2)) lPreferredSelector + (epsilon / n)
          ConstantArray[1, n]]

toPiecewise[wts_, x_] :=
  Piecewise[MapIndexed[{#1, x == #2[[1]]} &, wts]]

fSelectNext[aNumber_, n_Integer, epsilon_] :=
  Module[{wts = fProposalProbabilities[aNumber, n, epsilon], f},
    f = ProbabilityDistribution[toPiecewise[wts, x], {x, 1, n, 1}];
    RandomVariate[f, {1}]][[1]]

fGenerateSamples[numSamples_, aStartNumber_, n_Integer, epsilon_] :=
  NestList[fSelectNext[#, n, epsilon] &, aStartNumber, numSamples]

fGenerateTransitions[maxSamples_Integer, n_Integer, epsilon_] :=
```

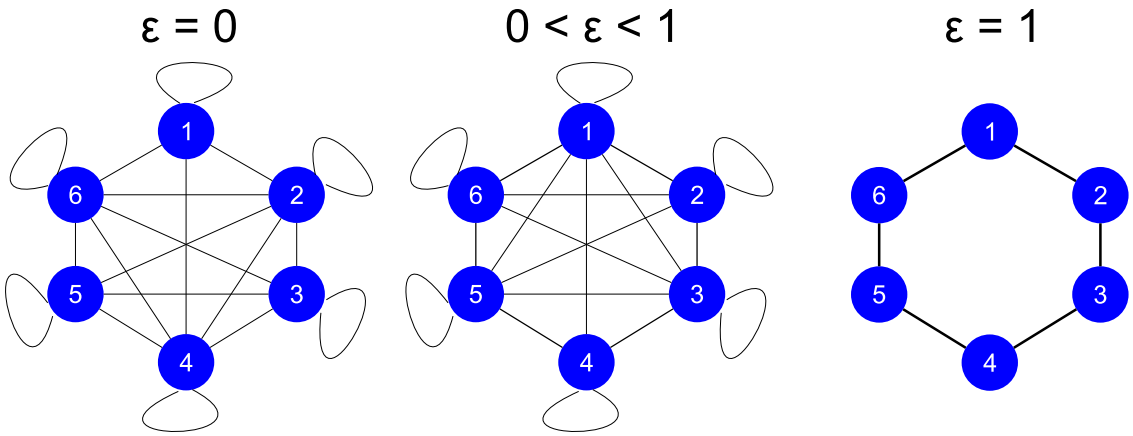

Figure 12.6: A Markovian die where ϵ determines the degree of dependence between throws of the die.

```
  Table[RandomVariate[ProbabilityDistribution[
    toPiecewise[fProposalProbabilities[i, n, epsilon], x], {x, 1, n,
1}], {maxSamples}], {i, 1, n, 1}]

fChain[numChains_Integer, numSamples_Integer, n_Integer, epsilon_] :=
  Module[{aNewEpsilon = 1 - epsilon, lAllTransitions, lTotal, lSteps},
    lAllTransitions = fGenerateTransitions[numChains numSamples,
                                                                    n, aNewEpsilor
    lSteps = Range@(numChains numSamples);
    lTotal = FoldList[fStep[#1, #2, lAllTransitions] &,
    RandomInteger[{1, n}], lSteps]; Partition[lTotal, numSamples]]

fStep[aCurrentNumber_Integer, aStepNumber_Integer, lSamples__] :=
  lSamples[[aCurrentNumber, aStepNumber]]

fVarianceEstimator[n_Integer, numChains_Integer, numSamples_, epsilon_] :=
  Variance[Mean /@ fChain[numChains, numSamples, n, epsilon]]
```

Problem 12.5.3. Compute the error in estimating the mean as ϵ is varied at a sample size of 5, 10,and 100.

The results of this are shown in Figure 12.7. As the sample size increases, the effect of dependence is less evident.

Problem 12.5.4. Find the effective sample size of 100 throws (when estimating the mean) for a die where $\epsilon = 1$. Comment on the effect of dependence on sampling.

First you need to find the average error for an independent die, which I find to be about 0.17. To make an equivalent error on a die with $\epsilon = 1$, I find about 210 samples are necessary. So dependence means that we need more samples to get the same quality of estimator. This problem is compounded as the size of the state space increases.

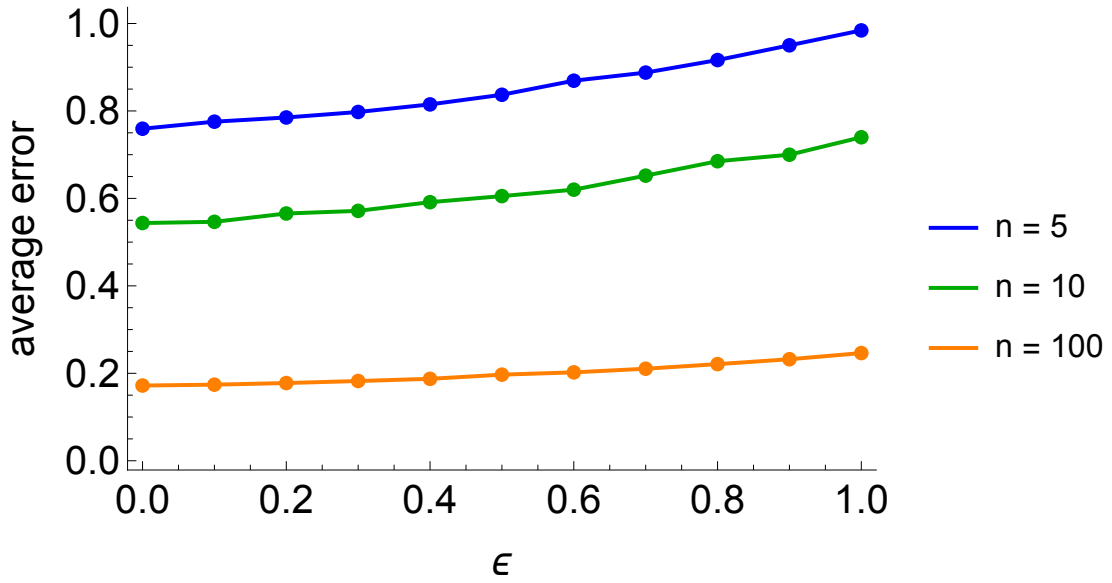

Figure 12.7: The effect of varying ϵ on the standard error in estimating the mean of a six-sided die.

Problem 12.5.5. Now suppose that the die starts always on side 2. Find the expectation of the die (not the running total, just the current value) at each time step. (Difficult.)

The key to doing this is to set up the problem as a discrete Markov chain. To do this we create a transition matrix $\boldsymbol{P}$:

$$\boldsymbol{P} = \begin{pmatrix} \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} \\ \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} \\ \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} \\ \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} \\ \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} \\ \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{1}{6}-\frac{\epsilon}{6} & \frac{\epsilon}{3}+\frac{1}{6} & \frac{1}{6}-\frac{\epsilon}{6} \end{pmatrix}$$

If we assume the die starts in state $\boldsymbol{s(0)} = (0, 1, 0, 0, 0, 0)$ (in other words on the number 2), then we can calculate the probabilities of each state at each time step $\boldsymbol{s}$:

$$\boldsymbol{s(t)} = \boldsymbol{s(0)}\boldsymbol{P}^t \tag{12.13}$$

Since we know the probability of each state as a function of time, we can calculate the expectation:

$$\boldsymbol{s(t)} = \boldsymbol{s(0)}\boldsymbol{P}^t \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix} \tag{12.14}$$

Problem 12.5.6. Following on from the last question find how long we need to leave the die before we are confident we are sampling from its *unconditional* distribution. (By "unconditional" here, we mean its probability distribution disregarding its start point.) (Difficult.)

Figure 12.8 shows the conditional expectation as a function of time for different ϵ values. For $\epsilon << 1$ we see that we get a steady state expectation after about 10-15 samples. We see that for $\epsilon = 1$ we never get a constant expectation, but we do approach (periodic) stationarity in the distribution after a sample of 15. Both of these cases indicate that we have reached the *unconditional* distribution, where we have essentially forgotten the starting value.

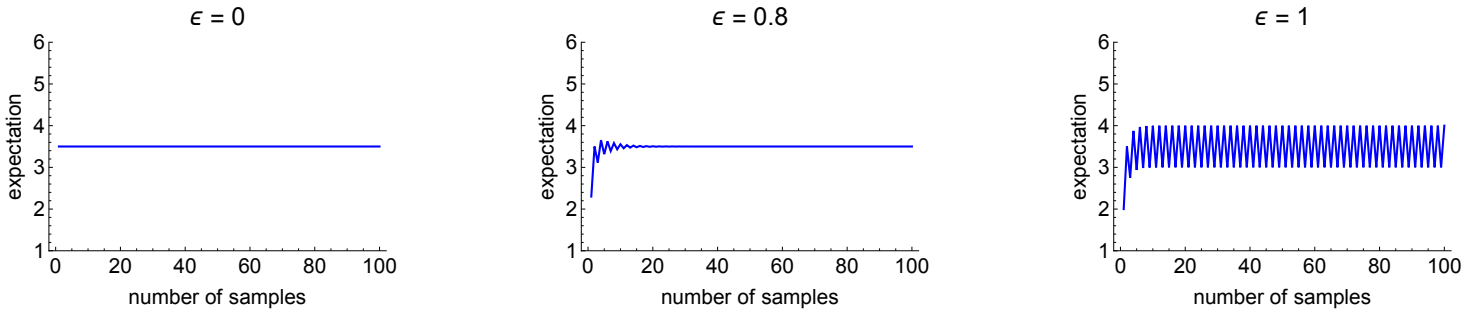

Figure 12.8: The effect of varying ϵ on the conditional expectation.

Problem 12.5.7. Carry out the above investigations but for a die with n sides. How does n affect the results?

As $n \uparrow$ the effective sample size decreases for a given ϵ. Figure 12.9 also shows that we get increased sensitivity to ϵ for a 20 sided die; intuitively this sensitivity should increase with n.

12.6 Turning a coin into a random-number generator

Suppose you have one coin that has equal probability of landings heads up versus tails up.

Problem 12.6.1. How can you use this coin to create a random variable X that has $Pr(X = 1) = 1/3$ and $Pr(X = 0) = 2/3$? (Hint: use rejection sampling.)

If you flip the coin twice and record its value Y on each flip you obtain,

$$Pr(00) = 1/4 \tag{12.15}$$
$$Pr(10) = 1/4 \tag{12.16}$$
$$Pr(01) = 1/4 \tag{12.17}$$
$$Pr(11) = 1/4 \tag{12.18}$$

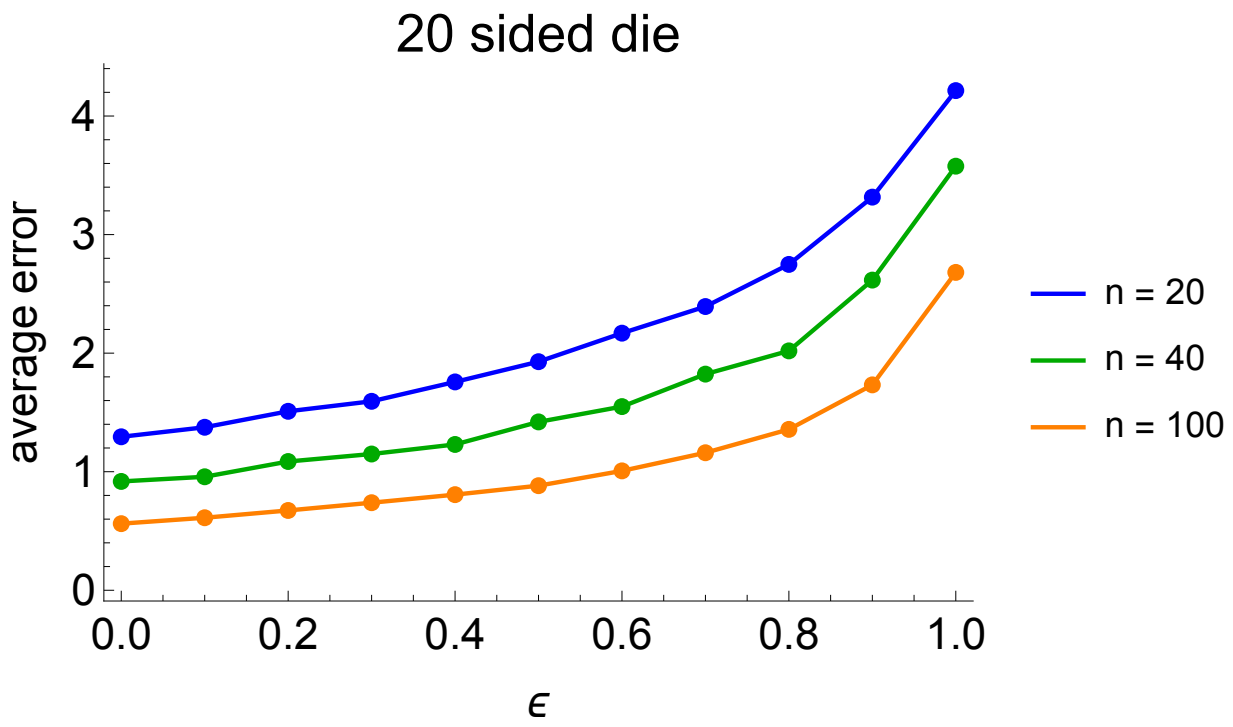

Figure 12.9: The effect of varying ϵ on the average error in estimating the mean for a 20-sided die.

so there are four outcomes, all of equal chance. What you want to do is convert something that has four outcomes into another thing that has three. So you assign: $00 \to X = 0$, and or$(10, 01) \to X = 1$. If the coin lands heads up twice, you just reject that sample, and repeat the exercise until you get one of the other three outcomes. This gives you the 1/3 to 2/3 probabilities of each outcome as you needed.

Problem 12.6.2. In R use a computational fair coin (i.e. a bernoulli distribution with $\theta = 0.5$) to create a random variable that is *approximately* distributed as a standard normal.

Use the central limit theorem,

$$\sqrt{n}(\overline{X} - 0.5) \xrightarrow{p} \mathcal{N}(0, \sigma) \tag{12.19}$$

So if we take enough samples the distribution of $\overline{X} \approx \mathcal{N}(0.5, \frac{\sigma}{\sqrt{n}})$. Doing this in R,

```
fCLT <- function(numSamples){
  X <- sapply(seq(1, numSamples, 1), function(i) rbinom(1, 1, 0.5))
  return(mean(X))
}

hist(sapply(seq(1, 1000, 1), function(i) fCLT(1000)), 100)
```

How to convert this into a standard normal we need to know $\sigma^2 = 0.5(1 - 0.5) \implies \sigma = 0.5$. Therefore the following should yield a standard normally-distributed rv,

```
fCLT1 <- function(numSamples){
  X <- sapply(seq(1, numSamples, 1), function(i) rbinom(1, 1, 0.5))
  return(sqrt(numSamples) * (mean(X) - 0.5) / 0.5)
}

lData <- sapply(seq(1, 1000, 1), function(i) fCLT1(1000))
hist(lData, 100)
sd(lData)
```

where the sd is close to 1.

Problem 12.6.3. Using the answer to the previous question create a variable that is approximately uniformly distributed between 0 and 1.

Just use the previous answer to generate CDF values from the theoretic $\mathcal{N}(0.5, \frac{\sigma}{\sqrt{n}})$ distribution,

```
fNormalQuantiles <- function(numSamples, numPerSample){
  X <- sapply(seq(1, numSamples, 1), function(i) fCLT(numPerSample))
  lCDF <- pnorm(X, 0.5, sqrt(0.5 * (1 - 0.5)) / sqrt(numPerSample))
  return(lCDF)
}
lCDF <- fNormalQuantiles(10000, 1000)
hist(lCDF)
```

12.7 Pseudo-random-number generators

Problem 12.7.1. A particular pseudo-random-number generator is known as the linear congruential generator which generates a sequence of pseudo-randomised numbers using the relation,

$$s_t = as_{t-1} + b \bmod M \tag{12.20}$$

where a, b and M are suitably chosen positive integers. What is the maximum period that such a sequence can have?

Since the sequence is bounded above my M this is the maximum number of unique values this sequence can take.

Problem 12.7.2. Write a function that implements the above recurrence relation and hence show that when $a = 2$, $b = 3$ and $M = 10$, where we begin with $s_0 = 5$ (the seed), the series has a period of 4.

Implementing this function in Mathematica we have the following,

```
fLinearCongruential[seed_Integer, a_Integer, b_Integer, M_Integer] :=
  Mod[a seed + b, M]

fLinearCongruentialSeries[numSamples_Integer, seed_Integer, a_Integer,
    b_Integer, M_Integer] :=
  NestList[fLinearCongruential[#, a, b, M] &, seed, numSamples]

fLinearCongruentialSeries[10, 5, 2, 3, 10]

{5, 3, 9, 1, 5, 3, 9, 1, 5, 3, 9}
```

which has repeats itself every fourth element.

Problem 12.7.3. Create a new function that has a maximum of one and a minimum of zero.

Simply divide the above output by M,

```
fLinearCongruentialUniform[numSamples_Integer, seed_Integer, a_Integer,
                           b_Integer, M_Integer] :=
  fLinearCongruentialSeries[numSamples, seed, a, b, M] / M
```

Problem 12.7.4. Use your newly created function with $a = 1229$, $b = 1$ and $M = 2048$ where we begin with $s_0 = 1$ (the seed) to generate 10,000 numbers between zero and one. Draw a histogram of the resultant sample. What sort of distribution does this look like?

It looks like a continuous uniform distribution (Figure 12.10).

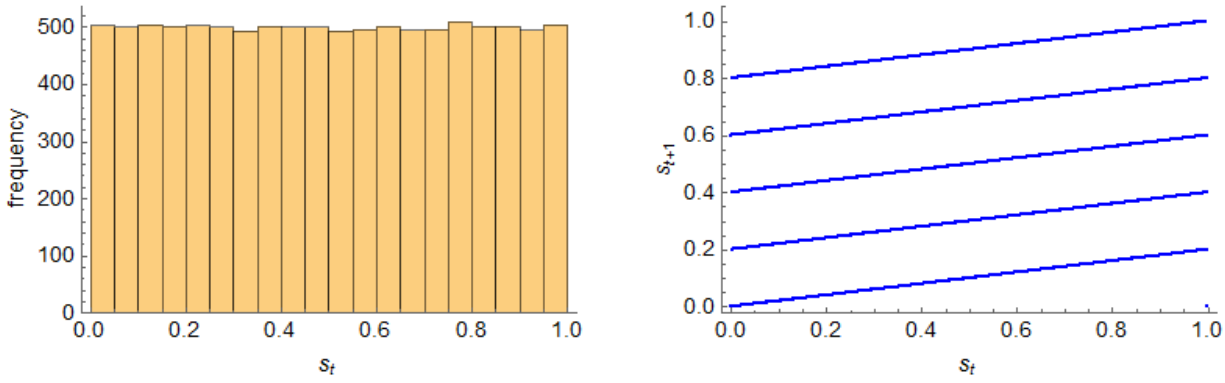

Figure 12.10: Left: the histogram and right: pairwise consecutive samples for the linear congruential generator with $a = 1229$, $b = 1$ and $M = 2048$ where we begin with $s_0 = 1$.

Problem 12.7.5. Draw a scatter plot of pairs of consecutive samples for the previously-generated series. Does this series look random?

No there is definitely patterning (Figure 12.10).

Problem 12.7.6. Now generate a series with $a = 1597$, $b = 51749$ and $M = 244944$, beginning with $s_0 = 1$, to generate 10,000 numbers between zero and one. Draw a histogram of the resultant sample. What sort of distribution does this look like? Does a scatter plot of consecutive pairs look random?

Again it looks like a continuous uniform distribution, and this time the consecutive pairs look more random (Figure 12.11).

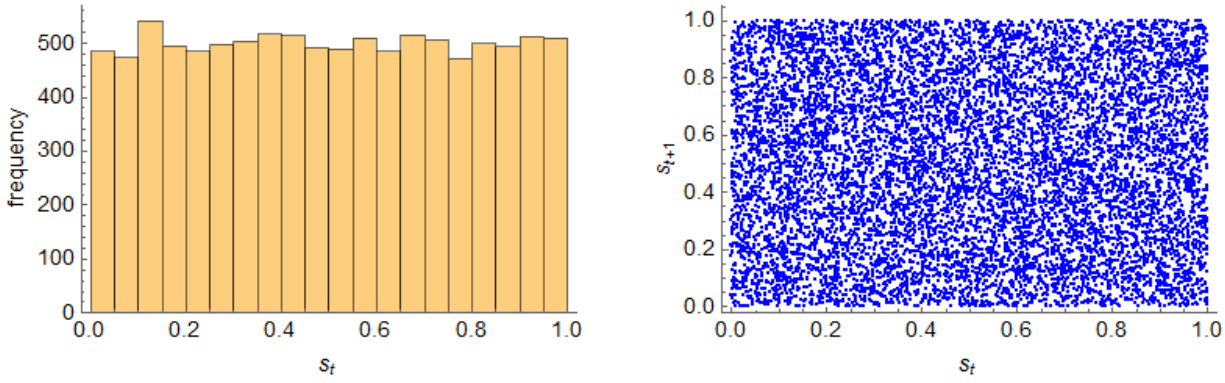

Figure 12.11: Left: the histogram and right: pairwise consecutive samples for the linear congruential generator with $a = 1597$, $b = 51749$ and $M = 244944$ where we begin with $s_0 = 1$.

Problem 12.7.7. Prove that inverse transform sampling works.

First we need to state the method itself mathematically. If $u \sim U(0, 1)$ then we suppose that $y = F^{-1}(u) \sim F$. To prove this,

$$\begin{aligned} Pr(y \leq x) &= Pr(F^{-1}(u) \leq x) && (12.21)\\ &= Pr(FF^{-1}(u) \leq F(x) && (12.22)\\ &= Pr(u \leq F(x)) && (12.23)\\ &= F(x) && (12.24) \end{aligned}$$

This means that $y \sim F$.

Problem 12.7.8. Use your most recent sequence of numbers from the linear congruential generator along with inverse transform sampling to generate pseudo-independent samples from the following density $F(x) = 1 - exp(-\sqrt{x})$.

We need to find the inverse function meaning,

$$u = 1 - exp(-\sqrt{x}), \tag{12.25}$$

which yields,

$$x = [\log (1 - u)]^2 . \tag{12.26}$$

Applying the above function to our sequence we obtain samples that look as if they come from the requisite distribution (remember to differentiate the CDF to yield the PDF).

Problem 12.7.9. Using the inverse transform method or otherwise use your sequence linear congruential generator to generate samples from a standard normal distribution.

To do with we need to find the inverse CDF for a standard normal. There is no analytic solution here and so numeric integration is the only approach. Once you have a numerical CDF versus a range of x you can create a interpolating function that returns the inverse-CDF.

An alternative approach is to use the Box-Muller transforms.

Chapter 13

Random Walk Metropolis

13.1 Ticked off

Imagine once again that you are investigating the occurrence of Lyme disease in the UK. This is a vector-borne disease caused by bacteria of species *Borrelia* which is carried by ticks. (The ticks pick up the infection by blood-feeding on animals or humans that are infected with *Borrelia*.) You decide to estimate the prevalence of this bacteria in ticks you collect from the grasslands and woodlands around Oxford.

You decide to use sample sizes of 100 ticks, out of which you count the number of ticks testing positive for *Borrelia*. You decide to use a binomial likelihood since you assume that the presence of *Borrelia* in one tick is independent of that in other ticks. Also because you sample a relatively small area you assume that the presence of *Borrelia* can be assumed to be identically-distributed across ticks.

Problem 13.1.1. You specify a $beta(1,1)$ distribution as a prior. Use independent sampling to estimate the prior predictive distribution (the same as the posterior predictive except using sampling from the prior in the first step rather than the posterior), and show that its mean is approximately 50.

This distribution is a discrete uniform distribution between 0 and 100 ticks. To estimate this distribution we first of all draw a value of $\theta_i \sim beta(1,1)$, then draw a random sample $X_i \sim \mathcal{B}(100, \theta_i)$. Repeating this exercise a few thousand times we get a reasonably accurate prior predictive distribution. To do this in R,

```
fPriorPredictive <- function(numSamples, a, b, N){
  lX <- vector(length=numSamples)
  for(i in 1:numSamples){
    theta <- rbeta(1, a, b)
    lX[i] <- rbinom(1, N, theta)
  }
  return(lX)
}
```

```
lX <- fPriorPredictive(1000, 1, 1, 100)
hist(lX)
mean(lX)
```

Problem 13.1.2. In a single sample you find that there are 6 ticks that test positive for *Borrelia*. Assuming a $beta(1,1)$ prior graph the posterior distribution, and find its mean.

Since the beta prior here is conjugate to the binomial likelihood the posterior is also a beta distribution. To transform a beta prior into a posterior, we use the rule: $beta(a,b) \to beta(a+X, b+n-X)$, where (a,b) are the prior parameters, X is the number of ticks collected and n is the sample size. Since we are using a beta(1,1) prior here, the posterior is a beta(7,95) distribution; which has a posterior mean of $\frac{7}{102} \approx 0.069$.

Problem 13.1.3. Generate 100 independent samples from this distribution using your software's inbuilt (pseudo-)random number generator. Graph this distribution. How does it compare to the PDF of the exact posterior? (Hint: in R the command is "rbeta"; in Matlab it is "betarnd"; in Mathematica it is "RandomVariate[BetaDistribution...]"; in Python it is "numpy.random.beta".)

After only 100 independent samples the estimated posterior is quite similar in shape to the actual distribution (Figure 13.1).

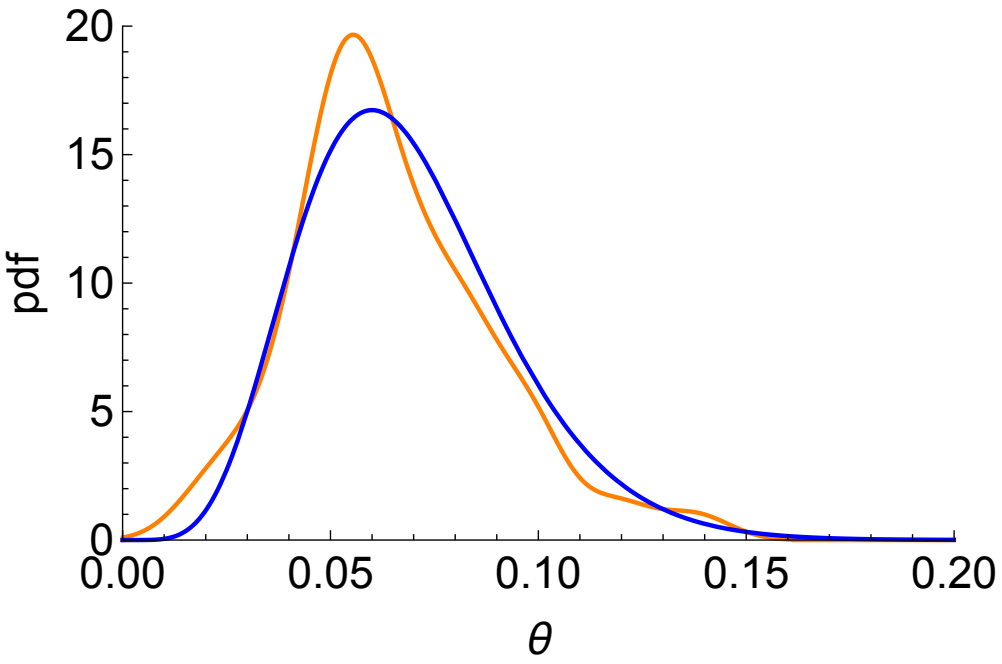

Figure 13.1: A PDF of the posterior estimated from independent samples (orange) versus the exact posterior (blue).

Problem 13.1.4. Determine the effect of increasing the sample size on using the independent sampler to estimate the posterior mean. (Hint: for each sample you are essentially comparing the sample mean with the true mean of the posterior.)

The error in using independent sampling to estimate the mean of a distribution is given by the (Lindberg-Lévy) central limit theorem:

$$(\bar{X} - \mathbb{E}[X]) \xrightarrow{d} \mathcal{N}(0, \sigma) \tag{13.1}$$

which means we can estimate the error for a large sample of size n by:

$$(\bar{X} - \mathbb{E}[X]) \approx \mathcal{N}(0, \frac{\sigma}{\sqrt{n}}) \tag{13.2}$$

This means that as the sample size increases the error in estimation decreases in accordance with $\sqrt{n}$ (Figure 13.2).

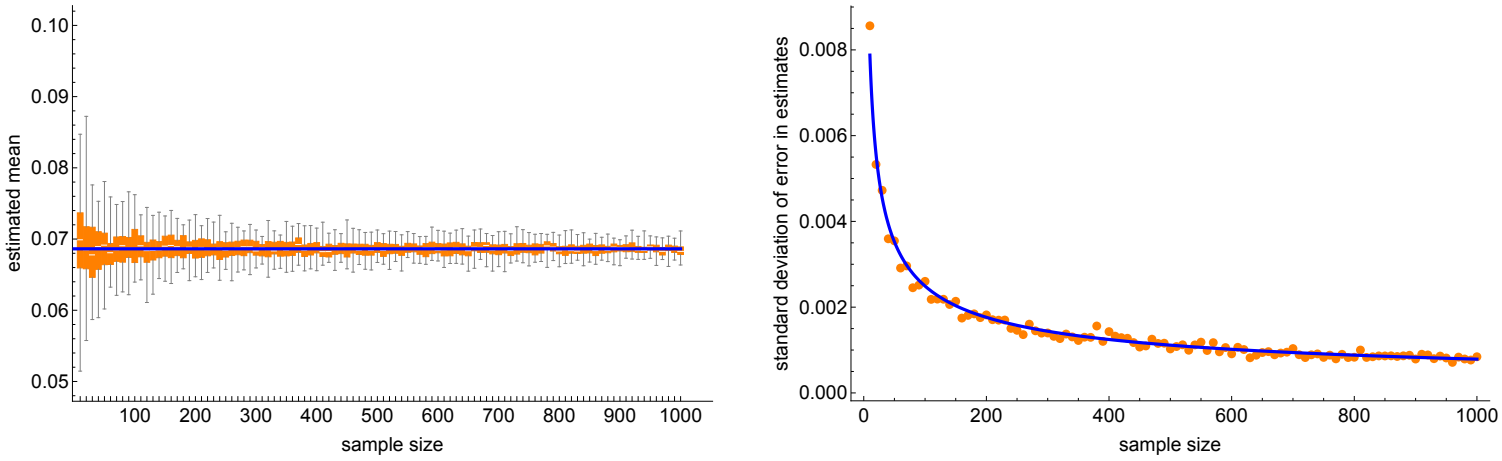

Figure 13.2: The estimated mean (left) and standard deviation in the error (right) using independent sampling to estimate the mean of the posterior for the ticks example.

Problem 13.1.5. Estimate the variance of the posterior using independent sampling for a sample size of 100. How does your sample estimate compare with the exact solution?

To do this generate an independent sample of size 100, and calculate the sample variance (see Figure 13.3.) Even after 100 samples we are able to estimate the posterior variance with quite reasonable resolution.

Problem 13.1.6. Create a proposal function for this problem that takes as input a current value of θ, along with a step size, and outputs a proposed value. For a proposal distribution here we use a normal distribution centred on the current θ value with a standard deviation (step size) of 0.1. This means you will need to generate a random θ from a normal distribution using your statistical software's inbuilt random number generator. (Hint: the only slight modification you need to make here is to ensure that we don't get $\theta < 0$ or $\theta > 1$ is to use periodic boundary conditions. To do this we use modular arithmetic. In particular we set $\theta_{proposed} = mod(\theta_{proposed}, 1)$. The command for this in R is $x\%\%1$; in Matlab the command is $mod(x, 1)$; in Mathematica it is $Mod[x, 1]$; in Python it is $x\%1$.)

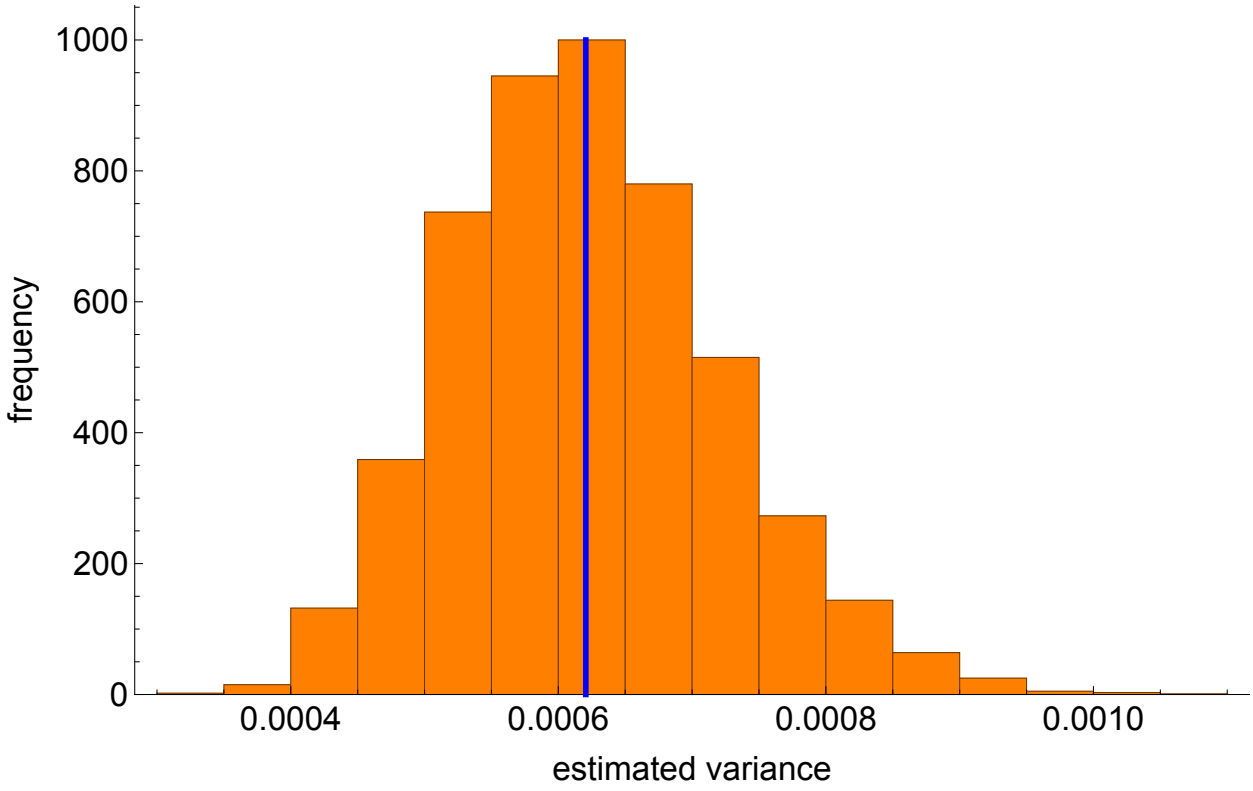

Figure 13.3: The estimated variance of the posterior for the ticks example versus the actual value (blue line) for an independent sample of size 100.

Problem 13.1.7. Create the "accept/reject" function of Random Walk Metropolis that accepts as input $\theta_{current}$ and $\theta_{proposed}$ and outputs the next value of θ. This is done based on a ratio:

$$r = \frac{p(X|\theta_{proposed}) \times p(\theta_{proposed})}{p(X|\theta_{current}) \times p(\theta_{current})} \tag{13.3}$$

and a uniformly-distributed random number between 0 and 1, which we call a. If $r > a$ then we update our current value of $\theta_{current} \to \theta_{proposed}$; alternatively we remain at $\theta_{current}$.

Problem 13.1.8. Create a function that combines the previous two functions; so it takes as input a current value of $\theta_{current}$, generates a proposed $\theta_{proposed}$, and updates $\theta_{current}$ in accordance with the Metropolis accept/reject rule.

Problem 13.1.9. Create a fully working Random Walk Metropolis sampler. (Hint: you will need to iterate the last function. Use a uniformly distributed random number between 0 and 1 as a starting point.)

Problem 13.1.10. For a sample size of 100 from your Metropolis sampler compare the sampling distribution to the exact posterior. How does the estimated posterior compare with that obtained via independent sampling using the same sample size?

The MCMC sample distribution is not as crisp as the independent sample distribution (Figure 13.4). This is because of the effects of dependence on the sampling efficiency. Intuitively, the information conveyed from each incremental sample is less than for the independent case.

There is also a slight bias in the MCMC posterior towards the starting point of the algorithm. This is because sampler hasn't had sufficient time to converge to the posterior. This bias can be removed by using more samples from the posterior and discarding those samples during the "warm-up" period.

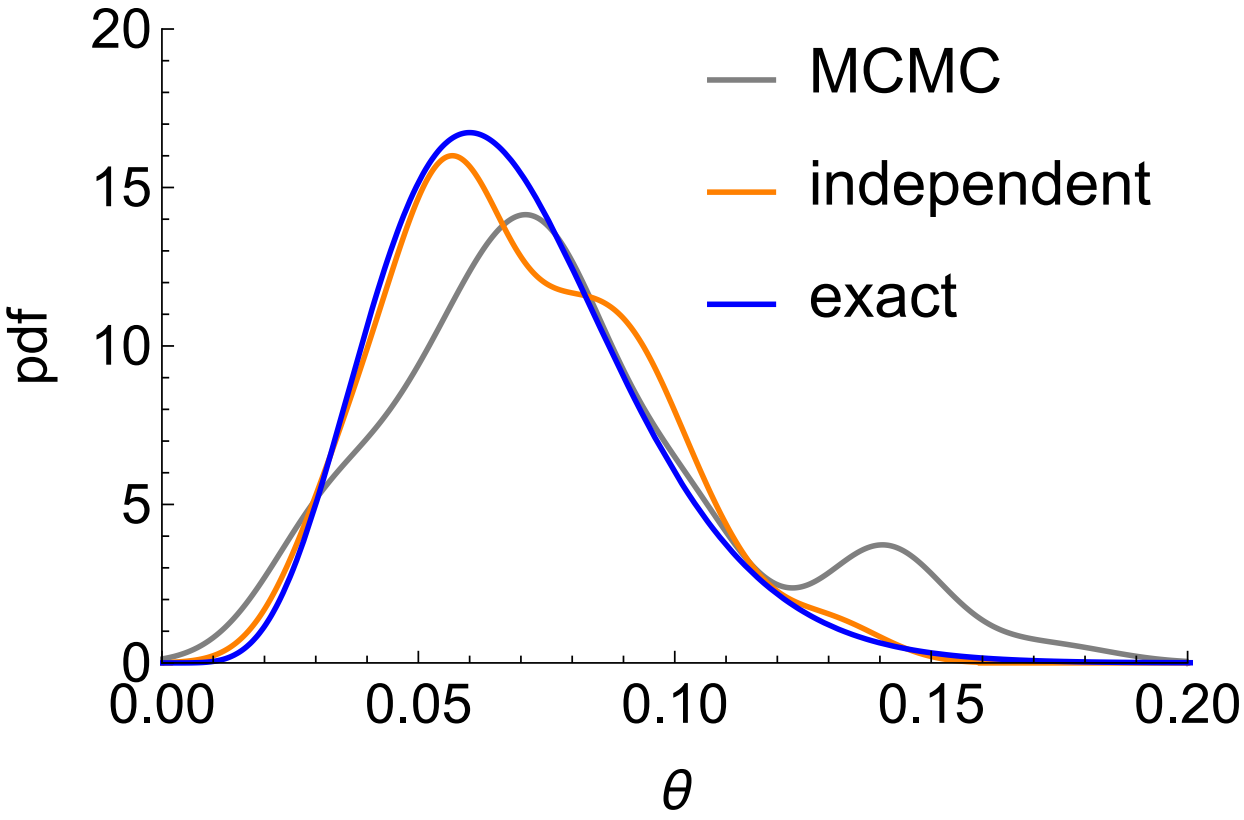

Figure 13.4: The estimated posterior via MCMC (orange) and independent (grey) sampling versus the exact posterior (blue).

Problem 13.1.11. Run 1000 iterations, where in each iteration you run a single chain for 100 iterations. Store the results in a 1000 x 100 matrix. For each iterate calculate the sample mean. Graph the resultant distribution of sample means. Determine the accuracy of the MCMC at estimating the posterior mean?

With only 100 samples we have not given the chains sufficient time to converge to the posterior density; specifically the effect of using a random start position that is *not* from the posterior means that our posteriors still reflect this. Therefore if we calculate the mean of our 100 posterior samples, it will tend to be upwardly biased of the true value because we haven't allowed for the "warm-up" period (Figure 13.5).

Problem 13.1.12. Graph the distribution of the sample means for the second 50 observations

of each chain. How does this result compare with that of the previous question? Why is there a difference?

The difference is solely due to the warm-up period being discarded (Figure 13.5). We have allowed the chains time to converge to the posterior, and hence by discarding the first 50 observations we reduce the effect of the random starting position. Since we are now using converged chains the estimator of the mean is unbiased.

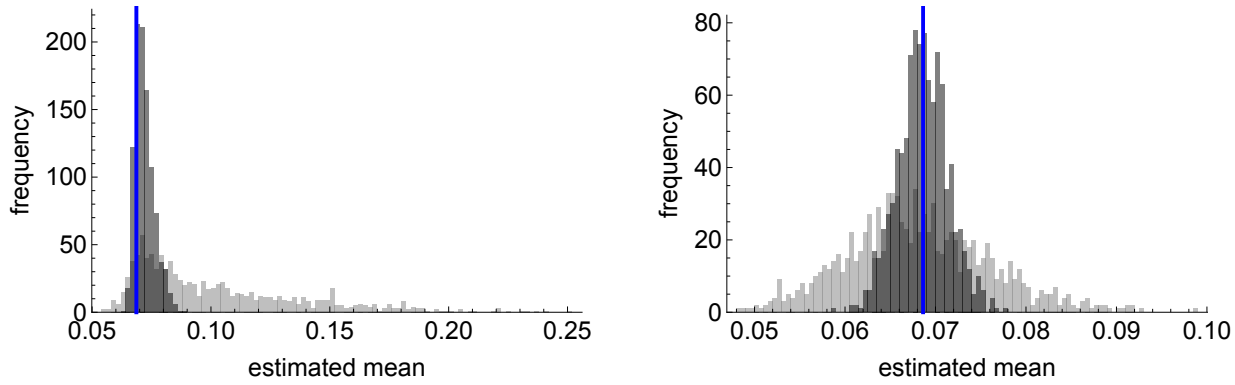

Figure 13.5: The sampling distribution of the sample mean of the MCMC runs for the ticks example, where (left) we use all 100 (light grey) or 1000 samples (dark grey) from each chain, and (right) we only use the second half of each.

Problem 13.1.13. Decrease the standard deviation (step size) of the proposal distribution to 0.01. For a sample size of 200, how the posterior for a step size of 0.01 compare to that obtained for 0.1?

A step size of 0.1 is able to find, then explore, the typical set at a much faster rate than the smaller step size (Figure 13.6); meaning that there is a lot of autocorrelation in the sampler's value. Intuitively, a sampler with a small step size is not able to move far from where it was at the end of the previous iteration!

Basically using a step size that is too low is equivalent to using a toothbrush in an archaeological dig. It takes you ages to find any hidden treasures!

Problem 13.1.14. Increase the standard deviation (step size) of the proposal distribution to 1. For a sample size of 200, how does the posterior for a step size of 1 compare to that obtained for 0.1?

Now the sampler is able to find the typical set fast enough. The trouble now is that it is inefficient at exploring it (Figure 13.7). Intuitively, the path of the sampler is characterised by a high rejection rate, since most of the proposed steps are a long way away from the region of high density.

Overall this means that the reconstructed probability mass at least lies in the correct region of parameter space. However, the reconstructed density has a high variance because there are relatively few unique samples relative to the density from a step size of 0.1.

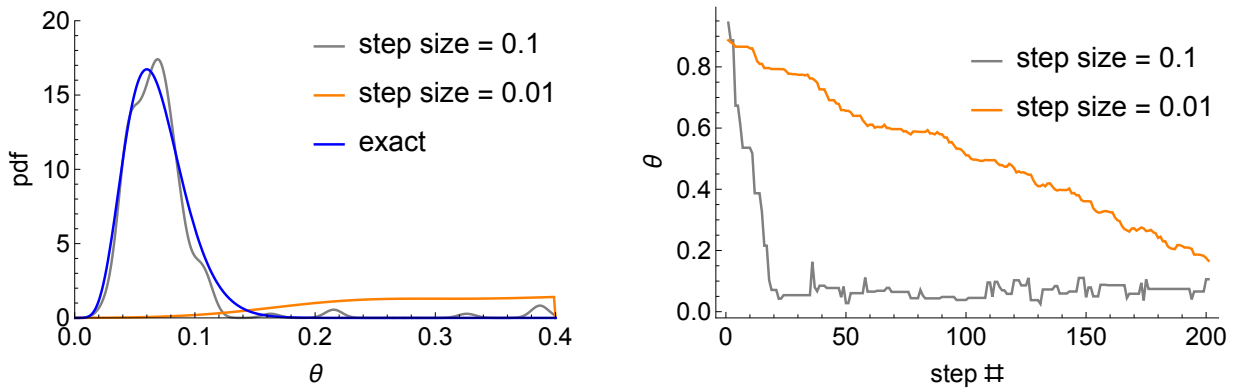

Figure 13.6: Left: the estimated posteriors for MCMC runs using two step sizes versus the actual. Right: the evolution of the path of each Markov Chain over time.

Basically using a step size that is too large is equivalent to using a digger in an archaeological dig; it finds the treasure fast enough but is too crude to save its finer details.

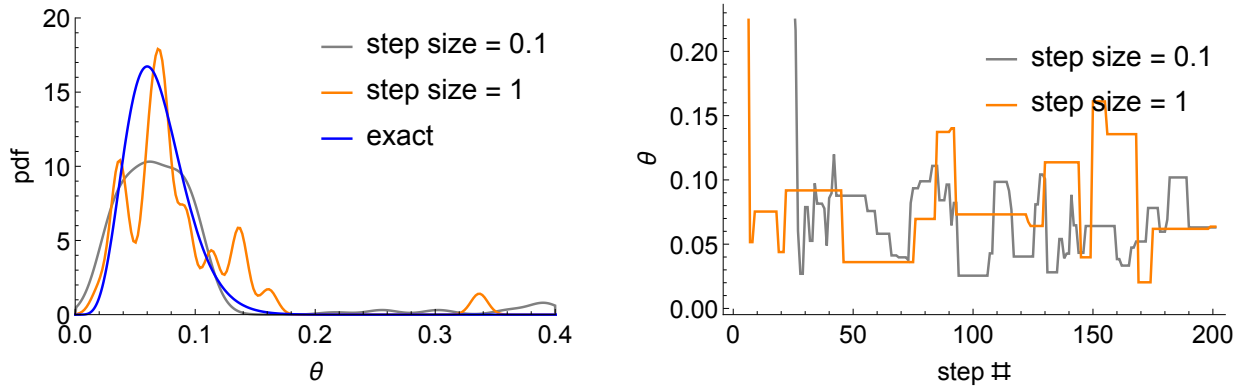

Figure 13.7: Left: the estimated posteriors for MCMC runs using two step sizes versus the actual. Right: the evolution of the path of each Markov Chain over time.

Problem 13.1.15. Suppose we collect data for a number of such samples (each of size 100), and find the following numbers of ticks that test positive for *Borrelia*: (3,2,8,25). Either calculate the new posterior exactly, or use sampling to estimate it. (Hint: in both cases make sure you include the original sample of 6.)

The posterior is a $beta(45, 457)$ distribution, which has a mean at about $\theta = 0.09$.

Problem 13.1.16. Generate samples from the posterior predictive distribution, and use these to test your model. What do these suggest about your model's assumptions?

Posterior predictive samples are unable to well-replicate either the minimum nor the maximum in the data (Figure 13.8). These suggest that either the assumption of independence or identical-distribution is violated; both of which there are good reasons for! (If one tick has the bacteria it

will infect nearby animals and in doing so, make it more likely for other ticks to become infected; meaning independence is likely violated. Following on from this we probably think that due to the contagious nature of the disease that there will be hotspots; meaning that the assumption of identical distribution is likely violated.)

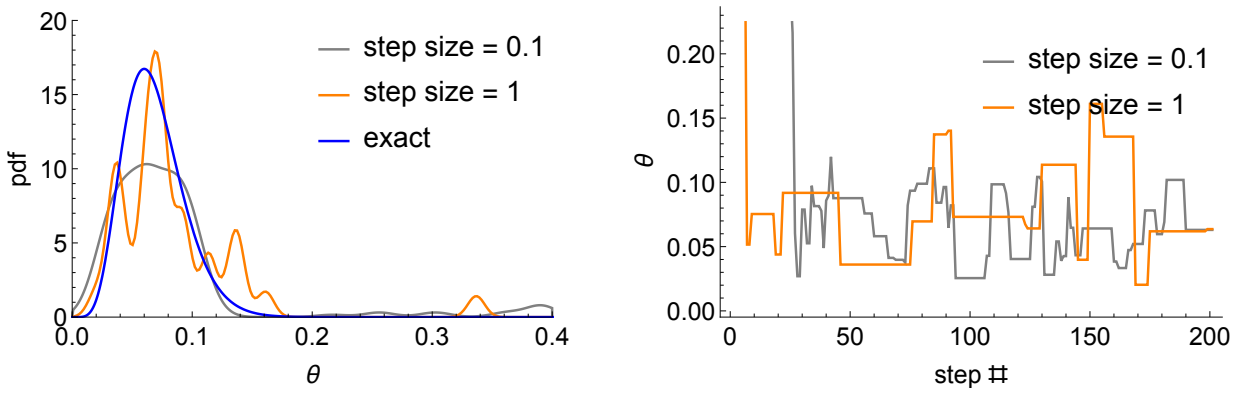

Figure 13.8: The posterior predictive distribution for a dataset of (6,3,2,8,25) *Borrelia*-positive ticks; each out of a sample of 100.

Problem 13.1.17. A colleague suggests as an alternative you use a beta-binomial likelihood, instead of the existent binomial likelihood. This distribution has two uncertain parameters $\alpha > 0$ and $\beta > 0$ (the other parameter is the sample size; $n = 100$ in this case), where the mean of the distribution is $\frac{n\alpha}{\alpha+\beta}$ Your colleague and you decide to use weakly informative priors of the form: $\alpha \sim \Gamma(1, \frac{1}{8})$ and $\beta \sim \Gamma(10, 1)$. (Here we use the parameterisation such that the mean of $\Gamma(a, b) = \frac{a}{b}$.) Visualise the joint prior in this case.

The joint prior is just the product of the individual priors because of independence between the parameters.

Problem 13.1.8. For this situation your colleague tells you that there are unfortunately no conjugate priors. As such, three possible solutions (of many) open to you are: 1. you use numerical integration to find the posterior parameters, or 2. use the Random Walk Metropolis-Hastings algorithm, or 3. you transform each of (α, β) so that they lie between $-\infty < \theta < \infty$. Why can't you use vanilla Random Walk Metropolis for (α, β) here?

The trouble is that the parameters are bounded to only be non-negative. Previously our parameter θ was bounded between 0 and 1, but we got around the issue of its bounds by using periodic boundary conditions; preserving the symmetry of the proposal distribution. This meant the we could just use vanilla Random Walk Metropolis with good sampling efficiency.

Here the problem is it is not possible to use periodic boundary conditions because there is only one boundary. This means that if we sample from (α, β) directly then we would ideally use an asymmetric proposal distribution (for example the log-normal); meaning we use the Metropolis-Hastings algorithm. Alternatively we transform (α, β) so that the transformed parameters are unbounded. Finally, because the model is fairly simple, having only two parameters, we can actually use numerical integration here to find the posterior.

Problem 13.1.19. By using one of the three methods above estimate the joint posterior distribution. Visualise the PDF of the joint posterior. How are α and β correlated here?

I used numeric integration here because the model is simple enough for it (see Figure 13.9). The Mathematica code is shown below,

```
fPrior1[\[Alpha]_, \[Beta]_] :=
  PDF[GammaDistribution[1, 8], \[Alpha]] PDF[
  GammaDistribution[8, 1], \[Beta]]

fLikelihood1[\[Alpha]_, \[Beta]_, x__, n_Integer] :=
  Likelihood[BetaBinomialDistribution[\[Alpha], \[Beta], n], x]

fPosteriorUnnormalised1[\[Alpha]_, \[Beta]_, x__, n_Integer] :=
  fPrior1[\[Alpha], \[Beta]] fLikelihood1[\[Alpha], \[Beta], x, n]

aInt = NIntegrate[
  fPosteriorUnnormalised1[\[Alpha], \[Beta], newdata,
    100], {\[Alpha], 0, \[Infinity]}, {\[Beta], 0, \[Infinity]}];

fPosterior1[\[Alpha]_, \[Beta]_, x__, n_Integer, aInt_] :=
  fPosteriorUnnormalised1[\[Alpha], \[Beta], x, n]/aInt

g1 = ContourPlot[
  fPosterior1[\[Alpha], \[Beta], newdata, 100, aInt], {\[Alpha], 0,
    2.1}, {\[Beta], 0, 20}, PlotRange -> Full, Evaluate@options3,
    FrameLabel -> {"\[Alpha]", "\[Beta]"}]
```

The parameters are positively correlated in Figure 13.9. This makes sense because the mean of the beta-binomial is $\frac{n\alpha}{\alpha+\beta}$. So if we increase α we need to increase β to ensure the mean is maintained, and we still are able to fit the data.

I have also used the other two methods; I used a log-normal jumping distribution for the Metropolis-Hastings set up. Here we use a log-normal whose mean is the current position of the Markov Chain. This is non-trivial because the mean of a log-normal isn't μ. The Mathematica code to do this is,

```
fStepMH[\[Alpha]_, \[Beta]_, aStepSize_] :=
  RandomVariate[
    LogNormalDistribution[1/2 (-aStepSize^2 + 2 Log[#]),
    aStepSize], {1}][[1]] & /@ {\[Alpha], \[Beta]}

fAcceptMH[lCurrent__, lProposed__, x__, n_Integer, aStepSize_] :=
  Module[{r = (
    fPosteriorUnnormalised1[lProposed[[1]], lProposed[[2]], x, n]/
    fPosteriorUnnormalised1[lCurrent[[1]], lCurrent[[2]], x, n])
    (PDF[LogNormalDistribution[
    1/2 (-aStepSize^2 + 2 Log[lProposed[[1]]]), aStepSize],
```

```
    lCurrent[[1]]] PDF[
    LogNormalDistribution[
    1/2 (-aStepSize^2 + 2 Log[lProposed[[2]]]), aStepSize],
    lCurrent[[2]]])/(PDF[
    LogNormalDistribution[
    1/2 (-aStepSize^2 + 2 Log[lCurrent[[1]]]), aStepSize],
    lProposed[[1]]] PDF[
    LogNormalDistribution[
    1/2 (-aStepSize^2 + 2 Log[lCurrent[[2]]]), aStepSize],
    lProposed[[2]]]), aRand = RandomReal[]},
    If[r > aRand, lProposed, lCurrent]]

fTakeStepMH[lCurrent__, x__, n_Integer, aStepSize_] :=
  Module[{lProposed =
    fStepMH[lCurrent[[1]], lCurrent[[2]], aStepSize]},
    fAcceptMH[lCurrent, lProposed, x, n, aStepSize]]

fMetropolisHastings[numSamples_Integer, lStart__, x__, n_Integer, aStepSize_] :=
  NestList[fTakeStepMH[#, x, n, aStepSize] &, lStart, numSamples]

n = 8000;
lSamples1 =
  Flatten[ParallelTable[
    fMetropolisHastings[n, RandomReal[{1, 2}, 2], newdata, 100, 0.5][[
      n/2 ;;]], {i, 1, 12, 1}], 1];

SmoothDensityHistogram[lSamples1, 0.5, PlotRange -> {{0, 2}, {0, 20}},
  Mesh -> 5, Evaluate@options3, FrameLabel -> {"\[Alpha]", "\[Beta]"}]
```

For the reparameterised model, used to allow us to do vanilla Metropolis, it is essential to include the Jacobian of the transformation in the new prior. In this case the Jacobian ends up being $e^{\alpha_1}e^{\beta_1}$ where $\alpha_1 = log(\alpha)$ and $\beta_1 = log(\beta)$. The Mathematica code for this is,

```
fLikelihood2[\[Alpha]1_, \[Beta]1_, x__, n_Integer] :=
  Likelihood[
    BetaBinomialDistribution[Exp[\[Alpha]1], Exp[\[Beta]1], n], x]

fPrior2[\[Alpha]1_, \[Beta]1_] :=
  PDF[NormalDistribution[0, 1], \[Alpha]1] PDF[
  NormalDistribution[0, 1], \[Beta]1] (Exp[\[Alpha]1] Exp[\[Beta]1])

fPosterior2Unnormalised[\[Alpha]1_, \[Beta]1_, x__, n_Integer] :=
  fLikelihood2[\[Alpha]1, \[Beta]1, x, n] fPrior2[\[Alpha]1, \[Beta]1]

fStep1[\[Alpha]1_, \[Beta]1_, aStepSize_] :=
  RandomVariate[
```

```
    MultinormalDistribution[{\[Alpha]1, \[Beta]1},
    aStepSize IdentityMatrix[2]], {1}][[1]]

fAccept1[lCurrent__, lProposed__, x__, n_Integer] :=
  Module[{r =
    fPosterior2Unnormalised[lProposed[[1]], lProposed[[2]], x, n]/
    fPosterior2Unnormalised[lCurrent[[1]], lCurrent[[2]], x, n],
    aRand = RandomReal[]}, If[r > aRand, lProposed, lCurrent]]

fTakeStep1[lCurrent__, x__, n_Integer, aStepSize_] :=
  Module[{lProposed = fStep1[lCurrent[[1]], lCurrent[[2]], aStepSize]},
    fAccept1[lCurrent, lProposed, x, n]]

fMetropolis1[numSamples_Integer, lStart__, x__, n_Integer, aStepSize_] :=
  NestList[fTakeStep1[#, x, n, aStepSize] &, lStart, numSamples]

fMetropolisTransformed[numSamples_Integer, lStart__, x__,
                       n_Integer, aStepSize_] :=
  Module[{lSample =
    fMetropolis1[numSamples, lStart, x, n, aStepSize]}, {Exp[#[[1]]],
      Exp[#[[2]]]} & /@ lSample]

n = 8000;
lSamples = ParallelTable[fMetropolisTransformed[n,
  RandomVariate[
    MultinormalDistribution[{0, 0}, IdentityMatrix[2]], {1}][[1]],
      newdata, 100, 0.05][[n/2 ;;]], {i, 1, 12, 1}];

SmoothDensityHistogram[Flatten[lSamples, 1], 0.5,
  PlotRange -> {{0, 2}, {0, 20}}, Mesh -> 5, Evaluate@options3,
  FrameLabel -> {"\[Alpha]", "\[Beta]"}]
```

Problem 13.1.20. Construct 80% credible intervals for the parameters of the beta-binomial distribution.

The posteriors for this example are shown in Figure 13.10. I actually found it easier to generate the quantiles via sampling here, and found that the 80% credible intervals were approximately:

$$0.59 \leq \alpha \leq 1.75$$
$$5.68 \leq \beta \leq 12.95$$

Problem 13.1.21. Carry out appropriate posterior predictive checks using the new model. How does it fare?

I used my sampled parameters here to simulate posterior predictive data (see Figure 13.11). The new posterior predictive data now encompasses the range seen in the actual data. Hence we can be more content with using this new model.

13.2 The fairground revisited

You again find yourself in a fairground, and where there is a stall offering the chance to win money if you participate in a game. Before participating you watch a few other plays of the game (by other people in the crowd) to try to determine whether you want to play.

Problem 13.2.1. In the most-boring version of the game, a woman flips a coin and you bet on its outcome. If the coin lands heads-up, you win; if tails, you lose. Based on your knowledge of similar games (and knowledge that the game must be rigged for the woman to make a profit!) you assume that the coin must be biased towards tails. As such you decide to specify a prior on the probability of the coin falling heads-up as $\theta \sim beta(2, 5)$. Graph this function, and – using your knowledge of the beta distribution – determine the mean parameter value specified by this prior.

The prior can be graphed using,

```
curve(dbeta(theta, 2, 5), xname='theta', xlab='theta', ylab='pdf')
```

whose mean is 2/7.

Problem 13.2.2. You watch the last 10 plays of the game, and the outcome is heads 3/10 times. Assuming a binomial likelihood, create a function that determines the likelihood for a given value of the probability of heads, θ. Hence or otherwise, determine the maximum likelihood estimate of θ.

The function is given below,

```
fLikelihood <- function(Z, theta, N){
  return(dbinom(Z, N, theta))
}
curve(fLikelihood(3, theta, 10), 0, 1, xname="theta", xlab='theta',
      ylab='likelihood')
```

The likelihood is peaked at 0.3 – the maximum likelihood estimate of the parameter value.

Problem 13.2.3. Graph the likelihood $\times$ prior. From the graph approximately determine the MAP θ estimate value.

This can be done using the previously-created likelihood function and the below function,

```
fLikelihoodTimesPrior <- function(Z, theta, N, a, b){
  return(fLikelihood(Z, theta, N) * dbeta(theta, a, b))
}

curve(fLikelihoodTimesPrior(3, theta, 10, 2, 5),
      xname='theta', xlab='theta')
```

which is peaked around 0.27. This can be determined with the below, although is not necessary for the approximate nature of this question,

```
optim(0.5, function(theta) -fLikelihoodTimesPrior(3, theta, 10, 2, 5),
      lower=0, upper=1, method="L-BFGS-B")
```

Note the "−" is necessary in the above because `optim` does minimisation by default.

Problem 13.2.4. By using R's `integrate` function find the denominator, and hence graph the posterior pdf.

The denominator can be found by the below,

```
integrate(function(theta) fLikelihoodTimesPrior(3, theta, 10, 2, 5), 0, 1)
```

which is approximately 0.164. Hence the posterior graph is just,

```
fPosterior <- function(Z, theta, N, a, b){
  aInt = integrate(function(theta1)
                   fLikelihoodTimesPrior(Z, theta1, N, a, b), 0, 1)[[1]]
  return((1 / aInt) * fLikelihood(Z, theta, N) * dbeta(theta, a, b))
}

curve(fPosterior(3, theta, 10, 2, 5), 0, 1, xname = 'theta',
      xlab = 'theta', ylab = 'pdf')
```

Problem 13.2.5. Use your posterior to determine your break-even/fair price for participating in the game, assuming that you win £1 if the coin comes up heads, and zero otherwise.

This is just the mean of the posterior. Using conjugate prior rules, the posterior is a beta(2+3,5+10-3) distribution, which has a mean of £5/17≈ 0.29. Alternatively, using your posterior, and R's numeric integration,

```
integrate(function(theta) theta * fPosterior(3, theta, 10, 2, 5), 0, 1)
```

which should give the same answer.

Problem 13.2.6. Another variant of the game is as follows: the woman flips a first coin – if it is tails you lose ($Y_i = 0$), and if it is heads you proceed to the next step. In this step, the woman flips another coin ten times, and records the number of heads, Y_i, which equals your winnings. Explain why a reasonable choice for the likelihood might be,

$$L(\theta, \phi|Y_i) = \begin{cases} (1-\theta) + \theta(1-\phi)^{10}, & \text{if } Y_i = 0 \\ \theta\binom{10}{Y_i}\phi^{Y_i}(1-\phi)^{10-Y_i}, & \text{if } Y_i > 0 \end{cases}$$

where θ and ϕ are the probabilities of the first and second coins falling heads-up, and Y_i is the score on the game.

Problem 13.2.7. Using the above formula, write down the overall log-likelihood for a series of N observations for $Y_i = (Y_1, Y_2, ..., Y_N)$.

Assuming conditional independence of the observations, the overall likelihood is given by,

$$L(\theta, \phi|Y_1, Y_2, ...Y_N) = \prod_{Y_i=0}\left[(1-\theta) + \theta(1-\phi)^{10}\right]\prod_{Y_i>0}\theta\binom{10}{Y_i}\phi^{Y_i}(1-\phi)^{10-Y_i} \tag{13.4}$$

$$= \left[(1-\theta) + \theta(1-\phi)^{10}\right]^{N_{Y_i=0}} \times \prod_{Y_i>0}\theta\binom{10}{Y_i}\phi^{Y_i}(1-\phi)^{10-Y_i} \tag{13.5}$$

where $N_{Y_i=0}$ is the number of times that $Y_i = 0$. Hence the log-likelihood is given by,

$$\log L(\theta, \phi|Y_1, Y_2, ...Y_N) = N_{Y_i=0}\log\left[(1-\theta) + \theta(1-\phi)^{10}\right] + N_{Y_i>0}\log\theta + \sum_{Y_i>0}\log\binom{10}{Y_i}\phi^{Y_i}(1-\phi)^{10-Y_i} \tag{13.6}$$

Problem 13.2.8. Using R's `optim` function determine the maximum likelihood estimate of the parameters for $Y_i = (3, 0, 4, 2, 1, 2, 0, 0, 5, 1)$.

Hint 1: Since R's `optim` function does minimisation by default, you will need to put a minus sign in front of the function to maximise it.

First of all we need a function that determines the log likelihood,

```
fLogLikelihoodHarderAll <- function(lY, theta, phi){
  N0 <- sum(lY == 0)
  N1 <- sum(lY > 0)
  lY1 <- lY[lY > 0]
  aLogLikelihood <- N0 * log((1 - theta) + theta * (1 - phi) ^ 10) +
                    N1 * log(theta) +
                    sum(sapply(lY1, function(Y) log(choose(10, Y) * phi ^ Y *
                                                      (1 - phi) ^ (10 - Y))))
```

```
  return(aLogLikelihood)
}
```

which we then use as an argument to,

```
lY <- c(3,0,4,2,1,2,0,0,5,1)
optim(c(0.2, 0.2),
      function(theta) -fLogLikelihoodHarderAll(lY, theta[1], theta[2]),
      lower = c(0.001, 0.001), upper=c(0.999, 0.999), method="L-BFGS-B")
```

where we have avoided the infinities by using bounds that are a bit away from the edge of the domain. The parameter estimates from this are $\theta \approx 0.75$ and $\phi \approx 0.24$.

Problem 13.2.9. Determine confidence intervals on your parameter estimates. (Hint 1: use the second derivative of the log-likelihood to estimate the Fischer Information matrix, and hence determine the Cramer-Rao lower bound. Hint 2: use Mathematica.)

Used Mathematica to do the differentiation here (I know this is cheating...).

```
fLogLikelihood[lY__, \[Theta]_, \[Phi]_] := Block[{N0 = Count[lY, 0],
  N1 = Count[lY, _?Positive],
  lY1 = Cases[lY, _?Positive]},
  N0 Log[(1 - \[Theta]) + \[Theta] (1 - \[Phi])^10] +
  N1 Log[\[Theta]] +
  Sum[Log[Binomial[10, Y] \[Phi]^Y (1 - \[Phi])^(10 - Y)], {Y, lY1}]]

fInformationMatrix[lY__, \[Theta]_, \[Phi]_] :=
  -D[fLogLikelihood[lY, \[Theta]1, \[Phi]1], {{\[Theta]1, \[Phi]1},
  2}] /. {\[Theta]1 -> \[Theta], \[Phi]1 -> \[Phi]}

fCRLB[lY_] := Block[{lParams =
  NMaximize[{fLogLikelihood[lY, \[Theta]1, \[Phi]1],
  0 <= \[Theta]1 <= 1, 0 <= \[Phi]1 <= 1},
  {\[Theta]1, \[Phi]1}][[2]], \[Theta], \[Phi]}, {\[Theta], \
  \[Phi]} = {\[Theta]1, \[Phi]1} /. lParams;
  1/Length@lY Inverse@fInformationMatrix[lY, \[Theta], \[Phi]]]
```

which when evaluated on the data yields a lower bound on the variances: $var(\theta) \approx 0.0025$ and $var(\phi) \approx 0.0003$, which are then used to calculate standard deviations: 0.05 and 0.02 for θ and ϕ respectively. Therefore the approximate 95% confidence intervals (based on the asymptotic normal approximation – multiply std. deviations by 1.96) are, $0.65 \leq \theta \leq 0.85$ and $0.21 \leq \phi \leq 0.27$.

Problem 13.2.10. Assuming uniform priors for both θ and ϕ create a function in R that calculates the unnormalised posterior (the numerator of Bayes' rule).

This is straightforward, since the priors are both unity, the numerator of Bayes' rule is just the likelihood. This function needs to be written in R however,

```
fLikelihoodHarderAll <- function(lY, theta, phi){
  N0 <- sum(lY == 0)
  N1 <- sum(lY > 0)
  lY1 <- lY[lY > 0]
  aLikelihood <- ((1 - theta) + theta * (1 - phi) ^ 10) ^ N0 *
    prod(sapply(lY1,
         function(Y) theta * choose(10, Y) * phi^Y * (1 - phi) ^ (10 - Y)))
  return(aLikelihood)
}

fUnnormalisedPosterior <- function(lY, theta, phi){
  return(fLikelihoodHarderAll(lY, theta, phi))
}
```

Problem 13.2.11. By implementing the Metropolis algorithm, estimate the posterior means of each parameter. (Hint 1: use a normal proposal distribution. Hint 2: use periodic boundary conditions on each parameter, so that a proposal off one side of the domain maps onto the other side.)

The various functions that are necessary to implement the sampling here are,

```
fProposal <- function(theta, phi, sigma){
  theta.prop <- rnorm(1, theta, sigma)
  phi.prop <- rnorm(1, phi, sigma)
  theta.prop <- theta.prop %% 1
  phi.prop <- phi.prop %% 1
  return(list(theta.prop=theta.prop, phi.prop=phi.prop))
}

fProposeAndAccept <- function(lY, theta, phi, sigma){
  lProposed <- fProposal(theta,phi,sigma)
  theta.prop <- lProposed$theta.prop
  phi.prop <- lProposed$phi.prop
  aCurrent <- fUnnormalisedPosterior(lY, theta, phi)
  aProposed <- fUnnormalisedPosterior(lY, theta.prop, phi.prop)
  r = aProposed / aCurrent
  if (r > runif(1)){
    theta.new = theta.prop
    phi.new = phi.prop
  }
  else{
    theta.new = theta
    phi.new = phi
  }
  return(list(theta = theta.new, phi=phi.new))
}
```

```
fMetropolis <- function(numIterations, lY, theta.start, phi.start, sigma){
  lTheta <- vector(length=numIterations)
  lPhi <- vector(length=numIterations)
  lTheta[1] <- theta.start
  lPhi[1] <- phi.start
  for(i in 2:numIterations){
    lParams <- fProposeAndAccept(lY, lTheta[i - 1], lPhi[i - 1], sigma)
    lTheta[i] <- lParams$theta
    lPhi[i] <- lParams$phi
  }
  return(list(theta=lTheta, phi=lPhi))
}
```

The sampler can be run for 100,000 samples, and a 2d density plot used to visualise the posterior samples using,

```
lSamples <- fMetropolis(100000, lY, 0.5, 0.5, 0.1)

library(ggplot2)
aDF <- data.frame(theta=lSamples$theta, phi=lSamples$phi)
ggplot(aDF, aes(x=theta, y=phi)) + geom_point() + geom_density2d()
```

and you should see much of the posterior weight around the maximum likelihood estimates we obtained previously. The posterior means are about 0.71 and 0.25 respectively.

Problem 13.2.12. Find the 95% credible intervals for each parameter.

This is easily done using the quantile function,

```
quantile(lSamples$theta, 0.025)
quantile(lSamples$theta, 0.975)
quantile(lSamples$phi, 0.025)
quantile(lSamples$phi, 0.975)
```

and you should get something like $0.42 \leq \theta \leq 0.96$ and $0.15 \leq \phi \leq 0.36$.

Problem 13.2.13. Using your posterior samples determine the fair price of the game. (Hint: find the mean of the posterior predictive distribution.)

A function that implements the game is,

```
fGenerateData <- function(N, theta, phi){
  lY <- vector(length=N)
  for(i in 1:N)
    lY[i] <- ifelse(theta > runif(1), rbinom(1, 10, phi), 0)
  return(lY)
```

```
}
```

Now all we do is feed in the respective values of θ and ϕ,

```
lY.posteriorPredictive <- vector(length=length(lSamples$theta))
for(i in 1:length(lSamples$theta)){
  lY.posteriorPredictive[i] <- fGenerateData(1, lSamples$theta[i],
                                             lSamples$phi[i])
}
hist(lY.posteriorPredictive)
mean(lY.posteriorPredictive)
```

which is about £1.75.

13.3 Malarial mosquitoes

Suppose that you work for the WHO where it is your job to research the behaviour of malaria-carrying mosquitoes. In particular, an important part of your research remit is to estimate adult mosquito lifespan. The lifespan of an adult mosquito is a critical determinant of the severity of malaria, since the longer a mosquito lives the greater the chance it has of a. becoming infected by biting an infected human; b. surviving the period where the malarial parasite undergoes a metamorphosis in the mosquito gut and migrates to the salivary glands; and c. passing on the disease by biting an uninfected host.

Suppose you estimate the lifespan of mosquitoes by analysing the results of a mark-release-recapture field experiment. The experiment begins with the release of 1000 young adult mosquitoes (assumed to have an adult age of zero); each of which has been marked with a fluorescent die. On each day (t) you attempt to collect mosquitoes using a large number of traps, and count the number of marked mosquitoes that you capture (X_t). The mosquitoes caught each day are then re-released unharmed. The experiment goes on for 15 days in total.

Since X_t is a count variable and you assume that the recapture of an individual marked mosquito is i.i.d., then you choose to use a Poisson model (as an approximation to the binomial since n is large):

$$\begin{aligned} X_t &\sim Poisson(\lambda_t) \\ \lambda_t &= 1000 \times exp(-\mu t)\psi \end{aligned}$$

where μ is the mortality hazard rate (assumed to be constant) and ψ is the daily recapture probability. You use a $\Gamma(2, 20)$ prior for μ (which has a mean of 0.1), and a beta(2,40) prior for ψ.

The data for the experiment is contained in the file `RWM_mosquito.csv`.

Problem 13.3.1. Using the data create a function that returns the likelihood. (Hint: it is easiest to first write a function that accepts (μ, ψ) as an input, and outputs the mean on a day t.)

The function that returns the mean on a given day is,

```
fMean <- function(mu, psi, t){
  return(1000 * exp(-mu * t) * psi)
}

curve(fMean(0.1, 0.05, t), 0, 20, xname='t')
```

Now creating a function that returns the likelihood,

```
fLikelihood <- function(mu, psi, lData){
  t <- lData$time
  X <- lData$recaptured
  lMean <- sapply(t, function(x) fMean(mu, psi, x))
  lLikelihood <- sapply(seq_along(t), function(i) dpois(X[[i]], lMean[[i]]))
  return(prod(lLikelihood))
}
```

Problem 13.3.2. Find the maximum likelihood estimates of (μ, ψ). (Hint 1: this may be easier if you create a function that returns the log-likelihood, and maximise this instead. Hint 2: use R's optim function.)

In Mathematica I found that ML estimates of $(\mu, \psi) = (0.097, 0.041)$. This was using the inbuilt NMaximise function which uses "Nelder-Meda" to find the maxima. To do this in R use,

```
fLogLikelihood <- function(params, lData){
  mu <- params[1]
  psi <- params[2]
  t <- lData$time
  X <- lData$recaptured
  lMean <- sapply(t, function(x) fMean(mu, psi, x))
  lLikelihood <- log(sapply(seq_along(t), function(i) dpois(X[[i]], lMean[[i]])))
  return(sum(lLikelihood))
}

optim(c(0.2, 0.1), function(params) -fLogLikelihood(params, lData),
      lower = c(0.001,0.001), upper=c(1,1), method='L-BFGS-B')
```

Problem 13.3.3. Construct 95% confidence intervals for the parameters. (Hint: find the information matrix, and use it to find the Cramer-Rao lower bound. Then find approximate confidence intervals by using the Central Limit Theorem.)

The point of this question is its difficulty to some extent. I want people to see how difficult it is to derive approximate estimates of the uncertainty of a parameter in Frequentist analyses. To do this you first of all find an estimate of the information matrix - essentially the negative of the Hessian matrix of second derivatives - at the ML estimates we found in the previous part. You then find its inverse, and the square root of its diagonal elements are the estimates of the parameter's standard error. To convert this to a confidence interval I am using a normal approximation, meaning that we

simply multiply the standard errors by 1.96 and add on to the parameter estimates. This results in the following:

$$0.07 \leq \mu \leq 0.12$$
$$0.033 \leq \psi \leq 0.049$$

Note these are approximate - I have used a normal approximation to derive these. This may not be particularly valid here since the parameters are close to zero. Also, note that these confidence intervals can contain negative values; to do things properly we should really transform to a unconstrained space then back to (0,1).

Problem 13.3.4. Write a function for the prior, and use this to create an expression for the un-normalised posterior.

Problem 13.3.5. Create a function that proposes a new point in parameter space using a log-normal proposal with mean at the current μ value, and a $beta(2+\psi, 40-\psi)$ proposal for ψ. (Hint: use a $log-\mathcal{N}(0.5(-\sigma^2+2log(\mu)),\sigma)$, where μ is the current value of the parameter.)

Problem 13.3.6. Create a function that returns the ratio of the un-normalised posterior at the proposed step location, and compares it to the current position.

Problem 13.3.7. Create a Metropolis-Hastings accept-reject function.

This isn't as trivial as for "vanilla" Metropolis - now we need to use the full Metropolis-Hastings accept-reject rule, which calculates the statistic:

$$r = \frac{p(\theta')}{p(\theta)} \times \frac{g(\theta|\theta')}{g(\theta'|\theta)} \tag{13.7}$$

where $g(\theta'|\theta)$ is the value of the PDF of the jumping kernel centred at current parameters, at the proposed parameter values. Since the two-proposal distributions are independent, we can find this just by multiplying together the log-normal and beta PDFs.

Problem 13.3.8. Create a Metropolis-Hastings sampler by combining your proposal and accept-reject functions.

Problem 13.3.9. Use your sampler to estimate the posterior mean of μ and ψ for a sample size of 4000 (discard the first 50 observations.) (Hint: if possible, do this by running 4 chains in parallel.)

The reject rate is pretty high here (probably because our proposal distribution takes no account of the posterior geometry), meaning that we are inefficient at exploring the posterior. However, after about 4000 samples we get a reconstructed posterior that looks similar to the exact one (Figure 13.12). The 80% credible intervals I obtain on each parameter are:

$$0.08 \leq \mu \leq 0.12$$
$$0.034 \leq \psi \leq 0.048$$

These are pretty similar to the approximate (95%) confidence intervals I obtained above.

Problem 13.3.10. By numeric integration compute numerical estimates of the posterior means of μ and ψ. How does your sampler's estimates compare with the actual values? How do these compare to the MLEs?

The exact values are $(\mu, \psi) = (0.096, 0.40)$ both of which lie right in the middle of the sampled credible intervals $\implies$ the sampler does a pretty good job here! The MLEs also look similar because we are using fairly wide priors here.

Problem 13.3.11. Carry out appropriate posterior predictive checks to test the fit of the model. What do these suggest might be a more appropriate sampling distribution? (Hint: generate a single sample of recaptures for each value of (μ, ψ) using the Poisson sampling distribution. You only need to do this for about 200 sets of parameter values to get a good idea.)

From the actual versus simulated data series it is evident that the posterior predictive samples do not reproduce the degree of variation we see in the data (Figure 13.13). In particular there are a number of days (2,7,8,10) where the actual recaptured value lies outside the posterior predictive range. This is because the assumption of independent recaptures, upon which the Poisson model is based, is likely violated. Intuitively individual mosquitoes will respond similarly to fluctuations in weather, which may cause them to be recaptured in "clumps". This lack of independence in the recaptures causes over-dispersion in the recapture data.

A more appropriate model that allows for the non-independence in recaptures is the negative binomial likelihood. This model becomes a Poisson distribution in the limit $\kappa \to \infty$, where $\frac{1}{\kappa}$ represents the degree of over-dispersion seen in the data.

Problem 13.3.12. An alternative model that incorporates age-dependent mortality is proposed where:

$$\lambda_t = 1000 \times exp(-\mu t^{\beta+1})\psi \tag{13.8}$$

where $\beta \geq 0$. Assume that the prior for this parameter is given by $\beta \sim exp(5)$. Using the same log-normal proposal distribution as for μ create a Random Walk Metropolis sampler for this new model. Use this sampler to find 80% credible intervals for the (μ, ψ, β) parameters.

The 80% credible intervals I obtained for the parameters were:

$$0.049 \leq \mu \leq 0.089$$
$$0.034 \leq \psi \leq 0.043$$
$$0.065 \leq \beta \leq 0.194$$

Problem 13.3.13. Look at a scatter plot of μ against β. What does this tell you about parameter identification in this model?

There is strong negative correlation between these parameter estimates (Figure 13.14). This suggests that it may be difficult to disentangle the effects of one parameter from another one. This makes intuitive sense; if $\mu \uparrow$ then $\beta \downarrow$ to allow lifespan to stay roughly constant.

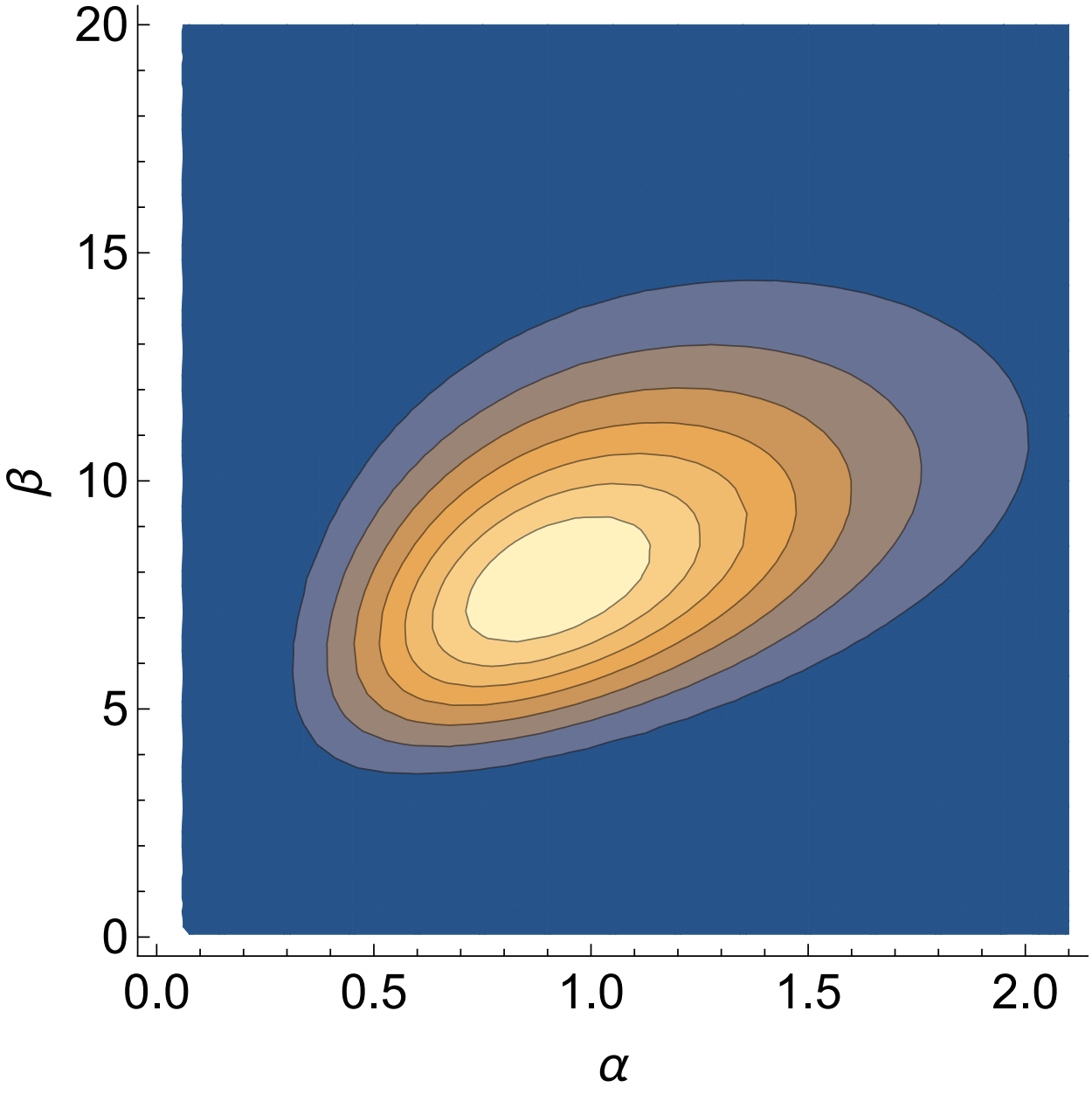

Figure 13.9: The joint posterior for the beta-binomial parameters.

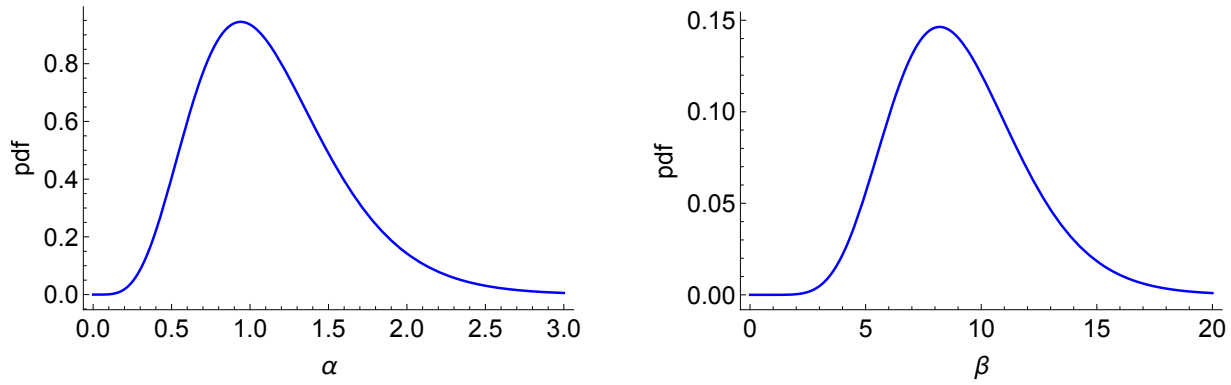

Figure 13.10: The posteriors for the beta-binomial parameters.

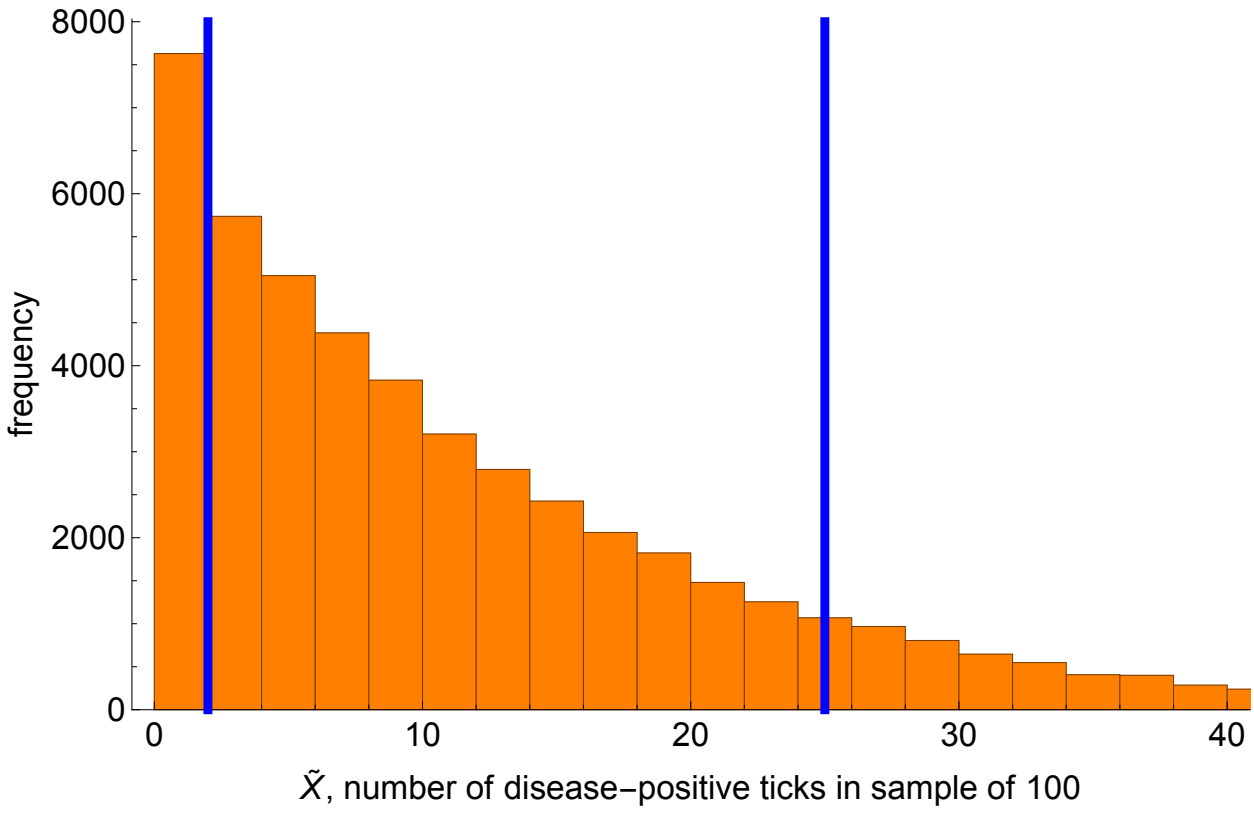

Figure 13.11: The posterior predictive distribution for the beta-binomial sampling model.

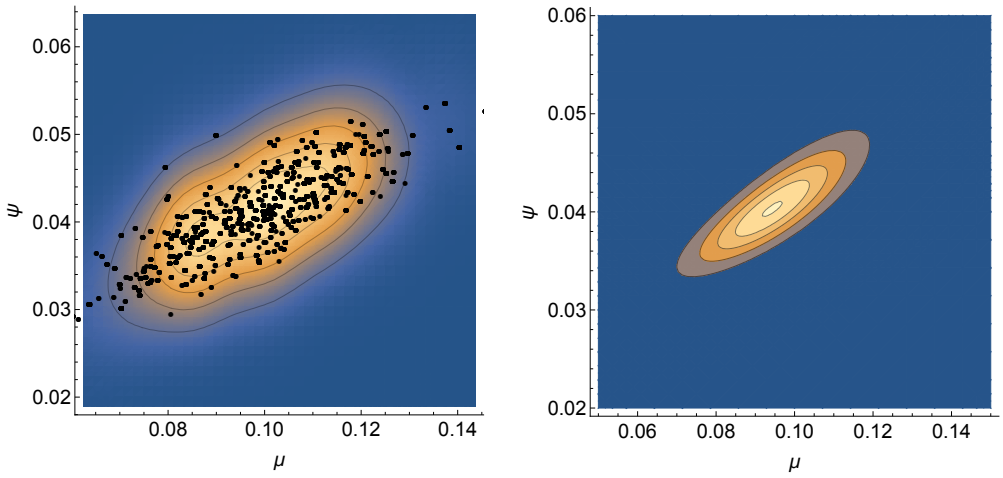

Figure 13.12: The sample-reconstructed posterior (left) versus the true posterior (right) for the mosquito question, where we assume a constant mortality rate.

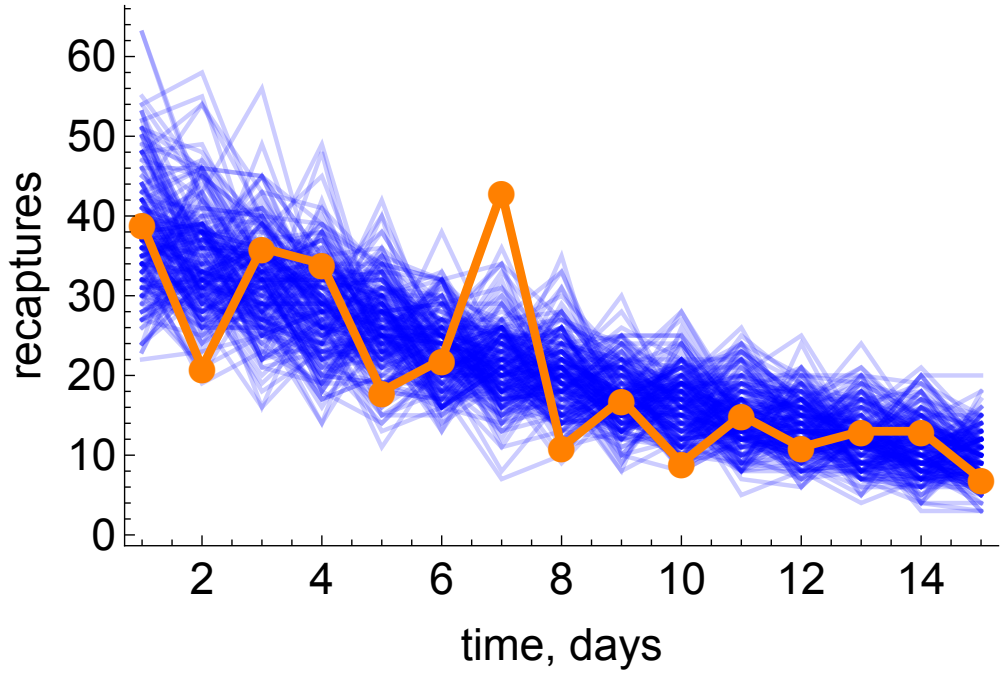

Figure 13.13: Samples from the posterior predictive distribution (blue) versus the actual recaptures (orange).

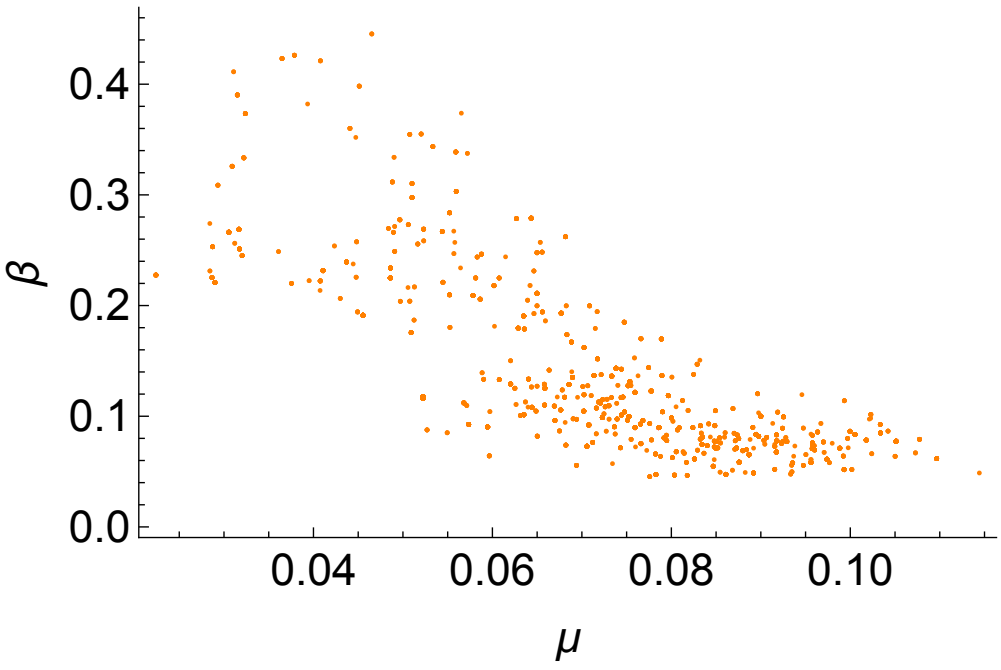

Figure 13.14: Posterior samples from (μ, β) for the mosquito model that incorporates age-dependent mortality.

Chapter 14

Gibbs sampling

14.1 The sensitivity and specificity of a test for a disease

Suppose that for a particular tropical disease no gold standard clinical test exists. Instead we have a test that is known to be imperfect; not always identifying a disease if the patient has the disease, and sometimes yielding false positives (patients that do not have the disease but test positive). However, by using this test in a clinical trial it is hoped that we can obtain better estimates for the disease sensitivity (S, the proportion of disease positive individuals who test positive) and specificity (C, the proportion of individuals who don't have the disease who test negative).

To do this we can construct a table of the observed and latent data for the test outcomes (see Table 14.1). In the table a and b are the number of observed positive and negative results respectively. Y_1 and Y_2 are latent variables that represent the gold standard – the true number of positive individuals out of a and b respectively.

Problem 14.1.1. Write down an expression for the likelihood, supposing that the prevalence for the disease is π. (Hint: multiply together the likelihoods corresponding to each of the cells in Table 14.1).

The likelihood of the top-left cell in Table 14.1 is $[\pi S]^{Y_1}$. This is the proportion of individuals with the disease (π) times the test specificity (S), giving the probability that we correctly identify one person as having the disease. This is raised to the power Y_1 since there are this many individuals with the disease in our sample.

Determining similar expressions for the other cells we obtain an overall likelihood,

		Truth		
		+	-	
Test	+	Y_1	$a - Y_1$	a
	-	Y_2	$b - Y_2$	b
		$Y_1 + Y_2$	$N - (Y_1 + Y_2)$	N

Table 14.1: Test outcomes versus true outcomes.

$$L(a,b|Y_1,Y_2,\pi,S,C) = \overbrace{[\pi S]^{Y_1}}^{\text{top-left: true +}} \times \overbrace{[\pi(1-S)]^{Y_2}}^{\text{bottom-left: false -}} \times \tag{14.1}$$

$$\underbrace{[(1-\pi)(1-C)]^{a-Y_2}}_{\text{top-right: false +}} \times \underbrace{[(1-\pi)C]^{b-Y_2}}_{\text{bottom-right: true -}} \tag{14.2}$$

Problem 14.1.2. Assuming priors of the form: $\pi \sim beta(\alpha_\psi, \beta_\pi)$, $S \sim beta(\alpha_S, \beta_S)$ and $C \sim beta(\alpha_C, \beta_C)$, it is possible to code up a Gibbs sampler for this problem [8] of the form,

$$Y_1|a,\pi,S,C \sim \mathcal{B}\left(a, \frac{\pi S}{\pi S + (1-\pi)(1-C)}\right) \tag{14.3}$$

$$Y_2|b,\pi,S,C \sim \mathcal{B}\left(b, \frac{\pi(1-S)}{\pi(1-S) + (1-\pi)C}\right) \tag{14.4}$$

$$\pi|a,b,Y_1,Y_2 \sim beta\left(Y_1 + Y_2 + \alpha_\pi, a + b - Y_1 - Y_2 + \beta_\pi\right) \tag{14.5}$$

$$S|Y_1,Y_2 \sim beta\left(Y_1 + \alpha_S, Y_2 + \beta_S\right) \tag{14.6}$$

$$C|a,b,Y_1,Y_2 \sim beta\left(b - Y_2 + \alpha_C, a - Y_1 + \beta_C\right) \tag{14.7}$$

Using the above expressions code up a working Gibbs sampler.

Problem 14.1.3. Suppose that out of a sample of 100 people, 20 of those tested negative and 80 positive. Assuming uniform priors on π, S and C, use Gibbs sampling to generate posterior samples for π. What do you conclude?

Since we have no idea what the true disease prevalence is, nor what the sensitivity or specificity of our test is, we cannot infer much here. So unsurprisingly, the posterior here is completely uninformative for π (left hand panel of Figure 14.1).

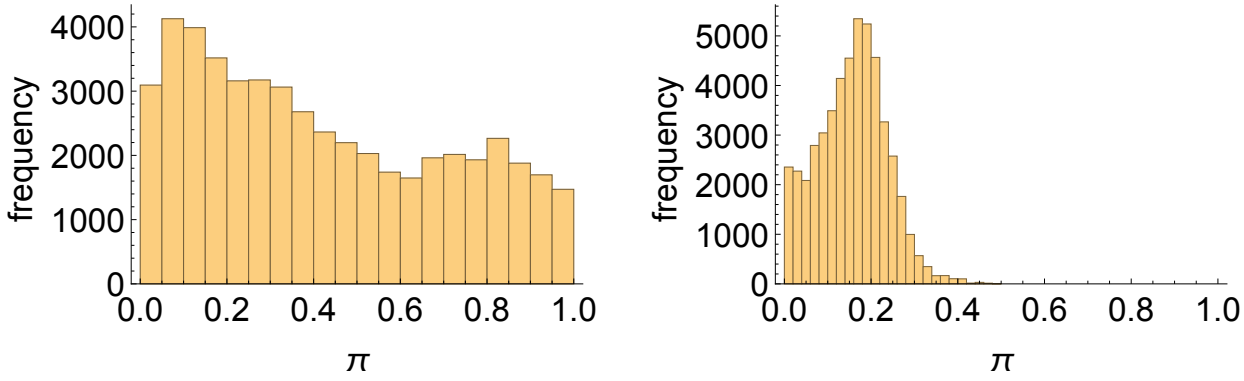

Figure 14.1: Posterior samples for left: uninformative, and right: informative priors being used for S and C.

Problem 14.1.4. Suppose that a previous study that compares the clinical test with a laboratory gold standard concludes that $S \sim beta(10,1)$ and $C \sim beta(10,1)$. Use Gibbs sampling to estimate the new posterior for π. Why does this look different to your previously-estimated distribution?

Since we have a reasonable knowledge of what the test sensitivity and specificity are, this means that we can now get a more concentrated estimate of what the disease prevalence is (right hand panel of Figure 14.1). This is the benefit of Bayesian statistics – we can put prior knowledge into a system and use that to determine better estimates of quantities of interest!

Problem 14.1.5. Suppose a previous analysis concluded that $\pi \sim beta(1, 10)$. Using this distribution as a prior, together with uniform priors on S and C, determine the posterior distributions for the test sensitivity and specificity respectively. Why does the test appear to be quite specific, although it is unclear how sensitive it is?

The posterior for S is fairly uniform between 0 and 1 (left hand panel of Figure 14.2). This is because using our prior for π we would assume that the true disease prevalence is somewhere around 10%,

```
hist(rbeta(100000, 1, 10))
```

So out of our sample of 100 individuals we would expect that only 10 of them have the disease. This is only a really small sample size, and even though it is potentially the case that we have identified all of these correctly, it is also possible that we have identified *none* of these correctly (if none of these fell within our 20 that tested positive). So intuitively, we still have no idea whether the people who tested positive are actually positive.

However, for C we expect that 90 people will not have the disease and we have predicted that 80 of them don't have the disease. This means regardless of who those people are, our test is pretty good at minimising false positives, and so we get a reasonably informative posterior for the specificity.

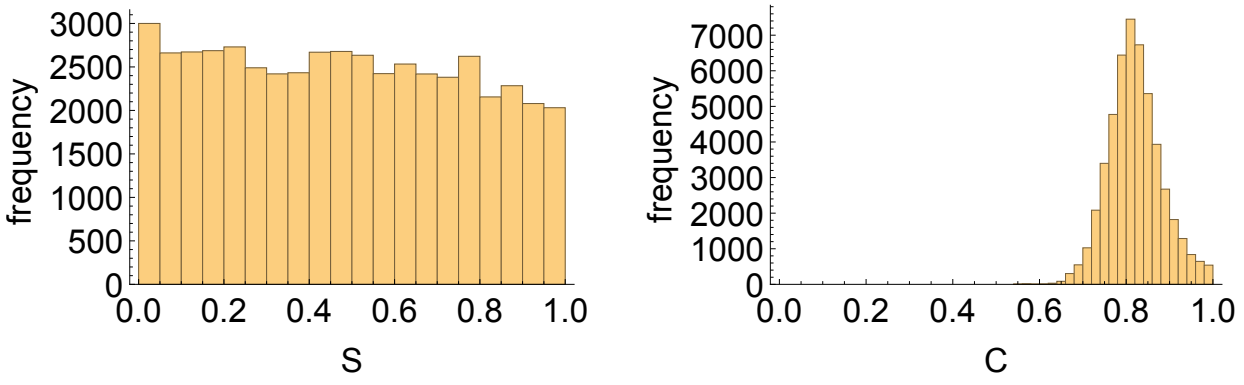

Figure 14.2: Posterior samples for left: test sensitivity (S), and right: specificity (C) assuming uniform priors on each of these parameters and $\pi \sim beta(1, 10)$.

Problem 14.1.6. Suppose that based on lab results you suppose that the test specificity $C \sim beta(10, 1)$, and $\pi \sim beta(1, 10)$, but the prior for S is still uniform. Explain the shape of the posterior for S now.

It is still uninformative (data not shown). Regardless of how specific our test is – there are still a large number of ways we could find 20/100 people having the disease. Only some of those ways actually include the 10 people that likely have the disease out of the positively-testing sample!

Problem 14.1.7. Now suppose that the sample size was 1000 people of which 200 tested positive. Using the same priors as the previous question, determine the posterior for S. What do you conclude about your test's sensitivity?

Still uninformative (data not shown). Nothing has changed from the previous question – we still have no idea whether the people who tested positive are actually positive. Collecting more data here won't help!

Problem 14.1.8. What do the previous results suggest is necessary to assess the sensitivity of a clinical test for a disease?

We need a gold standard to compare our test against! Intuitively, unless we know who the true positives are, we cannot hope to determine whether our test can detect the disease. Remember statistics will never give you a way to calculate something that can't be seen from a given experiment/data!

14.2 Coal mining disasters in the UK

The data in `gibbs_coal.csv` contains time series of the annual number of coal mining disasters in the UK from 1851-1961 [7]. In this question we are going to use Gibbs sampling to estimate the point in time when legislative and societal changes caused a reduction in coal mining disasters in the UK.

A model for the number of disasters D_t in a particular year t is,

$$D_t \sim \begin{cases} Poisson(\lambda_1), & \text{if } t \leq n \\ Poisson(\lambda_2), & \text{if } t > n \end{cases}$$

where λ_1 and λ_2 are the early and late mean disaster rates in the UK, and n is the time where the change in rates occurred.

Problem 14.2.1. Graph the data over time. Around what year (n) does it appear the disaster rate occurred?

This appears to occur roughly from 1880-1897.

Problem 14.2.2. Assuming the same $\lambda_i \sim \Gamma(a, b)$ priors for $i = \{1, 2\}$, and a discrete uniform prior for n between 1851-1861, determine out an expression for the full (un-normalised) posterior density.

The full density,

$$p(\lambda_1, \lambda_2, n|x_{1:N}, a, b) \propto p(x_{1:n}|\lambda_1)p(x_{n+1:N}|\lambda_2)p(\lambda_1|a, b)p(\lambda_2|a, b)p(n) \tag{14.8}$$

which can be explicitly calculated,

$$\begin{aligned} p(x_{1:n}|\lambda_1)p(x_{n+1:N}|\lambda_2)p(\lambda_1|a,b)p(\lambda_2|a,b)p(n) = & Poisson(x_{1:n}|\lambda_1)\times \\ & Poisson(x_{n+1:N}|\lambda_2)\times \\ & \Gamma(\lambda_1|a,b)\Gamma(\lambda_2|a,b)\times \\ & p(n) \end{aligned} \tag{14.9}$$

Problem 14.2.3. Determine the conditional distribution for λ_1 (i.e. $p(\lambda_1|x_{1:n}, n, a, b)$) by finding all those terms in the density that include λ_1, and removing the rest as constants of proportionality. (Hint: remember that a gamma prior is conjugate to a Poisson likelihood.)

We have that,

$$p(\lambda_1|x_{1:n}, n, a, b) \propto Poisson(x_{1:n}|\lambda_1)\Gamma(\lambda_1|a, b) \tag{14.10}$$

But we know that the gamma distribution is the conjugate prior to the Poisson likelihood, and so obtain,

$$\lambda_1|x_{1:n}, n, a, b \sim \Gamma(a + \sum_{t=1}^{n} x_t, b + n) \tag{14.11}$$

Problem 14.2.4. Using your answer to the previous problem write down the conditional distribution for λ_2.

By examining the previous answer we obtain,

$$\lambda_2|x_{n+1:N}, n, a, b \sim \text{gamma}(a + \sum_{t=n+1}^{N} x_t, b + N - n) \tag{14.12}$$

Problem 14.2.5. By collecting the terms that depend on n show that its conditional density can be written as,

$$p(n|x_{1:N}, \lambda_1, \lambda_2) \propto \lambda_1^{\sum_{t=1}^{n} x_t} e^{-n\lambda_1} \times \propto \lambda_2^{\sum_{t=n+1}^{N} x_t} e^{-(N-n)\lambda_2} \tag{14.13}$$

Taking only those terms that depend on n,

$$p(n|x_{1:N}, \lambda_1, \lambda_2) \propto Poisson(x_{1:n}|\lambda_1) \times Poisson(x_{n+1:N}|\lambda_2) \tag{14.14}$$

Note that the discrete uniform prior for n does not appear in the above because this is only a constant $(1/N)$. Now writing out each term explicitly, we have,

$$p(n|x_{1:N}, \lambda_1, \lambda_2) \propto \left(\prod_{t=1}^{n} \frac{\lambda_1^{x_t} e^{-\lambda_1}}{x_t!}\right) \times \left(\prod_{t=n+1}^{N} \frac{\lambda_2^{x_t} e^{-\lambda_2}}{x_t!}\right) \tag{14.15}$$

$$\propto \lambda_1^{\sum_{t=1}^{n} x_t} e^{-n\lambda_1} \times \propto \lambda_2^{\sum_{t=n+1}^{N} x_t} e^{-(N-n)\lambda_2} \tag{14.16}$$

where we note that we have dropped the $x_t!$ terms because these are constants.

Problem 14.2.6. Write a function in R that calculates the un-normalised expression for $p(n|x_{1:N}, \lambda_1, \lambda_2)$ for a single value of n. Hint: remember that the change point cannot occur at the last data point, and so return 0 for this case!

This can be done using,

```
fPointnConditional <- function(n, X, lambda.1, lambda.2){
  N <- length(X)
  aSum.1 <- sum(X[1:n])
  aSum.2 <- sum(X[(n+1):N])
  if(n == 111){
    return(0)
  }else{
    return(lambda.1 ^ aSum.1 * exp(-n * lambda.1) * lambda.2 ^ aSum.2 *
           exp(-(N - n) * lambda.2))
  }
}
```

Problem 14.2.7. Create a function that calculates the discrete probability distribution across all values of n. (Hint: remember this must be a valid probability distribution.)

All you need to do is calculate the point-wise un-normalised density, and then normalise it.

```
fAllnConditional <- function(X, lambda.1, lambda.2){
  lProb <- sapply(seq_along(X), function(n)
                                fPointnConditional(n, X, lambda.1, lambda.2))
  return(lProb / sum(lProb))
}
```

Problem 14.2.8. Create a function that independently samples from the discrete distribution for n you calculate. (Hint use the "sample" function.)

Such a function is,

```
fSampleN <- function(lambda.1, lambda.2, X){
  sample(seq_along(X), 1, prob=fAllnConditional(X, lambda.1, lambda.2))
}
```

For $\lambda_1 = 3$ and $\lambda_2 = 1$ we can draw a histogram of 1,000 draws from the distribution for n,

```
hist(sapply(1:1000, function(i) fSampleN(3, 1, X)), xlab='n')
```

which should be peaked around $n = 40$ which corresponds to the change-point being 1891 since we are using reasonable values for λ.

Problem 14.2.9. Write functions to independently sample from the conditional distributions for λ_1 and λ_2 that you previously determined.

Two such functions are,

```
fSampleLambda1 <- function(X, a, b, n){
  aSum <- sum(X[1:n])
  return(rgamma(1, (a + aSum), (b + n)))
}

fSampleLambda2 <- function(X, a, b, n){
  N <- length(X)
  aSum <- sum(X[(n + 1):N])
  return(rgamma(1, (a + aSum), (b + N - n)))
}
```

Problem 14.2.10. Combine all three previously created sampling functions to create a working Gibbs sampler. Hence estimate the change-point and its 95% central credible intervals.

I first create a function that does one iteration for the Gibbs sampler,

```
fGibbsSingle <- function(n, lambda.1, lambda.2, X, a, b){
  n <- fSampleN(lambda.1, lambda.2, X)
  lambda.1 <- fSampleLambda1(X, a, b, n)
  lambda.2 <- fSampleLambda2(X, a, b, n)
  return(list(n=n, lambda.1=lambda.1, lambda.2=lambda.2))
}
```

And then iterate using the above to create a Gibbs sampler,

```
fGibbs <- function(numIterations, n, lambda.1, lambda.2, X, a, b){
  mSamples <- matrix(nrow=numIterations, ncol=3)
  mSamples[1,] <- c(n, lambda.1, lambda.2)
  for(i in 2:numIterations){
    nPrev <- mSamples[(i - 1), 1]
    lambda.1Prev <- mSamples[(i - 1), 2]
    lambda.2Prev <- mSamples[(i - 1), 3]
```

```
    lParams <- fGibbsSingle(nPrev, lambda.1Prev, lambda.2Prev, X, a, b)
    mSamples[i,] <- c(lParams$n, lParams$lambda.1, lParams$lambda.2)
  }
  colnames(mSamples) <- c("n", "lambda.1", "lambda.2")
  return(as.data.frame(mSamples))
}
```

I start the Gibbs sampler at non-random values here, but to do this properly I should pick these using some sampling distribution,

```
mTest <- fGibbs(10000, 40, 3, 1, X, 1, 1)
hist(mTest$n)
```

The median estimate I obtain for n is 40/41, with its 2.5% and 97.5% quantiles being 36 and 46 respectively.

Problem 14.2.11. Using your sampler determine posterior median estimates for λ_1 and λ_2.

The posterior medians are roughly 3.06, and 0.93 respectively.

14.3 Bayesian networks

Suppose that when you leave your house in the morning you notice that the grass is wet. However you do not know whether the grass is wet because of last night's rain, or because the sprinkler went on in the night. You want to determine the cause of the wet grass, because this affects whether you need to water your plants in your windowsill. The causal pathway of the grass being wet is shown in Figure 14.3, along with the probabilities of the states. Here "cloudy" means that the night was completely overcast.

Problem 14.3.1. Show that the probability that it was cloudy last night conditional on the sprinkler being on, that it rained, and the grass being wet is approximately 0.444.

This can be done with the following computation,

$$p(c|r,s,w) = p(c|r,s) = \frac{p(r,s|c)p(c)}{p(r,s)} \tag{14.17}$$

$$= \frac{p(r|c)p(s|c)p(c)}{p(r|c)p(s|c)p(c) + p(r|\neg c)p(s|\neg c)p(\neg c)} \tag{14.18}$$

$$= \frac{0.8 \times 0.1 \times 0.5}{0.8 \times 0.1 \times 0.5 + 0.2 \times 0.5 \times 0.5} \tag{14.19}$$

$$\approx 0.444 \tag{14.20}$$

Problem 14.3.2. Show that the probability that it rained given that it was cloudy, the sprinkler was on, and the grass is wet is approximately 0.815.

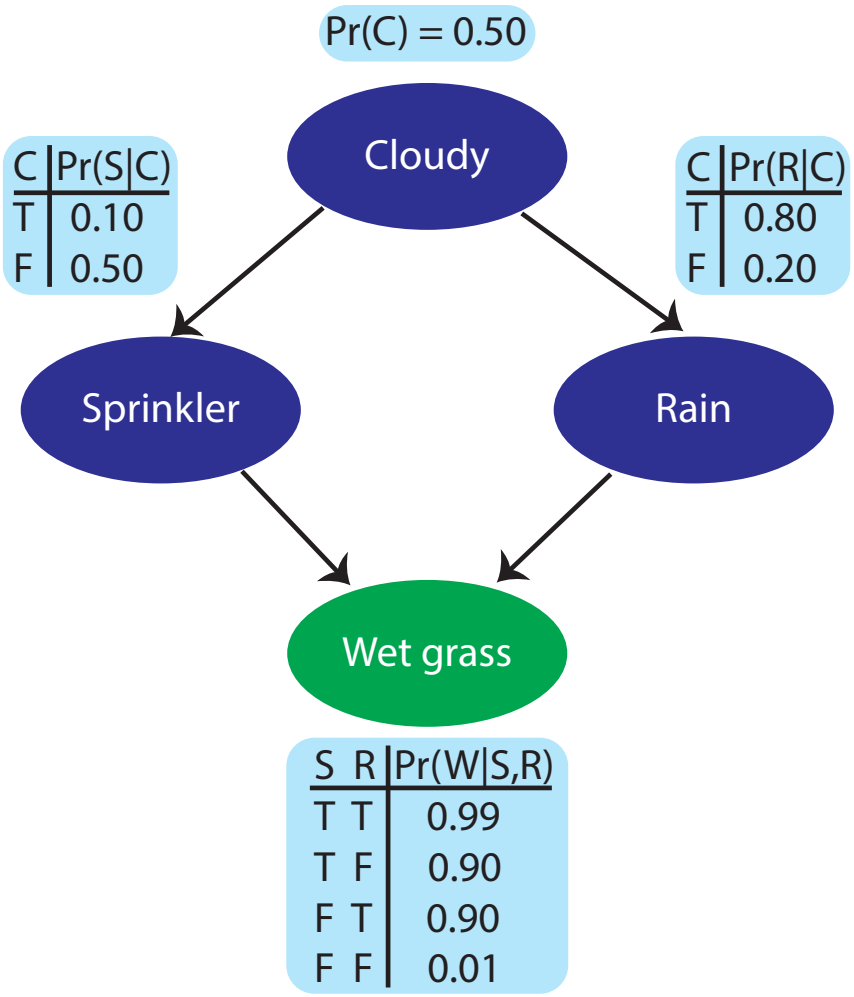

Figure 14.3: A Bayesian network for the wet grass example.

This can be done as follows,

$$p(r|c,s,w) = \frac{p(w|r,c,s)p(r|c,s)}{p(w|c,s)} \tag{14.21}$$

$$= \frac{p(w|r,s)p(r|c)}{p(w|c,s)} \tag{14.22}$$

$$= \frac{p(w|r,s)p(r|c)}{p(w|r,c,s)p(r|c) + p(w|\neg r,c,s)p(\neg r|c)} \tag{14.23}$$

$$= \frac{p(w|r,s)p(r|c)}{p(w|r,s)p(r|c) + p(w|\neg r,s)p(\neg r|c)} \tag{14.24}$$

$$= \frac{0.99 \times 0.8}{0.99 \times 0.8 + 0.90 \times 0.2} \tag{14.25}$$

$$\approx 0.815 \tag{14.26}$$

Problem 14.3.3. The remaining (non-trivial) conditional probabilities are $p(c|\neg r,s,w) = 0.048$

and $p(r|\neg c, s, w) = 0.216$. Suppose that when we walk outside we see that the grass is wet, and we also know that the sprinkler went on last night (we were woken by its noise). Create a Gibbs sampler to determine the unconditional probability that it was cloudy. Then find the unconditional probability as to whether it rained.

The conditional probabilities that we require to do this problem. These are shown in Figure 14.4. I implemented this sampler is steps. The first thing I did was create a transition function that accepts a probability and returns a binary outcome,

```
fTransition <- function(aProb){
  indicator <- rbinom(1, 1, aProb)
  return(indicator)
}
```

Then we need a function that calculates the probabilities of the various outcomes,

```
fProbCalculator <- function(cloud_or_rain, cloud, rain){
  if(cloud_or_rain){
    if(rain){
      aProb <- 0.444
    }else{
      aProb <- 0.048
  }
  }else{
    if(cloud){
      aProb <- 0.815
    }else{
      aProb <- 0.216
    }
  }
  return(aProb)
}
```

where `cloud_or_rain` is a boolean that indicates whether we are updating cloud (if true) or rain (if false). I then created two functions that update the state, and choose which state to update respectively,

```
fGibbsStep <- function(cloud_or_rain, cloud, rain){
  aProb <- fProbCalculator(cloud_or_rain, cloud, rain)
  aNewState <- ifelse(fTransition(aProb) == 1, T, F)
  if(cloud_or_rain){
    cloud <- aNewState
  }else{
    rain <- aNewState
  }
  return(list(cloud=cloud, rain=rain))
}
```

```
fGibbsSelector <- function(cloud, rain){
  cloud_or_rain <- ifelse(rbinom(1, 1, 0.5), T, F)
  return(fGibbsStep(cloud_or_rain, cloud, rain))
}
```

Finally we iterate over the latter function to create our working Gibbs sampler,

```
fGibbsComplete <- function(numIterations, cloud, rain){
  lCloud <- vector(length=numIterations)
  lRain <- vector(length=numIterations)
  lCloud[1] <- cloud
  lRain[1] <- rain
  for(i in 2:numIterations){
    lState <- fGibbsSelector(lCloud[i - 1], lRain[i - 1])
    lCloud[i] <- lState$cloud
    lRain[i] <- lState$rain
  }
  return(list(cloud=lCloud, rain=lRain))
}
```

Now running the sampler for 10,000 iterations we obtain unconditional probabilities of cloud and rain,

```
lTest <- fGibbsComplete(10000, T, T)
hist(ifelse(lTest$cloud, 1, 0))
hist(ifelse(lTest$rain, 1, 0))
mean(mean(ifelse(lTest$cloud, 1, 0))
mean(ifelse(lTest$rain, 1, 0))
```

so that $Pr(cloud) \approx 0.17$ and $Pr(rain) \approx 0.32$.

Problem 14.3.4. Using your Gibbs sampler determine the joint probability that it was cloudy and it rained.

This is straightforward to obtain from our sampler,

```
lTest <- fGibbsComplete(100000, T, T)
mean(ifelse(lTest$cloud, 1, 0)==1 & ifelse(lTest$rain, 1, 0)==1)
```

which should be roughly 0.14.

Problem 14.3.5. Visualise the path of your Gibbs sampler by creating a network graph showing the frequency of transitions between the four states. Hint 1: first create an adjacency matrix by calculating the frequency of transitions between all states. Hint 2: to visualise the graph in R use the iGraph package.

First create indicator variables representing the four states,

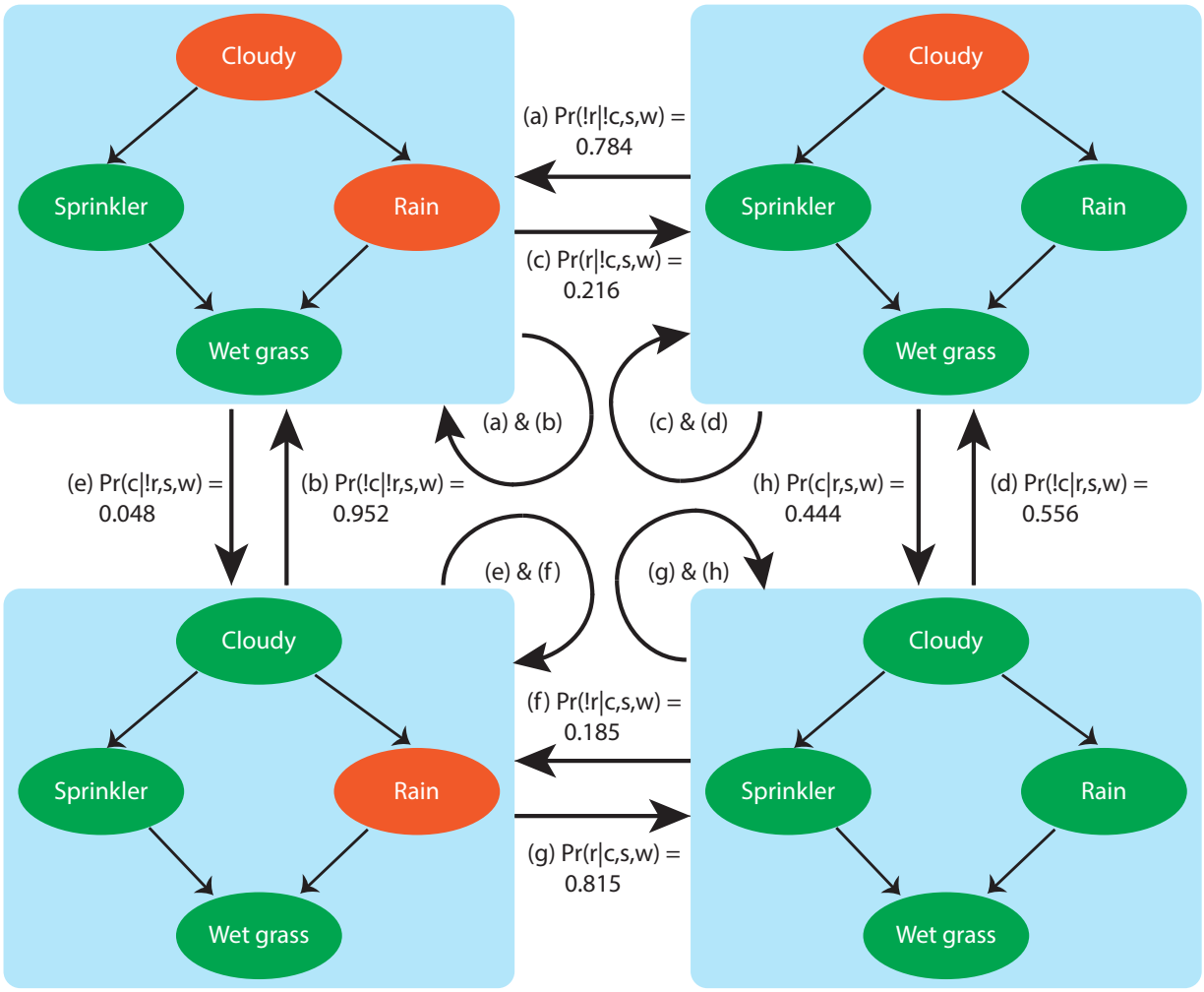

Figure 14.4: A schematic of a Gibbs sampler for the wet grass example. Green means that the event occurred, and orange that it didn't. **Replace ! with ¬ for consistency.**

```
aDF <- data.frame(cloud=lTest$cloud, rain=lTest$rain,
                  A=(!lTest$cloud) & (!lTest$rain),
                  B=(lTest$cloud) & (!lTest$rain),
                  C=(!lTest$cloud) & (lTest$rain),
                  D=(lTest$cloud) & (lTest$rain))
```

Then create a function that determines the transitions between two states,

```
fGetTransitions <- function(lTest, from, to){
  aLen <- length(lTest$cloud)
  vTransition <- vector(length=(aLen - 1))
  for(i in 2:aLen){
    if(aDF[i - 1, (2 + from)] & aDF[i, (2 + to)])
      vTransition[i] <- 1
```

```
    else
      vTransition[i] <- 0
  }
  return(vTransition)
}
```

Then determine all possible combinations of state transitions, and fill the adjacency matrix,

```
lCombs <- expand.grid(rep(list(1:4), 2))
mAdjacency <- matrix(nrow=4, ncol=4)
for(i in 1:16){
  print(i)
  mAdjacency[lCombs[i, 1], lCombs[i, 2]] <- sum(fGetTransitions(lTest,
                                                 lCombs[i, 1], lCombs[i, 2]))
}
```

Finally use iGraph to visualise the results,

```
library(igraph)
rownames(mAdjacency) <- c('C=0,R=0', 'C=1,R=0', 'C=0,R=1', 'C=1,R=1')
colnames(mAdjacency) <- rownames(mAdjacency)
ig <- graph.adjacency(mAdjacency, mode="directed", weighted=TRUE)
plot(ig, edge.width=E(ig)$weight / 10000 + 1, edge.label=round(E(ig)$weight, 3),
edge.arrow.size=1.5, edge.curved=TRUE)
```

The resultant graph should look something like Figure 14.5.

14.4 Proofs

Problem 14.4.1. Prove that the Gibbs sampler can be viewed as a case of Metropolis-Hastings.

The acceptance ratio for the Metropolis-Hastings is of the form,

$$r = \frac{p(\theta_{t+1}|data)}{p(\theta_t|data)} \times \frac{J(\theta_t|\theta_{t+1})}{J(\theta_{t+1}|\theta_t)}. \tag{14.27}$$

In Gibbs sampling the jumping probabilities are conditional distributions, so that,

$$J(\theta_{t+1}|\theta_t) = p(\theta_{t+1}|\theta_t, data). \tag{14.28}$$

This means we can rewrite the above ratio as,

$$r = \frac{p(\theta_{t+1}|data)}{p(\theta_t|data)} \times \frac{p(\theta_t|\theta_{t+1}, data)}{p(\theta_{t+1}|\theta_t), data} \tag{14.29}$$

$$= \frac{p(\theta_{t+1}, \theta_t|data)}{p(\theta_{t+1}, \theta_t|data)} \tag{14.30}$$

$$= 1 \tag{14.31}$$

So in other words we always accept, as we do in Gibbs sampling.

Problem 14.4.2. For the following bivariate normal density,

$$\begin{pmatrix} u \\ c \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} a \\ b \end{pmatrix}, \quad \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right],$$

show that the conditional densities are of the form,

$$u|c \sim \mathcal{N}(a + \rho(c - b), \sqrt{1 - \rho^2}) \tag{14.32}$$

$$c|u \sim \mathcal{N}(b + \rho(u - a), \sqrt{1 - \rho^2}) \tag{14.33}$$

There exist many versions of this proof online.

Problem 14.4.3. Show that the following changes of variables for a bivariate normal (with $\rho = 0.5$) as described in this Chapter result in uncorrelated components,

$$C = c - u \tag{14.34}$$

$$U = c + u \tag{14.35}$$

Examining first C, we have that its mean,

$$\mathbb{E}(C) = \mathbb{E}(c) - \mathbb{E}(u) \tag{14.36}$$

$$= b - a. \tag{14.37}$$

And its variance,

$$\text{var}(C) = \text{var}(c) + \text{var}(u) - 2\text{covar}(c, u) \tag{14.38}$$

$$= 2 - 2\rho \tag{14.39}$$

$$= 2(1 - \rho) \tag{14.40}$$

Examining the covariance between C and U,

$$\begin{aligned} \text{covar}(C, U) &= \text{covar}(c - u, c + u) && (14.41) \\ &= 0. && (14.42) \end{aligned}$$

As desired, this means that the bivariate normal for the transformed coordinates is of the form,

$$\begin{pmatrix} C \\ U \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} b - a \\ a + b \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right]$$

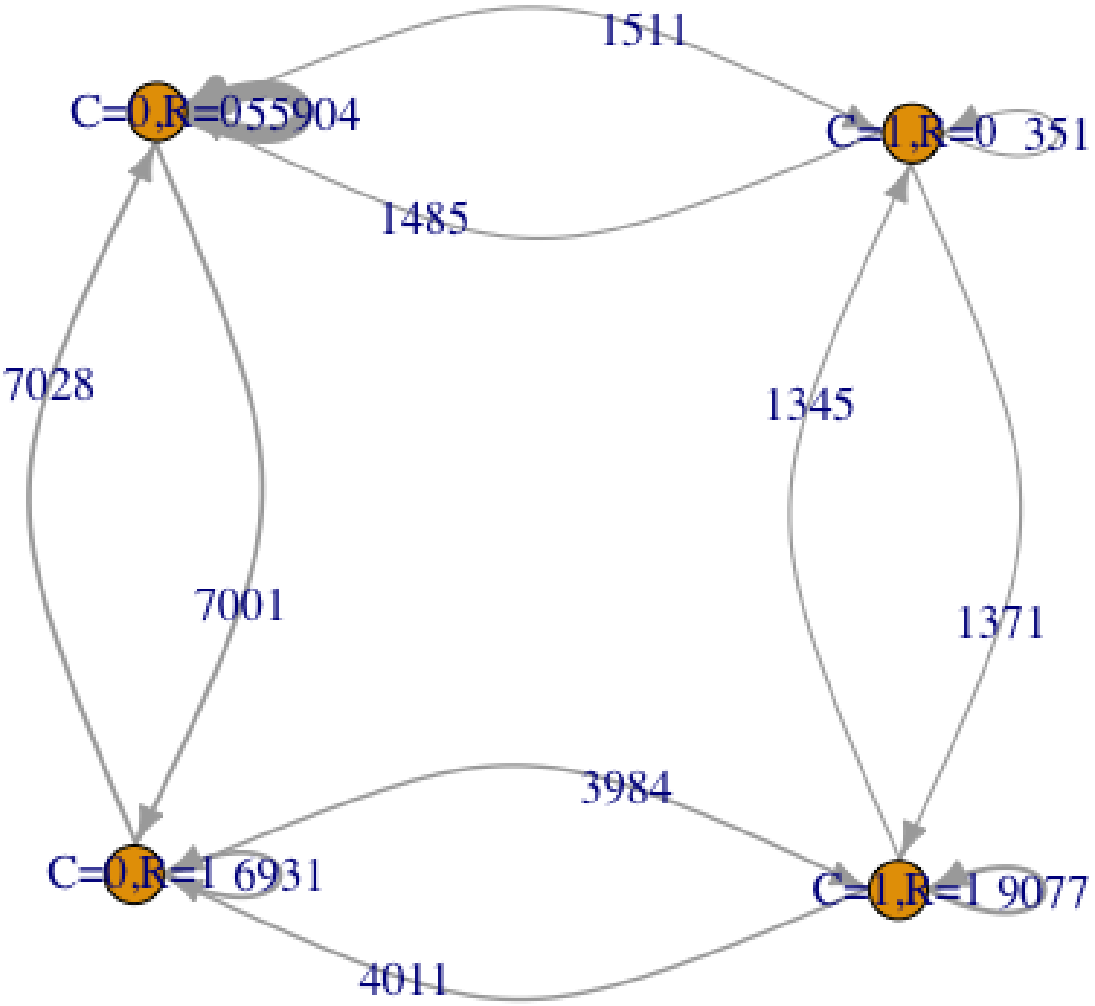

Figure 14.5: A graph representing the transitions between the four states for the Gibbs sampler of the cloud and rain example.

Chapter 15

Hamiltonian Monte Carlo

15.1 Cerebral malaria: coding up samplers

Suppose you work for the WHO researching malaria. In particular, it is your job to produce a model for the number of cases of cerebral malaria in a large country. Cerebral malaria is one of the most severe complications resulting from infection with Plasmodium falciparum malaria, and without treatment invariably causes death. However, even for patients receiving treatment there is still a significant chance of permanent cognitive impairment.

You decide to model the number of cases of cerebral malaria ($X = 5$) as being from a joint normal distribution along with the number of all malaria cases ($Y = 20$). The mean number of cases of cerebral malaria is μ_c, and the mean cases of all malaria is μ_t. If we assume an (improper) uniform prior distribution on these quantities and assume that the correlation between cerebral and total cases is known ($\rho = 0.8$) the posterior is:

$$\begin{pmatrix} \mu_t \\ \mu_c \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 20 \\ 5 \end{pmatrix}, \quad \begin{pmatrix} 2 & 0.8 \\ 0.8 & 0.5 \end{pmatrix} \right]$$

where all quantities are measured in units of "000s".

Note that this example does not test your ability to do Bayesian inference (because we have already provided the exact form of the posterior distribution). Rather its purpose is allow you to compare the performance of a number of different sampling algorithms.

Problem 15.1.1. Use your statistical software of choice generate 100 independent samples of (μ_t, μ_c). Draw a scatter plot of your (μ_t, μ_c) samples, with lines connecting consecutive points. How close are the sample-estimated means to the true means? (Hint: to do this in R you will need to use the MASS package:

```
library(MASS)
```

```
Sigma <- matrix(c(2, 0.8, 0.8, 0.5), 2, 2)
mvrnorm(n=100, c(20, 5), Sigma)
```

)

The independent sampler is the gold standard sampling routine here. Its samples quickly traverse the posterior space (Figure 15.2), meaning that we get an accurate (and unbiased) estimate of the posterior mean for only 100 samples (Figure 15.1).

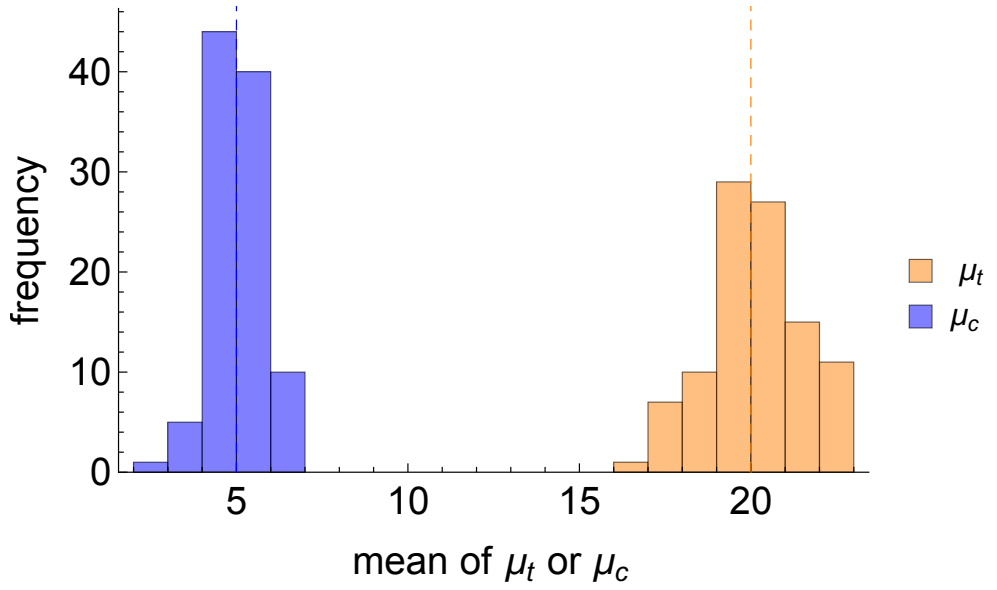

Figure 15.1: The sampling distribution for 100 samples from the posterior using the independent sampler.

Problem 15.1.2. Code up a Random Walk Metropolis sampler for this example. This is composed of the following steps:

1. Create a proposal function that takes a current value of $\theta = (\mu_t, \mu_c)$ and outputs a proposed value of these using a multivariate normal centred on the current estimates. (Here use a multivariate normal proposal with an identity covariance matrix.)

2. Create a function which takes as inputs $\theta^{current}$ and $\theta^{proposed}$, and outputs the ratio of the posteriors of the proposed value to the current one (Hint: to do this in R you will need to use the following to calculate the value of the posterior at (x, y):

```
library(mvtnorm)

Sigma <- matrix(c(2, 0.8, 0.8, 0.5), 2, 2)
dmvnorm(c(x, y), c(20, 5), Sigma)
```

).

3. Create an accept/reject function which takes as inputs $\theta^{current}$ and $\theta^{proposed}$, then uses the above ratio function to find: $r = \frac{\theta^{proposed}}{\theta^{current}}$; then compares r with a uniformly-distributed random number u between 0 and 1. If $r > u \implies$ output $\theta^{proposed}$; otherwise output $\theta^{current}$.

4. Combine the proposal function along with the accept/reject function to make a function that takes as input $\theta^{current}$, proposes a new value of θ, then based on r moves to that new point or stays in the current position.

5. Create a function called "RWMetropolis" that takes a starting value of θ and runs for n steps.

Use your "RWMetropolis" function to generate 100 samples from the posterior starting from $(\mu_t, \mu_c) = (10, 5)$. Draw a line plot of your (μ_t, μ_c) samples. How do your estimates of the posterior mean from Random Walk Metropolis compare with the true values? Why is there a bias in your estimates, and how could this be corrected?

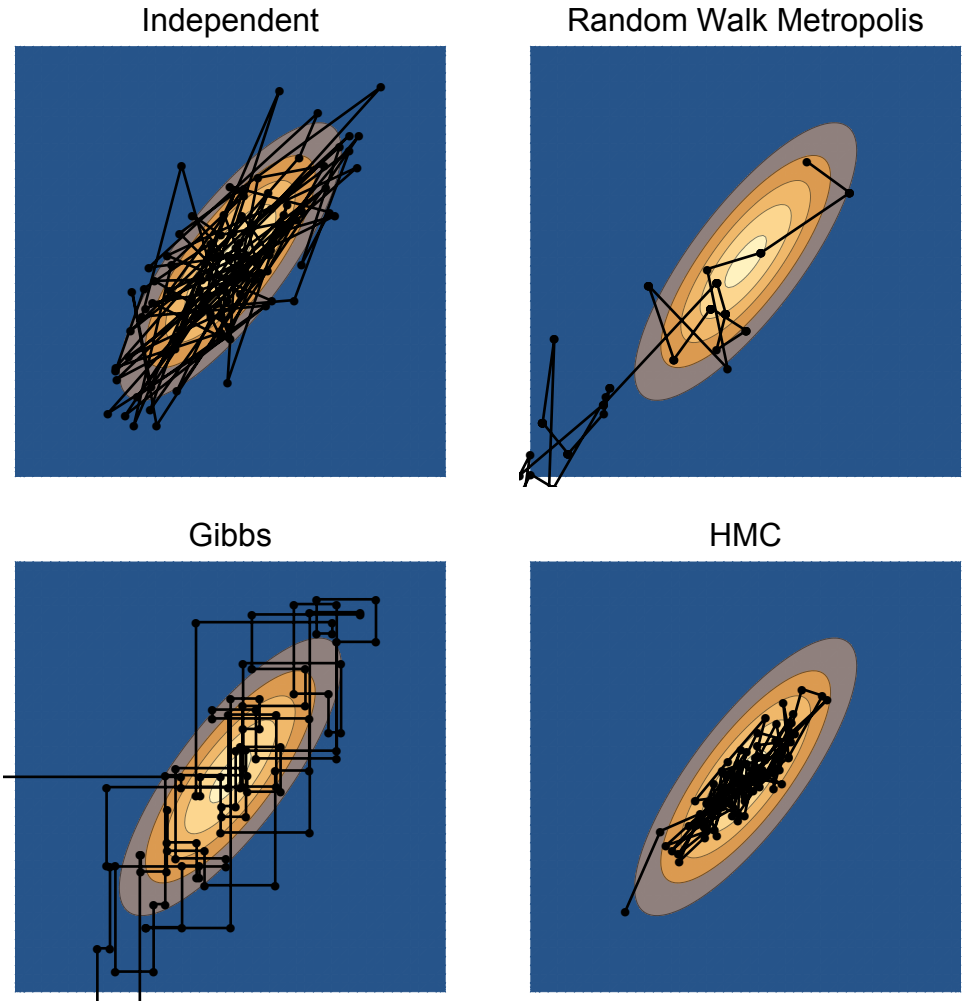

Figure 15.2: 100 samples from the posterior for the malaria example using four different sampling algorithms.

The Random Walk Metropolis sampler is significantly less efficient at exploring posterior space (Figure 15.2) than the independent sampler. This is because of its random walk nature; it essentially

ignores the posterior geometry! It is also because dependent sampling, in general, will never match the performance of an independent sampler.

The estimates of the mean of μ_t should be downwardly-biased (Figure 15.3) because we started the sampler at $\mu_t = 10 < 20$. What we should do is remove the first half (or so) iterations of the chain, where it has yet to converge to a stationary distribution.

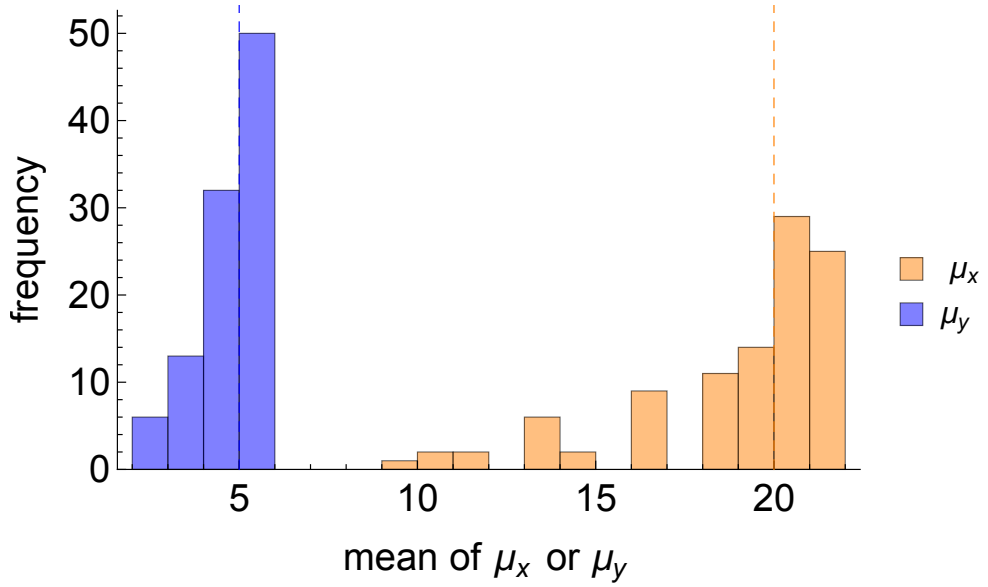

Figure 15.3: The sampling distribution for 100 samples from the posterior using Random Walk Metropolis sampler.

Problem 15.1.3. For your 100 samples using Random Walk Metropolis calculate the percentage of accepted steps.

This is about 30% (Figure 15.4). This relatively low acceptance rate (although not far from optimality for RWM) means that the sampler is slow to explore the posterior.

Problem 15.1.4. Create a function that calculates Gelman's $\hat{R}$ for each of (μ_t, μ_c) using:

$$\hat{R}(t) = \sqrt{\frac{W(t) + \frac{1}{T}(B(t) - W(t))}{W(t)}} \tag{15.1}$$

where,

$$W(t) = \frac{1}{m}\sum_{j=1}^{m} s(t)_j^2 \tag{15.2}$$

measures the within-chain variance at time t averaged over m chains, and $s(t)_j^2$ is the sample variance of chain j. And:

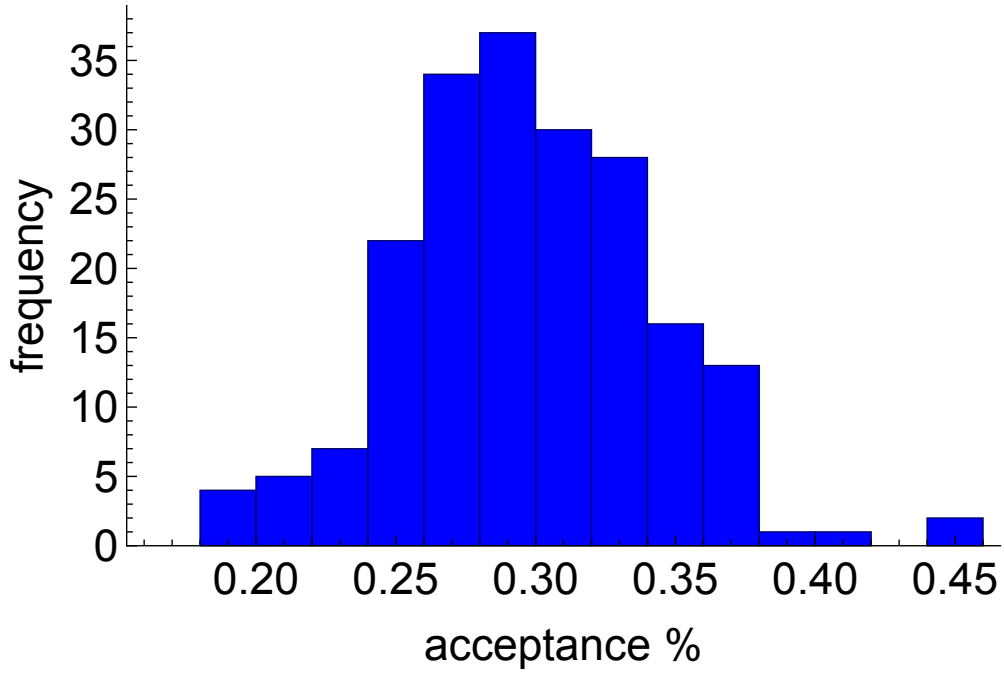

Figure 15.4: The acceptance rate across 200 Random Walk Metropolis Markov Chains, in each case using 100 steps.

$$B(t) = \frac{t}{m-1}\sum_{j=1}^{m}(\overline{\theta(t)}_j - \overline{\theta(t)})^2 \tag{15.3}$$

measures the between-chain variance at time t. Here $\overline{\theta(t)}_j$ is the average value of a parameter in chain j, and $\overline{\theta(t)}$ is the average value of a parameter across all chains. (Hint 1: first create two separate functions that calculate the within and between chain variance. Hint 2: you will obtain a value of $\hat{R}$ for each of (μ_t, μ_c).)

To do this first create a function in R that calculates the within chain variance,

```
fWithin <- function(lSamples){
  return(mean(sapply(lSamples, var)))
}

## Testing it
lSamples <- lapply(seq(1, 10, 1), function(i) rbinom(100, 100, 0.5))
fWithin(lSamples)
```

Then another that calculates the between chain variance,

```
fBetween <- function(lSamples){
  lMean <- sapply(lSamples, mean)
  aMean <- mean(lMean)
```

```
  m <- length(lSamples)
  t <- length(lSamples[[1]])
  return((t / (m - 1)) * sum((lMean - aMean) ^ 2))
}
```

Then putting these together we get a function to calculate $\hat{R}$,

```
fRhat <- function(lSamples){
  W <- fWithin(lSamples)
  B <- fBetween(lSamples)
  t <- length(lSamples[[1]])
  return(sqrt((W + (1 / t) * (B - W)) / W))
}
```

Problem 15.1.5. Start all eight chains at $(\mu_t, \mu_c) = (20, 5)$ and calculate $\hat{R}$ for a per chain sample size of 5. Does this mean we have reached convergence?

If the chains all begin in the same area of parameter space $\implies$ we get a false impression of convergence. This is why it's so important to start them at over-dispersed locations.

Problem 15.1.6. Using eight chains calculate $\hat{R}$ for each of (μ_t, μ_c) for a sample size of 100. This time make sure to start your chains in overdispersed positions in parameter space. Use a random number from a multivariate normal centred on the posterior means with a covariance matrix of 40 times the identity matrix.

The chains should be starting at dispersed locations in parameter space (Figure 15.5), otherwise the value of $\hat{R}$ obtained will be biased downwards. After a sample size of 100 the chains should be nearing convergence, and should have a value of $\hat{R}$ of about 1.05-1.06 (Figure 15.6).

Problem 15.1.7. After approximately how many iterations does Random Walk Metropolis reach $\hat{R} < 1.1$?

After about 150-250 iterations the chains should have roughly reached $\hat{R} < 1.1$ (Figure 15.6).

Problem 15.1.8. The conditional distributions of each variable are given by:

$$\mu_t \sim \mathcal{N}(20 + 1.6(\mu_c - 5), (1 - 0.8^2)2)$$
$$\mu_c \sim \mathcal{N}(5 + 0.4(\mu_t - 20), (1 - 0.8^2)0.5)$$

Use this information to code up a Gibbs sampler, again starting at $(\mu_t, \mu_c) = (10, 5)$. (Hint: in R use `rnorm`, or equivalent to create two functions: one that produces draws of μ_t given μ_c; and the other that produces draws of μ_c given μ_t. Then create a function that cycles between these updates. Make sure to always draw samples using the most recent values of (μ_t, μ_c)).

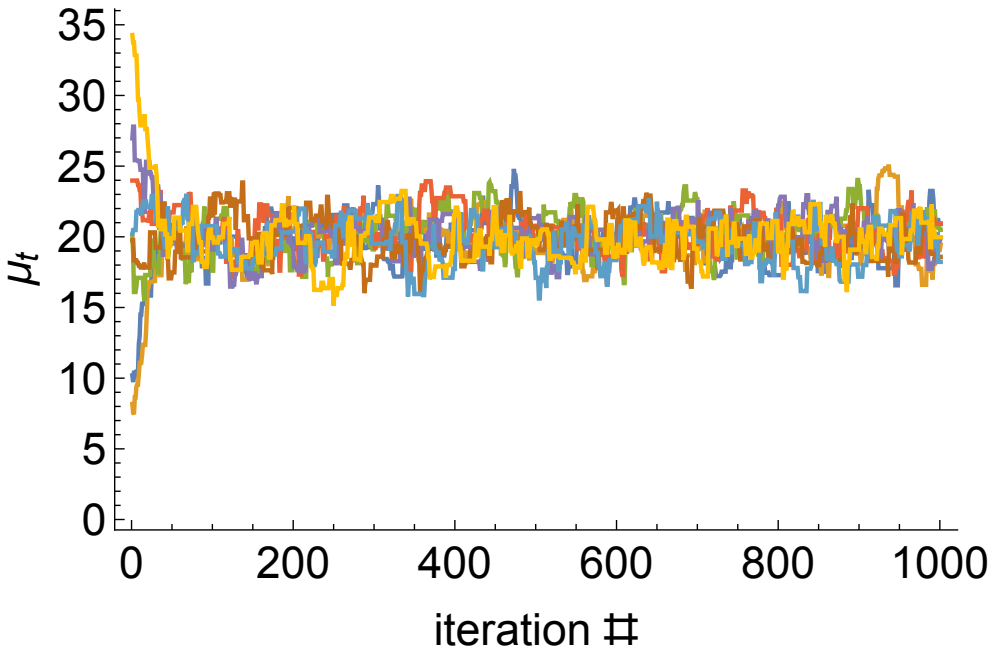

Figure 15.5: The path taken by each chain in μ_t space.

Problem 15.1.9. Use your Gibbs sampler to draw 100 samples. Draw a scatter plot of your (μ_t, μ_c) samples with lines connecting consecutive points. Discarding the first 50 observations, how do the estimates of the mean of each parameter compare with their true values?

After a short warm-up period the samples from the Gibbs algorithm quickly converge towards the posterior (Figure 15.2), with the resultant estimates of the mean reflecting this (Figure 15.7).

Problem 15.1.10. Generate 200 samples from each of your Random Walk Metropolis and Gibbs samplers. Discard the first 100 observations of each as warm-up. For each calculate the error in estimating the posterior mean of μ_t. Repeat this exercise 40 times; each time recording the error. How does their error compare to the independent sampler?

Each of the three algorithms is essentially unbiased after we discard the warm-up (Figure 15.8). I obtained an error of about 0.12 for the independent sampler; 0.51 for Random Walk Metropolis; and 0.40 for Gibbs.

Problem 15.1.11. Repeat Problem 15.10 to obtain the average error in estimating the posterior mean of μ_t across a range of sample sizes $n = 5$ to $n = 200$.

The error from Random Walk Metropolis and Gibbs is always higher than the independent sampler (Figure 15.9). After a sample size of about 20, the Gibbs outperforms Random Walk Metropolis.

Problem 15.1.12. Using the results from the previous question estimate the effective sample size for 150 observations of the Random Walk Metropolis and Gibbs samplers.

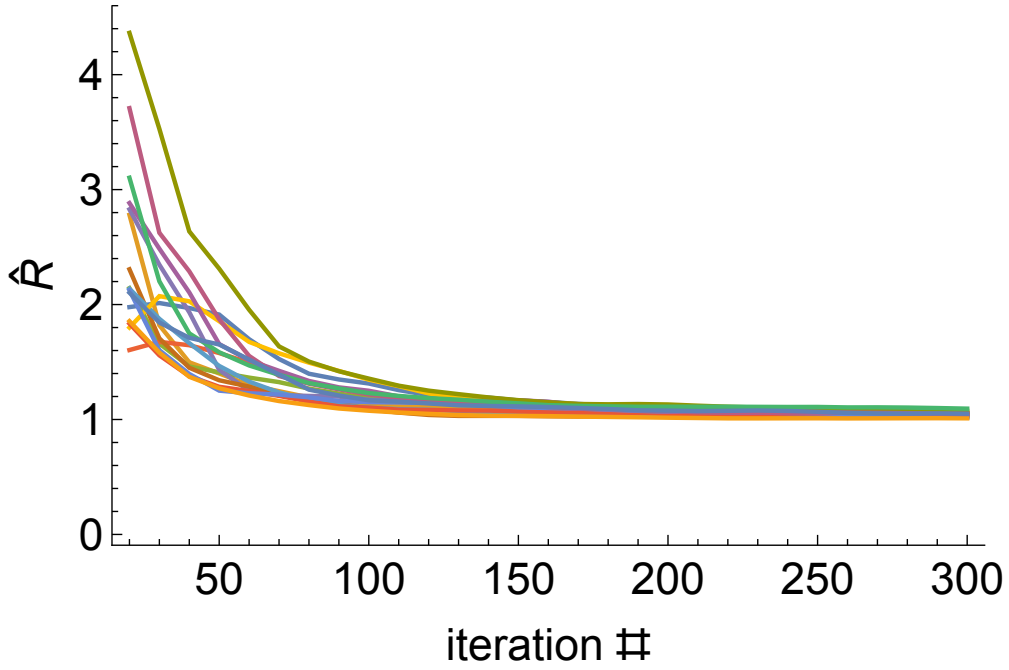

Figure 15.6: The values of $\hat{R}$ for μ_t across 16 different replicates, each with 8 chains running in parallel.

These estimates will be somewhat noisy, but I obtained an equivalent sample size of 11 for the Gibbs and 7 for RWM (Figure 15.9).

Problem 15.1.13. What do the above results tell you about the relative efficiency of each of the three samplers?

The Gibbs is more efficient than RWM. Although both are quite inefficient compared to the independent sampler; each with an effective sample size far less than 10% actual sample size.

Problem 15.1.14. Code up a Hamiltonian Monte Carlo sampler for this problem. (Alternatively, use the functions provided in the R file "HMC_scripts.R" adapted from [9]). Use a standard deviation of the momentum proposal distribution (normal) of 0.18, along with a step size $\epsilon = 0.18$ and $L = 10$ individual steps per iteration to simulate 100 samples from the posterior. How does the estimate of the mean compare with that from the Independent, Random Walk Metropolis and Gibbs samplers?

The performance of HMC is comparable to that of the independent sampler (Figure 15.9), and hence considerably more efficient at estimating the posterior mean.

Problem 15.1.15. What is the acceptance rate for HMC? How does this compare with RWM?

Based on 1,000 samples I obtain an acceptance rate of above 99% for HMC.

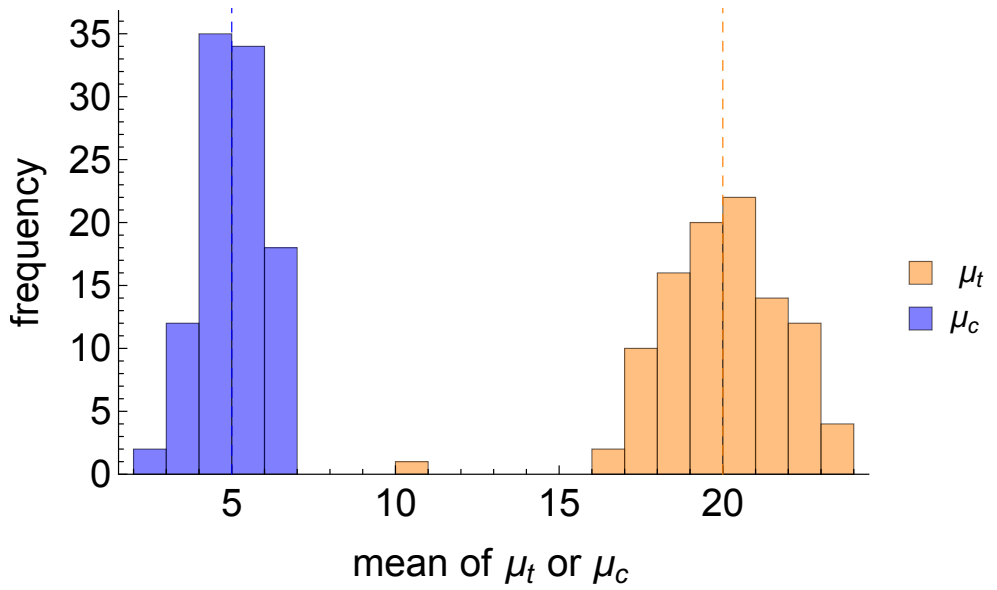

Figure 15.7: The sampling distribution for 100 samples from the posterior using a Gibbs sampler.

Problem 15.1.16. Gibbs sampling has an acceptance rate of 100%. How can HMC be more efficient than Gibbs given that its acceptance rate is less than 100%?

The Gibbs sampler moves in steps that are either vertical or horizontal (Figure 15.2). This is an inefficient way to explore the posterior which has a diagonal orientation. HMC tends to move in the diagonal with an acceptance rate comparable to that of Gibbs. By taking account of the posterior geometry it is much more efficient at exploring the typical set.

Problem 15.1.17. You receive new data that results in a change in the posterior to:

$$\begin{pmatrix}\mu_t\\ \mu_c\end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix}20\\ 5\end{pmatrix}, \quad \begin{pmatrix}2 & 0.99\\ 0.99 & 0.5\end{pmatrix}\right]$$

Using your Random Walk Metropolis sampler calculate $\hat{R}$ for 8 chains; each generating 100 samples for each.

With the higher correlation between the parameters the rate of convergence is slower. Intuitively this is because the model is poorly identified; it is impossible to disentangle one parameter's effect from another's. This causes problems with statistical inference, and hence with the sampling. This is an example of Gelman's "Folk Theorem" which states that any problem with MCMC is generally a problem with the underlying model.

In this case we can still get convergence although it occurs at a much slower rate than before (Figure 15.10), meaning that a sample size of about 500 is required.

Problem 15.1.18. Estimate the value of $\hat{R}$ for HMC on the posterior from the new data, for a sample size of 100. How does it compare to Random Walk Metropolis?

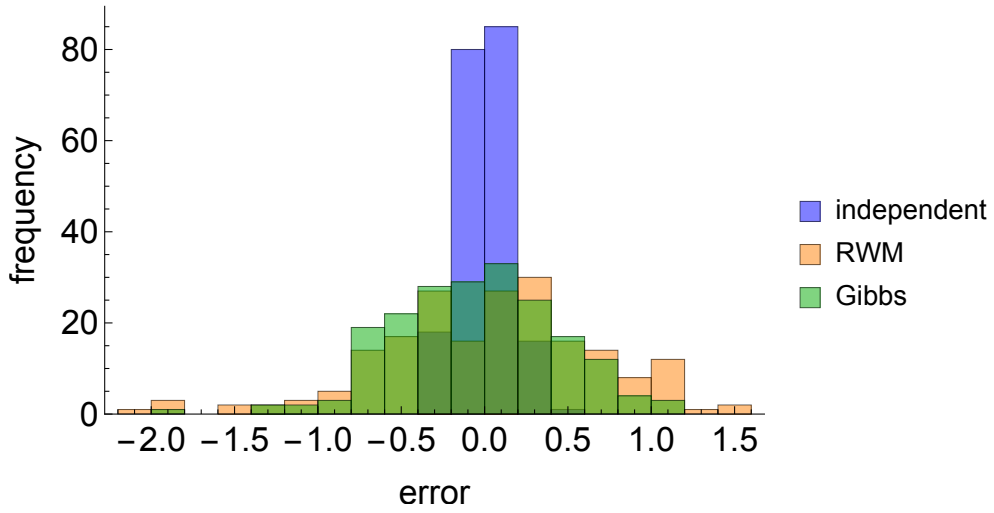

Figure 15.8: The sampling distribution in estimating the posterior mean of μ_t for three different sampling algorithms.

The rate of convergence will be significantly faster for HMC, and so we expect a value of $\hat{R}$ that is less than that for Random Walk Metropolis.

15.2 HMC and U-Turns

The code in `HMC_UTurn.R` uses simulates Hamiltonian dynamics for a single particle on the distribution described in the previous question:

$$\begin{pmatrix}\mu_t \\ \mu_c\end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix}20 \\ 5\end{pmatrix}, \quad \begin{pmatrix}2 & 0.8 \\ 0.8 & 0.5\end{pmatrix}\right]$$

In this question we will see how the efficiency of HMC depends on choice of the number of intermediate steps. In particular we investigate the propensity of a particle undergoing Newtonian dynamics to perform U-Turns.

Problem 15.2.1. Simulate a single particle starting at $(20, 5)$ for $L = 10$ steps with the following parameters $\epsilon = 0.18$ (step size), $\sigma = 0.18$ (momentum proposal distribution width). Plot the path in parameter space.

The particle seems to move without turning round (Figure 15.11).

Problem 15.2.2. Now try $L = 20, 50, 100$ steps, again plotting the results what do you notice about the paths?

As the number of steps taken increases the particle becomes more predisposed to U-turns (Figure 15.11).

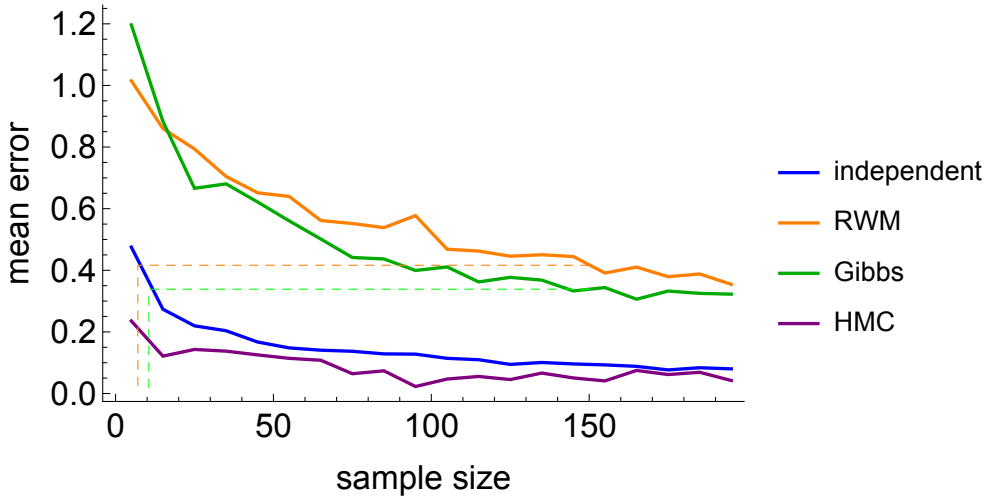

Figure 15.9: The mean error in estimating the posterior mean of μ_t for four different sampling algorithms. The dotted lines indicate the effective sample sizes for an actual sample size of 100 for the Gibbs and Random Walk Metropolis algorithms.

Problem 15.2.3. Simulate 100 iterations of the particle starting at $(20, 5)$, with each particle running for $L = 100$ steps. Examine the motion of the particle in one of the parameter dimensions, and hence determine an optimal number of steps for this distribution.

This can be done with the following R code,

```
nReplicates <- 100
nStep <- 100
mAll <- matrix(ncol=nReplicates, nrow=nStep)
for(i in 1:nReplicates){
  lTest <- HMC_keep(c(20, 5), U, grad_U, 0.18, nStep, 0.18)
  lTemp <- lTest$pos[, 1]
  aLen <- length(lTemp)
  mAll[, i] <- lTemp[1:(aLen - 1)]
}

library(reshape2)
mAll <- melt(mAll)
library(ggplot2)
ggplot(mAll, aes(x=Var1, colour=as.factor(Var2), y=value)) + geom_path() +
  theme(legend.position="none") +
  ylab('mu_t') + xlab('number of steps')
```

which should produce a plot qualitatively similar to Figure 15.12. From this we can see that we explore the most distance in this parameter dimension for about $L = 13 - 14$. Any longer and the particle turns round on its self.

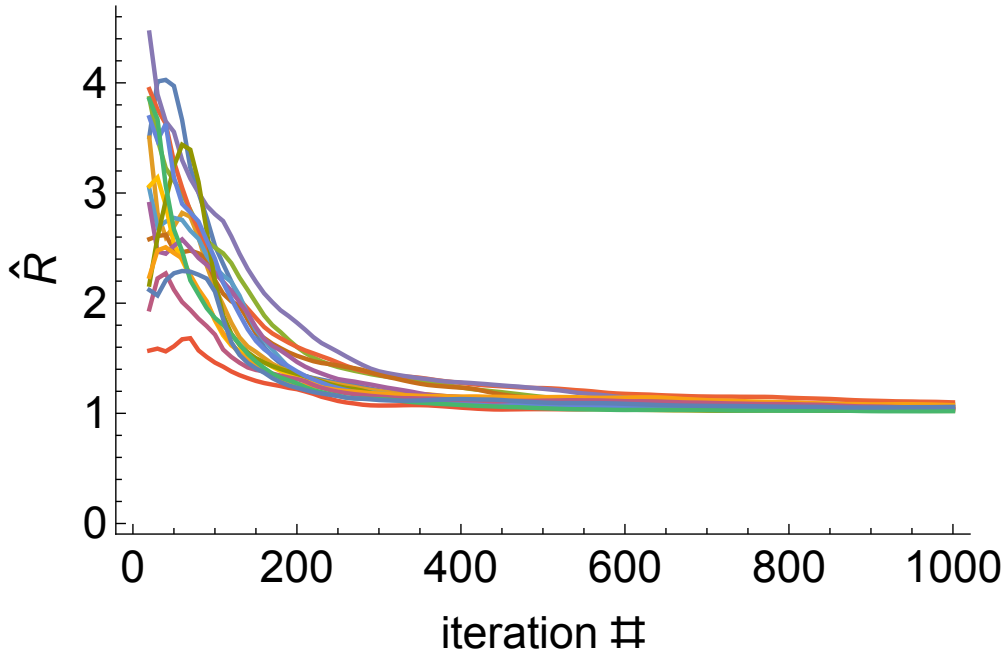

Figure 15.10: The values of $\hat{R}$ for 16 different replicates, each with 8 chains running in parallel for the case where $\rho = 0.99$.

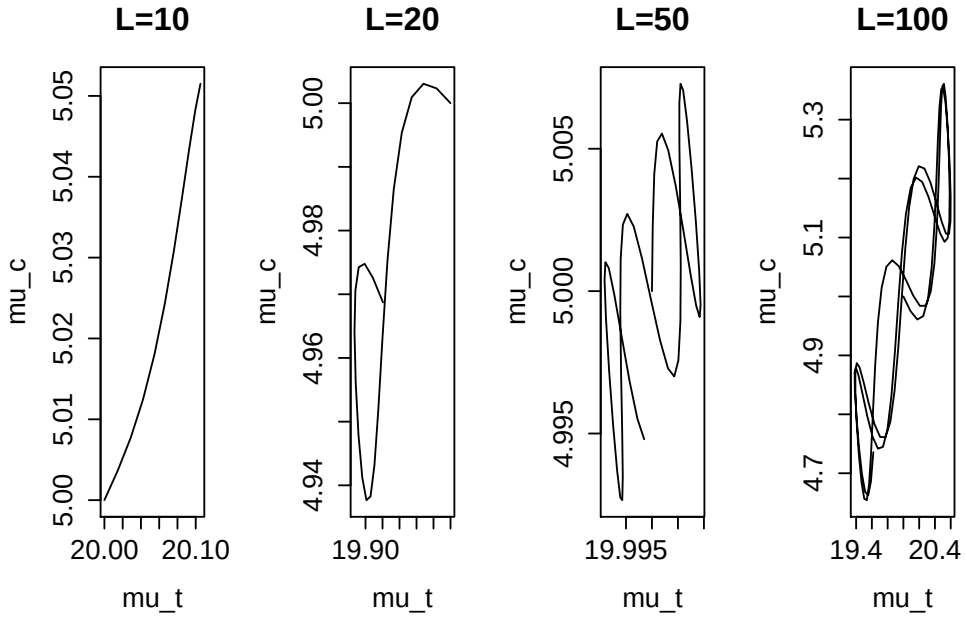

Figure 15.11: The path of a particle in parameter space for differing number of steps, L.

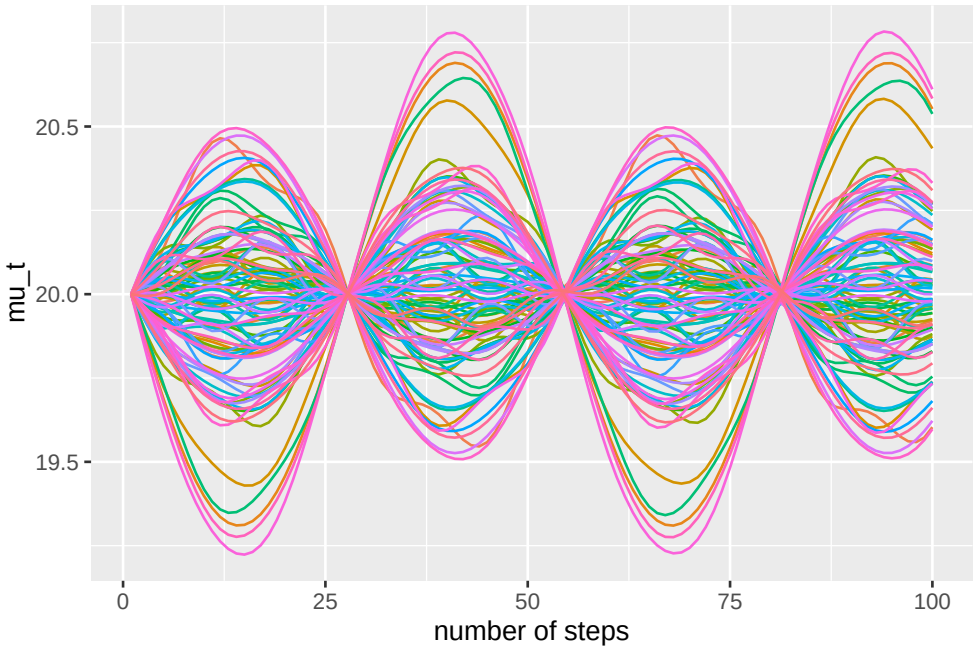

Figure 15.12: The path of 100 particle replicates over in the μ_t dimension.

Chapter 16

Stan

16.1 Discoveries data revisited

The file `evaluation_discoveries.csv` contains data on the numbers of "great" inventions and scientific discoveries (X_t) in each year from 1860 to 1959 [1]. In this question you will develop a model to explain the variation in scientific inventions over time. The simplest model here is to assume that (a.) one discovery is independent of all others, and (b.) the rate of occurrence of discoveries is the same in all years (λ). Since the data is discrete, these assumptions suggest the use a Poisson likelihood,

$$X_t \sim Poisson(\lambda) \tag{16.1}$$

Problem 16.1.1. Open a text editor and create a file called "discoveries.stan" in your working directory. In the file create three parameter blocks:

```
data {

}
parameters {

}
model {

}
```

Problem 16.1.2. Fill in the data and parameter blocks for the above model.

The completed model can be written as:

```
data {
  int N; // number of observations
  int<lower=0> X[N]; // vector of discoveries
}
```

```
parameters {
  real<lower=0> lambda;
}

model {
  X ~ poisson(lambda); // likelihood
  lambda ~ lognormal(2, 1); // prior
}
```

Problem 16.1.3. Using a $log-\mathcal{N}(2,1)$ prior for λ code up the `model` block; making sure to save your file afterwards.

Problem 16.1.4. Open your statistical software (R, Python, Matlab, etc.) and load any packages necessary to use Stan. (Hint: in R this is done by using "library(rstan)'; in Python this is done using "import pystan".)

Problem 16.1.5. Load the data into your software then put it into a structure that can be passed to Stan. (Hint: in R create a list of the data; in Python create a dictionary where the 'key' for each variable is the desired variable name.)

```
aDF <- read.csv('evaluation_discoveries.csv')
X <- aDF[, 2]
N <- length(X)
dataList <- list(N=N, X=X)
```

Problem 16.1.6. Run your model using Stan, with 4 chains, each with a sample size of 1000, and a warm-up of 500 samples. Set seed=1 to allow for reproducibility of your results. Store your result in an object called "fit".

```
fit <- stan(file='discoveries.stan', data=dataList, iter=1000, chains=4, seed=1)
```

Problem 16.1.7. Diagnose whether your model has converged by printing "'fit".

Lambda and lp should both have a value of $\hat{R} \approx 1$.

Problem 16.1.8. For your sample what is the equivalent number of samples for an independent sampler?

This is just the value of "n_eff" which I get to be around 600-900.

Problem 16.1.9. Find the central posterior 80% credible interval for λ.

This can be done in R by the following:

```
print(fit, pars='lambda', probs = c(0.1, 0.9))
```

Alternatively, you can extract lambda from the fit object then find its quantiles:

```
lLambda <- extract(fit, 'lambda')[[1]]
c(quantile(lLambda, 0.1), quantile(lLambda, 0.9))
```

Or in Python by doing:

```
import numpy as np
lambda_samples = fit.extract('lambda')['lambda']
np.percentile(lambda_samples, (10, 90))
```

In any case I get an interval of approximately $2.9 \leq \lambda \leq 3.3$.

Problem 16.1.10. Draw a histogram of your posterior samples for λ.

To do this in R, first extract the parameter, then draw the plot.

```
library(ggplot2)

lLambda <- extract(fit, 'lambda')[[1]]
qplot(lLambda)
```

Or in python:

```
import numpy as np
import matplotlib.pyplot as plt
lambda_samples = fit.extract('lambda')['lambda']
np.percentile(lambda_samples, (10, 90))
plt.hist(lambda_samples)
plt.xlabel("lambda")
plt.ylabel("frequency")
```

Problem 16.1.11. Load the `evaluation_discoveries.csv` data and graph the data. What does this suggest about our model's assumptions?

Both a time series and histogram are useful here (see Figure 16.1). To me the left hand plot suggests that there is some temporal autocorrelation in the data (perhaps invalidating an assumption of independence, and/or identical distribution). The histogram would seem to support this claim, since the variance is fairly obviously greater than the mean. I also plot an autocorrelogram of the data which suggests that there is autocorrelation in the series.

Problem 16.1.12. Create a `generated quantities` block in your Stan file, and use it to sample from the posterior predictive distribution. Then carry out appropriate posterior predictive checks to evaluate your model. (Hint: use the function `poisson_rng` to generate independent samples from your lambda).

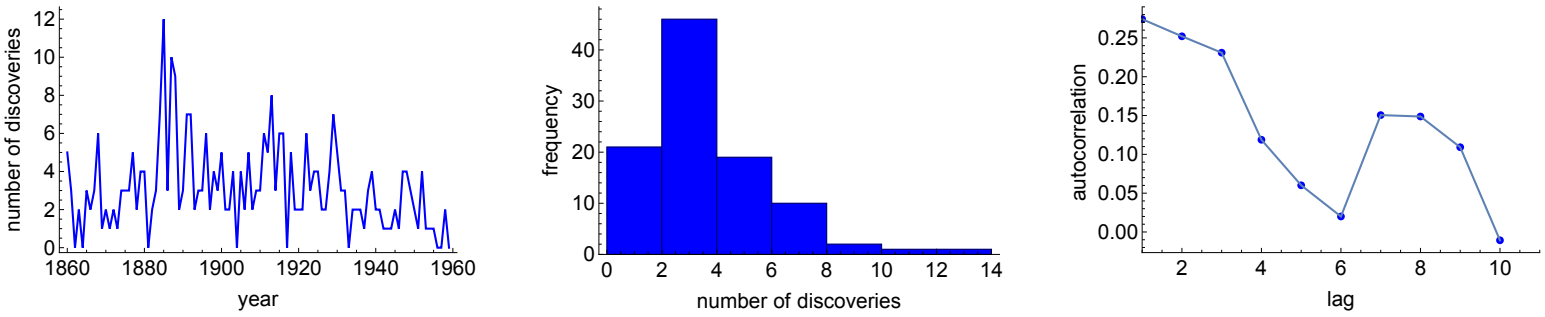

Figure 16.1: Characteristics of the "discoveries" data set.

```
generated quantities{
  int<lower=0> XSim[N];
  for (i in 1:N)
    XSim[i] <- poisson_rng(lambda);
}
```

One simple check is to compare the maximum of your posterior predictive simulations with that of the real data (which is 12 discoveries in 1885.) To do this in R do something like the following:

```
lXSim <- extract(fit, 'XSim')[[1]]
lMax <- apply(lXSim, 1, max)

library(ggplot2)
qplot(lMax)

sum(lMax >= 12) / length(lMax)
```

Or in Python by:

```
posterior_checks = fit.extract('XSim')['XSim']
posterior_checks_max = np.amax(posterior_checks, axis=1)
(posterior_checks_max >= 12).sum() / float(len(posterior_checks_max))
```

From the resultant histogram it is evident that our model would only very rarely produce 12 discoveries. I get 12+ discoveries in only about 1% of simulations.

Problem 16.1.13. A more robust sampling distribution is a negative binomial model:

$$X_i \sim NB(\mu,\ \kappa) \tag{16.2}$$

where μ is the mean number of discoveries per year, and $var(X) = \mu + \frac{\mu^2}{\kappa}$. Here κ measures the degree of over-dispersion of your model; specifically if $\kappa \uparrow$ then over-dispersion$\downarrow$.

Write a new Stan file called "discoveries_negbin.stan" that uses this new sampling model (Hint: use the Stan manual section on discrete distributions to search for the correct negative binomial

function name; be careful there are two different parameterisations of this function available in Stan!) Assume that we are using the following priors:

$$\mu \sim log - \mathcal{N}(2, 1) \tag{16.3}$$
$$\kappa \sim log - \mathcal{N}(2, 1) \tag{16.4}$$
$$\tag{16.5}$$

Draw 1000 samples across 4 chains for your new model. Has it converged to the posterior?

The full code should be something like:

```
data {
  int N;
  int<lower=0> X[N];
}

parameters {
  real<lower=0> mu;
  real<lower=0> kappa;
}

model {
  X ~ neg_binomial_2(mu, kappa);
  mu ~ lognormal(2, 1);
  kappa ~ lognormal(2, 1);
}
```

After 1000 samples across 4 chains I get a value of $\hat{R} \approx 1$, and an effective sample size around 1000.

Problem 16.1.14. Carry out posterior predictive checks on the new model. What do you conclude about the use of a negative binomial here versus the simpler Poisson?

The Stan code to generate the posterior predictive distribution samples is:

```
generated quantities{
  int<lower=0> XSim[N];
  for (i in 1:N)
    XSim[i] <- neg_binomial_2_rng(mu, kappa);
}
```

Now I obtain about 20% of posterior predictive samples that have a maximum value greater than or equal to 12 (that of the real data). This is much more reasonable than that from the Poisson model.

Problem 16.1.15. Find the central posterior 80% credible interval for the mean rate of discoveries μ from the negative binomial model. How does it compare with your results from the Poisson model? Why is this the case?

I get a similar answer to before, around: $2.8 \leq \lambda \leq 3.4$; it should be slightly wider because there is more sampling variability $\implies$ greater uncertainty over the parameter's value.

Problem 16.1.16. Calculate the autocorrelation in the residuals between the actual and simulated data series. What do these suggest about our current model?

If you look at the autocorrelation you see that there is a slight, yet persistent, positive autocorrelation (looking at the first lag). You can obtain these in R by the following:

```
lXSim <- extract(fit, 'XSim')[[1]]
mResid <- sweep(lXSim, 2, X)
lCorr <- sapply(seq(1, 200, 1),
                function(i) acf(mResid[i, ], lag.max=1)$acf[[2]])
qplot(lCorr)
```

Problem 16.1.17. Following on from the above suggest an alternative model formulation.

Clearly there is some persistence in the rate of discoveries over time. In other words, discoveries tend to clump together. This is clear from a simple time series plot of the data. One idea would be to allow an AR1 process for the mean of the process:

$$\mu(t) = \rho\mu(t-1) + (1-\rho)\bar{\mu} + \epsilon(t) \tag{16.6}$$

This allows for persistence in the rate over time and might remedy some of the issues. However, implementing it will be a bit tricky since $\mu(t)$ must always be positive. Perhaps the best thing to do here is to use a transformed parameter $exp(\mu)$ as the mean of the distribution. This will prevent it from being non-negative.

16.2 Hungover holiday regressions

The data in file `stan_hangover.csv` contains a series of Google Trends estimates of the search traffic volume for the term "hangover cure" in the UK between February 2012 to January 2016. The idea behind this problem is to determine how much more hungover are people in the "holiday season" period, defined here as the period between 10th December and 7th January, than the average for the rest of the year.

Problem 16.2.1. Graph the search volume over time, and try to observe the uplift in search volume around the holiday season.

```
plot(as.Date(hangover$date), hangover$volume, type='l', xlab='date',
     ylab='volume')
```

Shows obvious spikes during the holiday season period.

Problem 16.2.2. The variable "holiday" is a type of indicator variable that takes the value 1 if the given week is *all* holiday season, 0 if it contains none of it, and $0 < X < 1$ for a week that contains a fraction X of days that fall in the holiday season. Graph this variable over time so that you understand how it works.

Problem 16.2.3. A simple linear regression is proposed of the form,

$$V_t \sim \mathcal{N}(\beta_0 + \beta_1 h_t,\ \sigma) \tag{16.7}$$

where V_t is the search volume in week t and h_t is the holiday season indicator variable. Interpret β_0 and β_1 and explain how these can be used to estimate the increased percentage of hangovers in the holiday season.

β_0 is the average hangover search volume in weeks that aren't in the holiday season, and β_1 shows the uplift for a week that falls in the holiday season. The percentage increase is hence β_1/β_0.

Problem 16.2.4. Assuming $\beta_i \sim \mathcal{N}(0, 50)$ and $\sigma \sim$ half-$\mathcal{N}(0, 10)$ priors write a Stan model to estimate the percentage increase in hangoverness over the holiday period.

An example Stan program is shown below (not the most efficient since doesn't vectorise, but the below is simplest to understand),

```
data{
  int T;
  real V[T];
  real h[T];
}

parameters{
  real beta0;
  real beta1;
  real<lower=0> sigma;
}

model{
  for(t in 1:T)
    V[t] ~ normal(beta0 + beta1 *h[t], sigma);

  beta0 ~ normal(0, 50);
  beta1 ~ normal(0, 50);
  sigma ~ normal(0, 10);
```

```
}

generated quantities{
  real uplift;
  uplift = beta1 / beta0;
}
```

From this we estimate that with 50% probability, 77% $\leq$ uplift $\leq$ 86%!

16.3 Coding up a bespoke probability density

In the file `stan_survival.csv` there is data for a variable Y that we believe comes from a probability distribution $p(Y) = \frac{\sqrt[3]{b}}{\Gamma(\frac{4}{3})} exp(-bY^3)$ where $b > 0$ is a parameter of interest. In this question we are going to write a Stan program to estimate the parameter b even though this distribution is not amongst Stan's implemented distributions!

Problem 16.3.1. Explain what is meant by the following statement in Stan,

```
theta ~ beta(1, 1);
```

In particular, explain why this is equivalent to the following,

```
target += beta_lpf(theta | 1, 1);
```

where `target` is a Stan variable that stores the overall log-probability, and `+=` increments `target` by an amount corresponding to the RHS.

`~` statements in Stan do not mean sampling! They always mean increment the log probability by something, since HMC works in the (negative) log probability space. In this case it means increment the log probability by an amount corresponding to the probability density of a value "theta" from a beta(1,1) distribution. This is why it is equivalent to the second piece of code where we explicitly increment the log probability.

Note there is a subtle difference between the two which is that `~` statements drop all constant terms from the log probability update, whereas the target statements keep these. However for most purposes this is not important.

Problem 16.3.2. Work out by hand an expression for the log-probability of the density $p(Y) = \frac{\sqrt[3]{b}}{\Gamma(\frac{4}{3})} exp(-bY^3)$.

Just take the log of the expression,

$$log\ p = log\left(\frac{\sqrt[3]{b}}{\Gamma(4/3)}\right) - by^3 \tag{16.8}$$

Problem 16.3.3. Write a Stan function that for a given value of y and b calculates the log probability (ignoring any constant terms). Hint: Stan functions are declared as follows,

```
functions{
  real anExample(real a, real b){
    ...
    return(something);
  }
}
```

where in this example the function takes two reals as inputs and outputs something of type real.

A function to do this is shown below,

```
functions{
  real fCustomProb(real aY, real aB){
    real aConst;
    aConst = (aB ^ (1.0 / 3.0));
    return(log(aConst) - aB * (aY ^ 3));
  }
}
```

Problem 16.3.4. Use your previously created function to write a Stan program that estimates b, and then use it to do so with the y series contained within `stan_survival.csv`. (Hint: Stan functions must be declared at the top of a Stan program.)

An example Stan file is given below,

```
functions{
  real fCustomProb(real aY, real aB){
    real aConst;
    aConst = (aB ^ (1.0 / 3.0));
    return(log(aConst) - aB * (aY ^ 3));
  }
}

data{
  int N;
  real Y[N];
}

parameters{
  real<lower=0> b;
}

model{
  for(i in 1:N)
    target += fCustomProb(Y[i], b);
```

```
}
```

which when run estimates a posterior mean for $b \approx 2.42$.

16.4 Is a tumour benign or malignant?

Suppose that if a tumour is benign the result of a clinical test for the disease for individual i is $X_i \sim \mathcal{B}(20, \theta_b)$, whereas if the tumour is malignant $X_i \sim \mathcal{B}(20, \theta_m)$, where $\theta_b < \theta_m$. Suppose that we collect data on 10 patients' scores on this clinical test $X = \{4, 18, 6, 4, 5, 6, 4, 6, 16, 7\}$ and would like to infer the disease status for each individual, as well as the parameters (θ_b, θ_m).

Problem 16.4.1. Write down in pseudo-code the full model, where we suppose that we use uniform priors on (θ_b, θ_m) and discrete uniform priors on the disease status s_i of individual i.

This looks like,

$$s_i \sim \text{discrete-uniform}(1, 2) \tag{16.9}$$
$$\text{if} \quad (s_i = 1) \tag{16.10}$$
$$X_i \sim \mathcal{B}(10, \theta_b) \tag{16.11}$$
$$\text{else} \tag{16.12}$$
$$X_i \sim \mathcal{B}(10, \theta_m) \tag{16.13}$$

Problem 16.4.2. Assuming that $s_i \in [1, 2]$ is the disease status of each individual (1 corresponding to a benign growth, and 2 to a malignant one), use the `transformed parameters` block to calculate the log probability of each individual's data. (Hint: this will be a 10×2 matrix, where the 2 corresponds to two possible disease statuses for each individual.)

```
transformed parameters{
  matrix[10, 2] lp;
  for(i in 1:10)
    for(s in 1:2)
      lp[i,s] = log(0.5) + binomial_lpmf(X[i] | N, theta[s]);
}
```

Problem 16.4.3. The disease status of each individual $s_i \in [1, 2]$ is a discrete variable, and because Stan does not support discrete parameters directly it is not as straightforward to code up these problems as for continuous parameter problems. The way that to do this is by marginalising out s_i from the joint distribution,

$$p(\theta_b, \theta_m | X) = \sum_{s_1=1}^{2} p(\theta_b, \theta_m, s_1 | X) \tag{16.14}$$

where we have illustrated this for the disease status of individual 1. This then allows us to find an expression for the posterior density which we log to give lp, and then use `target+=lp` to increment the log probability. However, because we do this on the log density scale we instead do the following,

$$\log p(\theta_b, \theta_m|X) = \log \sum_{s_1=1}^{2} p(\theta_b, \theta_m, s_1|X) \tag{16.15}$$

$$= \log \sum_{s_1=1}^{2} exp\left(log\ p(\theta_b, \theta_m, s_1|X)\right) \tag{16.16}$$

$$= \text{log\_sum\_exp}_{s_1=1}^{2}(\log p(\theta_b, \theta_m, s_1|X)) \tag{16.17}$$

where `log_sum_exp(.)` (a function available in Stan) is defined as,

$$\text{log\_sum\_exp}_{i=1}^{K}\alpha = log \sum_{i=1}^{K} exp(\alpha) \tag{16.18}$$

and is a more numerically-stable way of doing the above calculation. Using this knowledge, write a full Stan model that implements this marginalisation, and use it to estimate θ_b and θ_m. (Hint: use the `binomial_logit_lpmf(X[i]|N,alpha[s])` function in Stan and define `ordered[2] alpha`, then transform from the unconstrained alpha to theta using `inv_logit`.)

The below code is one such implementation of this model,

```
data {
  int<lower=1> nStudy; // number studies
  int<lower=1> N; // samples per study
  int<lower=0, upper=N> X[nStudy]; // number successes
}

parameters {
  ordered[2] alpha;
}

transformed parameters{
  real<lower=0,upper=1> theta[2];
  matrix[nStudy, 2] lp;
  for(i in 1:2)
    theta[i] = inv_logit(alpha[i]);

for(n in 1:nStudy)
  for(s in 1:2)
    lp[n,s] = log(0.5) + binomial_logit_lpmf(X[n] | N, alpha[s]);
}

model {
```

```
  for(n in 1:nStudy)
    target += log_sum_exp(lp[n]);
}
```

This should yield $\theta_b \approx 0.2 - 0.3$ and $\theta_m \approx 0.8 - 0.9$.

Problem 16.4.4. We use the `generated quantities` block to estimate the probabilities of state $s = 1$ in each different experiment by averaging over all L posterior draws,

$$q(s = 1|X) \approx \frac{1}{L} \sum_{i=1}^{L} q(s = 1, \text{alpha}[s = 1]|X) \tag{16.19}$$

where $q(.)$ is the un-normalised posterior density. The averaging over all posterior draws is necessary to marginalize out the alpha parameter. To normalise the posterior density we therefore divide the above by the sum of the un-normalised probability across both states,

$$Pr(s = 1|X) = \frac{q(s = 1|X)}{q(s = 1|X) + q(s = 2|X)} \tag{16.20}$$

Using the above knowledge add a `generated quantities` block to your Stan model that does this, and hence estimate the probability that each individual's tumour is benign.

```
generated quantities{
  matrix[nStudy, 2] pstate;
  for(n in 1:nStudy)
    pstate[n] = exp(lp[n] - log_sum_exp(lp[n]));
}
```

Apart from unlucky individuals 2 and 9 where $p(benign) \approx 0$ all other individual's $p(benign) \approx 1$.

Problem 16.4.5. An alternative way to code this problem is to derive a Gibbs sampler. As a first step in this process write out the full joint posterior numerator. (Hint: now use a slightly-altered definition of $s_i \in [0, 1]$, where 1 indicates a benign tumour for individual i.)

Since the priors are uniform for all variables we have,

$$p(\theta_b, \theta_m|X, S) \propto \left(\prod_{i=1}^{10} \theta_b^{s_i X_i} (1 - \theta_b)^{s_i (20 - X_i)} \right) \left(\prod_{i=1}^{10} \theta_m^{(1-s_i) X_i} (1 - \theta_m)^{(1-s_i)(20-X_i)} \right) \tag{16.21}$$

$$= \left(\theta_b^{\sum_{i=1}^{10} s_i X_i} (1 - \theta_b)^{\sum_{i=1}^{10} s_i (20 - X_i)} \right) \left(\theta_m^{\sum_{i=1}^{10} (1-s_i) X_i} (1 - \theta_m)^{\sum_{i=1}^{10} (1-s_i)(20-X_i)} \right) \tag{16.22}$$

Problem 16.4.6. By removing those terms that don't depend on θ_b derive the conditional distribution $\theta_b|\theta_m, S, X$. Hence write down $\theta_m|\theta_b, S, X$

This amounts to removing the second half of the above yielding,

$$p(\theta_b|\theta_m, S, X) \propto \theta_b^{\sum_{i=1}^{10} s_i X_i}(1-\theta_b)^{20\sum_{i=1}^{10} s_i - \sum_{i=1}^{10} s_i X_i} \tag{16.23}$$

$$\sim beta(1 + \sum_{i=1}^{10} s_i X_i, 1 + 20\sum_{i=1}^{10} s_i - \sum_{i=1}^{10} s_i X_i) \tag{16.24}$$

Hence by symmetry we have that,

$$\theta_m|\theta_b, S, X \sim beta(1 + \sum_{i=1}^{10}(1-s_i)X_i, 1 + 20\sum_{i=1}^{10}(1-s_i) - \sum_{i=1}^{10}(1-s_i)X_i) \tag{16.25}$$

Problem 16.4.7. Show that the distribution for $s_i|s_{-i}, \theta_b, \theta_m, X$ can be written as,

$$s_i|s_{-i}, \theta_b, \theta_m, X \sim Bernoulli\left(\frac{1}{1 + \left[\frac{\theta_m}{1-\theta_m}/\frac{\theta_b}{1-\theta_b}\right]^{X_i}\left[\frac{1-\theta_m}{1-\theta_b}\right]^{20}}\right) \tag{16.26}$$

All the s_{-i} terms drop out leaving,

$$p(s_i|s_{-i}, \theta_b, \theta_m, X) \propto \theta_b^{s_i X_i}(1-\theta_b)^{s_i(20-X_i)}\theta_m^{(1-s_i)X_i}(1-\theta_m)^{(1-s_i)(20-X_i)} \tag{16.27}$$

This distribution only has two discrete values corresponding to $s_i = 0, 1$, so it can be deduced that this conditional density is a Bernoulli. This can written compactly after some algebra as,

$$s_i|s_{-i}, \theta_b, \theta_m, X \sim Bernoulli\left(\frac{1}{1 + \left[\frac{\theta_m}{1-\theta_m}/\frac{\theta_b}{1-\theta_b}\right]^{X_i}\left[\frac{1-\theta_m}{1-\theta_b}\right]^{20}}\right) \tag{16.28}$$

where as $\theta_m \to 1 \implies p(s_i = 1) \to 0$, or if $\theta_b \to 1 \implies p(s_i = 1) \to 1$, and *ceteris paribus* that as $X_i \uparrow \implies p(s_i) \downarrow$.

Problem 16.4.8. Using your three derived conditional distributions create a Gibbs sampler in R, and use it to estimate $(\theta_b, \theta_m, s_1, ..., s_{10})$.

I use three functions to do this,

```
fSampleThetaB <- function(X, S){
  aSumXS <- sum(X * S)
  aSumS <- sum(S)
  return(rbeta(1, (1 + aSumXS), (1 + 20 * aSumS - aSumXS)))
}

fSampleThetaM <- function(X, S){
  aSumXS <- sum(X * (1 - S))
  aSum1S <- sum(1 - S)
  return(rbeta(1, (1 + aSumXS),(1 + 20 * aSum1S - aSumXS)))
}

fSampleS <- function(thetaB, thetaM, X){
  S <- vector(length=10)
  logOddsM <- thetaM / (1 - thetaM)
  logOddsB <- thetaB / (1 - thetaB)
  aExtra <- ((1 - thetaM) / (1 - thetaB))
  for(i in 1:10){
    aProb <- 1 / (1 + ((logOddsM / logOddsB) ^ X[i]) * (aExtra ^ 20))
    S[i] <- rbinom(1, 1, aProb)
  }
  return(S)
}
```

Again you should to obtain a mean of around 0.3 for θ_b and 0.9 for θ_m, with all individuals apart from 2 and 9 having benign growths.

16.5 How many times did I flip the coin?

Suppose that I have a coin with θ denoting the probability of it landing heads-up. In each experiment I flip the coin N times, where N is unknown to the observer, and record the number of heads obtained Y. I repeat the experiment 10 times, each time flipping the coin the same N times, and record $Y = \{9, 7, 11, 10, 10, 9, 8, 11, 9, 11\}$ heads.

Problem 16.5.1. Write down an expression for the likelihood, stating any assumptions you make.

Assuming independent and identically-distributed observations we obtain,

$$Y_i \sim \mathcal{B}(N, \theta) \tag{16.29}$$

Problem 16.5.2. Suppose that the maximum number of times the coin could be flipped is 20, and that all other (allowed) values we regard *a priori* as equally probable. Further suppose that based on previous coin flipping fun that we specify a prior $\theta \sim beta(7, 2)$. Write down the model as a whole (namely, the likelihood and the priors).

$$Y_i \sim \mathcal{B}(N, \theta) \tag{16.30}$$
$$N \sim \text{discrete-uniform}(11, 20) \tag{16.31}$$
$$\theta \sim beta(7, 2) \tag{16.32}$$

Problem 16.5.3. This problem can be coded in Stan by marginalising out the discrete parameter N. The key to doing this is writing down an expression for the log-probability for each result Y_i conditional on an assumed value of N, and θ. Explain why this can be written in Stan as,

```
log(0.1) + binomial_lpmf(Y[i] | N[s], theta);
```

where `N[s]` is the sth element of a vector N containing all possible values for this variable.

Problem 16.5.4. In the `transformed parameters` block write code that calculates the log probability for each experiment and each possible value of N.

```
transformed parameters{
  vector[10] lp;
  for(s in 1:10)
    lp[s] = log(0.1) + binomial_lpmf(Y | N[s], theta);
}
```

The above uses the vectorised form of the RHS, which is important because it allows N to be the same across all experiments (I fell foul of this when I initially coded it up!)

Problem 16.5.5. Write a Stan program to estimate θ. (Hint: in the `model` block use `target+= log_sum_exp(lp)` to marginalise out N and increment the log probability.)

The Stan program is,

```
data{
  int K;
  int Y[K];
}

transformed data{
  int N[10];
  for(s in 1:10)
    N[s] = 10 + s;
}

parameters{
  real<lower=0, upper=1> theta;
}

transformed parameters{
```

```
  vector[10] lp;
  for(s in 1:10)
    lp[s] = log(0.1) + binomial_lpmf(Y | N[s], theta);
}

model{
  target += log_sum_exp(lp);
  theta ~ beta(7,2);
}
```

For a definition of `log_sum_exp(.)` see the answer to the next question. The posterior mean of θ is about 0.78, with a 95% HDI of $0.54 \leq \theta \leq 0.91$.

Problem 16.5.6. Use the `generated quantities` block to estimate the probabilities of each state.

This relies on us estimating the un-normalised density for the number of coin flips by averaging over all samples for θ,

$$q(N = 11|Y) \approx \frac{1}{L} \sum_{i=1}^{L} q(N = 11, \theta_i|Y) \tag{16.33}$$

where $q(.)$ is the un-normalised posterior density and L is the number of posterior samples. To normalise this density we then divide the above by the un-normalised density for all other possible values for N,

$$Pr(N = 11|Y) = \frac{q(N = 11|Y)}{\sum_{N=11}^{20} q(N = i|Y)} \tag{16.34}$$

To do this in Stan we use `log_sum_exp(.)` which is defined as,

$$\text{log\_sum\_exp}_{i=1}^{K} \alpha = log \sum_{i=1}^{K} exp(\alpha) \tag{16.35}$$

which allows us to marginalise out any dependence on N in log prob space because,

$$\log p(\theta) = \log \sum_{N=1}^{K} p(\theta, N) \tag{16.36}$$

$$= \log \sum_{N=1}^{K} exp\left(log\ p(\theta, N)\right) \tag{16.37}$$

$$= \text{log\_sum\_exp}_{N=1}^{K}(\log p(\theta, N)) \tag{16.38}$$

So implementing this in Stan we have,

```
generated quantities {
  simplex[10] pState;
  pState = exp(lp - log_sum_exp(lp));
}
```

which results in probabilities of each state shown in Figure 16.2. Note that it is better to use the expectation once, rather than do exp(something) divided by exp(something else) because this risks numerical accuracy.

Problem 16.5.7. An alternative way to estimate N and θ is to derive a Gibbs sampler for this problem. To do this first show that the joint (un-normalised) posterior distribution can be written as,

$$p(\theta, N|Y) \propto \left[\prod_{i=1}^{K} \binom{N}{Y_i} \theta^{Y_i}(1-\theta)^{N-Y_i}\right] \theta^{\alpha-1}(1-\theta^{\beta-1}) \tag{16.39}$$

where $K = 10$ and $(\alpha, \beta) = (7, 2)$ are the parameters of the prior distribution for θ.

The above is self-evident if you drop all the non-constant terms in the likelihood and prior.

Problem 16.5.8. Derive the conditional distribution $\theta|N, Y$. (Hint: remove all parts of the joint distribution that do not explicitly depend on θ.)

Removing those θ-independent parts of the joint distribution we obtain,

$$p(\theta|N, Y) \propto \theta^{\alpha-1+\sum Y_i}(1-\theta)^{\beta-1+KN-\sum Y_i} \tag{16.40}$$

which is recognisable as a $\theta|N, Y \sim beta(\alpha + \sum Y_i, \beta + KN - \sum Y_i)$ distribution!

Problem 16.5.9. Write an R function that independently samples from the conditional distribution $\theta|N, Y$.

An example R function is shown below,

```
fSampleTheta <- function(Y, N, alpha, beta){
  K <- length(Y)
  aSum.Y <- sum(Y)
  aSum.N <- N * K
  return(rbeta(1, alpha + aSum.Y, beta + aSum.N - aSum.Y))
}
```

Problem 16.5.10. Show that the conditional pmf $N|\theta, Y$ can be written as,

$$p(N|\theta, Y) \propto \left[\prod_{i=1}^{K} \binom{N}{Y_i}\right] (1-\theta)^{NK} \tag{16.41}$$

Removing all those parts of the joint distribution that do not depend on N we obtain the above.

Problem 16.5.11. Using the previously-derived expression, write a function that calculates the un-normalised conditional $N|\theta, Y$ for $N = 11, ..., 20$, which when normalised can be used to sample a value for N. Hint use the `sample` function in R.

```
fSampleN <- function(Y, theta){
  lUnnorm <- vector(length=10)
  K <- length(Y)
  for(i in 1:10){
    N <- i + 10
    lUnnorm[i] <- prod(sapply(Y, function(x) choose(N, x))) *
                                            (1 - theta) ^ (N * K)
  }
  lProb <- lUnnorm / sum(lUnnorm)
  return(sample(11:20, 1, prob=lProb))
}
```

Problem 16.5.12. Write a working Gibbs sampler using your two previously-created functions, and use this to estimate the probability distribution over θ and N.

I first create a function that randomly picks an updating order for θ and N, then samples from these.

```
fGibbsSingle <- function(Y, theta, N, alpha, beta){
  aInd <- rbinom(1, 1, 0.5)
  if(aInd==1){
    theta.new <- fSampleTheta(Y, N, alpha, beta)
    N.new <- fSampleN(Y, theta.new)
  }else{
    N.new <- fSampleN(Y, theta)
    theta.new <- fSampleTheta(Y, N.new, alpha, beta)
  }
  return(list(theta=theta.new, N=N.new))
}
```

Then string these together into a Gibbs routine,

```
fGibbsTotal <- function(numIter, Y, theta, N, alpha, beta){
  lTheta <- vector(length=numIter)
  lN <- vector(length=numIter)
  lTheta[1] <- theta
  lN[1] <- N
  for(i in 2:numIter){
    lParams <- fGibbsSingle(Y, lTheta[i - 1], lN[i - 1], alpha, beta)
    lTheta[i] <- lParams$theta
    lN[i] <- lParams$N
  }
```

```
  return(list(theta=lTheta, N=lN))
}
```

Then running this we obtain a distribution of values of N and θ sampled,

```
lSamples <- fGibbsTotal(10000, Y, 0.78, 11, 7, 2)
hist(lSamples$N)
hist(lSamples$theta)
```

which should have similar means to that of HMC for θ. The distribution for N is not the same, because in HMC we are sampling $p(N)$ not N itself!

Problem 16.5.13. Compare the rate of convergence in the mean of N sampled via Gibbs with that over that estimated from the $p(N)$ distribution that you sampled in HMC. Why is the rate of convergence so much faster for HMC? (Hint: this is not due to the standard benefits of HMC that I extolled in this chapter.)

First you will need to determine the expected value of N from $p(N)$ from Stan

```
fExpectation <- function(mStates){
  lN <- apply(mStates, 1, function(x) sum(x * seq(11, 20)))
  return(lN)
}
lExpected <- fExpectation(lPState)
```

Then examining a running mean of N over time from the Gibbs versus HMC we see that the latter is much faster to converge (Figure 16.3),

```
fRunningMean <- function(lSample){
  return(cumsum(lSample) / seq_along(lSample))
}
aDF <- data.frame(time=1:2000,
                  gibbs=fRunningMean(lSamples$N),
                  hmc=fRunningMean(lExpected))
aDF <- melt(aDF, id.vars='time')
ggplot(aDF, aes(x=time, y=value, colour=as.factor(variable))) + geom_line()
```

This is due to Rao-Blackwellisation – by marginalising out any dependence on N, and using the marginal distribution $p(\theta)$ to infer $p(N)$ we get a significant speed up.

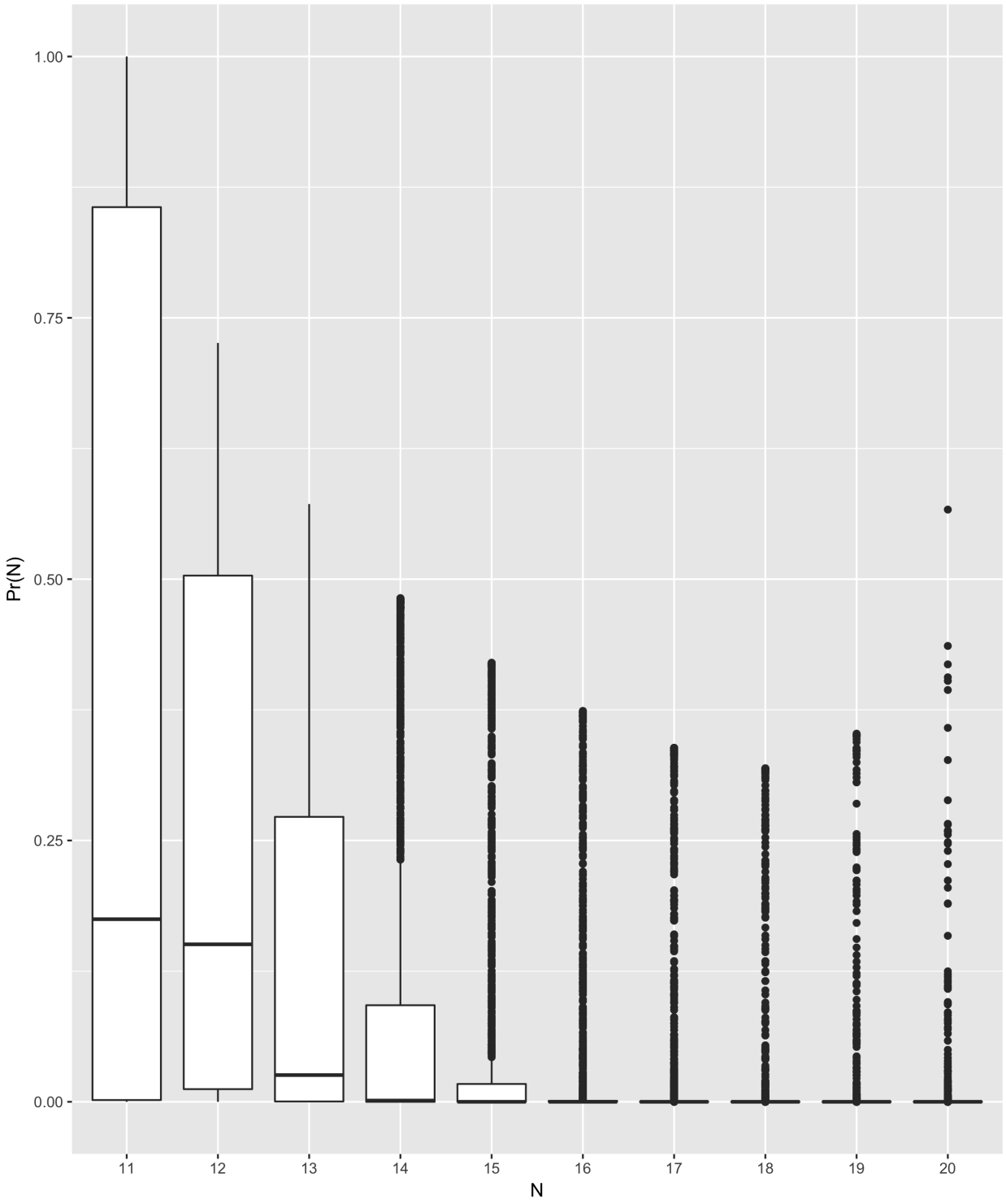

Figure 16.2: The estimated probabilities for each N in the coin flipping example.

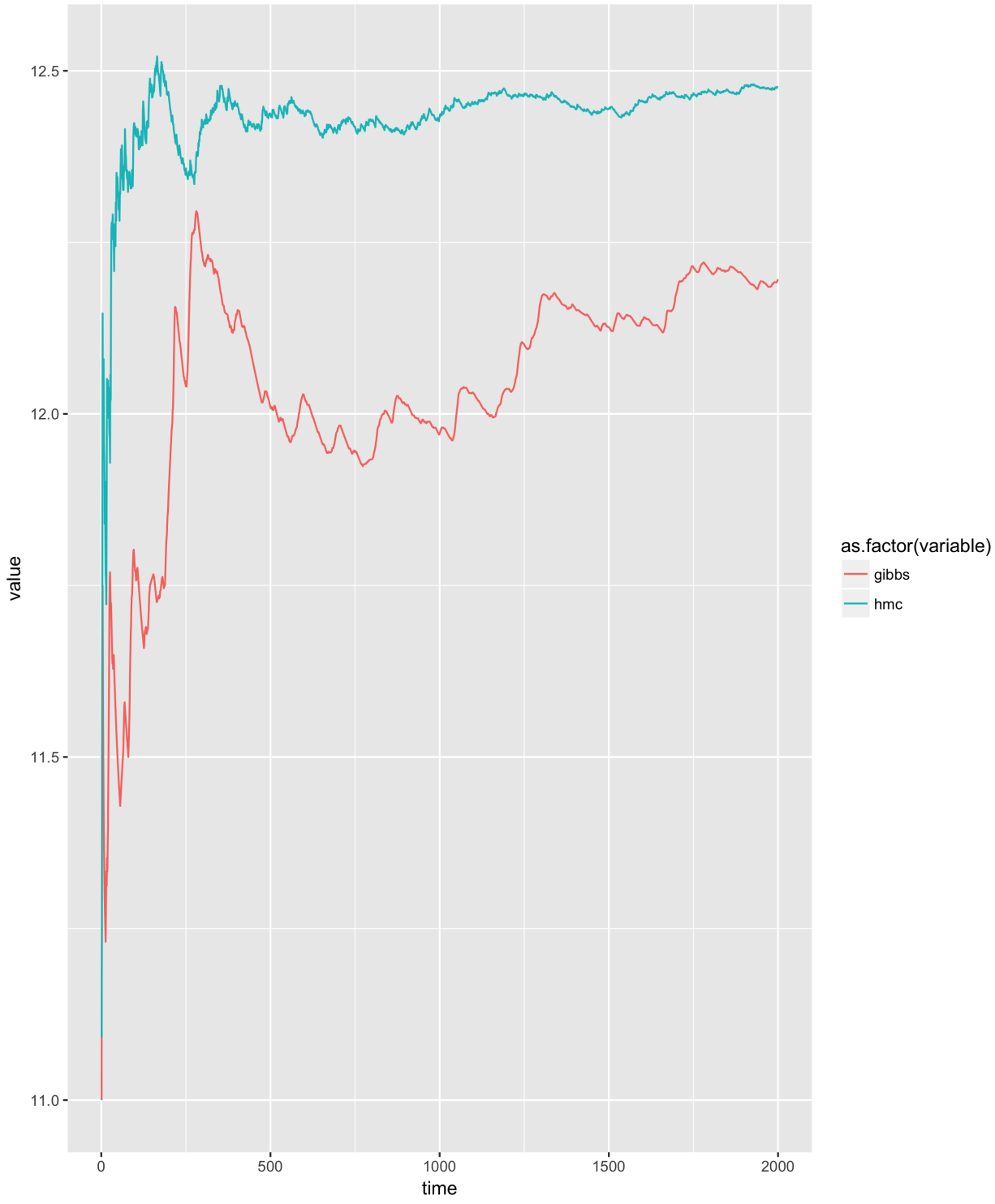

Figure 16.3: The rate of convergence of the mean N for the coin flipping example by HMC and Gibbs.

Chapter 17

Hierarchical models

17.1 A meta-analysis of beta blocker trials

Table 17.1 shows the results of some of the 22 trials included in a meta-analysis of clinical trial data on the effect of beta-blockers on reducing the risk of myocardial infarction [3]. The file `hierarchical_betaBlocker.csv` contains the full dataset.

The aim of this meta-analysis is to determine a robust estimate of the effect of beta-blockers by pooling information from a range of previous studies (this problem has been adapted from [10]).

Table 17.1: The data from the original study.

	Mortality: deaths/total	
Study	**Treated**	**Control**
1	3/38	3/39
2	7/114	14/116
3	5/69	11/93
4	102/1533	127/1520
...		
20	32/209	40/218
21	27/391	43/364
22	22/680	39/647

Problem 17.1.1. Start by assuming that the numbers of deaths in the control (r_i^c) and treated (r_i^t) groups for each trial are given by binomial distributions of the form:

$$r_i^c \sim \mathcal{B}(p_i^c, n_i^c) \tag{17.1}$$
$$r_i^t \sim \mathcal{B}(p_i^t, n_i^t) \tag{17.2}$$

where (n_i^t, n_i^c) are the numbers of individuals in the treatment and control datasets respectively. Further assume that the probabilities of mortality in the treatment and control datasets are given by:

$$logit(p_i^c) = \mu_i \tag{17.3}$$
$$logit(p_i^t) = \mu_i + \delta_i \tag{17.4}$$
$$\tag{17.5}$$

where $logit(x) = log\left(\frac{x}{1-x}\right)$, and we expect $\delta_i < 0$ if the beta-blockers have the desired effect. We assume the following diffuse priors for the parameters

$$\mu_i \sim \mathcal{N}(0, 10) \tag{17.6}$$
$$\delta_i \sim \mathcal{N}(0, 10) \tag{17.7}$$
$$\tag{17.8}$$

Estimate the posteriors for δ_i for the above model using Stan, or otherwise. Note: that for this model there is no inter-dependence between the studies. (Hint: use the Stan function `binomial_logit`.)

The code to estimate this model is given by:

```
data {
  int<lower=0> N;
  int<lower=0> nt[N];
  int<lower=0> rt[N];
  int<lower=0> nc[N];
  int<lower=0> rc[N];
}

parameters {
  vector[N] mu;
  vector[N] delta;
}

model {
  rt ~ binomial_logit(nt, mu + delta);
  rc ~ binomial_logit(nc, mu);
  delta  ~ normal(0, 10);
  mu ~ normal(0, 10);
}
```

The posteriors for δ_i in this model are fairly wide and contain non-zero densities at zero (Figure 17.1).

Problem 17.1.2. An alternative framework is a hierarchical model where we assume there to be a common over-arching distribution, across trials such that $\delta_i \sim \mathcal{N}(d, \sigma)$. By assuming the following

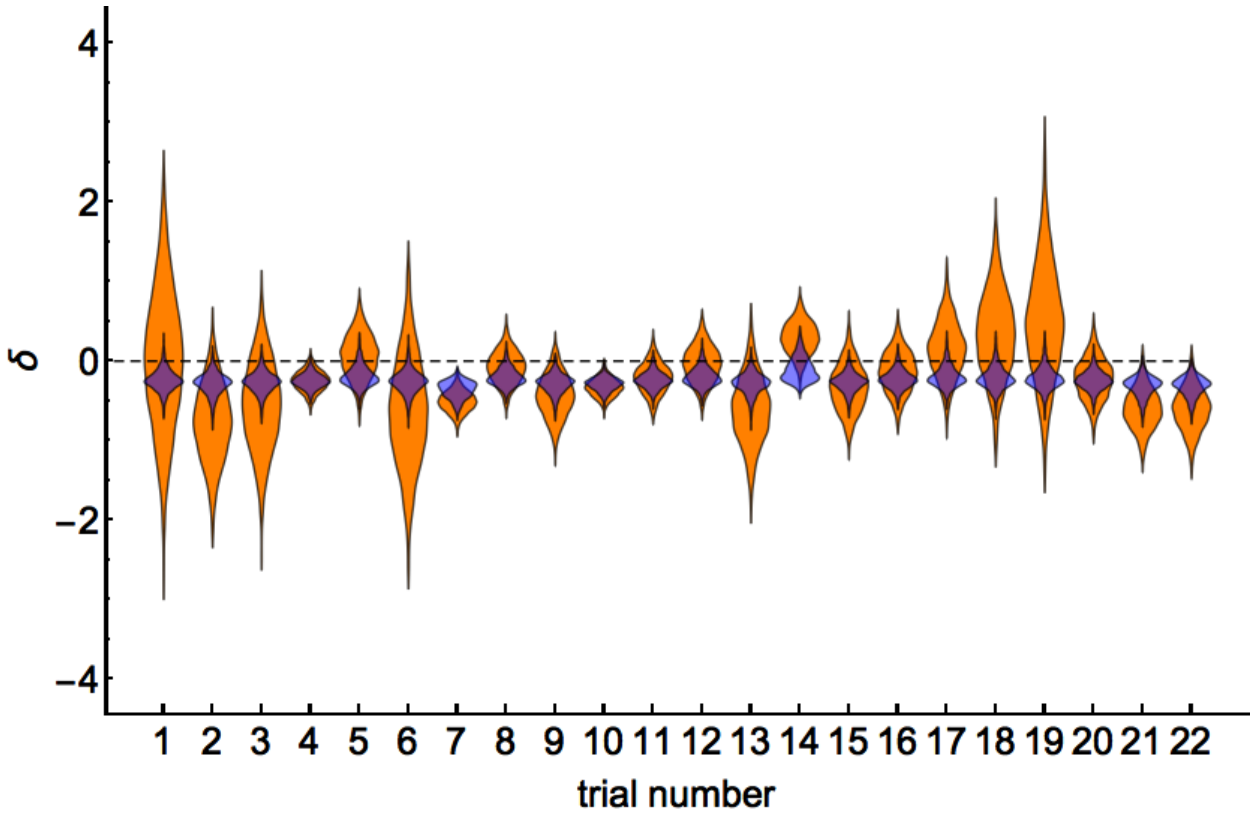

Figure 17.1: The posterior estimates of δ_i for the fully-heterogeneous (orange) and hierarchical (blue) models.

priors on these parameters estimate this model:

$$d \sim \mathcal{N}(0, 10) \tag{17.9}$$

$$\sigma \sim Cauchy(0, 2.5), \text{ for } \sigma \geq 0 \tag{17.10}$$

Estimate the posteriors for δ_i using Stan. How do these estimates compare to the non-hierarchical model?

The code for this problem is shown below:

```
data {
  int<lower=0> N;
  int<lower=0> nt[N];
  int<lower=0> rt[N];
  int<lower=0> nc[N];
  int<lower=0> rc[N];
}

parameters {
  real d;
  real<lower=0> sigma;
```

```
  vector[N] mu;
  vector[N] delta;
}

model {
  rt ~ binomial_logit(nt, mu + delta);
  rc ~ binomial_logit(nc, mu);
  delta  ~ normal(d, sigma);
  mu ~ normal(0, 10);
  d ~ normal(0, 10);
  sigma ~ cauchy(0, 2.5);
}
```

When I used the above code I ran into a few divergent iterations - this should **not** be ignored, and is best handled by increasing adapt_delta=0.95 when calling Stan. This should help the sampler avoid taking too large steps, and diverging in areas of high posterior curvature. The problematic regions of parameter space here are due to the correlation in estimates between d and the various δ_i; this is unsurprising, each trial only has a relatively small data sample. As such, it is difficult to disentangle the specific individual study effects from the overall effect d.

The hierarchical estimates of the effect of the drug are much more concentrated (Figure 17.1) - by pooling data across all studies we are better able to precisely estimate the effect of the drug. These indicate that the beta-blockers appear to act as desired - decreasing the probability of mortality.

Problem 17.1.3. Using the hierarchical model estimate the cross-study effect of the beta-blockers. (Hint: use the `generated quantities` code block.)

The code for to sample an overall δ is given below:

```
generated quantities {
  real delta_overall;
  delta_new = normal_rng(d, sigma);
}
```

Overall we estimate a negative value for δ (Figure 17.2). Whilst the posterior does overlap zero, we are fairly confident in concluding that $\delta < 0$.

Problem 17.1.4. For an out of sample trial suppose we know that $\mu_i = -2.5$. Using the cross-study estimates for δ estimate the reduction in probability for a patient taking the beta-blockers.

This is done by using the inverse-logit transformation (the logistic sigmoid). Essentially you want to evaluate logistic-sigmoid(-2.5) and compare it with logistic-sigmoid(-2.5-delta), across all the samples in your model. This results in a posterior distribution that is peaked at about 0.015 (Figure 17.3); indicating about a 1.5% reduction in mortality risk for those patients taking beta-blockers.

Problem 17.1.5. Estimate a model with a single, constant value of δ and μ across all trials. Graph the posterior for δ, and compare it with the cross-study hierarchical model estimate.

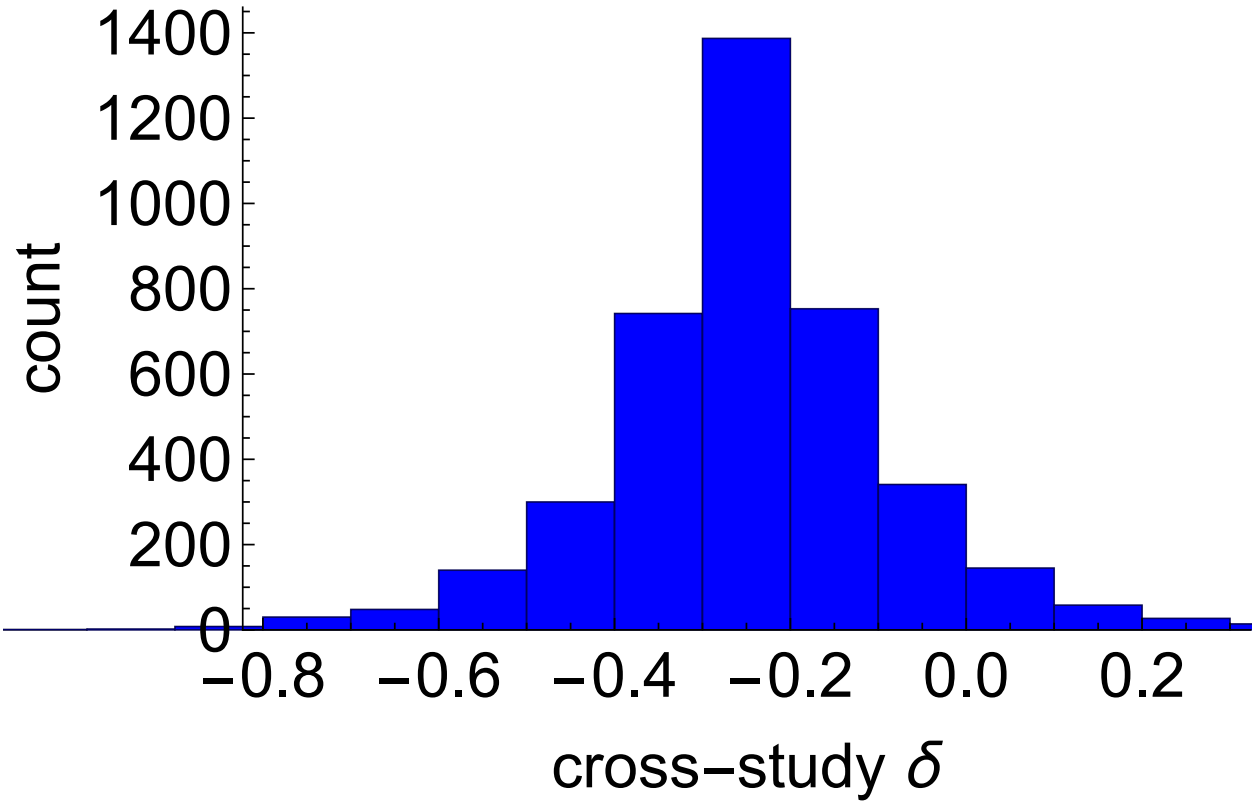

Figure 17.2: The posterior estimate of δ across all trials for the hierarchical model.

The non-hierarchical model gives us a false confidence in our estimates of δ, by assuming that the data from the individual studies are equivalent (exchangeable). This means that the estimate of δ obtained is more concentrated than for the hierarchical model (Figure 17.4).

Problem 17.1.6. Carry out appropriate posterior predictive checks on the homogeneous and hierarchical models, and hence conclude the preferred modelling choice.

One check to do here is to generate posterior predictive data sets of the same shape as the real data, and for each trial record whether the simulated value is greater than the actual. This can be done fairly easily using the `generated quantities` block (shown here for the homogeneous model):

```
generated quantities {
  int<lower=0> simTreatMort[N];
  int<lower=0> simContrMort[N];
  int indicatorTreat[N];
  int indicatorContr[N];
  for (i in 1:N) {
    simTreatMort[i] = binomial_rng(nt[i], inv_logit(mu + delta));
    simContrMort[i] = binomial_rng(nc[i], inv_logit(mu));
    indicatorTreat[i] = (simTreatMort[i] > rt[i]);
    indicatorContr[i] = (simContrMort[i] > rc[i]);
```

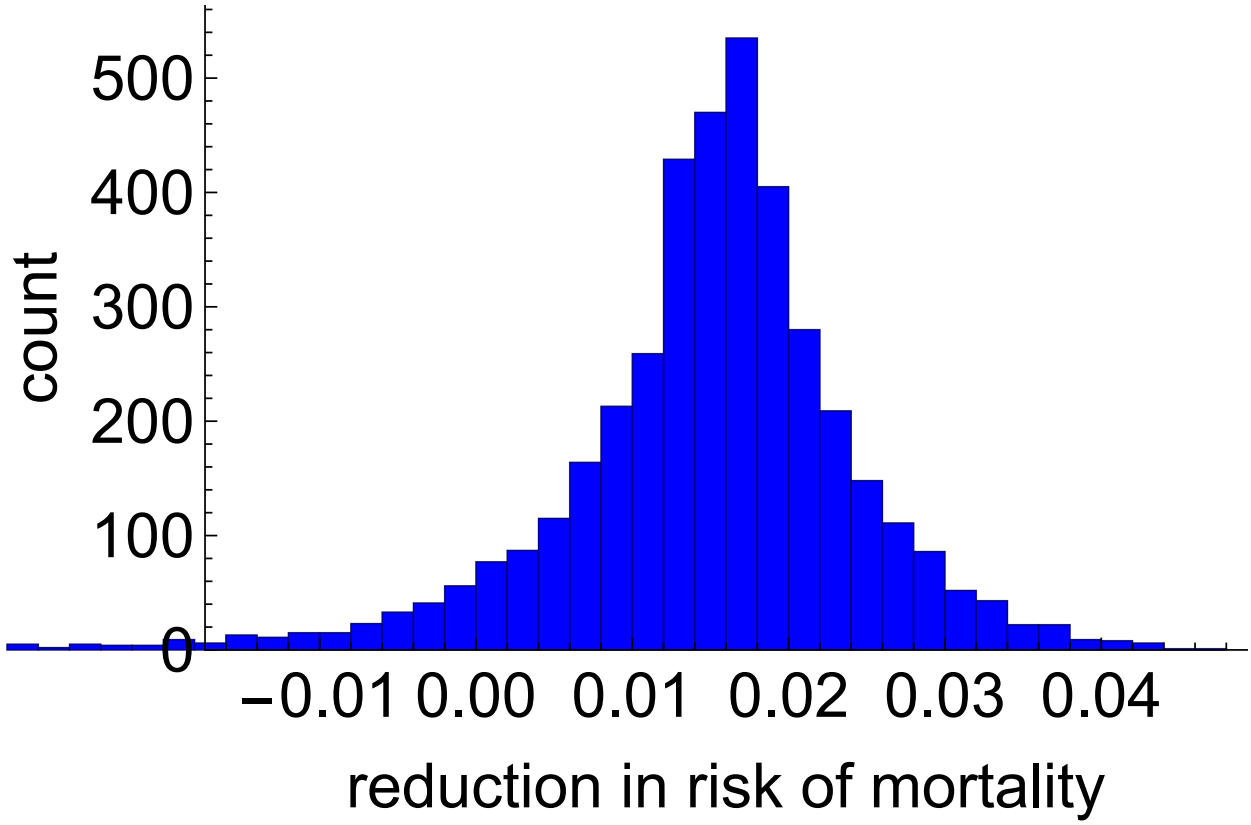

Figure 17.3: The posterior estimates of the reduction in mortality risk associated with taking a beta-blocker when $\mu = -2.5$.

```
  }
}
```

Doing this for both models we find that there are a range of Bayesian p values near 0 or 1 for the homogeneous model, whereas this is not the case for the hierarchical model (Figure 17.5.) Intuitively - by assuming that there was no difference between the data from each study - the homogeneous coefficient model is unable to replicate the degree of variation we see in the real data. We therefore prefer the hierarchical model.

17.2 I can't get no sleep

These data are from a study described in Belenky et al. (2003) [2] that measured the effect of sleep deprivation on cognitive performance. There were 18 subjects chosen from a population of interest (lorry drivers) who were restricted to 3 hours of sleep during the trial. On each day of the experiment their reaction time to a visual stimulus was measured. The data for this example are contained within `evaluation_sleepstudy.csv`, and contains three variables: Reaction, Days and Subject ID which measure the reaction time of a given subject on a particular day.

A simple model that explains the variation in reaction times is a linear regression model of the form:

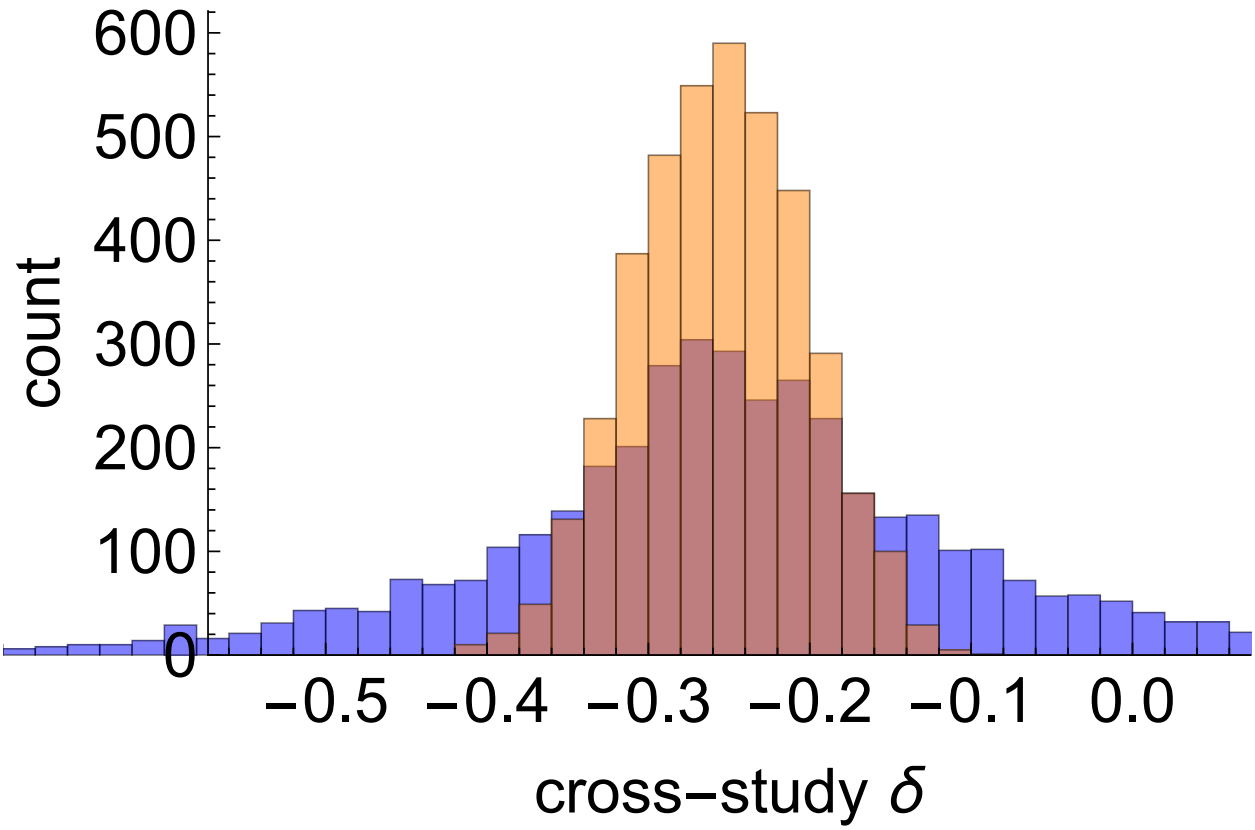

Figure 17.4: The posterior estimates of cross-study δ for the hierarchical (blue) and homogeneous (orange) models.

$$R(t) \sim \mathcal{N}(\alpha + \beta t, \sigma) \tag{17.11}$$

where $R(t)$ is the reaction time on day t of the experiment across all observations.

Problem 17.2.1. Assuming $\mathcal{N}(0, 250)$ priors on both α and β code up the above model in Stan. Use it to generate 1000 samples per chain, across 4 chains. Has the sampling algorithm converged?

```
data {
  int N; // number of observations
  matrix[N,2] X; // ones + days of sleep deprivation
  vector[N] R; // reaction times
}

parameters {
  vector[2] gamma;
  real<lower=0> sigma;
}
```

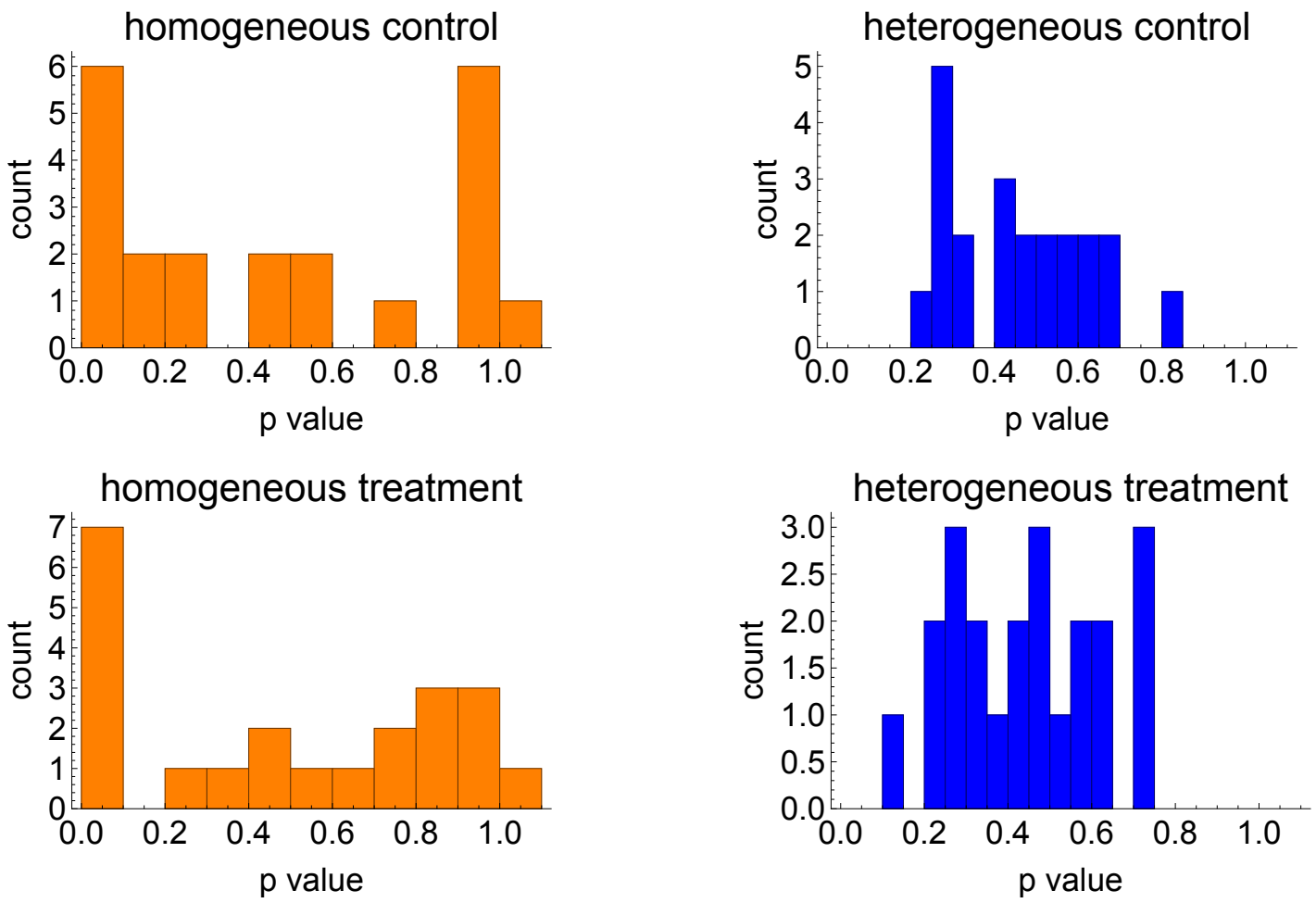

Figure 17.5: The distribution of Bayesian p values measuring whether the posterior predictive data exceeds the actual across each of the 22 trials for the homogeneous (orange) and hierarchical (blue) models.

```
model {
  R ~ normal(X * gamma, sigma);
 gamma ~ normal(0, 250);
}
```

After the requisite number of iterates the value for $\hat{R} < 1.1$, meaning it appears that the sampling distribution has converged.

Problem 17.2.2. Plot the posterior samples for α and β. What is the relationship between the two variables, and why?

There is a strong negative correlation between the estimates of these two variables. This is because to generate a line going through the centre of the dataset, if the intercept increases, the gradient must decrease.

Problem 17.2.3. By using the `generated quantities` code block or otherwise generate samples from the posterior predictive distribution. By overlaying the real time series for each individual on a graph of the posterior predictive comment on the fit of the model to data.

The posterior predictive distribution - whilst being a reasonable fit to the data for the anonymous data - is not able to fit well the data at the individual level (Figure 17.6).

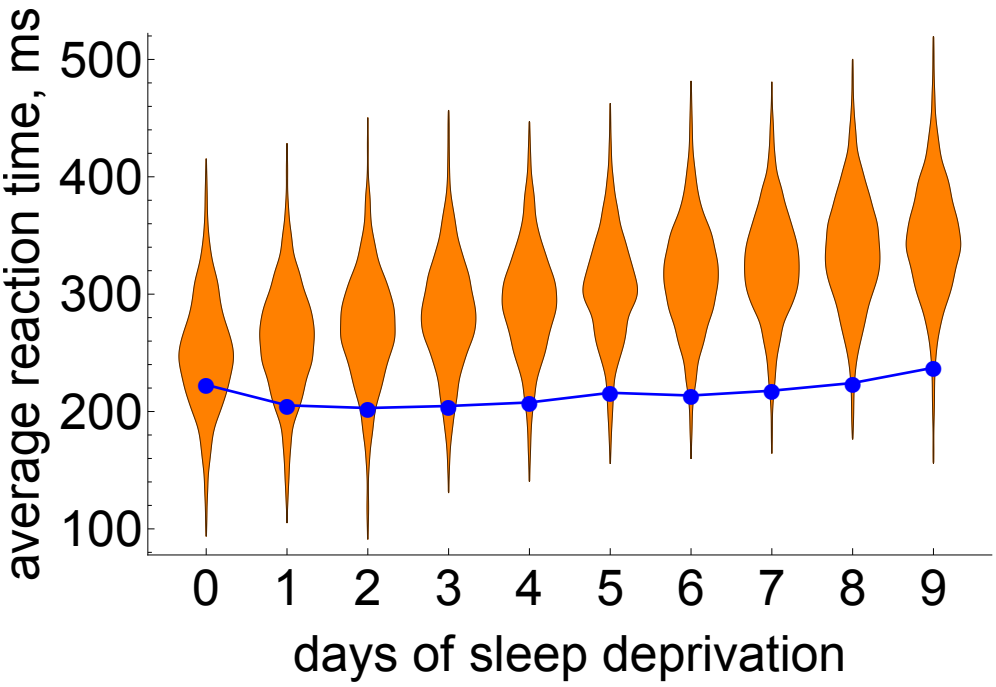

Figure 17.6: The posterior predictive distribution (orange) and the data for one individual in the sleep study (blue) using the "homogeneous coefficients" model.

Problem 17.2.4. Fit a model with separate (α, β) for each individual in the dataset. Use separate and independent $\mathcal{N}(0, 250)$ priors for the parameters. Again use 1000 samples per chain over 4 chains.

This is best done by creating an index of 1 to 18; corresponding to individual subject ID. The model can then be coded up as below:

```
data {
  int N; // number of observations
  vector[N] t; // days of sleep deprivation
  vector[N] R; // reaction times of individuals in the study
  int subject[N]; // subject ID
}

parameters {
  real alpha[18];
  real beta[18];
```

```
  real<lower=0> sigma;
}

model {
  for (i in 1:N)
    R[i] ~ normal(alpha[subject[i]] + beta[subject[i]] * t[i], sigma);

  alpha ~ normal(0, 250);
  beta ~ normal(0, 250);
  sigma ~ normal(0, 50);
}
```

Problem 17.2.5. Compute the posterior mean estimates of the β parameters for the new "heterogeneous-parameters" model. How do these compare to the single β estimate obtained for the homogeneous model?

The homogeneous estimate is about 10.4, with the heterogeneous estimates ranging from -2.8 (for subject 9) to 21.9 (for subject 1). Overall the heterogeneous estimates should have a mean that is roughly similar to the single estimate (it's not exactly so).

Problem 17.2.6. Using the `generated quantities` code block, or otherwise, generate samples from the posterior predictive distribution. By comparing individual subject data to the posterior predictive samples, comment on the fit of the new model.

The posterior predictive distribution can be generated in similar fashion to previously, but using the individual subject IDs as an array index.

```
generated quantities {
  vector[N] R_simulated; // store post-pred samples
  for (i in 1:N)
    R_simulated[i] = normal_rng(alpha[subject[i]] + beta[subject[i]] * t[i],
                                sigma);
}
```

The heterogeneous coefficients model is able to fit the data much more effectively at the individual data (Figure 17.7). This is unsurprising - essentially we may be guilty of overfitting the model to the data.

Problem 17.2.7. Partition the data into two subsets: a training set (of subjects 1-17) and a testing set (of subject 18 only). By fitting both models - the heterogeneous and homogeneous coefficients models - on the training sets, compare the performance of each model on predicting the test set data.

To do this I create two new data variables that hold only the data for the 18th subject. I then change the original data so that it only holds data for subjects 1-17. Of course these manipulations could be handled directly in the Stan code itself, but I prefer to do this outside. The code for the heterogeneous model is shown below:

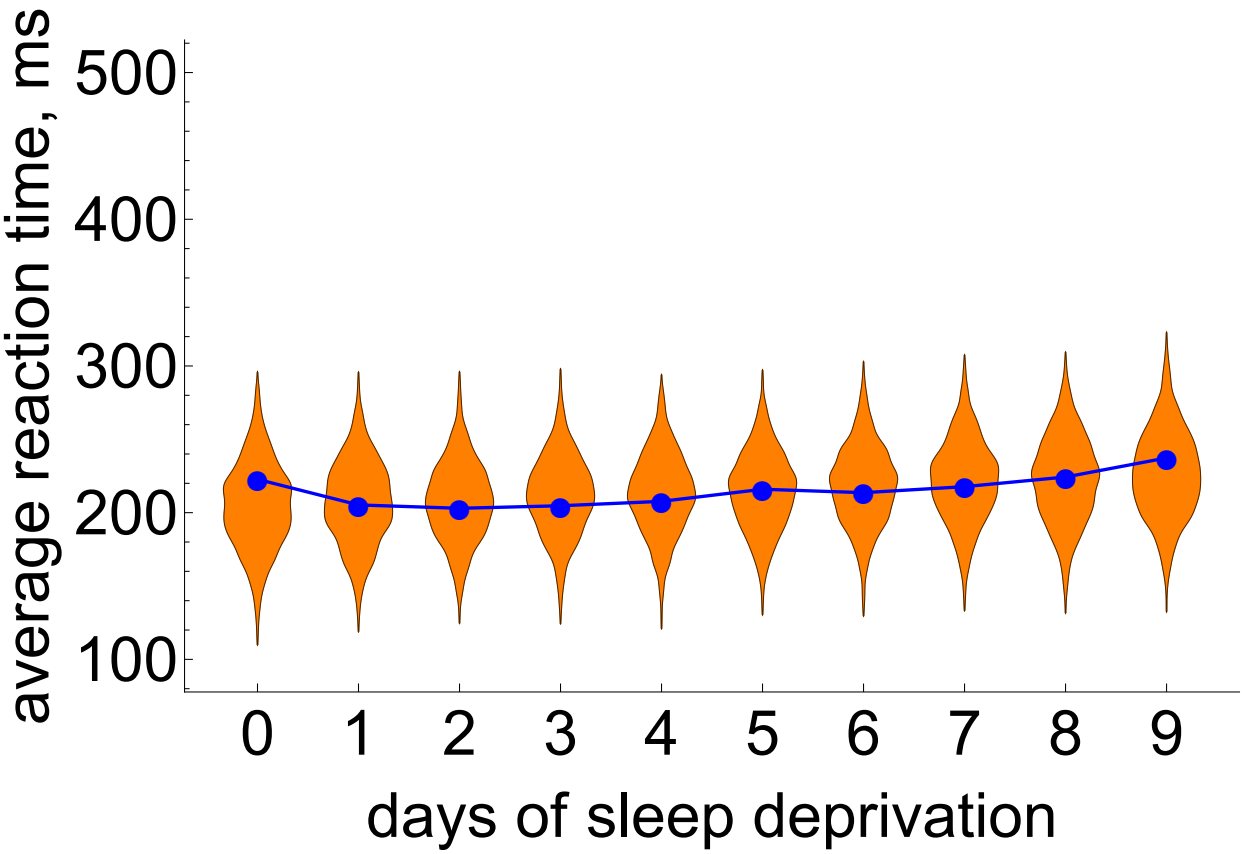

Figure 17.7: The posterior predictive distribution (orange) and the data for one individual in the sleep study (blue) using the "heterogeneous coefficients" model.

```
data {
  int N; // number of observations in training set
  vector[N] t; // days of sleep deprivation in training set
  vector[N] R; // reaction times of individuals in the training set
  int subject[N];
  int N2; // number of data points in the test set
  vector[N2] t2; // time obs in test set
}

parameters {
  real alpha;
  real beta;
  real<lower=0> sigma;
}

model {
  for (i in 1:N)
    R[i] ~ normal(alpha[subject[i]] + beta[subject[i]] * t[i], sigma);
```

```
  alpha ~ normal(0, 250);
  beta ~ normal(0, 250);
  sigma ~ normal(0, 50);
}

generated quantities {
  vector[N2] R_simulated; // store post-pred samples
  real aAlpha;
  real aBeta;

  aAlpha = normal_rng(0, 250);
  aBeta = normal_rng(0, 250);
  for (i in 1:N2)
    R_simulated[i] = normal_rng(aAlpha + aBeta * t2[i], sigma);
}
```

For the heterogeneous model there is really only one way to generate predictions for the test set - sample a value of the parameters from the priors, and using these parameter values to generate predictive datasets. Because the priors are wide this actually produces very poor predictions (Figure 17.8).

The homogeneous coefficients model however performs much better as is much more generalisable to new datasets. Intuitively the heterogeneous coefficients model is overfit to the data.

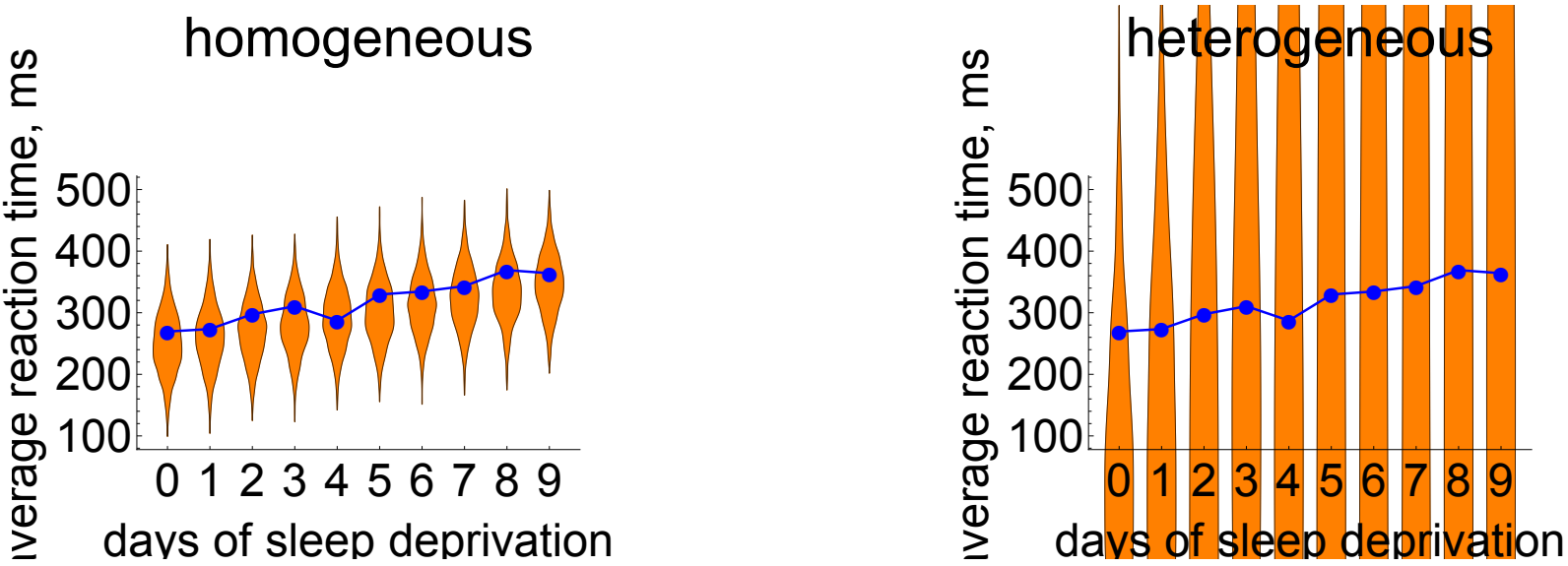

Figure 17.8: The posterior predictive distribution (orange) and the data for subject 18 for a model fitted to the other 17 subjects' data, across the homogeneous coefficients (left), and heterogeneous coefficients (right) models.

Problem 17.2.8. Alternatively we can fit a hierarchical model to the data which (hopefully) captures some of the best elements of each of the aforementioned models. Here we assume that the

individual (α, β) for each subject are allowed to vary, but there is some overarching "population-level" distribution from which they are drawn. Assume that the coefficients have the following relationships:

$$\alpha_i \sim \mathcal{N}(a, b) \tag{17.12}$$
$$\beta_i \sim \mathcal{N}(c, d) \tag{17.13}$$
$$a \sim \mathcal{N}(100, 100) \tag{17.14}$$
$$b \sim Cauchy(0, 5) \tag{17.15}$$
$$c \sim \mathcal{N}(10, 5) \tag{17.16}$$
$$d \sim Cauchy(0, 1) \tag{17.17}$$

Code up the above model and compare the posterior distribution for β for the hierarchical model, with those from the heterogeneous ones.

The posterior distribution for the parameters exhibits shrinkage towards the grand mean (Figure 17.9). In general those parameter estimates with a. the highest uncertainty, and b. lie furthest away from the mean, are shrunk the most in hierarchical models.

Problem 17.2.9. Graph the posterior distribution for β for another individual (not in the original dataset). How does this distribution compare to the value of β obtained from the homogeneous coefficient model? (Hint: use the `generated quantities` block to generate samples of β from the top-level parameters c and d.)

The code to sample a value of β for an out-of-sample individual is given below:

```
generated quantities {
  real aBeta;
  aBeta = normal_rng(c, d);
}
```

The posterior distribution for β has a mean of 10.2 (about the same as the original homogeneous estimate), but is wider (Figure 17.10).

17.3 Hierarchical ODEs: Bacterial cell population growth

The file `hierarchical_ode.csv` contains data for 5 replicates of an experiment in which bacterial cell population numbers were measured over time. The following model for bacterial population size is proposed to explain the data:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \alpha N(1 - \beta N). \tag{17.18}$$

However measurement of bacterial cell numbers is subject to random, uncorrelated measurement error:

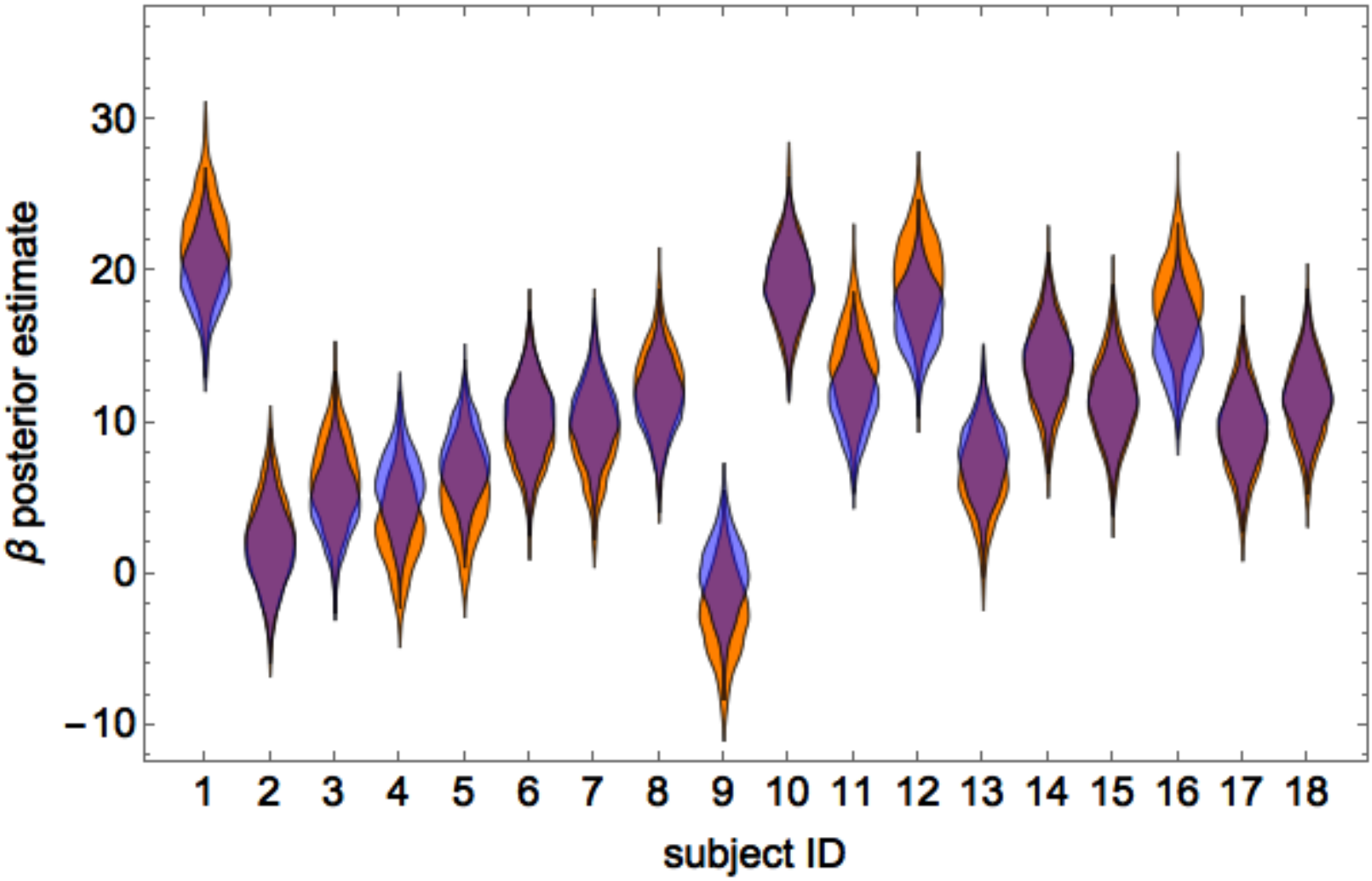

Figure 17.9: The posterior estimates of β for the heterogeneous estimates (orange) versus the hierarchical estimates (blue).

$$N^*(t) \sim \mathcal{N}(N(t), \sigma), \tag{17.19}$$

where $N^*(t)$ is the measured number of cells, and $N(t)$ is the true population size. Finally we suppose that the initial number of bacteria cells is unknown, and hence must be estimated.

Further we assume the following priors:

$$\begin{aligned} \alpha &\sim \mathcal{N}(0, 2) \\ \beta &\sim \mathcal{N}(0, 2) \\ \sigma &\sim \mathcal{N}(0, 1) \\ N(0) &\sim \mathcal{N}(5, 2) \end{aligned}$$

where all parameters have a lower value of zero.

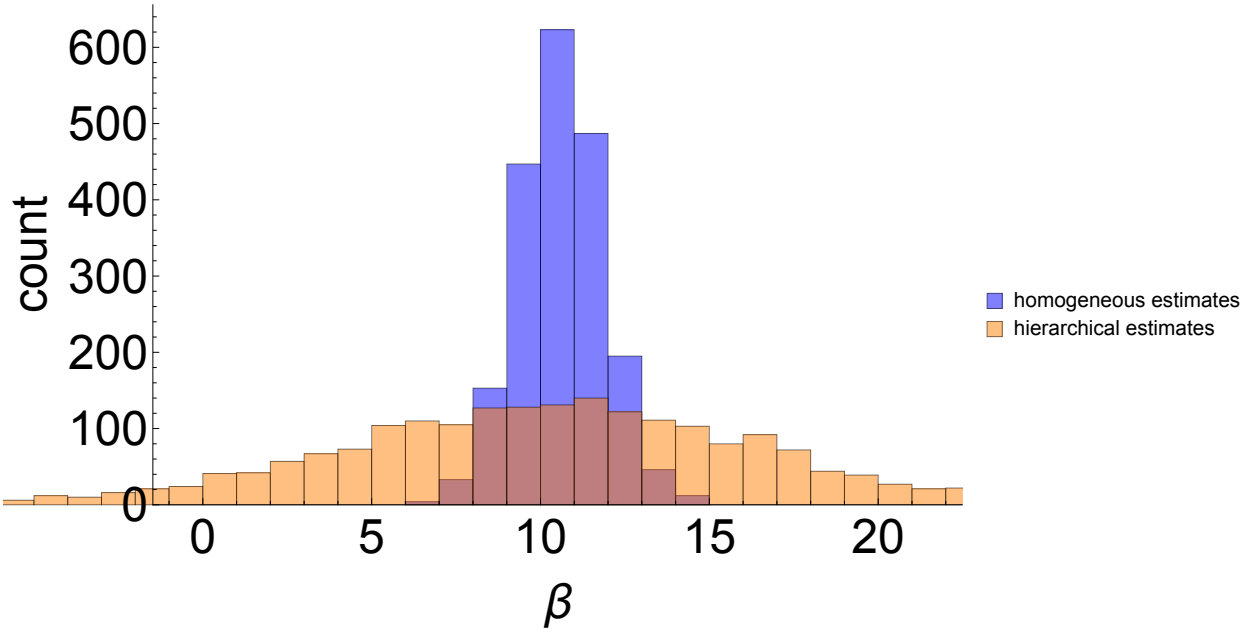

Figure 17.10: The posterior estimates of β for an out-of-sample individual for the homogeneous coefficients and hierarchical models.

Problem 17.3.1. Write a Stan function that returns $\frac{\mathrm{d}N}{\mathrm{d}t}$. (Hint 1: this will need to be done within the "functions" block at the top of the Stan file. Hint 2: the function must have a structure:

```
real[] bacteria_deriv(real t, real[] y, real[] theta, real[] x_r, int[] x_i)
```

where the variables x_i and x_r are not used here, but nonetheless need to be defined:

```
transformed data {
real x_r[0];
int x_i[0];
}
```

)

See answer to problem 17.3.2 for code.

Problem 17.3.2. Estimate a model where the parameters (α, β) are assumed to be the same across all experimental replicates.

```
functions {
  real[] bacteria_deriv(real t, real[] y, real[] theta, real[] x_r, int[] x_i) {
    real dydt[1];

    dydt[1] = theta[1] * y[1] * (1 - theta[2] * y[1]);
    return dydt;
  }
```

```
}

data {
  int<lower=1> T;
  int<lower=0> N;
  real t0;
  real ts[T];
  matrix[T,N] y;
}

transformed data {
  real x_r[0];
  int x_i[0];
}

parameters {
  real<lower=0, upper=2> theta[2]; // contains parameters (alpha,beta)
  real<lower=0> sigma;
  real<lower=0, upper=10> y0[1];
}

model {
  real y_hat[T, 1];
  sigma ~ cauchy(0, 1);
  theta ~ normal(0, 2);
  y0 ~ normal(5, 2);
  y_hat = integrate_ode(bacteria_deriv, y0, t0, ts, theta, x_r, x_i);
  for (i in 1:N)
    for (t in 1:T)
      y[t, i] ~ normal(y_hat[t, 1], sigma);
}

// use this to capture the log-likelihood for later parts of the question
generated quantities {
  vector[N * T] logLikelihood;
  int k;
  real y_hat[T, 1];

  k = 1;
  y_hat = integrate_ode(bacteria_deriv, y0, t0, ts, theta, x_r, x_i);

  for (i in 1:N){
    for (t in 1:T){
      logLikelihood[k] = normal_log(y[t, i], y_hat[t, 1], sigma);
      k = k + 1;
    }
  }
```

```
}
```

The posteriors for (α, β) are shown in Figure 17.11.

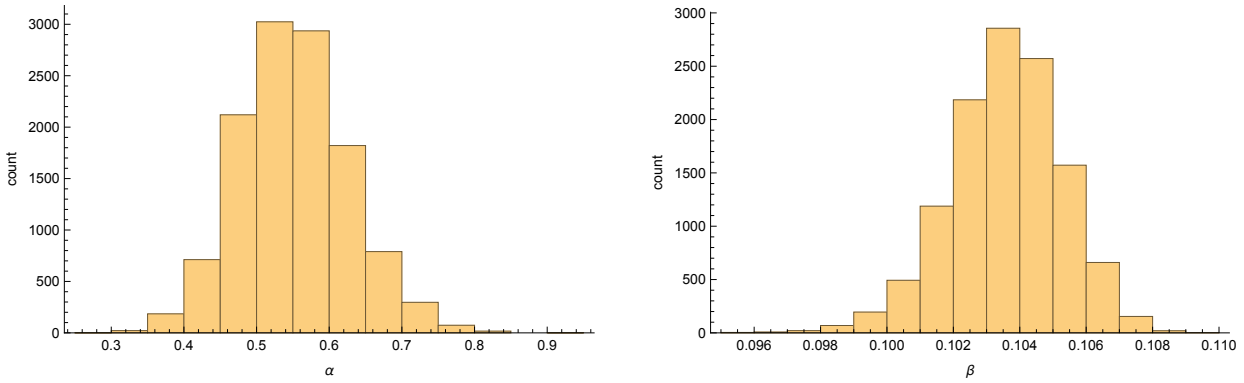

Figure 17.11: Homogeneous model estimates of (α, β).

Problem 17.3.3. By graphing the data, or otherwise, comment on the assumption of a common (α, β) across all replicates.

There is quite a clear variability between the replicates across different experiments (Figure 17.12). This makes the assumption of common parameter values across all replicates look quite weak. An alternative here would be to do some posterior predictive checks, but this isn't really needed here to be honest since the raw data plots are illuminating.

Problem 17.3.4. Now estimate a model that estimates separate values for (α, β) across all replicates. Graph the posterior distribution for each parameter.

```
parameters {
  real<lower=0, upper=2> theta[N, 2];
  real<lower=0> sigma;
  real<lower=0, upper=10> y0[1];
}

model {
  real y_hat[T, 1];
  sigma ~ cauchy(0, 1);
  y0 ~ normal(5, 2);

  for (i in 1:N){
    theta[i] ~ normal(0, 2);
    y_hat = integrate_ode(bacteria_deriv, y0, t0, ts, theta[i], x_r, x_i);
    for (t in 1:T)
      y[t, i] ~ normal(y_hat[t, 1], sigma);
  }
```

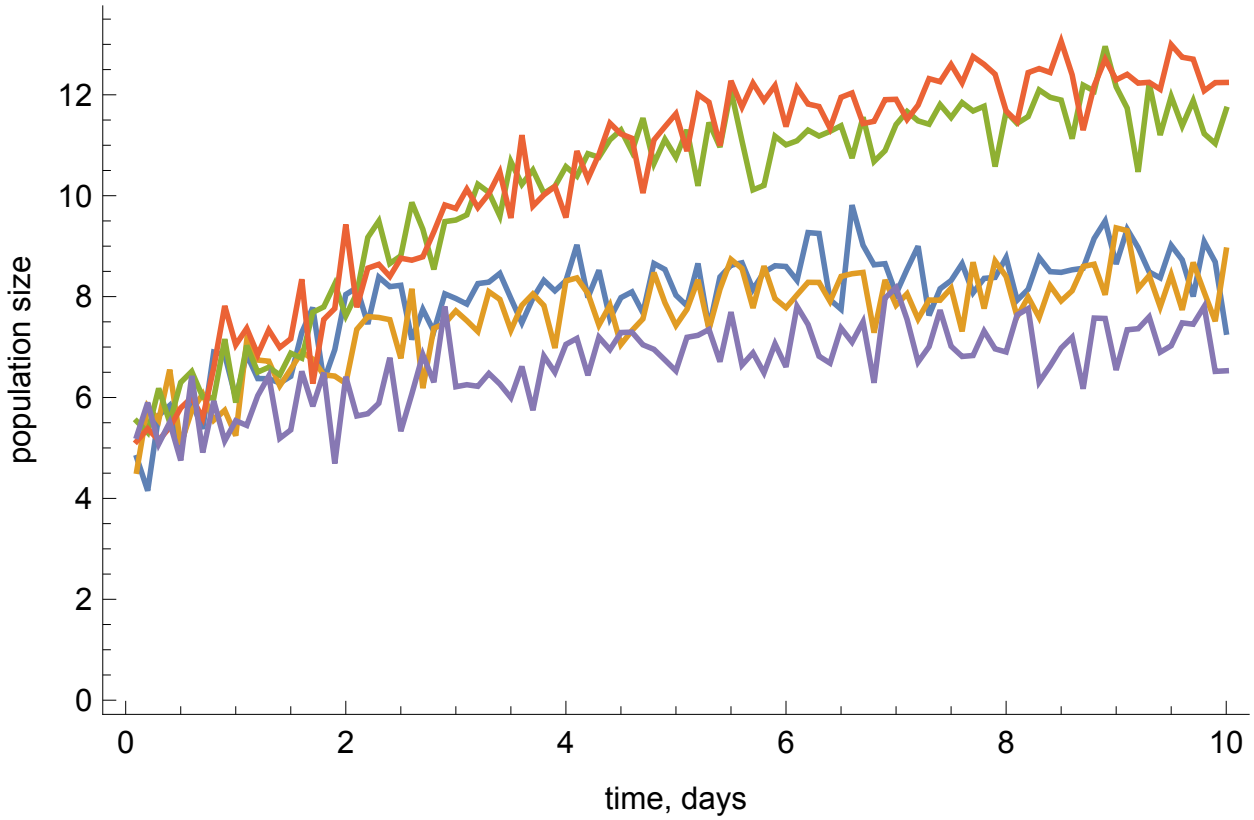

Figure 17.12: Time series plot of 5 experimental replicates.

```
}

generated quantities {
  vector[N * T] logLikelihood;
  int k;
  real y_hat[T, 1];

  k = 1;
  for (i in 1:N){
    y_hat = integrate_ode(bacteria_deriv, y0, t0, ts, theta[i], x_r, x_i);
    for (t in 1:T){
      logLikelihood[k] = normal_log(y[t, i],y_hat[t, 1], sigma);
      k = k + 1;
    }
  }
}
```

There is considerable heterogeneity in posterior estimates of (α, β) (Figure 17.13)

Problem 17.3.5. Estimate a hierarchical model assuming the following priors:

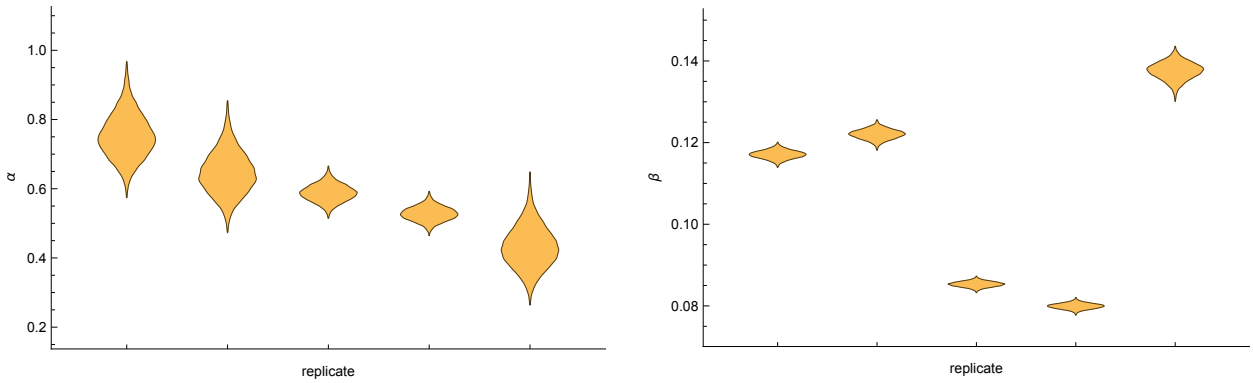

Figure 17.13: Posterior parameter estimates for heterogeneous model.

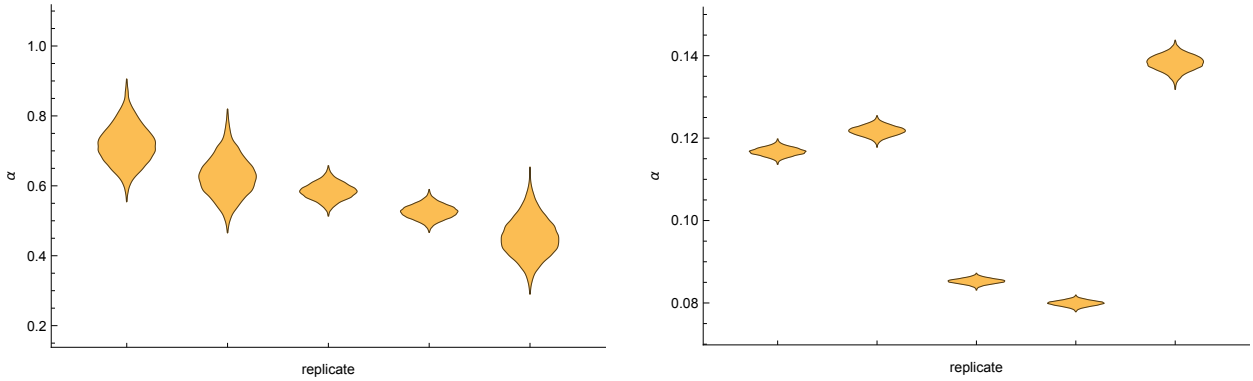

Figure 17.14: Posterior parameter estimates for hierarchical model.

$$
\begin{aligned}
\alpha &\sim \Gamma(a, b) \\
\beta &\sim \Gamma(c, d) \\
a &\sim \mathcal{N}(20, 5) \\
b &\sim \mathcal{N}(40, 5) \\
c &\sim \mathcal{N}(10, 3) \\
d &\sim \mathcal{N}(100, 5)
\end{aligned}
$$

Compare your estimates of (α, β) with those from the completely heterogeneous model.

```
functions {
  real[] bacteria_deriv(real t, real[] y, real[] theta, real[] x_r, int[] x_i) {
    real dydt[1];

    dydt[1] = theta[1] * y[1] * (1 - theta[2] * y[1]);
    return dydt;
```

```
  }
}

data {
  int<lower=1> T;
  int<lower=0> N;
  real t0;
  real ts[T];
  matrix[T, N] y;
}

transformed data {
  real x_r[0];
  int x_i[0];
}

parameters {
  real<lower=0> a1[2];
  real<lower=0> a2[2];
  real<lower=0, upper=2> theta[N, 2];
  real<lower=0> sigma;
  real<lower=0, upper=10> y0[1];
}

model {
  real y_hat[T, 1];
  a1[1] ~ normal(20, 5);
  a1[2] ~ normal(40, 5);
  a2[1] ~ normal(10, 3);
  a2[2] ~ normal(100, 5);
  sigma ~ cauchy(0, 1);
  y0 ~ normal(5, 2);

  for (i in 1:N){
    theta[i, 1] ~ gamma(a1[1], a1[2]);
    theta[i, 2] ~ gamma(a2[1], a2[2]);
    y_hat = integrate_ode(bacteria_deriv, y0, t0, ts, theta[i], x_r, x_i);
    for (t in 1:T)
      y[t, i] ~ normal(y_hat[t, 1], sigma);
  }
}

generated quantities {
  vector[N * T] logLikelihood;
  int k;
  real y_hat[T, 1];
  real aTheta[2];
```

```
  real y_hat_overall[T, 1];

  aTheta[1] = gamma_rng(a1[1], a1[2]);
  aTheta[2] = gamma_rng(a2[1], a2[2]);
  y_hat_overall = integrate_ode(bacteria_deriv, y0, t0, ts, aTheta, x_r, x_i);
  k = 1;
  for (i in 1:N){
    y_hat = integrate_ode(bacteria_deriv, y0, t0, ts, theta[i], x_r, x_i);
    for (t in 1:T){
      logLikelihood[k] = normal_log(y[t, i], y_hat[t, 1], sigma);
      k = k + 1;
    }
  }
}
```

There is very limited shrinkage versus the purely heterogeneous model (Figure 17.13 versus Figure 17.14.) This is because there is quite a lot of data for each replicate.

Problem 17.3.6. Estimate the overall (α, β) for the hierarchical model. How do these compare to the pooled model estimates?

The estimates reflect greater uncertainty compared to the pooled model (Figure 17.15 versus Figure 17.11). This is desirable since the pooled model understates uncertainty.

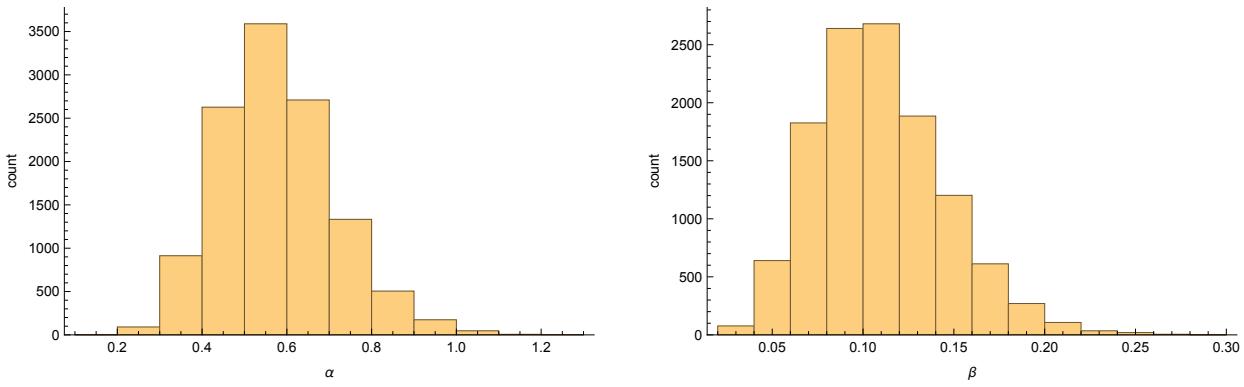

Figure 17.15: Overall parameter estimates for the posterior parameters in the hierarchical model.

Problem 17.3.7. By holding out one of your datasets, compare the predictive performance of each model.

This will favour the hierarchical model (the pooled model performs considerably worse using naive estimates of model performance.)

17.4 Bowel cancer model selection

The file `hierarchical_cancer.csv` contains (fictitious) data on the population size of a given county (N) and the number of bowel cancer cases in that county (X). In this question we aim to build a model to estimate the underlying rate of cancer occurrence λ.

Problem 17.4.1. A simple model is to assume that cancer occurrence is an independent event, and hence we use the following model,

$$X_i \sim Poisson(N_i\lambda) \tag{17.20}$$

where N_i is the population in county i, and X_i is the number of cases of bowel cancer in the same county. In Stan write a model to estimate the underlying rate of bowel cancer occurrence (λ), where we assume a prior of the form $\lambda \sim \mathcal{N}(0.5, 0.5)$.

```
data{
  int K;
  vector[K] N;
  int X[K];
}

parameters{
  real<lower=0> lambda;
}
model{
  X ~ poisson(lambda * N);
  lambda ~ normal(0.5, 0.5);
}
```

which should estimate $\lambda \approx 0.1$.

Problem 17.4.2. Using the `generated quantities` block record the estimated log-likelihood of each data point, for each posterior sample of λ.

```
generated quantities{
  vector[K] lLoglikelihood;
  for(i in 1:K)
    lLoglikelihood[i] = poisson_lpmf(X[i] | N[i] * lambda);
}
```

Problem 17.4.3. By using Stan's `optimizing` function to obtain the MAP estimate of λ, estimate the expected log pointwise predictive density (elpd) via a DIC method,

$$\widehat{\text{elpd}} = log\ p(X|\hat{\theta}_{Bayes}) - \underbrace{2V_{s=1}^{S} log\ p(X|\theta_s)}_{DIC} \tag{17.21}$$

where $V_{s=1}^{S} log\ p(X|\theta_s)$ is the variance in log-likelihood for all data points across S posterior draws. Hint: the latter part of the formula requires that we estimate the model by sampling.

The MAP estimates of the model log-likelihood can be determined using,

```
bFit <- optimizing(aModel, data=list(X=X, N=N, K=length(N)))
likelihoodBayes <- sum(bFit$par[2:1001])
```

Then estimating the model by sampling we can then obtain the p_{DIC} term,

```
fit <- sampling(aModel, data=list(X=X, N=N, K=length(N)), iter=200, chains=4)
lLoglikelihood <- extract_log_lik(fit, 'lLoglikelihood')
aLogLikelihood <- rowSums(lLoglikelihood)
pDIC <- 2 * var(aLogLikelihood)
```

which we then use to estimate $\widehat{\text{elpd}}$,

```
aDIC <- likelihoodBayes - pDIC
```

which should be $\approx$ -3996.

Problem 17.4.4. Estimate elpd using the AIC method. Hint: use Stan's `optimizing` function where the Stan file has had the prior commented out, to achieve the maximum likelihood estimate of the log-likelihood.

The AIC method penalises the estimated log-likelihood by one since there is only a single parameter in the model. Hence this estimate can be obtained by using,

```
bModel <- stan_model('poissonCancerML.stan')
bFit <- optimizing(bModel, data=list(X=X, N=N, K=length(N)))
likelihoodML <- sum(bFit$par[2:1001])
aAIC <- likelihoodML - 1
```

which should yield $\approx$ -3996 (it's slightly higher than that obtained from the DIC method).

Problem 17.4.5. Either manually or using the "loo" package in R estimate the elpd by a WAIC method. If you choose the manual method, this can be done with the following formula,

$$\widehat{\text{elpd}} = \underbrace{\sum_{i=1}^{N} log\left(\frac{1}{S}\sum_{s=1}^{S} p(X_i|\theta_s)\right)}_{\text{log pointwise predictive density}} - p_{WAIC} \tag{17.22}$$

where $p_{WAIC} = \sum_{i=1}^{N} V_{s=1}^{S} var_{post}\left[log\ p(X_i|\theta_s)\right]$.

Using the loo package this is quite straightforward,

```
library(loo)
lLoglikelihood <- extract_log_lik(fit, 'lLoglikelihood')
aWAIC <- waic(lLoglikelihood)
```

which should yield a value of ≈ -3999.

Alternatively, doing this manually,

```
library(matrixStats)
bWAIC_1 <- sum(sapply(seq(1, 1000, 1),
               function(i) logSumExp(lLoglikelihood[, i]) - log(400)))
bWAIC_p <- sum(sapply(seq(1, 1000, 1),
               function(i) var(lLoglikelihood[, i])))
bWAIC <- bWAIC_1 - bWAIC_p
```

where I have used `logSumExp` because it is more numerically stable than doing the piecewise application of the exponential. This should yield a value identical to that obtained via loo ≈ -3999.

Problem 17.4.6. By partitioning the data into 10 folds of training and testing sets (where one data point occurs in each testing set once only), estimate the out-of-sample predictive capability of the model. Hint 1: in R use the "Caret" package's `createFolds` to create 10 non-overlapping folds. Hint 2: adjust your Stan program to calculate the log-likelihood on the test set.

In R the folds can be created by,

```
lFolds <- createFolds(X)
```

The Stan program can be changed to the below which allows the estimation of out-of-sample predictive capability using,

```
data{
  int KTrain;
  vector[KTrain] NTrain;
  int XTrain[KTrain];

  // hold out set
  int KTest;
  vector[KTest] NTest;
  int XTest[KTest];
}

parameters{
  real<lower=0> lambda;
}

model{
  XTrain ~ poisson(lambda * NTrain);
  lambda ~ normal(0.5, 0.5);
```

```
}

generated quantities{
  vector[KTest] lLoglikelihood;
  for(i in 1:KTest)
    lLoglikelihood[i] = poisson_lpmf(XTest[i] | NTest[i] * lambda);
}
```

Then we create a loop in R that stores the log-likelihood for each fold,

```
vLogLikelihood <- vector(length=10, mode='list')
for(i in 1:10){
  print(i)
  aFold <- lFolds[[i]]
  XTrain <- X[-aFold]
  NTrain <- N[-aFold]
  XTest <- X[aFold]
  NTest <- N[aFold]
  aFit <- sampling(aModel, data=list(XTrain=XTrain,
                                     NTrain=NTrain,
                                     KTrain=length(NTrain),
                                     XTest=XTest,
                                     NTest=NTest,
                                     KTest=length(NTest)),
                                     iter=200, chains=4)
  vLogLikelihood[[i]] <- extract_log_lik(aFit, 'lLoglikelihood')
}
```

From which we can estimate the elpd by,

```
aLogTotal <- 0
for(i in 1:10){
  aLogLikeTemp <- vLogLikelihood[[i]]
  aLogTotal <- aLogTotal + sum(colMeans(aLogLikeTemp))
}
```

which should yield $\approx$ -3997. So in this case all measures look reasonably close to the value obtained by cross-validation.

Problem 17.4.7. A colleague suggests fitting a negative binomial sampling model to the data, in case over-dispersion exists. Using a prior $\kappa \sim log - \mathcal{N}(0, 0.5)$ on the dispersion parameter, change your Stan model to use this distribution, and estimate the out-of-sample predictive density using any of the previous methods. Which model do you prefer? Hint: use Stan's `neg_binomial_2` function to increment the log-probability.

The new Stan program should look something like,

```
data{
  int KTrain;
  vector[KTrain] NTrain;
  int XTrain[KTrain];

  // hold out set
  int KTest;
  vector[KTest] NTest;
  int XTest[KTest];
}

parameters{
  real<lower=0> lambda;
  real<lower=0> kappa;
}

model{
  XTrain ~ neg_binomial_2(lambda * NTrain, kappa);
  lambda ~ normal(0.5, 0.5);
  kappa ~ lognormal(0, 0.5);
}

generated quantities{
  vector[KTest] lLoglikelihood;
  for(i in 1:KTest)
    lLoglikelihood[i] = neg_binomial_2_lpmf(XTest[i] | NTest[i] * lambda,
                                            kappa);
}
```

And via manual cross-validation (the best method available here) I obtain an estimated elpd $\approx$ -2731. So the negative binomial model is a significantly better fit to the data on the face of it. However, to do this comparison correctly it is necessary to do pairwise comparison, which takes the variability in log likelihood into account, then compares a z score with a standard normal,

```
z <- sum(lLoglikelihood1 - lLoglikelihood2) / (sqrt(length(lLoglikelihood1)) *
         sd(lLoglikelihood1 - lLoglikelihood2))
p <- 1 - pnorm(z)
```

which will be tiny here.

Problem 17.4.8. A straightforward way to estimate the marginal likelihood is to use,

$$p(X) \approx \frac{1}{S} \sum_{s=1}^{S} p(X|\theta_s) \tag{17.23}$$

where $\theta_s \sim p(\theta)$. Either using Stan's `generated quantities` block or otherwise estimate the

marginal likelihood of the Poisson model. (Hint: if you use Stan then you need to use `log_sum_exp` to marginalise the sampled log probabilities.)

In Stan this can be done using the following block,

```
generated quantities{
  real loglikelihood;
  real<lower=0> lambda1;
  lambda1 = normal_rng(0.5, 0.5);
  while(lambda1 <= 0)
    lambda1 = normal_rng(0.5, 0.5);
  loglikelihood = 0;
  for(i in 1:K)
    loglikelihood = loglikelihood + poisson_lpmf(X[i] | lambda1 * N[i]);
}
```

Note in the above we are using samples from the prior **not** the posterior. The above is just using Stan like a random number generator. We can then estimate the marginal likelihood by doing the following in R,

```
library(matrixStats)
lMarginalLog <- extract(fit, 'loglikelihood')[[1]]
logSumExp(lMarginalLog)
```

which should yield a value $\approx$ -3991. However there is a large variance in this estimator for more complex models.

Problem 17.4.9. Estimate the marginal likelihood of the negative binomial model, and hence estimate the log Bayes Factor. Which model do you prefer?

This is best computed on the log scale (using the output from logSumExp), and should yield something like $\log BF \approx 1200$ in favour of the negative binomial model.

Chapter 18

Linear regression models

18.1 Crime and punishment

The data in `linearRegression_crimePunishment.csv` contains the murder rate per capita and the rate of automobile crimes per 100,000 individuals (both on the log scale) in the ten US States that have changed their legislation on capital punishment since 1960 (in all cases the states abolished capital punishment). We also include a dummy variable ("law") that is 1 if the state allows capital punishment in that year, and 0 otherwise. The crime data is from `http://www.disastercenter.com`.

Problem 18.1.1. Graph the data and comment on any trends.

The data is shown in Figures 18.1 and 18.2. There seems to be some association between the murder rate and auto crimes. In all cases it is difficult to visually discern an impact of the change in legislation.

Problem 18.1.2. A simple model for murder rates is of the form,

$$murder_{i,t} \sim \mathcal{N}\left(\alpha + \beta penalty_{i,t} + \gamma car_{i,t}, \sigma\right) \tag{18.1}$$

where we assume that the effect of having the death penalty is given by β, which is assumed to be the same across all states. We include $car_{i,t}$ – a measure of crimes on automobiles, as an independent variable to proxy for the contemporaneous underlying level of crime. Estimate this model and hence determine whether the death penalty acts as a deterrent to murder.

This model can be estimated using the following Stan code:

```
data{
  int N;
  int K;
  real murder[N];
  real car[N];
  int<lower=0, upper=1> law[N];
```

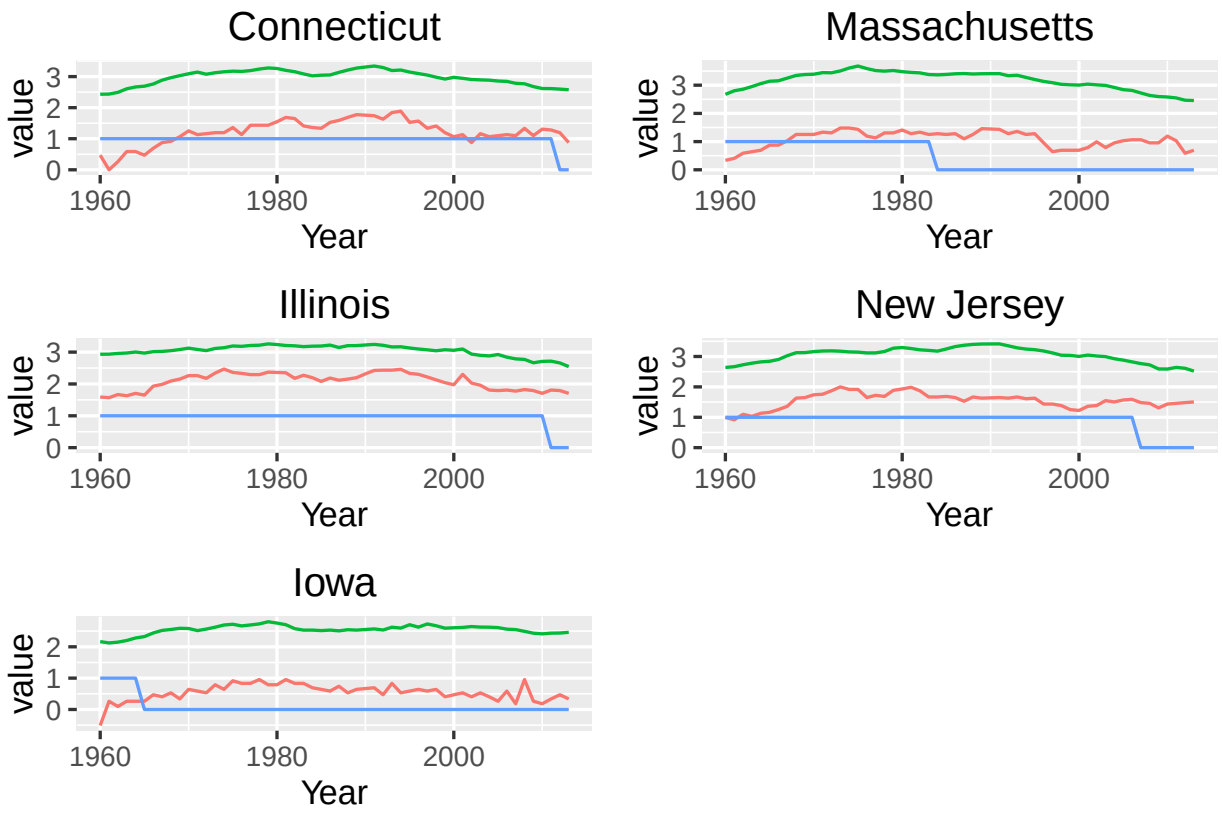

Figure 18.1: The log per capita murder rate (red) and auto crime rate (divided by 2 on log scale; green) versus the death penalty dummy (1 if state has death penalty; 0 otherwise; blue) in five of the US States which abolished the death penalty since 1960.

```
  int state[N];
}

parameters{
  real alpha;
  real beta;
  real gamma;
  real<lower=0> sigma;
}

model{
  for(i in 1:N)
    murder[i] ~ normal(alpha + beta * law[i] + gamma * car[i], sigma);

  alpha ~ normal(0, 1);
  beta ~ normal(0, 1);
  gamma ~ normal(0, 1);
  sigma ~ normal(0, 1);
}
```

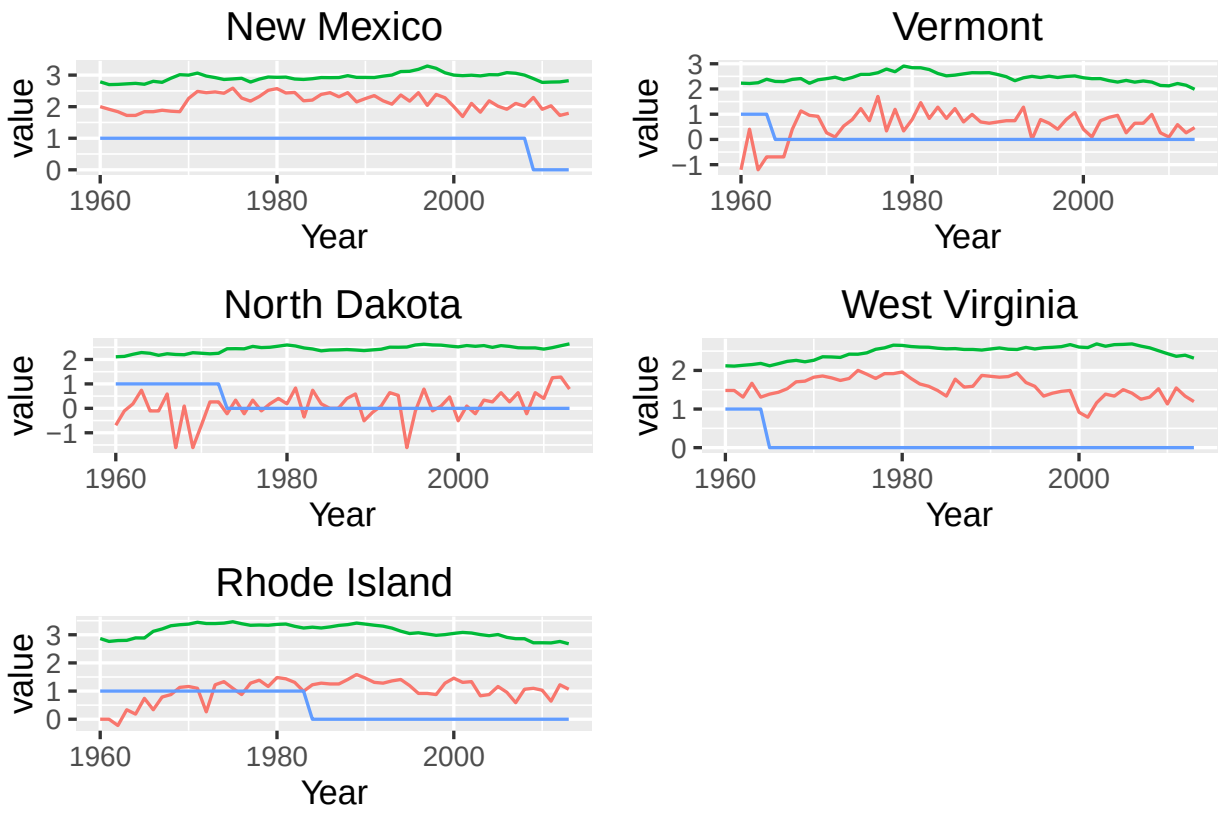

Figure 18.2: The log per capita murder rate (red) and auto crime rate (divided by 2 on log scale; green) versus the death penalty dummy (1 if state has death penalty; 0 otherwise; blue) in five of the US States which abolished the death penalty since 1960.

which when we run we obtain the following results,

```
## Print summary statistics
print(fit,probs = c(0.25, 0.5, 0.75))


mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha -1.48    0.02 0.19 -1.83 -1.62 -1.48 -1.35 -1.07   105 1.01
beta   0.24    0.00 0.06  0.10  0.20  0.24  0.27  0.34   219 1.00
gamma  0.45    0.00 0.04  0.38  0.43  0.45  0.48  0.52   105 1.01
sigma  0.62    0.00 0.02  0.59  0.61  0.62  0.64  0.66   345 0.99
```

Where we see that we estimate that the imposition of the death penalty on average *raises* the murder rate by on average 24%!

Problem 18.1.3. An alternative model allows there to be state-level effects,

$$murder_{i,t} \sim \mathcal{N}\left(\alpha_i + \beta_i penalty_{i,t} + \gamma_i car_{i,t}, \sigma_i\right), \tag{18.2}$$

where we assume that $\alpha_i \sim \mathcal{N}(\bar{\alpha}, \sigma_\alpha)$, $\beta_i \sim \mathcal{N}(\bar{\beta}, \sigma_\beta)$ and $\gamma_i \sim \mathcal{N}(\bar{\gamma}, \sigma_\gamma)$ (we assume fully heterogeneous estimates for σ). Estimate the above model and compare the results with the homogeneous

coefficient model.

This model can be estimated using the following code,

```
data{
  int N;
  int K;
  real murder[N];
  real car[N];
  int<lower=0, upper=1> law[N];
  int state[N];
}

parameters{
  real alpha[K];
  real beta[K];
  real gamma[K];
  real<lower=0> sigma[K];
  real alpha_top;
  real<lower=0> alpha_sigma;
  real beta_top;
  real<lower=0> beta_sigma;
  real gamma_top;
  real<lower=0> gamma_sigma;
}

model{
  for(i in 1:N)
    murder[i] ~ normal(alpha[state[i]] + beta[state[i]] * law[i]
                       + gamma[state[i]] * car[i], sigma[state[i]]);

  alpha ~ normal(alpha_top, alpha_sigma);
  beta ~ normal(beta_top, beta_sigma);
  gamma ~ normal(gamma_top, gamma_sigma);
  alpha_top ~ normal(0, 1);
  beta_top ~ normal(0, 1);
  gamma_top ~ normal(0, 1);
  alpha_sigma ~ normal(0, 1);
  beta_sigma ~ normal(0, 1);
  gamma_sigma ~ normal(0, 1);
  sigma ~ normal(0, 1);
}

generated quantities{
  real alpha_average;
  real beta_average;
  real gamma_average;
```

```
  alpha_average = normal_rng(alpha_top, alpha_sigma);
  beta_average = normal_rng(beta_top, beta_sigma);
  gamma_average = normal_rng(gamma_top, gamma_sigma);
}
```

which when we estimate the above yields,

```
## Print summary statistics
print(fit,probs = c(0.25, 0.5, 0.75))


mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha_average -1.31    0.06 1.09 -3.27 -2.00 -1.37 -0.72  1.09   347 1.01
beta_average  -0.26    0.02 0.31 -0.90 -0.45 -0.27 -0.08  0.38   400 1.00
gamma_average  0.46    0.01 0.18  0.06  0.36  0.47  0.56  0.78   400 0.99
```

with a mean effect size of a 26% reduction in murder rates although with a much wider range of effects.

Problem extra. (Not in main text but wanted to include) Another model allows there to be time trends in the data,

$$murder_{i,t} \sim \mathcal{N}\left(\alpha_i + \delta_i t + \beta_i penalty_{i,t} + \gamma_i car_{i,t}, \sigma_i\right), \tag{18.3}$$

where $\delta_i \sim \mathcal{N}(\bar{\delta}, \sigma_\delta)$. Again estimate this model and compare the effect size of the death penalty across the three models.

This model is only a slight modification of the above one and yields an estimated effect of,

```
## Print summary statistics
print(fit,probs = c(0.25, 0.5, 0.75))


mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha_average -1.76    0.06 1.04 -3.66 -2.44 -1.74 -1.07  0.30   348 1.00
beta_average  -0.18    0.01 0.26 -0.80 -0.32 -0.16 -0.04  0.35   331 1.00
gamma_average  0.54    0.01 0.15  0.19  0.45  0.55  0.63  0.84   400 1.01
```

And so an estimated reduction in the murder rate by 18% on average.

Problem 18.1.4. Compare the predictive fit of the models using the estimated leave-one-out cross-validation from the "loo" package. Which of the three models do you prefer? Hence conclude as to whether the death penalty acts as a deterrent of murder.

This can be done by recording the log likelihood of each data point in the `generated quantities` block. For example for the homogeneous coefficient model,

```
generated quantities{
  real logLikelihood[N];
  for(i in 1:N)
    logLikelihood[i] = normal_lpdf(murder[i] | alpha + beta * law[i] +
                                   gamma * car[i], sigma);
}
```

If we do this for all three models we obtain an estimated expected log (pointwise) likelihood from each using the `loo` function,

- Homogeneous: -512.2
- State-level, no time trend: -57.7
- State-level, with time trend: -3.9

Which when using the "compare" function we see that the best-fitting model is the State-level models with a time trend. In this model the estimated effect of the death penalty is to decrease the murder rate by 18% on average, with a 50% credible interval of $4\% \leq effect \leq 32\%$. So even though on average we see that there is quite a strong effect of the law, we are quite uncertain as to its size in a given state.

Problem 18.1.5. Critically evaluate the best-performing model and hence any conclusions that can be drawn from this analysis.

A criticism is that we have failed to include other omitted factors that may affect the murder rate but are also correlated with the abolishment of the death penalty. It is quite possible that contained within α_i there are factors that may affect both the murder rate and be correlated with the redaction of the law. A potential improvement would be to use first-differences regression (or fixed effects),

$$\Delta murder_{i,t} \sim \mathcal{N}\left(\delta_i \Delta t + \beta_i \Delta penalty_{i,t} + \gamma_i \Delta car_{i,t}, \sigma_i\right), \tag{18.4}$$

where Δ signifies the first difference of the variable. At least in the above model we have removed any individual (time-invariant) heterogeneity from affecting our estimates of the deterrent size. In fact, if we do estimate a model similar to the one above we do not see such any significant impact of the legislation (results not shown).

An alternative approach would be to include other factors in the model that explain α_i.

Chapter 19

Generalised linear models and other animals

19.1 Seatbelts

The file `glm_seatbelts.csv` contains data on the monthly total of car drivers killed (on a log 10 scale) in Great Britain between January 1969 and December 1984, see:

https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/UKDriverDeaths.html

It also contains a measure of petrol prices over the same period, as well as a variable that represents the month on a scale of 1-12.

During the period for which the data runs there was a change in the law that meant it became a legal requirement to wear seatbelts in cars. In this question we are going to estimate when this event occurred by examining the data.

Problem 19.1.1. Plot the data. Can you see by eye when the legislation was likely enacted?

It looks like there is a structural break in the series around 1983 (see Figure 19.1), which happens to be when the law was enacted (at the end of January that year).

Problem 19.1.2. A model is proposed of the form,

$$deaths(t) \sim \mathcal{N}\left(\alpha + \beta petrol(t) + \sum_{i=1}^{11} \delta_i D(i,t) + Gamma(t,s), \sigma\right) \tag{19.1}$$

where,

$$\gamma = \begin{cases} 0, & \text{if } t < s \\ \gamma_0, & \text{if } t \geq s, \end{cases} \tag{19.2}$$

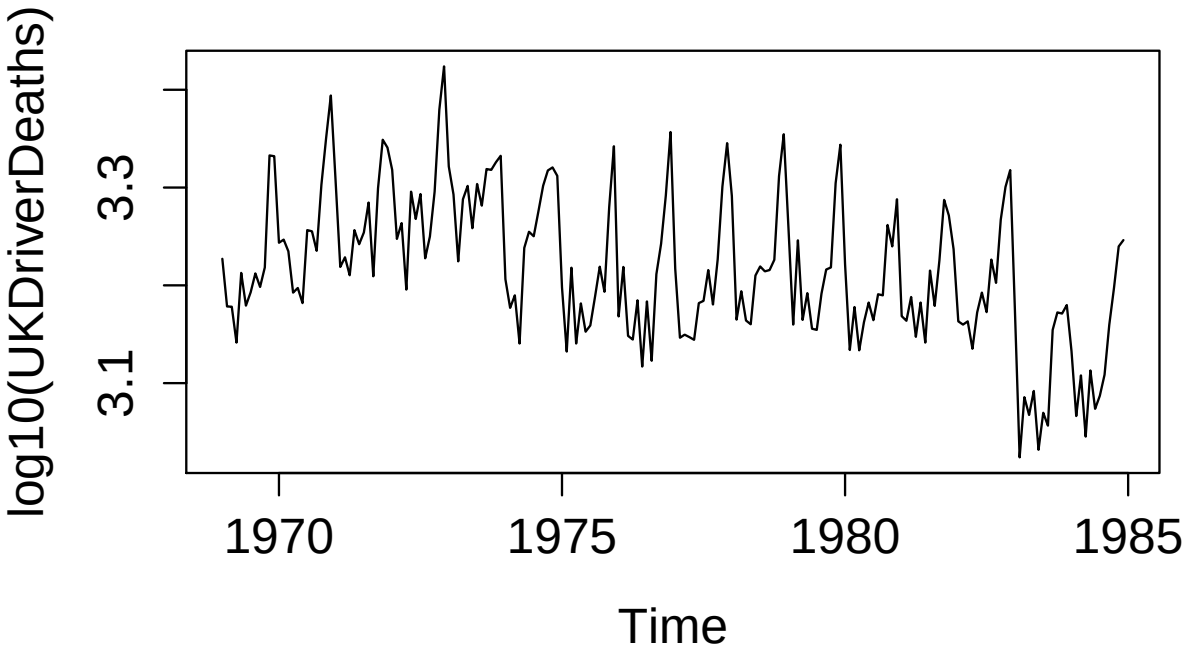

Figure 19.1: The number of car drivers killed in car accidents in Great Britain.

and $\gamma_0 < 0$ represents the effect of the seatbelt legislation on the numbers of car drivers killed after some implementation date s; $D(i,t)$ is a dummy variable for month i equal to 1 if and only if the date t corresponds to that month, and is zero otherwise.

Implement the above model in Stan, and hence estimate the effect that the seatbelt legislation had on car driver deaths.

This can be done with the following code,

```
functions{
  // function that returns a dummy variable from 1-11 if in that month.
  // For december return zero to avoid falling into the dummy variable trap
  real dummySelector(int aMonth, real[] dummies){
    real aDummy;
    if(aMonth < 12){
      return(dummies[aMonth]);
    }else{
      return(0.0);
    }
  }
}
```

```
data{
  int N;
  real deaths[N];
  real petrol[N];
  int month[N];
}

transformed data{
  real log_unif;
  log_unif = - log(N);
}

parameters{
  real beta;
  real delta[11];
  real alpha;
  real<upper=0> gamma;
  real<lower=0> sigma;
}

transformed parameters{
  vector[N] lp;

  // discrete uniform prior on s
  lp = rep_vector(log_unif, N);
  for(s in 1:N)
    for(t in 1:N)
      lp[s] = lp[s] + normal_lpdf(deaths[t] | t < s ? (alpha +
              beta * petrol[t] + dummySelector(month[t], delta)) :
              (alpha + beta * petrol[t] +
               dummySelector(month[t], delta) + gamma), sigma);
}

model{
  alpha ~ normal(0, 1);
  beta ~ normal(0, 1);
  delta ~ normal(0, 1);
  gamma ~ normal(0, 1);
  sigma ~ normal(0, 1);

  // marginalise out s
  target += log_sum_exp(lp);
}
```

The median estimate of the effect size (gamma) is around an 8% reduction in car driver deaths.

Problem 19.1.3. Using the `generated quantities` block estimate the date when the legislation

was enacted.

This can be done using the softmax function along with a random sample from a categorical distribution,

```
generated quantities {
  int<lower=1, upper=N> s;
  s = categorical_rng(softmax(lp));
}
```

If we run the code we get an output of the form shown in Figure 19.2, which has a median of January 1983 as hoped.

Figure 19.2: The model estimated date when the seatbelt legislation was enacted.

19.2 Model choice for a meta-analysis

Suppose that the data contained in `glm_metaAnalysis.csv` contains the (fictitious) result of 20 trials of a new drug. In each trial 10 patients with a particular disorder treated with the drug, and the data records the number of individuals cured in each trial.

Problem 19.2.1. Graph the data across all 20 trials. What does this suggest about a potential model to explain the data?

There is much more variability in the data than could be explained using a binomial model with a single θ value.

Problem 19.2.2. Suppose that we have two models that we could use to describe the data,

$$X_i \sim \mathcal{B}(10, \theta), \tag{19.3}$$

or alternatively,

$$X_i \sim beta - binomial(10, a, b), \tag{19.4}$$

where X_i is the number of successes in trial $i \in [1, 20]$. Write two Stan programs to fit each of the above models to the data, and use the estimated LOO-CV (use the "loo" package for R) to choose between the above models. (Assign $\theta \sim beta(1, 1)$ and $a, b \sim \mathcal{N}(2, 5)$ for priors for each model, where a and b are constrained to be positive.)

The code to estimate each model is shown below,

```
data{
  int N;
  int n;
  int X[N];
}

parameters{
  real<lower=0> a;
  real<lower=0> b;
}

model{
  X ~ beta_binomial(n, a, b);
  a ~ normal(2, 5);
  b ~ normal(2, 5);
}

generated quantities{
  vector[N] logLikelihood;
  for(i in 1:N)
    logLikelihood[i] = beta_binomial_lpmf(X[i] | n, a, b);
}
```

and,

```
data{
  int N;
  int n;
  int X[N];
}

parameters{
  real<lower=0,upper=1> theta;
}

model{
  X ~ binomial(n, theta);
  theta ~ beta(1, 1);
}

generated quantities{
  vector[N] logLikelihood;
  for(i in 1:N)
    logLikelihood[i] = binomial_lpmf(X[i] | n, theta);
}
```

If we estimate the above we obtain estimates of the elpd of -49.6 and -45.4 for the binomial and beta-binomial models respectively. Using the "compare" function from "loo" we obtain a difference of 4.2 with a standard error of 3. This has a p value well above the threshold for statistical significance. Therefore By this criterion there is nothing to choose between these models.

Problem 19.2.3. An alternative way to choose between these models is to use Bayes factors. Rather than determine the marginal likelihoods explicitly, this can actually be done in Stan by allowing a discrete model choice parameter $s \in \{1, 2\}$ that dictates which model to use. Code up this model in Stan, and by examining the posterior distribution for $Pr(s)$ determine which sampling distribution fits the data best. (Hint: assign equal probability to each model *a priori* and marginalise out s to obtain the log-probability.)

The following code estimates this model,

```
data{
  int N;
  int n;
  int X[N];
}

parameters{
  real<lower=0> a;
  real<lower=0> b;
  real<lower=0, upper=1> theta;
}
```

```
transformed parameters{
  vector[2] lp;
  for(s in 1:2){
    if(s==1)
      lp[s] = log(0.5) + binomial_lpmf(X | n, theta);
    else
      lp[s] = log(0.5) + beta_binomial_lpmf(X | n, a, b);
  }
}

model{
  target += log_sum_exp(lp);
  theta ~ beta(1, 1);
  a ~ normal(2, 5);
  b ~ normal(2, 5);
}

generated quantities{
  vector[2] lProbs;
  lProbs = exp(lp - log_sum_exp(lp));
}
```

The posterior distribution for the $Pr(s = 2|X)$ has a mean of 0.99. In this case we strongly prefer the beta-binomial model. The two approaches use different criteria to choose. They both tend towards the same answer, that the beta-binomial model is better.

Problem 19.2.4. An alternative approach is to use the binomial likelihood, but use a hierarchical model where each θ_i is drawn from some population-level distribution. Comment on whether you would prefer this approach or the beta-binomial model. (Hint: do not estimate the hierarchical model.)

The hierarchical model is really the same as the beta-binomial case, since the latter is essentially, $X_i \sim \mathcal{B}(10, \theta_i)$ where $\theta_i \sim beta(a, b)$. This is the same as the hierarchical model.

19.3 Terrorism

In this question we are going to investigate the link between the incidence of terrorism and a country's level of income. The data in `glm_terrorism.csv` contains for one hundred countries (those for which the latest data was available) the following series,

- **count**: the number of acts of terrorism perpetrated in each country from 2012 to 2015, as compiled by START [4].
- **gdp**: the gross domestic product of each country in 2015 as compiled by the World Bank.
- **population**: the population of each country in 2015 as compiled by the World Bank.

- **gdpPerCapita**: the GDP per capita in each country.
- **religion**, **ethnic**, **language**: measures of fractionalisation with respect to each of these measures, obtained from: `http://www.anderson.ucla.edu/faculty_pages/romain.wacziarg`.
- **law** and **corruption**: measures of the rule of law and corruption (actually an inverse measure) as compiled by the World Bank in their 2016 World Governance Indicators report.
- **democracy** and **autocracy**: indicators of democracy and autocracy respectively from the polity4 database.
- **region** and **region_numeric**: the region to which a country belongs out of Asia, Europe, Middle East and North Africa, Sub-Saharan Africa, South America, and North America.

Problem 19.3.1. Graph the data. What does this tell you about the processes?

Using the `pairs` plotting function in R where we have logged the count, GDP, and population we obtain Figure 19.3. From this plot it is clear that there is a fairly strong relationship between terror count and population size. Otherwise it is hard to see any strong associations with the terror variable although perhaps there is a negative correlation between terrorism and rule of law and the corruption variable (actually an inverse measure of corruption, meaning there is a positive association between corruption and terrorism). Otherwise there is some covariance between GDP per capita and rule of law and corruption (a higher GDP per capita means a lower corruption level). There is similarly strong associations between linguistic and ethnic fractionalisation, and also between rule of law and corruption.

Problem 19.3.2. A simple model for the terrorism count is the following,

$$count_i \sim Poisson(\alpha + \beta_1 population_i + \beta_2 gdpPerCapita_i), \tag{19.5}$$

where i corresponds to one of the countries in our dataset. Code up this model in Stan, and use it to obtain estimates of the effect of a country's income level on the incidence of terrorism.

Problem 19.3.3. Now include corruption, religion and ethnic as further variables in the above generalised linear model. What is the impact of each of these variables on the terrorism count?

Problem 19.3.4. Conduct posterior predictive checks to assess the applicability of the model to the data. What do these tests suggest?

19.4 Eurovision

The data in `Eurovision.csv` contains historical data of the outcome of the Eurovision song contest from 1976 to 2015 for the twenty countries who have featured most consistently in the finals throughout the years. Along with the results from the contest we also include data on the distance between pairs of countries, whether those countries share a common language, and if one was ever colonised by the other. In this question we ask you to develop a model to help explain the way in which countries award points to one another.

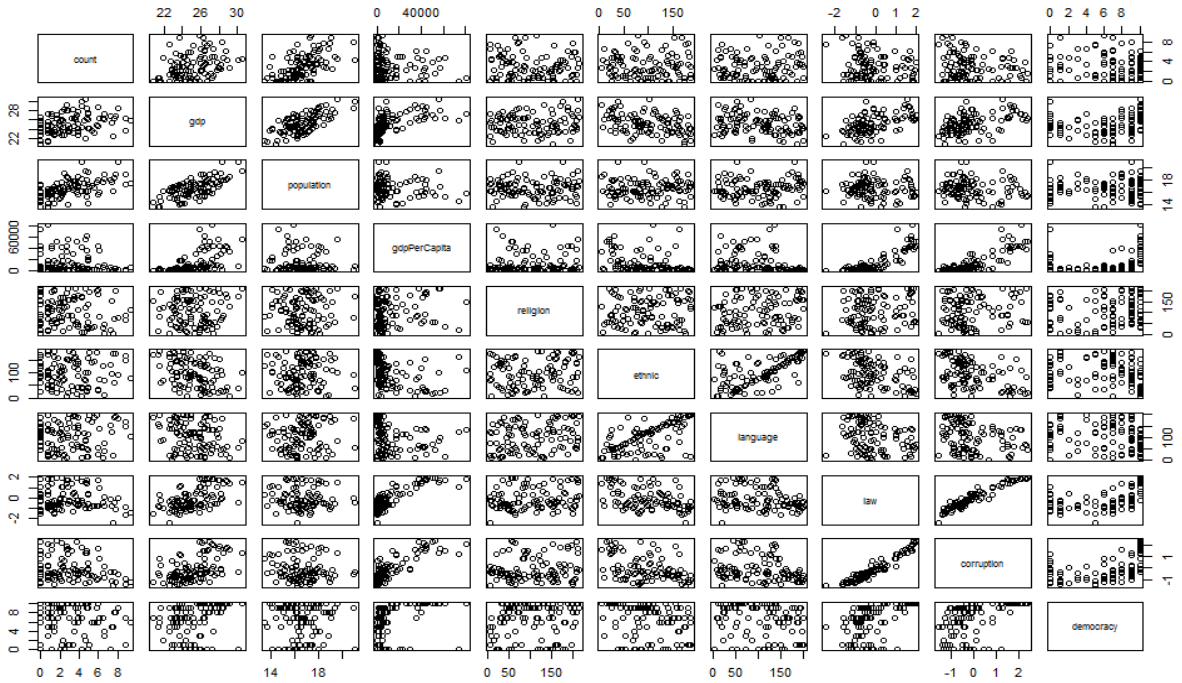

Figure 19.3: Scatter plots of the data for the terrorism example. Note that the count, GDP, and population are logged.

19.5 More terrorism (harder)

The data in `terrorism.csv` contains historical pairwise counts of terrorist attacks perpetrated by citizens of an origin country against a target country, compiled by Alan Krueger see:

http://krueger.princeton.edu/pages/

assembled from the U.S. State Department's annual list of significant international terrorist incidences (PGT). In this question we ask students to develop a model to explain the incidence of such attacks using data on the attributes of each country (the origin and target).

Bibliography

[1] *The World Almanac and Book of Facts*. 1975.

[2] Gregory Belenky, Nancy J Wesensten, David R Thorne, Maria L Thomas, Helen C Sing, Daniel P Redmond, Michael B Russo, and Thomas J Balkin. Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: A sleep dose-response study. *Journal of sleep research*, 12(1):1–12, 2003.

[3] John B Carlin. Meta-analysis for 2× 2 tables: A bayesian approach. *Statistics in medicine*, 11(2):141–158, 1992.

[4] National Consortium for the Study of Terrorism and Responses to Terrorism (START). Global terrorism database. 2016.

[5] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[6] Andrew Gelman et al. Objections to bayesian statistics. *Bayesian Analysis*, 3(3):445–449, 2008.

[7] RG Jarrett. A note on the intervals between coal-mining disasters. *Biometrika*, 66(1):191–193, 1979.

[8] Lawrence Joseph, Theresa W Gyorkos, and Louis Coupal. Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard. *American Journal of Epidemiology*, 141(3):263–272, 1995.

[9] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

[10] David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

[11] Peter F Thall and Stephen C Vail. Some covariance models for longitudinal count data with overdispersion. *Biometrics*, pages 657–671, 1990.