

5

Introduction to the Bayesian Approach to Regression Modelling with Spatial and Spatial-Temporal Data

5.1 Introduction

This chapter provides an important bridge between the material that has been covered so far in the book and the analytical material that follows in Parts II and III. We have three key objectives. The first is to provide the reader with an introduction to Bayesian inference from a theoretical perspective (what do we mean by “Bayesian inference”), from a model building perspective (how do we construct Bayesian models to tackle the problem in hand) and from a computational perspective (how do we implement/fit a Bayesian model using WinBUGS). The second objective is to demonstrate how to apply Bayesian inference to analyse spatial data of the type that we illustrated in the examples in Chapter 1 (Section 1.3.2). Finally, to help fix ideas, the third objective is to provide some illustrative examples using spatial data.

There are three key features of the Bayesian approach that offers itself as an efficient and pragmatic way to statistical inference (i.e. learning about the unknown parameters that we are interested in). First, the Bayesian approach allows us to utilise all sources of information that are available to us (e.g. information from the data that we observe; information from expert opinion; and/or information from previous studies). Second, the Bayesian approach represents various sources of information using probability distributions. Third, the application of Bayes’ theorem combines all the available information for learning (or making inference) about the underlying process and the associated parameters. Bayes’ theorem is a simple theorem concerning conditional probabilities of the form $\Pr(A | B)$ – the probability of an event A occurring given that an event B has occurred. Yet, as we shall see in this chapter, Bayes’ theorem allows us to form a probability distribution (called the posterior distribution) that encapsulates all the available information about the underlying process and the parameters.

This chapter is structured as follows. In Section 5.2 we will, through examples, introduce the three key components of any Bayesian model: the prior distribution, the likelihood function and the posterior distribution. In a Bayesian model, prior distributions need to be assigned to all unknown model parameters. A prior distribution is a probability distribution that represents our knowledge about an unknown parameter before the data are analysed. The data we observe contain another source of information about the unknown parameters, and this information is represented through the likelihood, a probability distribution for the data. We then use Bayes’ theorem to combine the prior information with the likelihood to form the posterior distribution. The posterior distribution can be viewed as the updated knowledge about the unknown parameters in light of the data. Inference

about an unknown parameter is based on its posterior distribution. In Section 5.3, we will discuss some issues associated with summarising the posterior distribution, and we will introduce WinBUGS, a flexible statistical programme to implement Bayesian models. We will discuss the Bayesian implementation of regression models to spatial data in Section 5.4. These regression models will lay the foundation for the more complex spatial and spatial-temporal models that will be described in subsequent chapters. Section 5.5 discusses model comparison and model evaluation, two important elements in any statistical modelling within the Bayesian framework. As we shall see, prior specification is a key element in Bayesian inference and it is particularly so in modelling spatial and spatial-temporal data. In Section 5.6 we look at different prior specifications and in particular “non-informative” and “informative” priors and see how informative priors can be used to express geographical-substantive as well as spatial knowledge (we will draw a distinction), and the nature as well as the source of that knowledge.

5.2 Introducing Bayesian Analysis

5.2.1 Prior, Likelihood and Posterior: What Do These Terms Refer To?

To introduce these three terms and the relationship between them, consider the simple example of learning about the probability of obtaining a head when a coin is flipped once. The quantity of interest is the unknown probability of getting a head from a flip of a given coin and is denoted as θ (the Greek letter theta). To learn about this unknown probability, θ , an obvious way forward is to conduct an experiment by flipping this coin several times to see how many heads we observe. However, *before* carrying out any experiment, we may be able to say something about θ based on either our own belief or knowledge. For example, if the experimenter is a coin expert, then after noting that the coin is a genuine British one pound coin, she forms the belief that the chance of getting a head should be quite close to 50%, say, between 45–55%. However, suppose close inspection of the coin reveals scratches on both sides of the coin which may affect its rotation in the air. The coin expert may now revise her belief, asserting that “ θ is most likely to be 0.5 and is unlikely to be outside the range 0.4 to 0.6.” By contrast, suppose the experimenter knows absolutely nothing about coins, has never seen a British one pound coin before and is sceptical about whether it is genuine. This experimenter may say “ θ will take any value between 0 and 1.” Why 0 and 1? It is because θ is a probability. Both of those two statements about θ are valid,¹ and they both represent beliefs/knowledge about the unknown parameter θ *prior to* seeing the results from the experiment (i.e. the data). Clearly, the first statement implies stronger prior belief (or less prior uncertainty) about θ than the second one.

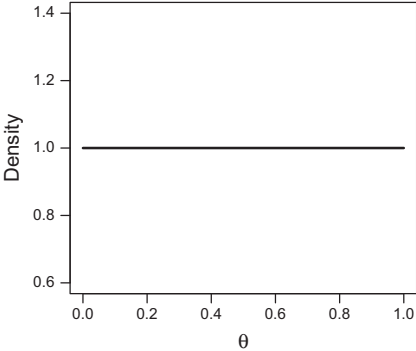
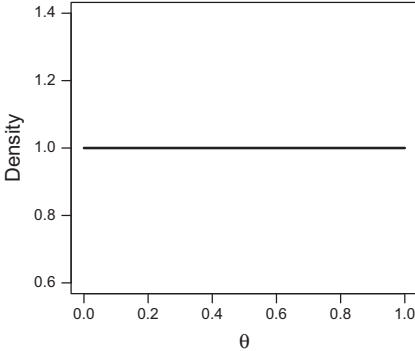
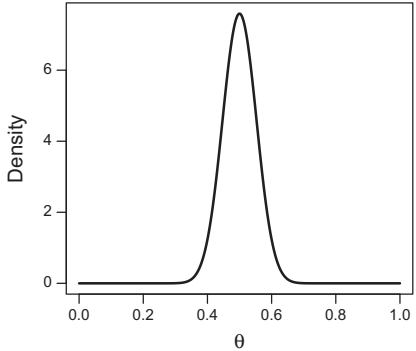
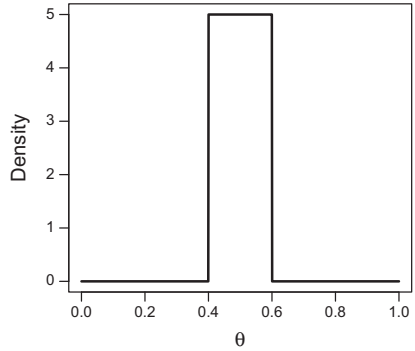
Prior information can also come from knowledge generated from previous experiments. For example, if other experimenters have flipped coins that were produced from the same batch as the one in hand, such results, say 48 heads out of 100 flips, can be used in the current analysis as a form of prior information. We will say more about this in Section 5.6.

In a Bayesian model, the analyst specifies a *prior distribution*, a probability distribution to represent the prior information for an unknown parameter. Hereafter, we will use “a

¹ If you say “ θ will likely be between -1 and 9 ”, then this is not a valid statement about your prior knowledge about θ because θ clearly cannot be negative and cannot go beyond 1!

TABLE 5.1

The Probability Density Functions for the Beta Distribution and the Uniform Distribution with Some Examples

$\theta \sim \text{Beta}(a, b)$	$\theta \sim \text{Uniform}(c, d)$
Probability Density Function $\Pr(\theta a, b) = \frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} \cdot \theta^{a-1} \cdot (1-\theta)^{b-1}$, with θ defined between 0 and 1.	Probability Density Function $\Pr(\theta c, d) = \frac{1}{d-c}$, with θ defined between c and d .
<p style="text-align: center;">Beta(1, 1)</p> 	<p style="text-align: center;">Uniform(0, 1)</p> 
<p style="text-align: center;">Beta(45.5, 45.5)</p> 	<p style="text-align: center;">Uniform(0.4, 0.6)</p> 

prior distribution” and “a prior” interchangeably. For example, to represent the non-expert belief, we can assign either the uniform distribution $\text{Uniform}(0,1)$ or the Beta distribution $\text{Beta}(1,1)$ as a prior for θ . Mathematically, we use the notation $\theta \sim \text{Beta}(1,1)$ to mean that θ follows the $\text{Beta}(1,1)$ distribution. Note that the $\text{Beta}(1,1)$ distribution is equivalent to $\text{Uniform}(0,1)$ in the sense that both distributions present the same prior information, that is, all values between 0 and 1 are assumed to be equally likely for the unknown parameter θ before seeing any data.

Table 5.1 shows the shapes of both distributions as well as their probability density functions. The expert opinion – the parameter θ is most likely to be 0.5 but unlikely to be outside the interval between 0.4 and 0.6 – can be represented using either the $\text{Beta}(45.5, 45.5)$ distribution or the $\text{Uniform}(0.4, 0.6)$ distribution.² As opposed to a flat distribution from the

² The former may capture the expert’s opinion more closely, as the latter implies (a) all values between 0.4 and 0.6 are equally likely and (b) it is impossible for θ to go outside the interval defined by this distribution.

non-expert belief, these two prior distributions are more *informative* about where θ lies: it is more likely to lie between 0.4 and 0.6 than anywhere else (Table 5.1). The two distributions arising from the expert's belief are called *informative priors*, whereas the two distributions arising from the non-expert's belief are called *vague* or *weakly-informative priors*. We will have more to say about prior specification in Section 5.6. Exercise 5.1 shows how to derive $Beta(45.5, 45.5)$ based on the expert's opinion.

The other piece of information about θ comes from data. Suppose an experiment has been carried out in which this coin was flipped 10 times and six heads were observed. This is data. In a Bayesian model, data are considered to be fixed and non-random, but how many heads we observe is governed by the process of flipping this particular coin 10 times. This process itself depends upon the unknown parameter, θ . As a result, data are considered as *realizations* of the process, hence providing some information about the parameter that we want to learn about. In order to link the data to the process and thus to the parameter θ , a *likelihood* function – a probability distribution for describing the data – is specified. For the coin data, a natural choice for the likelihood function is the binomial distribution, which describes the chance of observing y successes ($y = 0, 1, \dots, n$) out of n trials with the outcome of each trial being either a success or a failure. So, the observation $y = 6$ heads is a realization from a binomial distribution $Binomial(\theta, n)$ where θ is the unknown parameter and $n = 10$, the number of flips in the experiment. We write the likelihood as $y \sim Binomial(\theta, n)$, and the probability density function of a binomial distribution is:

$$\Pr(y | \theta, n) = \binom{n}{y} \theta^y (1 - \theta)^{n-y} \quad (5.1)$$

where $\binom{n}{y}$ is the binomial coefficient that computes the number of ways of having y successes out of n trials.

Now we need to combine the two sources of information, the prior information and the information from the experimental data, to form *the posterior distribution*, a probability distribution that contains our updated/current knowledge about the unknown parameter θ . This combination is done using Bayes' theorem. Proposed by Thomas Bayes in 1763 (Bayes, 1763), Bayes' theorem is defined through two events, A and B:

$$\Pr(A | B) = \frac{\Pr(B | A) \times \Pr(A)}{\Pr(B)} \quad (5.2)$$

In the above expression, $\Pr(A | B)$ is a conditional probability of event A given event B. The conditional probability $\Pr(A | B)$ can be seen as the probability of something that we are interested in, i.e. event A occurring (or not) given the information that something else, i.e. event B, has happened. Bayes' theorem allows us to express that conditional probability in the form given on the right-hand side of Eq. 5.2. When dealing with parameters and data, we can consider parameters as event A and data as event B in the sense that parameters are the "something" that we wish to learn about whilst data are the "something" that have been observed. Applying Bayes' theorem, we have

$$\Pr(parameters | data) = \frac{\Pr(data | parameters) \times \Pr(parameters)}{\Pr(data)} \quad (5.3)$$

By “ignoring” the normalizing constant $\Pr(data)$ in the denominator,³ Eq. 5.3 simplifies to:

$$\Pr(parameters | data) \propto \Pr(data | parameters) \times \Pr(parameters), \quad (5.4)$$

where “ \propto ” means “is proportional to”, due to the removal of the normalizing constant. $\Pr(parameters | data)$ is the posterior distribution and is given by the product of $\Pr(data | parameters)$, known as the likelihood function, and $\Pr(parameters)$, the prior distribution. Hence, Eq. 5.4 is often written as

$$posterior \propto likelihood \times prior \quad (5.5)$$

In words, the posterior distribution is proportional to the product of the likelihood and the prior.

Return now to the coin flipping example. Using the prior distribution *Beta* (a, b), where a and b can be replaced by the corresponding numbers (Table 5.1) depending on whether a vague prior or an informative prior is used, the posterior distribution for θ , the probability of getting a head from flipping the coin in question once, is given by

$$\Pr(\theta | y, n, a, b) \propto \left[\binom{n}{y} \theta^y (1-\theta)^{n-y} \right] \times \left[\frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} \cdot \theta^{a-1} (1-\theta)^{b-1} \right] \quad (5.6)$$

The two pairs of square brackets in Eq. 5.6 enclose the binomial likelihood and the Beta prior, respectively. Combining like terms, we have:

$$\Pr(\theta | y, n, a, b) \propto \frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} \cdot \binom{n}{y} \cdot \theta^{y+a-1} (1-\theta)^{(n-y)+(b-1)} \quad (5.7)$$

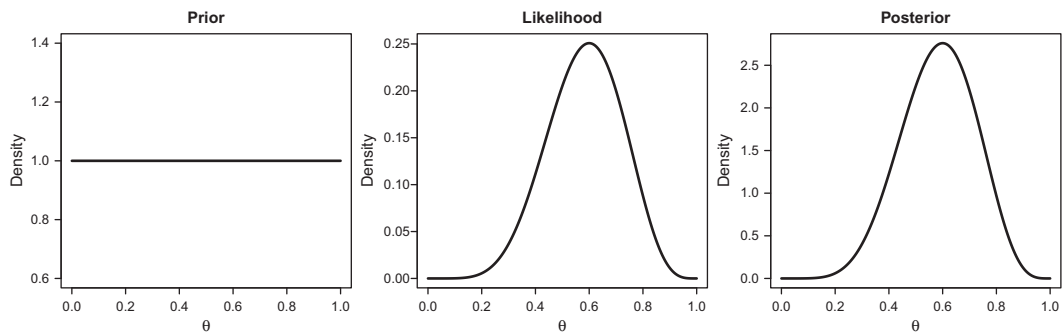
Since the four quantities, a , b , n and y , are known numbers, the term $\frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} \cdot \binom{n}{y}$ is simply a multiplicative constant that does not involve θ . Thus, the posterior distribution for θ can be simplified to

$$\Pr(\theta | y, n, a, b) \propto \theta^{y+a-1} (1-\theta)^{(n-y)+(b-1)} \quad (5.8)$$

Comparing Eq. 5.8 to the probability density function of the Beta distribution in Table 5.1 (ignoring the multiplicative constant $\frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)}$), the posterior distribution for θ is just another Beta distribution with parameters $(y+a)$ and $(n-y+b)$.

When the posterior distribution and the prior distribution are of the same probability distribution (but with different values for the parameters), then the prior distribution is called a *conjugate* prior for the chosen likelihood. Thus, the Beta distribution is a conjugate prior for the binomial likelihood since the resulting posterior distribution is also a Beta distribution. The Uniform distribution, on the other hand, is not a conjugate prior (thus

³ The normalizing constant $\Pr(data)$ ensures the resulting posterior distribution is a proper distribution, namely, integrates to one. This multiplicative constant can be removed since (a) the shape of the posterior distribution is unaffected and (b) WinBUGS, the software that carries out all the required computation of the posterior distribution, does not need to know this constant (WinBUGS will be introduced in Section 5.3).

**FIGURE 5.1**

The prior, the likelihood and the posterior for the coin flip example.

called a *non-conjugate* prior) for the binomial likelihood. A benefit of using a conjugate prior is that the posterior distribution is a known distribution. There are closed-form expressions for various summaries (such as the mean and different percentiles) of the posterior distribution. Using a conjugate prior, as we will see in Chapters 7 and 8, also allows us to gain a better understanding of some spatial models. However, the availability of numerical methods (e.g. Markov chain Monte Carlo) and software (e.g. WinBUGS) allows us to use various forms of prior distributions, conjugate or not, that are appropriate for dealing with the problem in hand. We defer the discussion of Bayesian computation to Section 5.3.

Using the vague prior $Beta(1,1)$ (i.e. $a = 1$ and $b = 1$), and with the data $y = 6$ and $n = 10$, Figure 5.1 shows the prior, the binomial likelihood and the resulting posterior distribution $\theta | y = 6, n = 10, a = 1, b = 1 \sim Beta(7,5)$. The likelihood and the posterior distribution are very similar in shape. That is because, compared to the likelihood, the $Beta(1,1)$ prior contains very little information about θ . As a result, the posterior distribution is largely dominated by the likelihood. In general, when vague priors are used, the results from a Bayesian analysis are similar to those from a frequentist analysis.

To summarise, in Bayesian modelling, we need to specify a likelihood function to link the observed data to a probability model and a set of prior distributions for all the unknown parameters in that model. Using Bayes' theorem, we then combine the likelihood with the priors to form the posterior distribution, a probability distribution for the parameters. Encapsulating information from both the data and the priors, the posterior distribution allows us to learn about the unknown parameters. Before we talk about how to summarise the posterior distribution, we will look at an example of Bayesian regression modelling.

5.2.2 Example: Modelling High-Intensity Crime Areas

This example analyses a set of binary 0/1 outcome values across 337 census output areas (COAs) in Sheffield. These binary outcome values indicate whether a COA was considered as a high-intensity crime area ($= 1$; labelled as a PHIA) or not ($= 0$; labelled as a non-PHIA) based on police perceptions. Here we want to investigate whether ethnic heterogeneity affects the likelihood of being considered as a PHIA, hence a regression type of analysis.

Again, we need to specify a likelihood function for the binary outcome values and a prior distribution for each of the unknown parameters involved. For COA i where $i = 1, \dots, 337$, let y_i be the binary value indicating whether COA i is a PHIA ($y_i = 1$) or not ($y_i = 0$). Each of these binary outcome values can be modelled using a Bernoulli distribution. If a random variable X follows a Bernoulli distribution with parameter π , then X takes the

value 1 with probability π and takes the value 0 with probability $1 - \pi$. So, the likelihood is written as $y_i \sim \text{Bern}(\pi_i)$, where π_i is the probability of COA i being considered as a PHIA. For each COA, we have an index, labelled as $x_{i,\text{ethnic}}$ that quantifies the level of ethnic heterogeneity. This COA-level index of ethnic heterogeneity takes a value between 0 and 1, and the larger the value the greater the ethnic mix in that COA. We want to assess whether this observable covariate on ethnic heterogeneity can be used to explain the COA-level PHIA probabilities. To do that, a logistic regression model can be used, and it specifies a regression relationship as follows:

$$\text{logit}(\pi_i) = \alpha + \beta \cdot x_{i,\text{ethnic}} \quad (5.9)$$

In Eq. 5.9, the logit transformed π_i (i.e. $\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1 - \pi_i}\right)$) is expressed as a function of the covariate on ethnic heterogeneity. The logit transformation is used to ensure $\pi_i = \frac{\exp(\alpha + \beta \cdot x_{i,\text{ethnic}})}{1 + \exp(\alpha + \beta \cdot x_{i,\text{ethnic}})}$ always lies between 0 and 1. In this model, there are two unknown parameters: α , the intercept, and β , the regression coefficient. It is β that we are interested in as it measures the effect of the covariate $x_{i,\text{ethnic}}$ on the outcome. As we do not have any prior information on these two parameters, a typical choice of a vague prior for the intercept and regression coefficient(s) is a normal distribution with mean 0 and a large variance, say, 1000000 (see Section 5.6 for more detail). Therefore, for the prior specification, $\alpha \sim N(0, 1000000)$ and $\beta \sim N(0, 1000000)$. Combining the likelihood with the priors, the (joint) posterior distribution for α and β is

$$\Pr(\alpha, \beta \mid \text{data}) \propto \left[\prod_{i=1}^{337} \pi_i^{y_i} \times (1 - \pi_i)^{1 - y_i} \right] \times \left[e^{-\frac{\alpha^2}{2 \times 1000000}} \right] \times \left[e^{-\frac{\beta^2}{2 \times 1000000}} \right] \quad (5.10)$$

On the right-hand side of Eq. 5.10, the product in the first pair of square brackets gives the likelihood and the following two pairs of square brackets are the priors for the two parameters. Putting the mathematical detail aside, the posterior distribution in Eq. 5.10 is more complicated than that from the coin flip example (i.e. Eq. 5.8) for two reasons. First, Eq. 5.10 is a multivariate probability distribution over multiple parameters. This is often the case in spatial and spatial-temporal models where there are many unknown parameters and hence the posterior distribution is higher dimensional. Second, the posterior distribution in Eq. 5.10 is not a known probability distribution, so we have no known formula to work out, for example, the posterior mean of the regression coefficient, β . In Bayesian regression modelling, determining the posterior summary of unknown parameters is done numerically via Markov chain Monte Carlo. So, to learn about β , we need to discuss the topic of Bayesian computation.

5.3 Bayesian Computation

5.3.1 Summarising the Posterior Distribution

In Section 5.2, we established that all knowledge about an unknown parameter is contained in the posterior distribution. An advantage of the Bayesian approach is, in addition

to reporting a point and an interval estimate of a parameter, we can also construct any probability statements regarding the unknown parameter. We can also provide posterior estimates about any transformation of that parameter. All can be done easily using the posterior distribution.

Typically, we report the posterior mean (i.e. the mean of the posterior distribution) as a point estimate of an unknown parameter and use the 2.5 percentile and the 97.5 percentile of the posterior distribution to form the 95% *credible interval*. For example, based on the posterior distribution $Beta(7,5)$ from the coin example, the posterior mean of θ , the probability of getting a head from a single coin flip, is about 0.58 and the 95% credible interval of θ is (0.31,0.83). These values can be calculated using either known formulae for the Beta distribution (see Exercise 5.2) or through Monte Carlo integration (see Section 5.3.2 later). A 95% credible interval of the form (a,b) corresponds to a probability statement, meaning that there is a 0.95 probability that the unknown parameter falls between the two numbers a and b . This is different from the interpretation of a 95% confidence interval from the frequentist approach. A 95% confidence interval (a,b) implies that if we obtain a large number of datasets through repeating the same experiment many times, then 95% of these datasets would yield values of the unknown parameter that go between the two numbers a and b . A confidence interval is based on the idea of repeated experiments (Section 1.4.2).

Since we have the entire probability distribution, we can also report probability statements such as “what is the chance that the given coin is unbiased if a coin is considered to be unbiased when its probability of getting a head from a single flip is highly likely to be between 0.45 and 0.55” or “what is the chance that the covariate on ethnic heterogeneity plays a role in explaining the observed binary values on whether a COA is considered to be a PHIA or not”. These two questions can be answered by calculating the following two *posterior probabilities*, $\Pr(0.45 < \theta < 0.55 | data)$ and $\Pr(\beta > 0 | data)$, respectively. Figure 5.2 represents these two posterior probabilities graphically.

For the coin flip example, $\Pr(0.45 < \theta < 0.55 | data) = 0.22$ (see Exercise 5.3 for the calculation). In other words, the posterior probability that θ lies between 0.45 and 0.55 is 0.22, a value that is perhaps too low to be considered as “highly likely” whilst not small enough to say the coin is biased. Basically, we cannot draw any firm conclusion based on the limited amount of data (e.g. small number of flips) and the vagueness of the prior

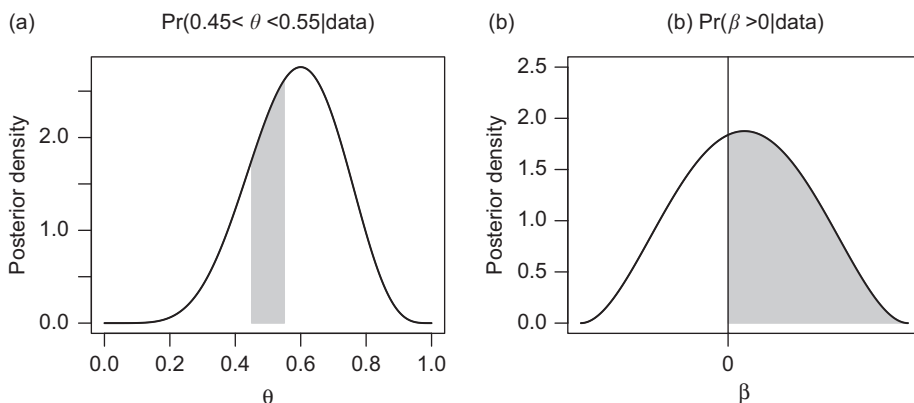


FIGURE 5.2

Graphical representation of the two posterior probabilities, $\Pr(0.45 < \theta < 0.55 | data)$ in panel (a) and $\Pr(\beta > 0 | data)$ in panel (b). The shaded area in each plot represents the corresponding posterior probability.

information – a situation that we often encounter in modelling spatial and spatial-temporal data (see Chapter 7). Getting more data (e.g. doing more coin flips) or incorporating additional information through the prior (e.g. expert opinion and/or previous experiments) are ways to provide more information. We will see how we can achieve the latter through modelling in Part II and Part III of this book.

For the HIA example, although we do not yet have the computational tools to calculate the posterior probability $\Pr(\beta > 0 | \text{data})$, we can still make a few comments. In regression modelling, we want to assess whether a covariate is associated with the outcome or not. The frequentist approach would answer that by testing whether the corresponding regression coefficient β is equal to 0 – the value 0 indicates no effect. But would it be meaningful to calculate the posterior probability: $\Pr(\beta = 0 | \text{data})$? The answer is no because that probability is always equal to 0 for a continuous-valued parameter. Instead we calculate the posterior probability that the regression coefficient β is greater than 0. If the resulting posterior probability is very large, say larger than 0.95, then this means that the majority of the posterior distribution of β lies above 0. That would imply a COA with a larger value of the ethnic heterogeneity index (i.e. with a greater ethnic mix) tends to be more likely to be considered as a PHIA. If, on the other hand, the resulting posterior probability is very small, say less than 0.05, then the majority of the posterior distribution is lying below 0. The result would then indicate that a COA with a greater ethnic mix would be less likely to be considered as a PHIA. If the posterior probability is not far away from 0.5, then β is not different from 0, suggesting that ethnic heterogeneity does not play a role in explaining the observed outcome.

In some situations, we are interested in some transformation of a parameter, in addition to the parameter itself. For example, we may be interested in (predicting) the number of heads observed if we flip the coin in question 100 times more. When using a logistic regression model, as in the HIA example, we are interested in the exponentiated coefficient β , i.e. $OR = e^\beta$, which is interpreted as the odds ratio (OR). From the odds ratio, we can calculate the change in odds (e.g. the probability of being a PHIA divided by the probability of being a non-PHIA) for a one-unit change in the corresponding covariate. So, in the HIA example, if the posterior mean of OR is 1.003 then a one-unit increase in the index of ethnic heterogeneity would be associated with a $0.3\% = (1.003 - 1) \times 100$ increase in the odds of being a PHIA. We can also derive a 95% credible interval for such a change because β has a posterior probability and any transformation of β also has a posterior probability. However, when the transformation is nonlinear, e.g. the exponential transformation, it must be carried out within the model fitting – we will see how to do that in Sections 5.3.2 and 5.4.2. This is because the exponential transformation of the posterior mean of β is not equal to the posterior mean of e^β . The reader is encouraged to demonstrate that.

5.3.2 Integration and Monte Carlo Integration

Mathematically, summaries of the posterior distribution can be written in the form of a definite integral. Consider the simple situation where we only have a single parameter, θ , the form of the integral is given by

$$\int_H g(\theta) \cdot \Pr(\theta | \text{data}) d\theta \quad (5.11)$$

where $\Pr(\theta | \text{data})$ is the posterior distribution of θ , $g(\theta)$ is any function of θ and H denotes the interval over which the integral is evaluated. Eq. 5.11 gives various posterior summaries

of θ through specifying $g(\theta)$ and H accordingly. For example, setting $g(\theta) = \theta$ and the interval H to $\pm\infty$, the posterior mean of θ is given by

$$\mathbb{E}(\theta \mid \text{data}) = \int_{-\infty}^{+\infty} \theta \cdot \Pr(\theta \mid \text{data}) d\theta \quad (5.12)$$

The posterior probability of θ greater than some threshold C is calculated by

$$\Pr(\theta > C \mid \text{data}) = \int_C^{+\infty} \Pr(\theta \mid \text{data}) d\theta \quad (5.13)$$

where $g(\theta) = 1$ and the integral is evaluated between C and $+\infty$.

Through Eq. 5.11, we can also compute posterior summaries of some transformation of θ . For example, the posterior mean of e^θ and the posterior probability of e^θ greater than a given threshold Q can be calculated respectively by

$$\mathbb{E}(e^\theta \mid \text{data}) = \int_{-\infty}^{+\infty} e^\theta \cdot \Pr(\theta \mid \text{data}) d\theta \quad (5.14)$$

and

$$\Pr(e^\theta > Q \mid \text{data}) = \int_Q^{+\infty} \Pr(\theta \mid \text{data}) d\theta \quad (5.15)$$

In general, however, there are no closed-form solutions for these integrals (except for conjugate models). These integrals need to be computed numerically via Monte Carlo integration.

The idea of Monte Carlo integration goes as follows. Suppose we can sample M random values independently from the posterior distribution. Then the empirical distribution (e.g. the histogram) of these sampled values approximates the posterior distribution and the integral in Eq. 5.11 can be approximated by the average of the function $g(\theta)$ calculated using the sampled values within the interval H . That is, denoting these values as $\theta^{(1)}, \dots, \theta^{(M)}$,

$$\int_H g(\theta) \cdot \Pr(\theta \mid \text{data}) d\theta \approx \frac{1}{M} \sum_{m=1}^M g(\theta^{(m)}) I(\theta^{(m)} \in H) \quad (5.16)$$

where $I(\theta^{(m)} \in H)$ is an indicator function that returns 1 if the sampled value $\theta^{(m)}$ falls within the interval H and returns 0 otherwise. The larger the M , the better the approximation becomes. Therefore, we can approximate the posterior mean and the 95% credible interval of θ by the mean of the sampled values $\theta^{(1)}, \dots, \theta^{(M)}$ and the 2.5 and 97.5 percentiles of the sampled values. Similarly, using the exponentiated sampled values $e^{\theta^{(1)}}, \dots, e^{\theta^{(M)}}$, the same idea can be applied to approximate the posterior mean and the 95% credible interval of e^θ . The posterior probability of the form $\Pr(\theta > C \mid \text{data})$ is approximated by the proportion of the sampled values that are greater than the given threshold C .

To illustrate, consider again the coin example, where θ is the unknown probability of getting a head when flipping the given coin once. The R code given in Figure 5.3 first samples M random values independently from the posterior distribution $\theta \mid \text{data} \sim \text{Beta}(7, 5)$, then uses these sampled values to approximate the posterior distribution, the posterior mean, the 95% credible interval and the posterior probability, $\Pr(0.45 < \theta < 0.55 \mid \text{data})$. Lines 53 to 64 in Figure 5.3 calculate the above posterior summaries using the closed-form solutions

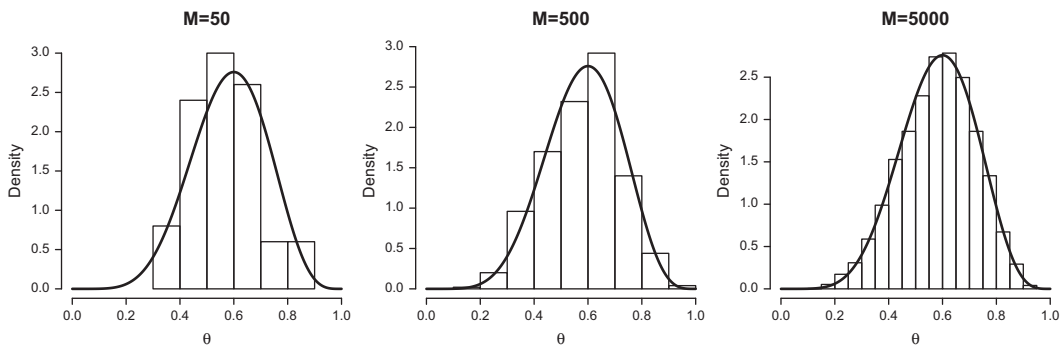
```

1 #####
2 # specify number of random values to be sampled
3 # independently from the posterior distribution
4 #####
5 M <- 50
6
7 #####
8 # sample M random values independently from the posterior
9 # distribution Beta(7,5)
10 #####
11 sampled.values <- rbeta(M,7,5)
12
13 #####
14 # the histogram of the sampled values
15 # Note that in the hist function, the argument xlim sets the
16 # minimum and maximum values when plotting the histogram and
17 # freq=FALSE tells R to plot the probability density (so
18 # that we can superimpose the beta(7,5) density)
19 #####
20 hist(sampled.values,xlim=c(0,1),freq=FALSE)
21
22 #####
23 # superimpose the posterior distribution Beta(7,5)
24 #####
25 # first generate a sequence of values between 0 and 1 for
26 # calculating the Beta density
27 x <- seq(0,1,length.out=1000)
28 # calculate the density at each of the 1000 values generated
29 # from the previous step
30 beta.density <- dbeta(x,7,5)
31 # superimpose the curve of the posterior distribution
32 lines(x,beta.density)
33
34 #####
35 # using the sampled values to approximate the posterior
36 # mean, the 95% credible interval and the posterior
37 # probability Pr(0.45<theta<0.55|data)
38 #####
39 mean(sampled.values) # posterior mean
40 # 95% credible interval using the quantile function
41 quantile(sampled.values,c(0.025,0.975))
42 # select the sampled values within the required range
43 v <- which(sampled.values>0.45 & sampled.values<0.55)
44 # calculate Pr(0.45<theta<0.55|data); the R function length
45 # counts the number of values in the object v
46 length(v)/M
47
48 #####
49 # calculating the above summaries using closed-form
50 # solutions (some via built-in functions in R)
51 #####
52 # the mean of Beta(a,b) is a/(a+b)
53 7/(7+5)
54 qbeta(0.025,7,5) # For a random variable X~Beta(a,b), the x
55 # solutions function returns x such that
56 # Pr(X<x) is equals the 1st argument of
57 # the function; the 2nd and 3rd
58 # arguments specify the parameter values
59 # of the Beta distribution, giving the
60 # lower bound of the 95% credible interval
61 qbeta(0.975,7,5) # the upper bound of the 95% CI
62 # the pbeta function (below) returns the prob. of X less
63 # than the value given by the 1st argument
64 pbeta(0.55,7,5) - pbeta(0.45,7,5)

```

FIGURE 5.3

R code to carry out a Monte Carlo integration and calculate various summary statistics from the posterior distribution $\theta | \text{data} \sim \text{Beta}(7,5)$ using closed-form solutions (via built-in R functions).

**FIGURE 5.4**

Comparing the approximation to the posterior distribution $\theta | \text{data} \sim \text{Beta}(7, 5)$ through M random values independently drawn from the posterior distribution. See Figure 5.3 for detail.

via some built-in functions in R. Figure 5.4 compares the true posterior distribution to the histogram of the sampled values across different sample sizes. As the size of the sample M increases, the approximation becomes better. This is also evident in the numerical summaries tabulated in Table 5.2: the approximated values through Monte Carlo integration are getting close to the true values calculated from $\text{Beta}(7, 5)$ as M increases.

It should be noted that we are able to compare the approximated distribution to the true posterior distribution and various approximated summary statistics to their true values because the posterior distribution in this particular example is a known probability distribution, a direct consequence of using a conjugate prior. When using non-conjugate priors, as is often the case in practice, the true values of various posterior summaries cannot be calculated analytically. Thus, Monte Carlo integration becomes essential. Figure 5.4 also highlights the importance of having a large enough set of sampled values in order to approximate the posterior distribution well. This relates to the topic of efficiency that we will return to in Section 5.3.5.3.

More generally, when modelling spatial and spatial-temporal data, there are multiple parameters. The same Monte Carlo integration idea applies, although the discussion becomes more complicated because instead of having just one single parameter θ , we have a vector of k parameters, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_k)$. We use a boldface letter to denote a vector of parameters. The idea proceeds as follows. Suppose we can sample M sets of values independently from the (multivariate) posterior distribution $\Pr(\boldsymbol{\theta} | \text{data})$ and denote the i^{th} set as $\boldsymbol{\theta}^{(i)} = (\theta_1^{(i)}, \dots, \theta_k^{(i)})$, with $i = 1, \dots, M$. The posterior distribution $\Pr(\boldsymbol{\theta} | \text{data})$ is known as the joint posterior distribution, as it is defined for all the parameters jointly. Then, using the

TABLE 5.2

Posterior Summaries for θ Using Either the Closed-Form Solutions for the Beta Distribution or through Monte Carlo Integration with Different Sizes of Sampled Values

		Posterior Mean	95% Credible Interval	$\Pr(0.45 < \theta < 0.55 \text{data})$
From closed-form solutions		0.58	(0.31, 0.83)	0.22
MC integration	$M = 50$	0.57	(0.36, 0.85)	0.30
	$M = 500$	0.58	(0.30, 0.83)	0.21
	$M = 5000$	0.58	(0.30, 0.83)	0.21

sampled values associated with a particular parameter, say θ_j (where the index j takes an integer value between 1 and k), i.e. $\theta_j^{(1)}, \dots, \theta_j^{(M)}$, we can approximate the *marginal posterior distribution* of this parameter $\Pr(\theta_j | \text{data})$ and any posterior summaries as in the single parameter case.

The prerequisite for Monte Carlo integration is that we can sample values (or sets of values) from a (multidimensional) posterior distribution that is not of a standard form (e.g. not a Beta distribution or a univariate/multivariate normal distribution and so on). As a result, these samples cannot be drawn directly using, say, the `rbeta` (or `rnorm` or `rmvnorm`) function in R. One way to deal with this problem is through the use of Markov chain sampling, a topic that we discuss next.

5.3.3 Markov Chain Monte Carlo with Gibbs Sampling

Monte Carlo integration, as we have observed, is a technique for numerical integration using values sampled from a given posterior distribution. Markov chain sampling is a powerful way to provide the values required to carry out that integration technique. Sampling using Markov chains allows us to sample values from any given posterior distribution, and that distribution can be of high dimension (i.e. with many parameters). There are many Markov chain sampling methods, but we are focusing on Gibbs sampling here because it is the sampling algorithm that WinBUGS uses. Before that, we will first define what a Markov chain is and describe the general idea of sampling using Markov chains.

A sequence of random variables, X_1, X_2, \dots , forms a Markov chain if the conditional distribution of X_{t+1} given X_1, \dots, X_t only depends on X_t . When applied to sampling, the Markovian property tells us to (a) explore the posterior distribution iteratively and (b) sample the next value from the distribution based on the current value. This iterative sampling idea is illustrated graphically in Figure 5.5. When this iterative procedure is carried out for long enough (over a sufficient number of iterative steps), we are able to examine the entire posterior distribution. Using these sampled values, Monte Carlo integration can then be used to provide various summaries of the posterior distribution.

Markov chain Monte Carlo (MCMC) is a collection of computational methods (or algorithms) that combine both Markov chain sampling and Monte Carlo integration to provide posterior summaries of unknown parameters. There are a large number of theoretical results on MCMC, but the discussion of these results is well beyond the scope of this book. We refer interested readers to van Ravenzwaaij et al. (2018) for an introduction and Gilks

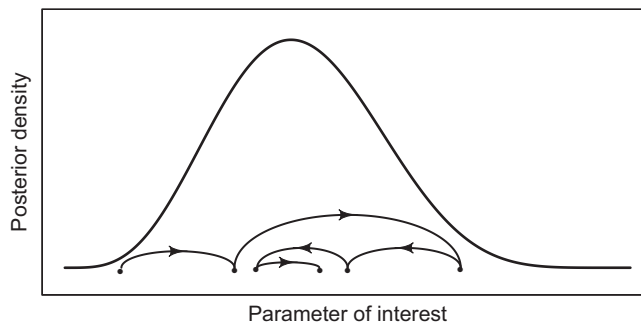


FIGURE 5.5

Illustrating the iterative nature and the Markovian property of Markov chain sampling.

et al. (1996) for more technical details and applications. We now turn our attention to Gibbs sampling.

Let $\theta = (\theta_1, \dots, \theta_k)$ denote a vector of k unknown parameters. Gibbs sampling generates a Markov chain by sampling from the *full conditional distributions*. For a parameter θ_j in the vector θ , its full conditional distribution is given by

$$\Pr(\theta_j | \theta_1, \theta_2, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_k, \text{data}),$$

which is the conditional distribution of θ_j given all other parameters and the observed data. Gibbs sampling draws values from the posterior distribution through the following three steps.

Step 1: Choose a set of initial (or starting) values for every single parameter in θ . We

denote the set of initial values as $\theta^{(0)} = (\theta_1^{(0)}, \dots, \theta_k^{(0)})$, where the superscript of each element in $\theta^{(0)}$ denotes the iteration number and the subscript is the parameter index.

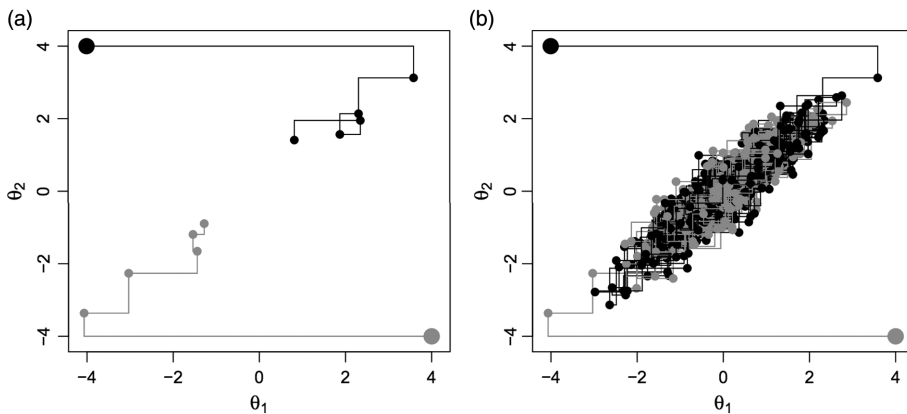
Step 2: Update the values of the parameters in turn using the full conditional distributions. This updating step goes as follows. Starting with θ_1 , we sample a new

value $\theta_1^{(1)}$ from its full conditional distribution $\Pr(\theta_1 | \theta_2 = \theta_2^{(0)}, \dots, \theta_k = \theta_k^{(0)}, \text{data})$, where all other parameters are fixed at their initial values. Similarly, for θ_2 , we sample a new value $\theta_2^{(1)}$ from its full conditional distribution $\Pr(\theta_2 | \theta_1 = \theta_1^{(1)}, \theta_3 = \theta_3^{(0)}, \dots, \theta_k = \theta_k^{(0)}, \text{data})$ but, in this case, θ_1 is fixed at the updated value $\theta_1^{(1)}$, whilst all other parameters are still fixed at their initial values. We then follow the same procedure to update θ_3 , θ_4 and so on until all k parameters have been updated. Then we have finished one MCMC iteration and $\theta^{(1)} = (\theta_1^{(1)}, \dots, \theta_k^{(1)})$ is the set of values sampled at Iteration 1.

Step 3: Repeat the updating in Step 2 thousands of times.

The many sets of sampled values are then used to produce posterior summaries of the unknown parameters as described in Section 5.3.2, providing that we have satisfied the checks that we will come to in Section 5.3.5.

For a simpler situation where there are just two parameters, $\theta = (\theta_1, \theta_2)$, Figure 5.6 illustrates various stages in the sampling of a (bivariate) posterior distribution using Gibbs sampling. Specifically, Panel (a) of Figure 5.6 shows that two MCMC chains are used to explore the posterior distribution, and the two MCMC chains have different initial values: one with $\theta_1^{(0)} = 4$ and $\theta_2^{(0)} = -4$, whilst the other with $\theta_1^{(0)} = -4$ and $\theta_2^{(0)} = 4$. Each solid dot represents the set of values sampled at each iteration, and the zig-zag shape of the two chains is the result of the conditional updating at Step 2. For example, from its starting position, each chain first gets a new value for θ_1 whilst keeping θ_2 at its initial value so the chain moves horizontally. A new value for θ_2 is then drawn whilst keeping θ_1 at its current position so that the chain moves vertically. After many iterations, the two MCMC chains appear to come together, sampling values within the same region (Figure 5.6(b)). There are, however, a number of checks that need to be carried out before we use the sampled values to compute posterior summaries of the two parameters. We will return to these checks after we have introduced WinBUGS, the software that performs Bayesian inference through the use of Gibbs sampling.

**FIGURE 5.6**

A graphical illustration of Gibbs sampling with two parameters. Two MCMC chains are shown. Panel (a) shows the first few iterations and panel (b) shows many hundreds of iterations, including those in panel (a).

5.3.4 Introduction to WinBUGS

WinBUGS is a flexible programme for carrying out Bayesian inference. WinBUGS is based on the BUGS language where the acronym BUGS stands for *Bayesian inference Using Gibbs Sampling*. In WinBUGS, the analyst specifies a likelihood function for the observed data and priors for the unknown parameters. WinBUGS then automatically constructs the resulting posterior distribution, draws samples from the posterior distribution via Gibbs sampling and subsequently produces posterior summaries of the parameters. WinBUGS is an open-source software that is freely available online. Throughout this book, Version 1.4.3 is used.⁴

We consider the simple coin example discussed in Section 5.2.1 to illustrate how to fit a model in WinBUGS. Figure 5.7 shows the full model in its mathematical form and the corresponding implementation in WinBUGS syntax.

To fit a Bayesian model in WinBUGS, we first need a model file, a file that contains the specifications of the likelihood function, the priors and possibly other quantities of interest. Every model file starts with the keyword `model`, and the model description is enclosed within a pair of curly brackets `{...}` (see Lines 1 and 11 in Figure 5.7). Lines 2 and 4 specify the binomial likelihood for the observed data and the Beta prior for θ , respectively. The two quantities `y` and `theta` are called *stochastic nodes*, and each stochastic node is associated with a probability distribution via the tilde sign `~`. Typically, a stochastic node is used to specify the likelihood (e.g. Line 2) or a prior distribution for an unknown parameter (e.g. Line 4). The WinBUGS syntax for a probability distribution starts with the letter `d` followed by the abbreviation for that distribution, so `dbin` and `dbeta` are the WinBUGS syntax for the binomial distribution and the Beta distribution, respectively.

There is another type of node in WinBUGS called the *logical node*. For example, `diff` and `pGT.7` on Lines 7 and 10 (Figure 5.7) are logical nodes. A logical node stores the result from a set of deterministic calculations such as arithmetic operations (e.g. Line 7) and logical operations (e.g. Line 10). The left arrow sign, `<-`, links a logical node to its calculation. Specifically, on Line 7, the logical node `diff` stores the difference between a sampled

⁴ OpenBUGS is another version of the BUGS language. Apart from some exceptions (see for example Section 8.2.1.2), the syntax between WinBUGS and OpenBUGS is similar. All material discussed in this chapter applies to both packages.

	WinBUGS code to implement the coin model with the Beta(1,1) prior for	Model in its mathematical form
1	model{	$y \sim \text{Binomial}(\theta, n)$
2	y ~ dbin(theta,n) # the binomial likelihood	
3	# for y heads in n flips	
4	theta ~ dbeta(a,b) # the Beta prior on theta,	$\theta \sim \text{Beta}(a, b)$
5	# the probability of head	
6	# compute the difference between theta and 0.7	
7	diff <- theta - 0.7	
8	# compute the posterior probability that theta	
9	# is greater than or equal to 0.7	
10	pGT.7 <- step(diff)	$\Pr(\theta \geq 0.7 \text{data})$
11	}	

FIGURE 5.7

The WinBUGS implementation of the coin example with the $\text{Beta}(1,1)$ vague prior. Note that the line numbers are included for reference and they are not part of the WinBUGS code. Neither are the equations listed in the last column. The mathematical form is provided for explanation only.

value for `theta` and a fixed value 0.7 at each MCMC iteration. The difference then enters the `step` function on Line 10, where the `step` function returns 1 if the argument `diff` is non-negative and 0 otherwise. Therefore, the posterior mean of `pGT.7`, i.e. the proportion of the MCMC iterations where θ is greater than or equal to 0.7, gives the required posterior probability, i.e. $\Pr(\theta \geq 0.7 | \text{data})$. As we will see in later examples, the `step` function is often used to calculate posterior probabilities. We calculate the posterior probability $\Pr(\theta \geq 0.7 | \text{data})$ to simplify the discussion here. The calculation of $\Pr(0.45 < \theta < 0.55 | \text{data})$ also uses the `step` function but is slightly more complicated – see Exercise 5.4.

Two more features of the WinBUGS syntax from this simple example should be noted. First, WinBUGS is case-sensitive so, for example, `pGT.7` and `pgt.7` refer to two different quantities. Second, the hash sign, `#`, is the annotation symbol in WinBUGS, so comments after the `#` sign, on that line, are ignored. It is always good practice to annotate code.

We now need to enter the observed data and the values for the two parameters in the Beta prior. This is done using a data list as follows:

1	<code>list(y=6, n=10</code> # observed data: y=6 heads in n=10 flips
2	<code>,a=1, b=1)</code> # defines the Beta(1,1) vague prior

Data entry starts with the keyword `list` and the pair of parentheses encloses the data values.

Both the model file and the data list fully define the model. What we need now are the initial values to initiate the MCMC chains. Typically, two MCMC chains are run so that we can assess convergence of the chains (see Section 5.3.5.2). The idea of checking convergence is to ensure that the MCMC samples used to obtain summaries are from the posterior distribution. Below are two lists with initial values for the two chains:

1	# initial value for theta for MCMC chain 1
2	<code>list(theta=0.1)</code>

1	# initial value for theta for MCMC chain 2
2	<code>list(theta=0.9)</code>

The model file, the data file and the two lists of initial values are typeset and saved in separate plain text files with a .txt extension. Figure 5.8 shows all four files open in WinBUGS (to arrange the files, go to Window then choose Tile Horizontal or Tile Vertical).

To run a model in WinBUGS, follow the steps listed below (adapted from Lunn et al., 2012, p.17–20). We now go through these steps in turn.

Step 1. Open all the files containing model, data and initial values in WinBUGS.

Step 2. Open the Specification Tool (Figure 5.9) from Model -> Specification ...

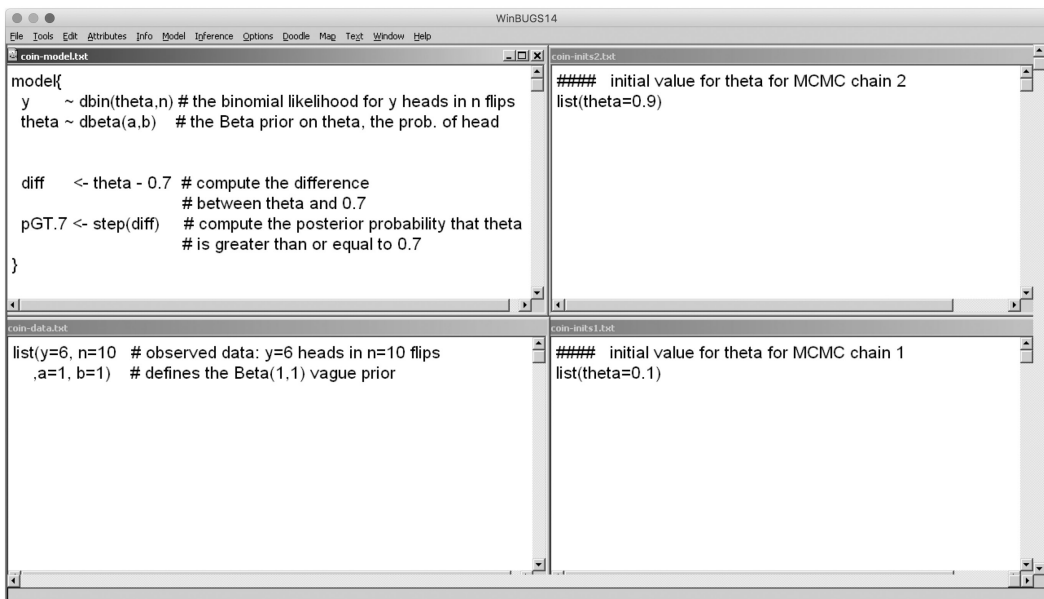


FIGURE 5.8

The four text files (the model file, the data file and the two lists of initial values) in WinBUGS in a tiled format.

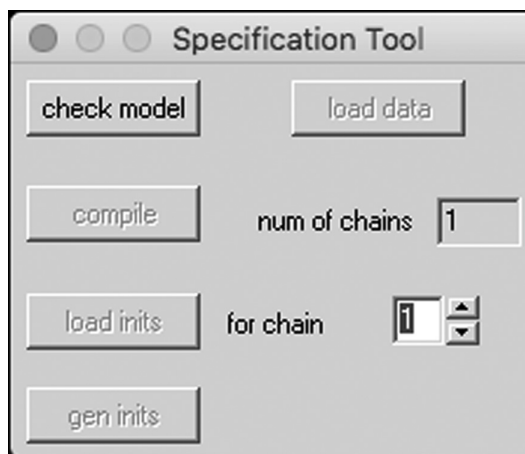


FIGURE 5.9

The specification Tool window.

- Step 3.** Activate the window containing the model code by clicking the banner of the model file (when a window is activated, the banner with the file name is in blue), then click on `check model` in the Specification Tool; WinBUGS displays any message at the bottom left-hand corner. If the model is correctly specified, it will display “model is syntactically correct”.
- Step 4.** Activate the window with the data and click on `load data` in the Specification Tool; the bottom left-hand corner will read “data loaded”.
- Step 5.** Type 2 in the text box labelled `num of chains` in the Specification Tool to specify running two MCMC chains.
- Step 6.** Now click `compile` in the Specification Tool and you will see the message “model compiled”.
- Step 7.** Activate the window with the initial value for chain 1, then click on `load inits` in the Specification Tool. At this point, two things happen: (a) the message shows “chain initialized but other chain(s) contain uninitialized variables” and (b) the box next to the label “for chain” in the Specification Tool automatically changes to 2.
- Step 8.** Activate the window with the initial value for chain 2, then click on `load inits` in the Specification Tool and the display message will read “model is initialized”.

At this point, WinBUGS knows that we are using the binomial likelihood and the Beta prior and internally formulates the (Beta) posterior distribution for θ . We have instructed WinBUGS to form two MCMC chains to sample from the posterior distribution with the first chain starting at $\theta = 0.1$ and the second one at $\theta = 0.9$. If you make a mistake at any point during this process (steps 1 to 8), you need to go through the steps again *from the beginning*.

Before updating the MCMC chains, Step 9 below informs WinBUGS of the quantities for which the sample values are to be stored or monitored:

- Step 9.** Open the Sample Monitor Tool from Inference -> Samples ... and type `theta` into the text box labelled `node` and then click `set`. This sets the monitor for `theta`. Repeat for `pGT.7`.

With the monitors set, Step 10 performs the MCMC updating:

- Step 10.** Open Update Tool from Model -> Update ... and enter 10000 in the text box labelled `updates` and click on `update`.

The number of iterations (or updates) depends on convergence and efficiency (see Section 5.3.5), but in general, the more complex a model, the more updates are required. For this simple example, 10000 iterations are sufficient. As WinBUGS updates the two chains, the number in the text box labelled `iteration` in the Update Tool window is being “refreshed” at a regular frequency defined in the refresh text box (the default is to refresh every 100 iterations). This feature shows how quickly (or slowly) the model is running.

Once the updating has completed, Step 11 obtains various posterior summaries of the parameters of interest, namely, `theta` and `pGT.7`.

- Step 11.** Type `*` in the Sample Monitor Tool to select all quantities that have been monitored, then click `density` – to produce the marginal posterior distribution for each parameter approximated using the sampled values (Figure 5.10) – and `stats` – to obtain various posterior summaries of the parameters (Figure 5.11).

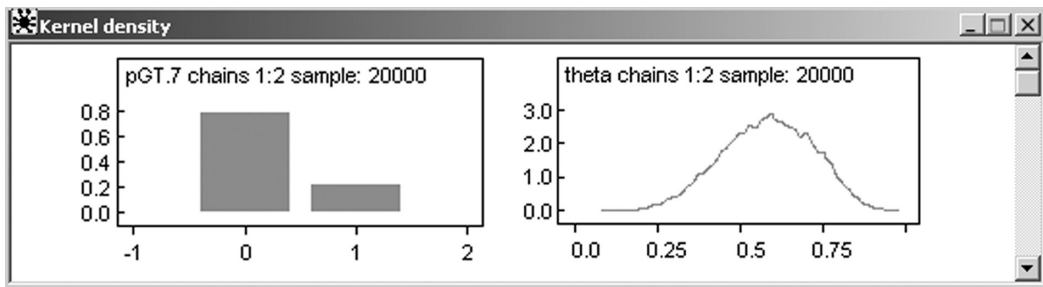


FIGURE 5.10

The marginal posterior distribution (or referred to as kernel density in WinBUGS) for each of the two parameters, θ and $p_{GT.7}$.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
pGT.7	0.2154	0.4111	0.002982	0.0	0.0	1.0	1	20000
theta	0.5832	0.1372	0.001069	0.3056	0.588	0.8307	1	20000

FIGURE 5.11

Various numerical summaries for both parameters from the posterior distribution.

Figure 5.11 shows that the posterior mean for θ is 0.58 and the 95% credible interval for θ is (0.31, 0.83). The latter two values are the 2.5th and the 97.5th percentiles from the posterior distribution for θ (the values from the 5th and the 7th column in Figure 5.11). The numerical summaries based on the 20000 sampled values (MCMC samples) from two MCMC chains are in agreement with those from the closed-form solutions given in Table 5.2. The posterior probability $\Pr(\theta \geq 0.7 | \text{data})$ is estimated to be about 0.22, the posterior mean of $p_{GT.7}$. The form of the posterior density for $p_{GT.7}$ (Figure 5.10) reflects the fact that $p_{GT.7}$ is obtained from the step function that returns either 1 or 0. As we shall see in Parts II and III of this book, WinBUGS can deal with more complex and realistic models for analysing spatial and spatial-temporal data. The posterior distributions for those models are no longer in closed form. However, the 11 steps described above are generic in running any model in WinBUGS.

5.3.5 Practical Considerations when Fitting Models in WinBUGS

In this section, we will discuss a number of issues that arise when fitting a model in WinBUGS. Readers are also referred to the paper by Brooks (1998) for a more general discussion. We start with the issue of how to set the initial values.

5.3.5.1 Setting the Initial Values

Initial values are required for all parameters that are given prior distributions. These initial values provide a set of starting values for the Gibbs sampler (see Section 5.3.3). There is no right or wrong initial value for a parameter, but the general advice is to choose the initial values sensibly! If a chosen initial value is near where the posterior distribution is, the MCMC chain may approach the posterior distribution faster compared to a chosen initial

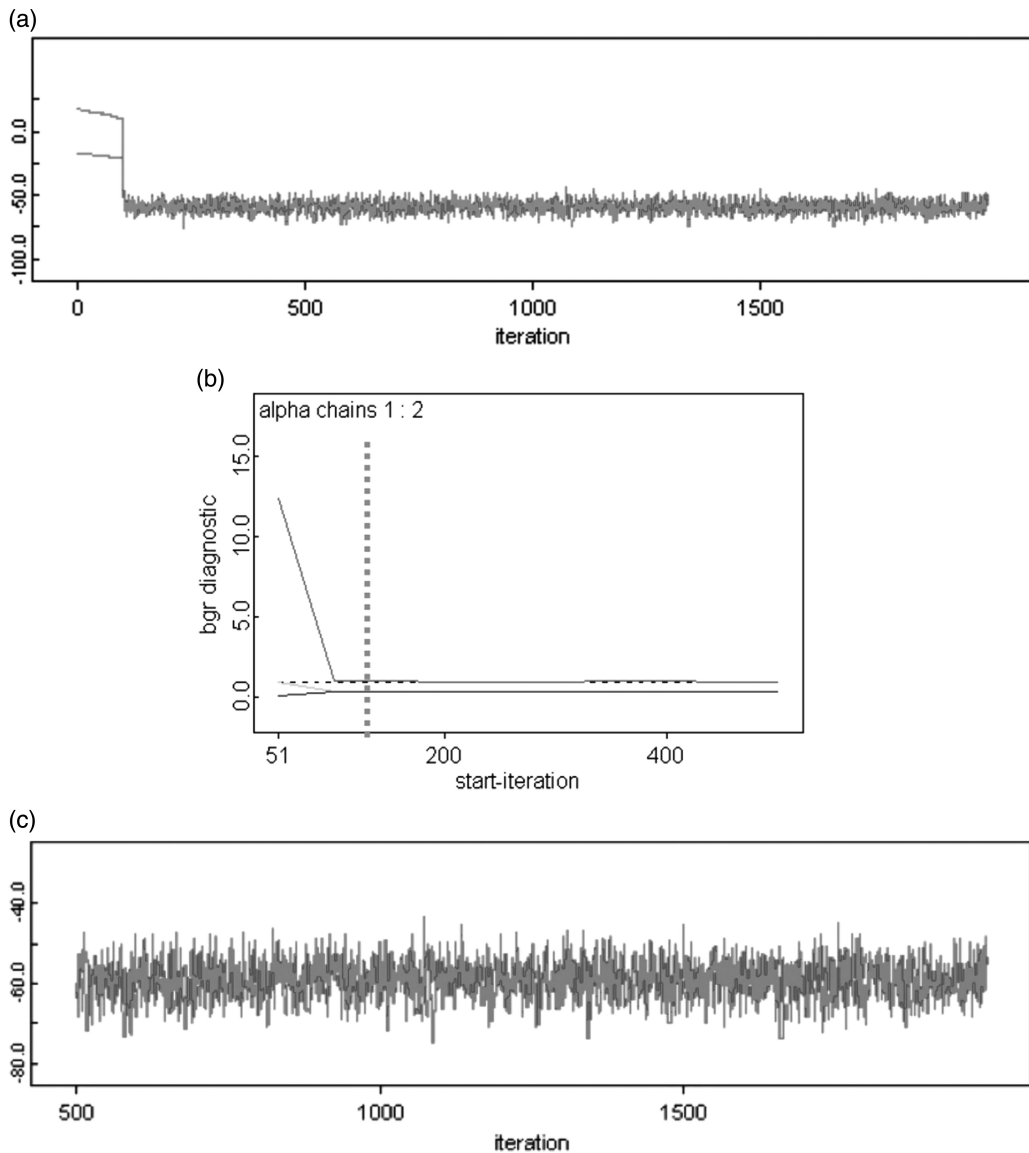
value that is far away. A badly chosen initial value may sometimes lead to slow or non-convergence. In practice, the analyst can explore the data to arrive at an “educated” guess for the initial values. For instance, if the unknown parameter is the population mean, one can use the sample mean from the data as an initial value. For regression coefficients, initial values can be taken as some small values, say ± 0.001 , or estimates from fitting the model in the frequentist framework (e.g. using the `lm` or `glm` function in R). If the parameter is a variance, then the sample variance can be used. To assess convergence reliably, Gelman and Rubin (1992) suggest choosing overdispersed initial values for different chains. Here, overdispersed means that the initial values need to be very different whilst still being sensible. Intuitively, we would be more confident of convergence if our two MCMC chains had started from very different initial positions but come together to sample from the same parameter space.

5.3.5.2 Checking Convergence

Convergence checking is concerned with ensuring the MCMC chains are sampling from the target posterior distribution. Here we will focus on the convergence diagnostics available in WinBUGS whilst we refer the reader to Brooks (1998, p.75–77) for a more general discussion of the topic. Convergence can be checked visually using the history plot. A history plot shows the sampled values against iteration numbers. Once a chain has converged, the history plot should show the sampled values scattering randomly around a stable mean value (Figure 5.12(c)). The converse, however, may not be true. Imagine two chains have run and the history of both chains shows a random scatter but about two different mean values. In that case, convergence has not been reached because these two chains may well have become trapped in some local modes of a multimodal posterior distribution, or there may be issues with the model itself, e.g. parameters are not individually identifiable.⁵ For these reasons, it is important to run two or more MCMC chains (typically we take the minimum number of two) to ensure convergence is reliably assessed and the posterior distribution is explored fully.

In addition to the visual inspection of the history plot, the Brooks-Gelman-Rubin (BGR) diagnostic (Gelman and Rubin, 1992 and Brooks and Gelman, 1997), a formal statistic for detecting non-convergence, is implemented in WinBUGS: the `bgr diag` button in the Sample Monitor Tool. Running multiple chains is required to carry out the BGR diagnostic, and running two chains meets the requirement. The basic idea is that the values sampled from multiple chains may have come from the same underlying distribution if, for a given value, we can no longer tell which chain this value is from. To formalise this, the BGR diagnostic calculates the ratio of the overall variability (by pooling all sampled values together) to the averaged within-chain variability. Once chains have converged, the BGR diagnostic should be close to 1 (Gelman et al., 2014, p.285). In practice, convergence is achieved if the BGR diagnostic is below 1.05 for *all* parameters (Lunn et al., 2012, p.75). Figure 5.12(b) shows an example of the BGR plot from WinBUGS. In the BGR plot, we are looking for the k^{th} iteration, beyond which the blue and the green lines (representing the overall variability and the within-chain variability, respectively) are stable and the red line (the BGR diagnostic) is stable and close to the horizontal dashed line at 1. So, the BGR plot in Figure 5.12(b) suggests that the two chains have reached convergence after around

⁵ Two parameters are not individually identifiable if we only have information on some combination of the two. For example, a and b cannot be estimated individually if we only know the value of $(a + b)$.

**FIGURE 5.12**

(See colour insert.) (a) A history plot of the 2000 iterations from two MCMC chains where the beginning of the two chains is clearly not from the target posterior distribution; (b) the BGR plot from WinBUGS showing that after the 150th iteration all three lines are stable and the red line is close to 1 (the grey vertical line is superimposed for ease of interpretation and is not part of the BGR plot); (c) after discarding the first 500 iterations the resulting history plot has the required form (more iterations than suggested by the BGR plot have been discarded to be on the safe side).

150 iterations, after which the two chains start to come together and settle around a stable mean thereafter as shown in the history plot in Figure 5.12(a).

Iterations before convergence are known as “burn-in”, and they should be removed (or discarded) before making posterior summaries. To discard the burn-in, enter the beginning iteration for any posterior summary into the text box labelled `beg` in the Sample Monitoring Tool. For example, to discard the first 500 iterations in Figure 5.12(a), set the

beginning iteration to 501 so that all iterations from (and including) the 501st iteration will be used to calculate the posterior summary for this parameter.

5.3.5.3 Checking Efficiency

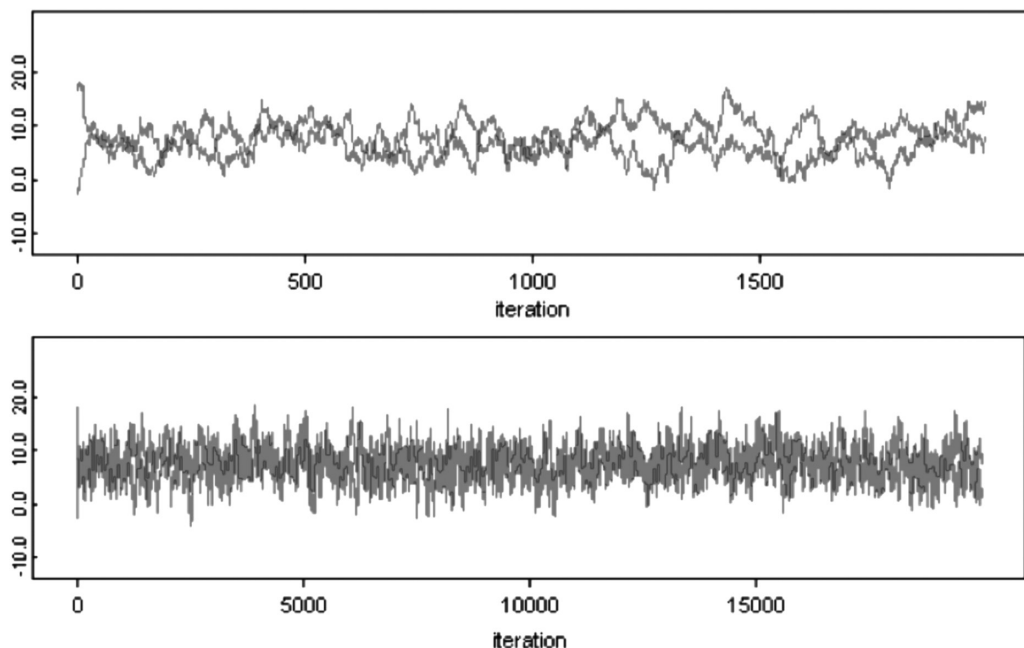
After convergence, we now need to obtain samples (from the posterior distribution) to compute posterior summaries. The more iterations we have, the better the posterior distribution is approximated (see the discussion of the sample size M in Section 5.3.2). But we cannot and do not want to run the model indefinitely. So, we need some criterion to tell us to stop sampling once the set of sampled values becomes sufficiently large to provide a good approximation to the posterior distribution. This is the check of efficiency. The Monte Carlo (MC) error is a standard output from WinBUGS for this purpose. The MC error is the standard error of the mean based on the MCMC samples as an estimate of the true posterior mean. The MC error reduces with increasing sample size because, if we assume the MCMC samples are independent, the formula for the MC error would be s / \sqrt{n} , where s and n are, respectively, the sample standard deviation and the number of MCMC samples. Unfortunately, MCMC samples are not independent, but rather they are autocorrelated (because of the Markovian property; also see below). So, the MC error based on the autocorrelated MCMC samples is larger than s / \sqrt{n} , and how much larger depends on the strength of the autocorrelation – see Section 6.3.2.1 for the definition of the effective sample size in the context of modelling spatial data. Nevertheless, we can see that the MC error is inversely related to the sample size. We want the mean of the MCMC samples to be as close to the true mean as possible, hence we need the MC error to be small. As suggested by Lunn et al. (2012, p.78–89), we can stop updating when the MC error goes below 5% of the corresponding posterior standard deviation. More iterations are required if we are interested in estimating the tail probability of the posterior distribution. In such cases, it is recommended to run the model till the MC error is less than 1.5% of the posterior standard deviation (Raftery and Lewis, 1992). For the output in Figure 5.13, a sample of 1000 MCMC iterations (second row in the output) is sufficiently large to meet the 5% criterion if we are interested in estimating the parameter beta. But if the interest is in estimating the upper tail probability of beta greater than 0.4 (roughly three standard deviations away from the mean), then we would need about 9000 iterations to meet the more stringent criterion of 1.5%.

As mentioned above, MCMC samples are autocorrelated. The greater the autocorrelation, the more samples are required to meet the below 5% (or below 1.5%) efficiency criterion. The `auto_cor` button in the Sample Monitor Tool calculates the autocorrelation at various lags. The two MCMC chains in the top panel of Figure 5.14 are highly autocorrelated (showing a “snake-like” behaviour), and the mixing for these two chains is poor. In general, high-autocorrelation and poor mixing are not problems, and running the two chains longer often helps solve any issue (see the bottom panel of Figure 5.14). In addition,

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
beta	0.1329	0.09812	0.01029	-0.01505	0.1201	0.3514	501	100
beta	0.124	0.09138	0.002902	-0.05298	0.1232	0.3067	501	1000
beta	0.13	0.09625	0.00105	-0.05274	0.1278	0.3241	501	9000

FIGURE 5.13

Posterior summary of a parameter with various number of MCMC iterations. The MC error (4th column) reduces as the number of MCMC samples increases (the last column).

**FIGURE 5.14**

Two MCMC chains with high autocorrelation (top panel), and the same two chains but run for longer (20000 iterations), showing the required form (bottom panel).

the option of thinning, i.e. using every k^{th} iteration instead of every single iteration to produce the posterior summary, also helps reduce autocorrelation. To use every 10th iteration, for example, we enter 10 into the text box labelled `thin` in the Sample Monitor Tool.

5.4 Bayesian Regression Models

In this section, we discuss the Bayesian implementation of a regression model. Regression modelling is a technique to study the relationship between the outcome values that we have observed and a set of explanatory variables. Models considered in this book are mostly from the family of generalized linear models (GLM), a class of models developed by Nelder and Wedderburn (1972) and McCullagh and Nelder (1989) to deal with both continuous- and discrete-valued (including count and binary) outcome data.

Regression modelling, as we shall see, forms the basis of many analyses that we will discuss in Parts II and III of this book. When analysing spatial and spatial-temporal data, building a Bayesian regression model involves specifying a *likelihood function* for describing the observed outcome values; a *process model* that specifies a function of the explanatory variables together with spatial and spatial-temporal random effects; and finally, a *set of prior distributions* for all the unknown parameters (see Section 1.4.3.1).

In general, specification of the likelihood function depends on the type of outcome values (continuous- or discrete-valued) and the statistical properties of the outcome values (for example, a roughly symmetric, bell-shaped histogram of the response values suggests a normal

likelihood function). In some situations, the context of the application and/or data availability provides hints as to which likelihood might be suitable. For example, when modelling counts of events with rare occurrence, such as annual counts of domestic burglary or street assaults involving violence over a set of small areas, either the Poisson or the binomial distribution may seem to be a reasonable choice. However, using the binomial distribution as the likelihood function for modelling the number of street assaults involving violence across a city may be difficult. This is because the binomial distribution requires us to specify the quantity n_i , the population at-risk in sub-area i of the city, which is not well-defined in that situation (see Section 9.5). For domestic burglary counts, on the other hand, the at-risk population can be defined as the number of houses in each spatial unit, so both the Poisson and the binomial distributions could be used and, for rare events, both would produce comparable results.

The process model considered in this section only includes explanatory variables (also known as covariates or predictors, and these terms will be used interchangeably hereafter). It specifies a relationship between the outcome and the observable covariates. The choice of covariates to be included in a regression analysis is typically based on theoretical grounds and data availability. In practice, when modelling spatial and spatial-temporal data, we typically include spatial and spatial-temporal random effects to account for the effects of unobserved/unmeasured covariates, in addition to allowing for information sharing spatially and/or temporally. We defer the descriptions of these random effect models till Parts II and III.

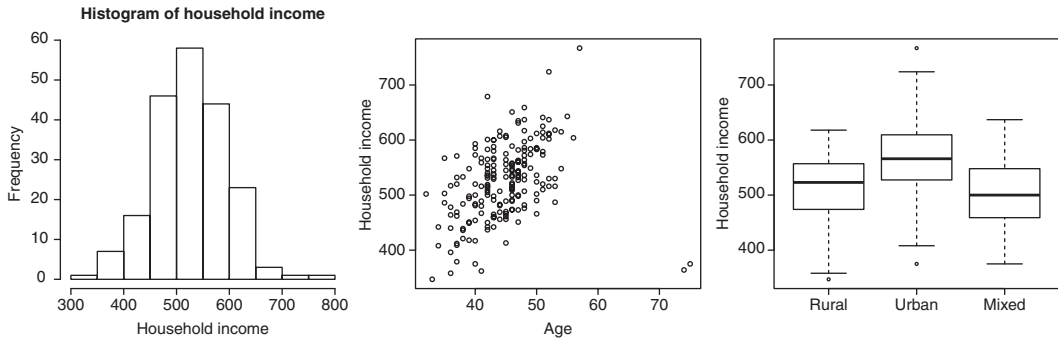
Finally, in the illustrative examples below, vague priors are assigned to parameters so that posterior inferences are primarily based on the observed data. Specifically, a normal distribution with mean 0 and a large variance 1000000 (i.e. 10^6) is used as a vague prior for the intercept and each of the regression coefficients. When applicable, a vague Gamma prior, $\text{Gamma}(0.001, 0.001)$, is assigned to the data precision, which is the inverse of the data variance. The support of a Gamma distribution is on the positive real line, a feature that satisfies the requirement of a variance parameter, which must be strictly positive. For a reader who is unfamiliar with the Gamma distribution, see Exercise 5.7 for some more detail. We will return to the discussion of prior specification in Section 5.6.

We have seen an example of the logistic regression model in Section 5.2.2. In the next two sections, we look at two examples. The first example illustrates the specification of a Bayesian linear regression with a normal likelihood. We pay particular attention to the implementation of this model in WinBUGS as well as the practical issues discussed in Section 5.3.5 when performing Markov chain Monte Carlo in WinBUGS. The second example deals with a set of small-area burglary counts. This example gives an illustration of Bayesian Poisson regression modelling. We will look at how to make posterior inference on some transformation of parameters in WinBUGS and talk about model evaluation using posterior predictive checks.

5.4.1 Example I: Modelling Household-Level Income

The dataset used in this example is a subset of the survey data on household-level weekly total gross income in Newcastle introduced in Section 1.3.1. We extracted data on 200 households with the following three variables:

y_i :	Weekly total gross income (in £s) for household i
age_i :	Age of the head of household
r_i :	Rurality index, a categorical variable indicating whether household i is in a rural area (= 1), an urban area (= 2) or a mixed rural-urban area (= 3)

**FIGURE 5.15**

Exploring the household-level income data in Newcastle: the histogram of the household-level income values; a scatterplot of the income values against age; and a boxplot for assessing the difference in mean income across the three area types: rural, urban and mixed.

The subscript i indicates household with $i = 1, \dots, 200$. The aim of the analysis here is to assess the effects of the two covariates, age_i and r_i , on household income.

Figure 5.15 explores the data graphically. The bell-shaped and symmetric distribution of the income values shown in Figure 5.15(a) suggests that a normal likelihood is reasonable. Figure 5.15(b) indicates that household income tends to increase as the head of household gets older, with the exception of the two data points lying in the bottom right corner. These two outlier points may pull the regression line downwards, so we need to bear this feature in mind when building the model. For the association between rurality and income, Figure 5.15(c) shows that urban households tend to have higher levels of income compared to rural and mixed areas, while the income levels of the latter two appear to be similar.

Based on the observations from the exploratory plots, a regression model is given as follows. For $i = 1, \dots, 200$,

$$y_i \sim N(\mu_i, \sigma^2) \quad (5.17)$$

$$\mu_i = \alpha + \beta \cdot age_i + \eta_{r_i}$$

In the above model, the normal distribution is used as the likelihood function. The regression relationship (the process model) expresses household income as a linear combination of the intercept, α , and the effects from the two covariates, $\beta \cdot age_i$ and η_{r_i} . Specifically, the regression coefficients β and η (η is a Greek letter pronounced eta) quantify the age effect and the rurality effect on income, respectively. Note that the three-level categorical variable r_i enters the model through the subscript on η . As a result, η represents three elements, η_1 , η_2 and η_3 . The corresponding element is then selected depending on the rurality of a household. Here, η_1 is fixed to 0 (i.e. setting the rural category as the reference category) so that the intercept α can be estimated. Thus, η_2 and η_3 are measuring the income difference between urban and rural areas and between mixed and rural areas, respectively. Readers are encouraged to make the distinction between the two formulations of the rurality covariate, η_{r_i} or $r_i \cdot \eta$.⁶ Finally, σ^2 is the residual variance, measuring the between household variability that is not explained by the two covariates in the model.

⁶ See, for example, Lunnn et al. (2012, p.104–106) and Gelman and Hill (2007, p.66–68) for more detail on how to handle categorical covariates in regression modelling in general.

To complete the model specification, priors need to be assigned to the five unknown parameters: α , β , η_2 , η_3 and σ^2 . A vague normal prior with mean 0 and a large variance 10^6 is assigned to the intercept α and the regression coefficients β , η_2 and η_3 . For the residual variance σ^2 , we use a vague Gamma prior with parameters 0.001 and 0.001 on the precision τ , which is $1/\sigma^2$. Figure 5.16 shows the WinBUGS model, the structure of the data and the initial values for two separate MCMC chains.

```

1 #####
2 # Block 1:
3 # The WinBUGS code for specifying the normal regression
4 # model for the income data
5 #####
6 model{
7   # looping through all households in the data
8   for (i in 1:N) {
9     y[i] ~ dnorm(mu[i],tau) # normal likelihood for income
10    # specifying the regression relationship
11    mu[i] <- alpha + beta*age[i] + eta[r[i]]
12  }
13  # setting the rural category as the reference
14  eta[1] <- 0
15  # specifying priors on the intercept and the regression
16  # coefficients
17  alpha ~ dnorm(0,0.000001)
18  beta ~ dnorm(0,0.000001)
19  eta[2] ~ dnorm(0,0.000001)
20  eta[3] ~ dnorm(0,0.000001)
21  # prior on residual precision (=1/variance)
22  tau ~ dgamma(0.001,0.001)
23  # calculate the residual variance from precision
24  residual.variance <- pow(tau,-1)
25  # calculate the residual standard deviation from variance
26  residual.SD <- pow(residual.variance,0.5)
27  # quantifying income difference between mixed and urban
28  eta[4] <- eta[3] - eta[2]
29 }
30 #####
31 # Block 2:
32 # the dataset on household-level income with two
33 # household-level covariates in the WinBUGS format
34 #####
35 list(N = 200
36      ,y = c(604, 500, 467, 542, ... , 466, 565, 379, 556)
37      ,age = c(51, 46, 41, 47, ... , 50, 43, 37, 45)
38      ,r = c(1, 1, 1, 1, ..., 1, 1, 1, 2, 2, ..., 2, 2, 2, 3
39             ,3, ... , 3, 3)
40 )
41 #####
42 # Block 3:
43 # two sets of initial values
44 #####
45 # initial values for MCMC chain 1
46 list(alpha=100, beta=1, eta=c(NA,10,10,NA),tau=0.001)
47 # initial values for Chain 2
48 list(alpha=700, beta=-1, eta=c(NA,5,-10,NA),tau=0.01)

```

FIGURE 5.16

The WinBUGS code for specifying the regression model (Block 1), the income data (Block 2) and two sets of initial values for running two MCMC chains (Block 3).

In Figure 5.16, `eta` (η) contains four elements. The first three elements, `eta[1]`, `eta[2]` and `eta[3]`, correspond to η_1 , η_2 and η_3 , as defined in Eq. 5.17. The fourth element, `eta[4]`, the difference between `eta[3]` and `eta[2]` (Line 28 in Figure 5.16), quantifies the income difference between mixed and urban areas. Both `eta[1]` and `eta[4]` are logical nodes – both are on the right-hand side of the `<` symbol (Section 5.3.4) – and do not need initial values, so NA is placed in their positions in the two sets of initial values (Lines 46 and 48 in Figure 5.16). But initial values are required for the two stochastic nodes `eta[2]` and `eta[3]`, as they both have priors (Lines 19 and 20 in Figure 5.16). Another point to note is that `dnorm`, the WinBUGS syntax to define a normal distribution (see Line 9), is parameterised using mean (the first argument in `dnorm`) and *precision* (the second argument in `dnorm`), where the precision is the inverse of a variance. The vague normal prior, $N(0, 1000000)$, is therefore implemented as `dnorm(0, 0.000001)` on Lines 17–20. Save the three blocks to three separate files then follow the 11 steps described in Section 5.3.4 to run the model. The results below are based on running the two MCMC chains over 10000 iterations.

First, we need to check convergence. Figure 5.17 shows the history plots and the BGR diagnostic plots for two chains. After a few hundred iterations, all parameters, apart from `residual.SD`, have reached convergence: all three lines in each BGR plot remain stable, and the red line virtually overlaps the dotted line 1; each history plot shows a random scatter of the sampled values around a stable mean; and the two MCMC chains overlap nicely, as seen in the history plot, showing a good mixing. For `residual.SD`, both the blue and green lines only become stable after around 5000 iterations. Hence, we will discard the first half of the 10000 iterations in each chain as burn-in by setting the `beg` in the Sample Monitor Tool to 5001. The remaining iterations are used for posterior summary.

We next check efficiency. Table 5.3 shows that all MC errors (the values in the fourth column) are less than 5% of the corresponding posterior standard deviations (the values in the third column), meaning that the total 10000 iterations (5000 from each of the two MCMC chains) are sufficient to provide a good approximation to the posterior distribution. At this point, we can proceed to interpret the estimates of the parameters tabulated in Table 5.3.

The regression coefficient β (`beta`), measuring the age effect, is estimated to be 4.00 (the posterior mean), with a 95% credible interval of 2.57–5.43, suggesting a strong effect of age on income. A £4.00 increase in weekly household income is estimated for each additional year for the head of the household, whilst the 95% credible interval tells us that this increase can be as small as £2.57 or can be as large as £5.43. These results agree with what we have observed in Figure 5.15(b).

Compared to rural areas, households in urban areas have considerably higher income, and the income difference, quantified by η_2 (`eta[2]`), is about £48.5 (95% credible interval: 28.5–68.4). While there appears to be no income difference between rural and mixed areas (the posterior mean of η_3 (`eta[3]`) is –13.4 with a 95% credible interval of –33.3–7.2, thus including the value 0), the income level in mixed areas tends to be much lower than that in urban areas (η_4 (`eta[4]`) estimated to be –61.8 with a 95% credible interval from –83.1 to –40.5; excluding the value 0).

As a form of model checking, Figure 5.18 plots observed household-level income, the fitted values (i.e. the posterior means of μ_i in Eq. 5.17) from the regression model, against the age of the head of household. Although the fitted line captures the overall pattern that income increases with age, it is perhaps pulled down by the two outliers in the bottom right corner. To minimize the influence of the outliers, one might either remove them from the data or modify the model. Unless we know that those two observations are errors created by, for example, mistakes in data entry, modifying the model is better. In Exercise 5.9,

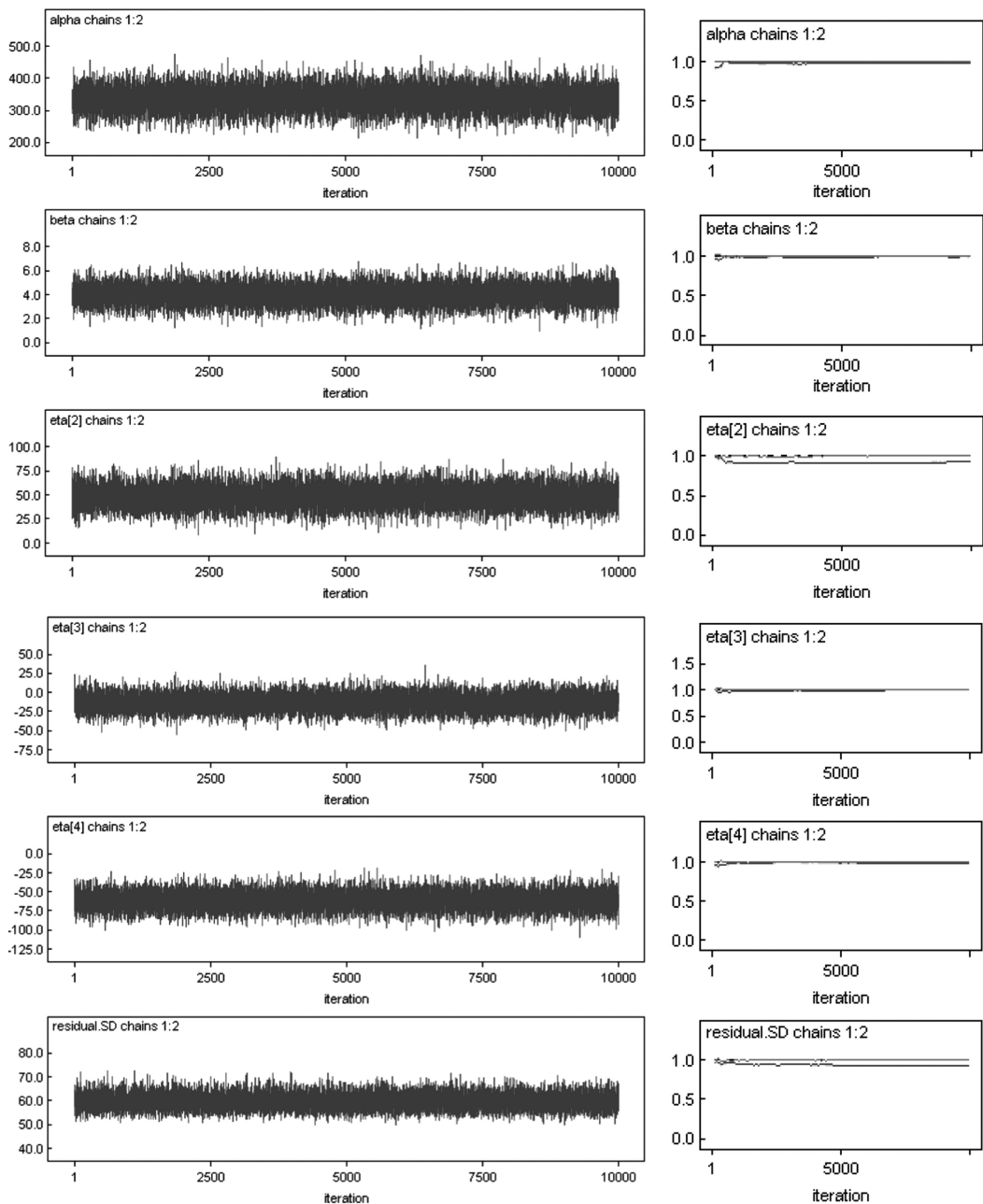


FIGURE 5.17

(See colour insert.) Checking convergence: history plots and the BGR diagnostic plots.

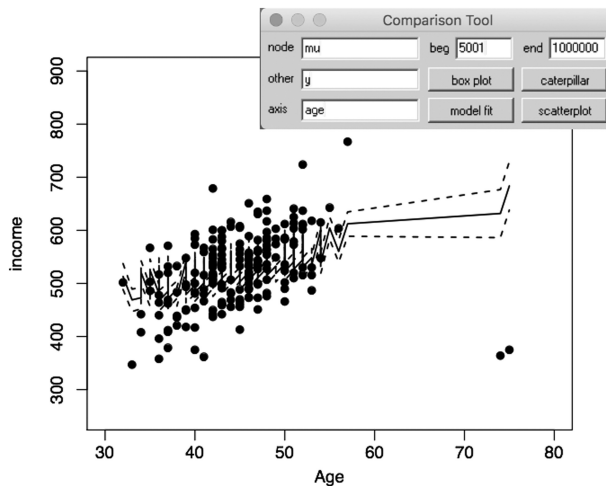
the normal likelihood in Eq. 5.17 is replaced by a Student- t distribution with four degrees of freedom to make the regression more robust to (i.e. less affected by) outliers.

As illustrated here, model checking plays a key role in regression modelling (in fact, any kind of statistical modelling): it helps to identify potential issues with the model in hand and, in turn, inform possible model improvements.

TABLE 5.3

Summary Statistics of the Posterior Distributions

node	mean	sd	MC error	2.50%	median	97.50%	start	sample
alpha	336.6	33.69	0.3135	271.6	336.5	401.5	5001	10000
beta	3.996	0.7387	0.007304	2.569	3.992	5.429	5001	10000
eta[2]	48.47	10.24	0.1052	28.49	48.38	68.43	5001	10000
eta[3]	-13.36	10.27	0.1155	-33.3	-13.41	7.232	5001	10000
eta[4]	-61.83	10.83	0.1071	-83.06	-61.77	-40.48	5001	10000
residual.SD	59.65	3.019	0.03227	54.09	59.54	66.0	5001	10000

**FIGURE 5.18**

Model checking: A scatterplot of the household-level income values against age of the households with the fitted regression line (with uncertainty) superimposed. The black dots are the actual observations, the solid line is the fitted regression line (the posterior means of μ_i in Eq. 5.17) and the two dashed lines represent the 95% uncertainty band (i.e. the 95% credible intervals of the fitted values). The window insert shows how this plot is produced in WinBUGS.

The plot in Figure 5.18 is produced within WinBUGS using The Comparison Tool (open through Inference -> Compare...). The inserted window shows the detail.

5.4.2 Example II: Modelling Annual Burglary Rates in Small Areas

This example uses a dataset containing the numbers of burglary events recorded in each of the 452 census output areas (COAs) in Peterborough, UK, in 2005 (see Section 1.6 in Chapter 1). The aim here is to construct a Poisson regression model that estimates the effects of three selected covariates on small area burglary rates. For each COA i ($i = 1, \dots, 452$), the dataset contains the following information:

O_i :	The number of recorded burglary events during 2005
n_i :	The number of at-risk houses
$x_{i,own}$:	The percentage of owner-occupied houses

$x_{i,det}$:	The percentage of detached/semi-detached houses
$x_{i,ethnic}$:	An index measuring ethnic heterogeneity; this index ranges between 0 and 1, and the larger the value the greater the ethnic mix

Here we specify the Poisson distribution as the likelihood function to model the variation in burglary counts. For each COA i , the observed burglary count is modelled as

$$\begin{aligned} O_i &\sim \text{Poisson}(\mu_i) \\ \mu_i &= n_i \cdot \theta_i \end{aligned} \tag{5.18}$$

where n_i is the number of at-risk houses and θ_i is the underlying burglary rate for that COA. Note that Eq. 5.18 is expressed according to a syntax restriction in WinBUGS where any calculation of the Poisson mean cannot be done within `dpois`, the WinBUGS function to define a Poisson distribution. In other words, we cannot do `O[i] ~ dpois(n[i]*theta[i])` in WinBUGS, but instead we have to rewrite it as `O[i] ~ dpois(mu[i])`, then define the Poisson mean on a separate line, e.g. `mu[i] <- n[i] * theta[i]`. See Lines 9 and 10 in Figure 5.19 for the implementation. Otherwise, Eq. 5.18 is equivalent to $O_i \sim \text{Poisson}(n_i \cdot \theta_i)$.

Our interest is to see how the COA-level burglary rates are affected by the three COA-level covariates. To do that, the following regression relationship (the process model) is formulated:

$$\log(\theta_i) = \alpha + \beta_1 \cdot x_{i,own} + \beta_2 \cdot x_{i,det} + \beta_3 \cdot x_{i,ethnic} \tag{5.19}$$

Eq. 5.19 models the log-transformed burglary rate in each COA as a function of the intercept and the effects of the three COA-level covariates. The log transformation (known as the log link function (McCullagh and Nelder, 1989)) is used to ensure the burglary rate, θ_i , is a positive quantity. For prior specification, each of the four unknown parameters in the model, α and β_k ($k = 1, 2, 3$), has a vague normal prior with mean 0 and variance 10^6 . Apart from the use of the Poisson likelihood, the model specification is largely the same as that for the income example in Section 5.4.1. But we want to focus on the following two points, (a) making posterior inference on some transformation of parameters and (b) model checking using posterior predictive probabilities. We now discuss the two in turn.

For a Poisson regression, in addition to the regression coefficients β_k , we are also interested in the exponentiated β_k , i.e. e^{β_k} , which are interpreted as the rate ratios (RRs). For example, $RR_1 = e^{\beta_1}$ measures the change in the burglary rate due to a one percent increase in owner-occupied housing. An estimate of the rate ratio that is much higher (lower) than 1 would suggest an increase (decrease) in the burglary rate with a unit increase in the corresponding covariate. If the rate ratio is estimated to be close to 1, then the corresponding covariate does not play an important role in explaining the variation in burglary rates at the COA level. To obtain the posterior estimates of the rate ratios, the nonlinear exponential transformation needs to be carried out within WinBUGS. We will comment on how this is done in Figure 5.19.

Posterior predictive check is a useful tool to check how consistent a model is with the observed data. As stated by Gelman et al. (2014, p.143), “If the model fits, then replicated data generated under the model should look similar to observed data. To put it another way, the observed data should look plausible under the posterior predictive distribution.” In this example, we want to see how well the Poisson regression model defined in Eqs. 5.18 and 5.19 describes the observed burglary counts. To do that, for each COA, we simulate

```

1 #####
2 # The WinBUGS code for specifying the Poisson regression
3 # model for the burglary data
4 #####
5 model {
6   # looping through the 452 COAs (N=453)
7   for (i in 1:N) {
8     # the Poisson likelihood for burglary counts
9     O[i] ~ dpois(mu[i])
10    mu[i] <- n[i] * theta[i] # defining the Poisson mean
11    # the regression relationship
12    log(theta[i]) <- alpha
13                      + beta[1]*(x_own[i]-mean(x_own[]))
14                      + beta[2]*(x_det[i]-mean(x_det[]))
15                      + beta[3]*(x_ethnic[i]-mean(x_ethnic[]))
16  }
17  # vague priors on the intercept and the regression
18  # coefficients
19  alpha ~ dnorm(0,0.000001)
20  for (k in 1:3) {
21    beta[k] ~ dnorm(0,0.000001)
22  }
23  # compute the rate ratios (RRs) associated with the
24  # covariates
25  RR_own <- exp(beta[1])
26  RR_det <- exp(beta[2])
27  RR_ethnic <- exp(beta[3])
28  # posterior predictive check against the observed
29  # COA-level case counts
30  for (i in 1:N) { # go through each COA
31    O.pred[i] ~ dpois(mu[i]) # predict the burglary
32                           # count for each COA
33    # calculate the difference between the predicted
34    # and the observed counts
35    diff[i] <- O.pred[i] - O[i]
36    # calculating ppp[i]; see main text for detail
37    ppp[i] <- equals(O[i],0)*equals(O.pred[i],O[i])
38              + (1- equals(O[i],0))*step(diff[i])
39  }
40 }
41 #####
42 # two sets of initial values
43 # (specified only partially; see the main text)
44 #####
45 # initial values for MCMC chain 1
46 list(alpha=-2, beta=c(0.01,0.01,0.01))
47 # initial values for Chain 2
48 list(alpha=-4, beta=c(-0.01,-0.01,-0.01))

```

FIGURE 5.19

The WinBUGS code to fit the Poisson regression model defined in Eq. 5.18 and Eq. 5.19 for the COA-level burglary counts in Peterborough reported during 2005.

O_i^{pred} , the number of burglary cases predicted from the fitted model, then compare the posterior distribution of O_i^{pred} to the actual observed case count in that COA. Note that O_i^{pred} has a posterior distribution since it is a function of the unknown parameters as formulated in Eq. 5.19. Any discrepancy between the model and the observed value can be highlighted via the so-called posterior predictive p-value:

$$\begin{aligned} \Pr(O_i^{pred} \geq O_i | data) & \text{ if } O_i > 0 \\ \Pr(O_i^{pred} = O_i | data) & \text{ if } O_i = 0 \end{aligned} \quad (5.20)$$

An extremely large (or small) posterior predictive p-value would indicate that the model is producing too many (or too few) burglary cases – the reader is encouraged to visualize this interpretation (hint: using Figure 5.2). In addition to checking against the observed outcome values, posterior predictive check can also be used to check against some particular characteristics of the dataset (see for example Sections 9.1 and 9.3 in Chapter 9 and Section 16.5 in Chapter 16).

Figure 5.19 shows the WinBUGS implementation of the Poisson regression model for the burglary data, featuring both the calculation of the rate ratios and the posterior predictive check. There are a few points to comment on. First, on Lines 13 to 15, each of the three covariates is mean-centred, subtracting each covariate value from its mean with `mean(x_own[])` being the syntax to calculate the mean of the covariate `x_own`. Mean-centring the covariates sometimes helps speed up convergence and results in better mixing. It also gives a more meaningful interpretation for the intercept. For example, in this implementation, `alpha` is interpreted as the log burglary rate when all covariates are set at their mean values.

Second, Lines 25 to 27 calculate the rate ratios. For each of the three regression coefficients, `beta[1]`, `beta[2]` and `beta[3]`, a value is sampled from the posterior distribution at each MCMC iteration. This sampled value is then exponentiated to give the value for the corresponding rate ratio. Posterior summaries for the rate ratios are then obtained using these exponentiated values over the MCMC samples.

Third, Line 31 predicts the number of burglary cases for each COA. Then Lines 37 and 38 calculate the posterior predictive p-value. Since there is no `if` statement in WinBUGS, the calculation may seem convoluted. The `equals` function in WinBUGS returns 1 when its two arguments are equal and returns 0 otherwise. Therefore, `equals(O[i], 0) * equals(O.pred[i], O[i])` contributes to the posterior predictive p-value of a COA where zero cases are observed (i.e. the second line in Eq. 5.20). For a COA with at least one case, `equals(O[i], 0) * equals(O.pred[i], O[i])` is 0, whilst the part on Line 38, `(1 - equals(O[i], 0)) * step(diff[i])`, contributes towards the calculation of the first line in Eq. 5.20.

Finally, the two sets of initial values are only partially specified because we still need to assign initial values for each `O.pred[i]`, the predicted number of burglary events. There are two options to accomplish this. The first option is to explicitly specify a list of 453 integer values for each chain, e.g. `O.pred=c(1,1,1,...,1)` for chain 1 and `O.pred=c(3,3,3,...,3)` for chain 2. The second option is to use the `gen_inits` button in the Specification Tool. Proceed from Step 1 to Step 8 as described in Section 5.3.4. After loading each set of initial values in (i.e. Step 7 and Step 8), WinBUGS displays the message “This chain contains uninitialized variables” in the left-hand bottom corner. Once both sets of initial values have loaded in (i.e. after Step 8), press the `gen_inits` button in the Specification

TABLE 5.4

Posterior Summaries of the Estimated Rate Ratios Associated with the Three COA-Level Covariates

Rate Ratios	Posterior Mean (95% credible interval)
RR_{own}	0.988 (0.983,0.992)
RR_{det}	0.996 (0.993,1.000)
RR_{ethnic}	0.997 (0.991,1.004)

Tool so that initial values are generated by WinBUGS internally for all the parameters that do not have starting values (see also the WinBUGS manual). WinBUGS will display “initial values generated, model initialized”. We can then proceed onto the subsequent steps. We will discuss some of the results below, deferring the details of the WinBUGS implementation to Exercise 5.10.

Table 5.4 summarises the posterior estimates of the rate ratios across the three covariates. The posterior means of all three rate ratios are estimated to be below 1, suggesting that a one-unit increase in each of the covariates would be associated with a decrease in burglary rate. However, only the covariate on the percentage of owner-occupied houses appears to be statistically significant in the sense that the 95% credible interval does not include 1. The change in the burglary rate can also be quantified. A one percent increase in owner-occupied housing (roughly one or two houses more in a COA occupied by the owners) is associated with a reduction in the burglary rate of 1.2% ($= (0.988 - 1) \times 100$, with the negative value indicating a reduction). The same calculation yields the 95% credible interval of that reduction: (0.8%,1.7%). Note that the above calculation is a linear transformation of the posterior summaries of the rate ratio and hence not required to be carried out within the WinBUGS fit. There is weak evidence of an effect associated with the percentage of detached/semi-detached houses since the upper bound of the 95% credible interval is 1. Ethnic composition does not appear to be a strong factor for explaining variability in small area burglary rates.

How well does this model fit the data? The estimated posterior predictive p-values for 423 COAs fall in a reasonable range between 0.05 and 0.95, a criterion proposed by Gelman et al. (2014, p.150–151) to assess discrepancy. These estimates suggest that the regression model considered here fits the observed burglary counts in these 423 COAs adequately. For the remaining 29 COAs, however, the model predicts either too many (two COAs) or too few (27 COAs) cases of burglary, indicating a lack of fit. Exercise 5.10 investigates these 29 COAs in more detail.

We finish this example by noting that we will see many aspects of the modelling presented here again in Parts II and III. The regression model considered in this example is certainly much simpler than the spatial and spatial-temporal models that we will discuss later in the book, but it is always advisable to start simple. A simple model often offers some insights into the dataset in hand, e.g. what the likely effects are from the covariates and which parts of the data the model fits well and which parts it fits poorly. Some of the information helps us to refine the model to provide a better description of the data.

5.5 Bayesian Model Comparison and Model Evaluation

For any given dataset, we might often come up with several different models. We might, for example, be undecided as to which of several models should be preferred on theoretical

grounds; we might progress in our analysis from fitting a rather simple model to more complex models. In these circumstances, a question arises: “how do I compare these models and decide between them?”

In the Bayesian framework, models can be compared using the Deviance Information Criterion (DIC; Spiegelhalter et al., 2002), which takes the following form:

$$DIC = \bar{D} + pD \quad (5.21)$$

In Eq. 5.21, \bar{D} is the posterior mean of the deviance, measuring the goodness of fit (of the model to the data), and pD is the effective number of parameters, measuring model complexity. A smaller \bar{D} value indicates a better fit to the data. A model with more “effective parameters” is a more complicated model. For non-hierarchical models (all the models considered in this chapter are non-hierarchical), pD is very close to the number of actual unknown parameters in the model. However, for hierarchical models, to which most of the spatial and spatial-temporal models that we will discuss in Parts II and III of this book belong, a better way to measure model complexity is through pD , rather than counting the number of actual unknown parameters. This is because some of these unknown parameters (e.g. random effects) are not independent but correlated and hence the *effective* number of parameters in the model may be fewer than the actual number.

For comparing models invoking Occam’s razor, our preference is usually for simple rather than complicated models, providing the simpler model does not result in a poor fit to the data. DIC is the Bayesian equivalent of Akaike’s Information Criterion (AIC), which is often used in the frequentist approach for model comparison. As in the case of AIC, a model with a smaller DIC value is better supported by the data. When comparing two models, a rule of thumb is that where there is a difference between two DIC values greater than 5, the model with the smaller DIC value is preferred. A difference in DIC less than 5 suggests the two models are indistinguishable (based on DIC) (Lunn et al., 2012, p.166–167).

It is important to note that whilst the DIC comparison helps us to decide which model or set of models best describes the data, DIC itself does not tell us how *well* the chosen model(s) describes the observed data. The best model amongst the models considered can still be a poor model. For this reason, it is important to perform appropriate model checks. We have seen some forms of model checking in Sections 5.4.1 and 5.4.2 by comparing the fitted values to the observed values. We shall see other forms of model checking in subsequent chapters (see for example Chapter 9, Section 11.2 and Section 16.5).

5.6 Prior Specifications

The choice of prior is, in principle, subjective. It reflects our knowledge about the parameter before seeing the observed data in hand (it expresses our “prior” knowledge). We divide our discussion on the prior into two situations depending on how much information we have about the parameter(s) before seeing the data.

5.6.1 When We Have Little Prior Information

If the analyst feels she knows very little about the possible value, then a vague prior should be chosen. Such a prior is also referred to as a diffuse or weakly-informative prior.

Throughout this book, we will refer to a prior as “vague”, “diffuse” and “weakly-informative” interchangeably. The rationale for using a vague prior “is often said to be ‘to let the data speak for themselves’ so that inferences are unaffected by information external to the current data” (Gelman et al., 2014, p.51). In these circumstances a prior is selected which is vague with respect to the likelihood, meaning that the prior distribution spreads diffusely over the range of values for the parameter that are supported by the likelihood. If we use a vague prior, estimates from Bayesian analysis tend to be similar to maximum likelihood estimates because the posterior distribution is dominated by the likelihood (see the use of the vague $Beta(1,1)$ prior in the coin example in Section 5.2.1). Depending on what parameter we are dealing with, there are some typical probability distributions that we can use for specifying a vague prior.

For a location parameter, θ (for example a mean or a regression parameter), the uniform distribution on the whole real line from $-\infty$ to $+\infty$, denoted as $Uniform(-\infty, +\infty)$, looks to be a good candidate for being a vague prior because all possible values of θ are assumed to be equally likely. Posterior inference is then based on the likelihood, $\Pr(\text{data} | \theta)$. However, $Uniform(-\infty, +\infty)$ is an improper probability distribution because it does not integrate to 1, a requirement for any proper probability distribution. Using an improper distribution as a prior should usually be avoided, but there are exceptions (see Section 8.2.1.1 when specifying the intrinsic conditional autoregressive model for spatial data). There are options to make an approximation to this improper distribution. One is to use $Uniform(a, b)$ as a prior for θ where a and b are chosen to reflect a “wide” range (say $a = -1000$ and $b = 1000$). Another possible option is to use the normal distribution with mean 0 and a “large” standard deviation. For example, $\theta \sim N(0, \sigma^2)$ with σ set to say 1000. It is easier to conceptualize a normal distribution in terms of standard deviation because of the so-called three-sigma rule: almost all values covered by a normal distribution $N(0, \sigma^2)$ are within $\pm 3\sigma$ (Pukelsheim 1994), giving us a rough idea of the range of a normal distribution.

But how “wide” is wide and how “large” is large? One principle we can adopt is to write down what we think the range of values that θ should take, then stretch the range out. For example, if we expect θ to lie in the range between 1 and 10, then $Uniform(-1000, 1000)$ or $N(0, 100^2)$ represent vague priors for θ . On the other hand, if we expect θ to lie in the range between 500 and 1000, then neither of the two priors before are vague (they are in fact very informative – too informative!). In that case, better choices (giving a more spread out range) would be $Uniform(-100000, 100000)$ or $N(0, 10000^2)$. WinBUGS parameterizes the normal distribution in terms of mean and *precision*, which is $1/\text{variance}$. So the WinBUGS code to implement the prior specification $\theta \sim N(0, 10000^2)$ is `theta ~ dnorm(0, 0.00000001)`.

For a variance parameter, a prior needs to have a positive support. Typical choices as a vague prior are

- A $Gamma(\epsilon, \epsilon)$ distribution with ϵ set to be small, say 0.001, on the precision τ (see Exercise 5.7 for more information about the Gamma distribution). For example, the WinBUGS implementation of $\tau \sim Gamma(0.001, 0.001)$ is `tau ~ dgamma(0.001, 0.001)`.
- A uniform distribution, $Uniform(a, b)$, on the standard deviation σ where a is typically chosen to be a small value close to but greater than 0, say 0.0001, and b is set to be a large value, say 1000.
- A half normal distribution with mean 0 and a large variance on the standard deviation σ (Gelman 2006). “Half” means that the normal distribution is bounded

below by 0 such that σ is strictly positive. For example, $\sigma \sim N_{+\infty}(0, 10^2)$, where $N_{+\infty}$ denotes a half normal distribution bounded below by 0. In WinBUGS: `sigma ~ dnorm(0, 0.01) I(0,)`, where `I(0,)` restricts the distribution to be defined only on the positive real line.

Similar to the discussion for a location parameter, the range of a prior distribution for a variance parameter should be sufficiently wide to be vague. When dealing with hierarchical models, which we shall introduce in Parts II and III, we may want to consider assigning moderately informative priors (e.g. using a smaller standard deviation on the half normal prior or a narrower range for the uniform prior) to help the estimation. We will see some examples later (see for example Section 16.4).

Whatever strategy the analyst adopts to specify priors, sensitivity analysis should form part of any model fitting process. Different priors should be tested to ensure that an apparently innocuous uniform prior is not introducing substantial information into the fitting. The analyst needs to consider, and try out, different specifications of a vague prior and widen the range of a uniform prior (or increase the variance of a normal prior).

5.6.2 Towards More Informative Priors for Modelling Spatial and Spatial-Temporal Data

Bayesian inference becomes particularly interesting in those cases where we believe, with justification, that we have some prior knowledge about one or more of the model's parameters. Bayesian inference allows us to make use of that knowledge rather than simply discarding it and proceeding as if it is only the present dataset that matters for the problem at hand. The reader is reminded that in Section 5.2.1 we provided some rather general examples where, in the case of a coin flipping experiment, we might move away from a vague prior towards a more informative prior. The examples we provided fell into two categories: (i) where we are able to argue about the probable range of the parameter from first principles (about coins and how they behave when flipped); (ii) where we are able to call on previous relevant experimentation (with coins like the one under study) or other relevant empirical research experience. When the analyst feels that they are an expert on the subject (based on either (i) or (ii) or some combination of both) and feels that they have a great deal of prior knowledge, they are likely to choose an informative prior – one that introduces a source of information, in addition to that from data, into the estimation of the parameter. Between the two extremes of “I know nothing” and “I am an expert”, we can select priors that are to some degree informative. That degree of informativeness becomes stronger as we move to the latter state. Before we can do that we must have a clear understanding of what the parameter means (its interpretation). This will depend on the problem we are studying.

So what kind of prior knowledge is likely to be relevant when analysing spatial and spatial-temporal data? We suggest two types of prior knowledge, which we refer to as geographical-substantive knowledge and spatial knowledge.

By geographical-substantive knowledge we mean the kind of knowledge that derives from previous work in the substantive field – studying areas of a city prone to high levels of crime (Section 5.2.2); studying variation in household income (Section 5.4.1); studying why some areas of a city have higher levels of household burglary than others (Section 5.4.2). Typically, such substantive knowledge is used to specify the covariates that are

included in the process model rather than proscribing, in some sense, the values of one or more of the regression parameters. So this type of prior knowledge, at least in many areas of the social sciences, informs *the specification of the process model* rather than the inclusion of an *informative prior distribution* on one or more of the parameters.⁷

The second type of prior knowledge, what we have termed spatial knowledge, is interesting in a different way. In Section 3.3.2 we discussed dependence and heterogeneity and saw that they constituted two of the fundamental properties of spatial and spatial-temporal data. Data values close together in space (and in space-time) tend to be more alike than data values that are further apart. If we are willing to impose that observation about data onto a set of parameters, then we can assume that parameters close together in space (and space-time) tend to have values that are more alike than parameters that are further apart. This second observation provides the basis for specifying a type of prior distribution (or prior model) on parameters that leads to information borrowing amongst spatial or space-time units that are close together. We shall see examples of spatial prior models in Chapter 8 and spatial-temporal prior models in Chapter 15 that enable such information borrowing and, in the process, how the data sparsity challenge is addressed. As a form of “expert” knowledge, spatial knowledge, as we have defined here, is a hybrid between types (i) and (ii) described above – combining knowledge that expresses how things vary in space and space-time from first principles with knowledge that has been observed in many empirical contexts about how social science and environmental phenomena vary in space and space-time.

It is worth noting that there can be a link here with geographical-substantive knowledge. Consider a situation where previous work suggests the need to include a particular covariate in the process model but data on this covariate is either unavailable or incomplete. One way to approach this situation is to introduce random effects into the process model. To reflect our spatial knowledge about the covariate for which data are missing or incomplete, appropriate prior distributions can be specified so that the random effects are spatially structured and/or spatially unstructured (see for example Sections 1.3.2.1 and 9.2).

5.7 Concluding Remarks

One of the aims of this chapter has been to introduce Bayesian inference from a theoretical perspective, a model building perspective and a computational perspective. As we have seen, the Bayesian approach provides the joint probability distribution for all the parameters. This feature brings several benefits that we briefly summarize here. We will illustrate these benefits in many different spatial and spatial-temporal modelling contexts in the course of Parts II and III.

First, Bayesian inference incorporates information not only from data (via the data model) but also from other sources external to the observed data (via the prior distributions). For the analysis of spatial and spatial-temporal data, this latter feature becomes particularly useful, because as we have noted in 5.6.2 and will describe in depth in Chapters 8 and 15, prior distributions provide a way of capturing dependence structures across space and space-time. Second, the joint posterior distribution for all parameters

⁷ See Section 3.2 for reasons as to why our knowledge about the association is often quite limited.

allows various sources of uncertainty to propagate throughout the statistical analysis. The estimation of regression coefficients takes into account the uncertainty arising from, for example, the presence of missing data and/or the errors in measuring the outcome data. Third, since parameter inference is based on the entire posterior distribution, in addition to the conventional point and interval estimates, probability statements can be readily obtained in order to directly address the question(s) in hand. For example, what is the probability that the average household income level in an area is below a given poverty threshold, or how likely is it that the overall burglary rate in the treatment group is less than in the control group? Fourth, in addition to the parameters in the model, inference on interpretable quantities that are some transformation of the model parameters can be obtained easily via Markov chain Monte Carlo (MCMC). We saw examples of this in Sections 5.2.2 and 5.4.2 (where the parameter of interest in each case is a non-linear transformation of the regression coefficient as opposed to the coefficient itself) and Section 1.3.2.4 (where we were interested in posterior inference on $[\exp(b) - 1] \times 100$, the percentage change in the burglary rate compared to the controls, rather than b ; see Section 12.3.5.4 for more detail). Finally, as we shall see many times, the Bayesian approach offers a convenient way to construct complex models – a feature that enables the analyst to construct a range of models to examine, for example, the robustness of the findings or to explore different features of the data.

Much of the rest of this book focuses on Bayesian regression modelling, so it is appropriate at this point to draw attention to a few points that the modeller of spatial and spatial-temporal data ought to be aware of and that the reader will encounter at various points in Parts II and III. First, the choice of a likelihood is certainly not unique. Certain features of the outcome values may require us to consider alternatives. For example, the Student's t distribution with ν degrees of freedom is often used as an alternative to a normal likelihood in robust regression when outliers are present in response values (see Section 5.4.3 and Exercise 5.9; see also Gelman and Hill, 2007, p.124–125). When dealing with count data, the issue of overdispersion, where the variance exceeds the mean in the data, is prevalent in the case of spatial and spatial-temporal count data (see Section 3.3.3). The Poisson likelihood does not allow overdispersion in data since both the mean and the variance are assumed to be equal under a Poisson distribution. The negative binomial distribution offers an alternative (e.g. Hilbe, 2011), while including random effects within the Poisson likelihood is also used (e.g. Gschlößl and Czado, 2008). In application three in Chapter 9, we will discuss some choices of likelihood to tackle the issue of zero-inflation (where the chosen likelihood cannot accommodate the large number of 0 values in the observed data) when analysing count data in a spatial setting.

With a selection of likelihood choices to hand, which one should we use first? We recommend “starting simple”. Some likelihoods (e.g. normal, binomial and Poisson) are considered to be “standard” while others (e.g. the Student's t and the negative binomial) are typically used once you are more familiar with the data.⁸ Start with the standard one, then consider other alternatives. Placing the regression analysis in the Bayesian framework makes it easy to examine different likelihoods, and the change of the likelihood function only requires you to modify a few lines of WinBUGS code. This gives you the flexibility to fully explore the features in the data. Over the course of model building, model checking and model comparison are vital to a statistical analysis.

⁸ The t_ν and negative binomial distributions are not in the exponential family, so the corresponding regression models are not in the class of GLMs.

The other main aim of this chapter has been to show how Bayesian inference is applied to model spatial data. We have approached this through the three examples in this chapter – the HIA example in Section 5.2.2 and the income and the burglary examples in Section 5.4. Those three examples provide a starting point where a Bayesian analysis has been implemented on spatial data. Beyond specification of the process model, the modelling that we have seen so far is not spatial in any distinctive or “special” way. However, in Section 5.6 and Section 5.6.2 in particular, we have started to explore how that most characteristic feature of Bayesian inference, embedding prior knowledge into data analysis, might proceed in the case of spatial data.

5.8 Exercises

Exercise 5.1. For the coin example in Section 5.2.1, justify the use of a Beta distribution, $Beta(a,b)$, as a prior distribution for θ , the probability of obtaining a head when a coin is flipped once. The expert’s opinion that θ is most likely to be 0.5 but unlikely to be outside the interval between 0.4 and 0.6 can be expressed mathematically as the mean of the Beta distribution is 0.5 and the standard deviation of the Beta distribution is 0.05 (why 0.05? Hint: the so-called three sigma rule). We can form two simultaneous equations with two unknown quantities, a and b , then solve for a and b . (Hint: if $\theta \sim Beta(a,b)$, then the mean of θ is $\frac{a}{a+b}$ and the variance of θ is $\frac{ab}{(a+b)^2(a+b+1)}$).

Exercise 5.2. Again, for the coin flip example, derive the posterior distribution for θ using the $Beta(a,b)$ distribution as a prior for θ . Calculate the posterior mean and the posterior variance for θ when using (a) the vague $Beta(1,1)$ prior and (b) the informative $Beta(45.5, 45.5)$ prior. Compare and comment on your results.

Exercise 5.3. Use R to simulate a set of 10000 random values from the distribution $Beta(7,5)$, the posterior distribution for θ when using the vague prior $Beta(1,1)$. (Hint: to simulate n values randomly from the distribution $Beta(a,b)$ in R, use the function `rbeta(n,a,b)`, then use the function `length` to see how many of those simulated values fall between 0.45 and 0.55; see also the R code given in Figure 5.3).

Exercise 5.4. Follow the steps outlined in Section 5.3.4 to carry out the analysis for the coin flip data. First, run the model exactly as given in Figure 5.7 and make sure you can obtain the results as presented in Figure 5.10 and Figure 5.11. Then modify the WinBUGS code in Figure 5.7 to calculate the posterior probability $\Pr(0.45 < \theta < 0.55 | data)$. (Hint: you may need to multiply the output from two step functions, i.e. `step(diff1)*step(diff2)`, where `diff1 <- theta - 0.45` and `diff2 <- 0.55 - theta`).

Exercise 5.5. For the coin example, suppose we are interested in the probability of observing more than 20 heads if the same coin is flipped a further 30 times. Calculate this probability under the following two scenarios: (a) fixing θ (the probability of getting a head if this coin is flipped once) to the mean of the posterior distribution $Beta(7,5)$ when using the vague $Beta(1,1)$ prior; and (b) taking the entire posterior distribution of θ (i.e. $Beta(7,5)$) into account when calculating the above

probability. Comment on the results and explain why the resulting probabilities are different. Hint: for (a), one can use the R command `1-pbinom(20,30,0.58)` to calculate the required probability, whilst for (b), one needs to add the following six lines of WinBUGS code to Figure 5.7 (between Line 10 and Line 11 there) to do the prediction then calculate the required probability:

```
# make a prediction for the future 30 flips
y.pred <- dbin(theta,30)
# why subtract y.pred from 20.7 (not 20)?
d <- y.pred - 20.7
# Pr(number of heads over 30 flips > 20 | data))
prob.gt.20 <- step(d)
```

Exercise 5.6. Comment on the similarity between Exercise 5.5 above and Exercise 2.6 in Chapter 2.

Exercise 5.7. Use R to explore the properties (the support, the distributional shape, the mean and the variance) of the following Gamma distributions: (a) *Gamma*(0.01,0.01); (b) *Gamma*(1,1); (c) *Gamma*(10,10); and (d) *Gamma*(10,5). If $X \sim \text{Gamma}(a,b)$, then the mean of X is a/b and the variance of X is a/b^2 . Some useful R functions are

- 1) `rgamma(n,a,b)` simulates n values randomly from a *Gamma*(a,b) distribution.
- 2) `dgamma(x,a,b)` calculates the density function for a *Gamma*(a,b) distribution at a given value x .
- 3) `pgamma(x,a,b)` calculates the probability $\Pr(X \leq x)$ for a given value of x and $X \sim \text{Gamma}(a,b)$.
- 4) `qgamma(p,a,b)` returns a value x such that $\Pr(X \leq x) = p$ for a given probability of p .

Exercise 5.8. Carry out the analysis of the income data in WinBUGS using the model presented in Eq. 5.17; pay particular attention to checks of convergence and efficiency and the form of model checking given in Figure 5.18. Monitor the DIC value.

Exercise 5.9. Repeat the modelling of the income data but replace the normal distribution in the likelihood by the t_4 distribution. Compare the results (e.g. the posterior estimates of the model parameters and the fit to the data) with those obtained from the model with the normal likelihood (see Exercise 11.2 in Chapter 11 for more detail).

Exercise 5.10. Analyse the burglary count data at the census output area (COA) level in Peterborough in 2005 using the Poisson regression model presented in Eq. 5.18 and Eq. 5.19. Investigate the goodness of fit to the data from the model via the posterior predictive check (Eq. 5.20). Comment on which parts of the data the model fits well and which parts of the data the model fits poorly. (Hint: for an area where the model fits poorly, compare the posterior predictive distribution of the case count with the actual observed count to understand whether the model predicts too many or too few cases.)

Exercise 5.11. The dataset `PHIA_with_covariates.csv`, on the book's website, contains the binary 0/1 outcome values that we discussed in Section 5.2.2. Also contained within that dataset are four COA-level covariates:

- 1) ethnic: index of ethnic heterogeneity that varies between 0 and 1, where the larger the value the greater the ethnic mix in a COA
- 2) carvan: percentage of households in a COA without a car or van
- 3) turnover: percentage of population living in a COA who in the previous year were living at another address either in the same COA or elsewhere
- 4) lonepar: percentage of households in a COA classified as a single-parent family

Extend the logistic regression model presented in Section 5.2.2 to include all four covariates and fit that model to the PHIA data. Comment on the estimated covariate effects. (Hint: the WinBUGS syntax $y[i] \sim \text{dbern}(\pi[i])$ specifies the Bernoulli likelihood for modelling the 0/1 binary outcome value $y[i]$ and $\text{logit}(\pi[i]) \leftarrow \alpha + \beta \text{ethnic}[i]$ specifies the regression relationship on the logit scale in Eq. 5.9.)

Exercise 5.12. Repeat the analysis in Exercise 5.11 but using the dataset `EHIA_with_covariates.csv`, which contains another set of COA-level binary outcome values that indicate whether a COA was considered as an HIA based on recorded crime data (the letter E in EHIA represents “empirical”). We will return to this dataset with more detail in Section 9.2.

Exercise 5.13. Suppose the values in each of the following three sets of data are independently drawn from a common normal distribution with an unknown mean and a known variance:

Set 1: 18, 4, 16, 1, -11, 0, -14, -32, 16, -19

Set 2: 842, 1166, 1346, 1091, 767, 1073, 1331, 1017, 996, 731, 816, 1219, 984, 797, 955, 916, 694, 919, 883, 937

Set 3: 19899, 33191, 13677, 33565, 20708, 24951, 26319, 50313, 40523, 34858, 37719, 35145, 40998, 33051, 34282, 17261, 22102, 7501, 41055, 27525

For each set of data, determine if each of the probability distributions listed below is a vague prior, a moderately informative prior, an informative prior or a prior distribution that may not be appropriate for the unknown mean:

(a) $N(0, 10^2)$; (b) $N(0, 100^2)$; (c) $N(0, 10000^2)$; (d) $N(0, 100000^2)$;

(e) $\text{Uniform}(0, 100)$; (f) $\text{Uniform}(0, 10000)$; and (g) $\text{Uniform}(0, 1000000)$.

Exercise 5.14. In Chapter 1, we stated the structure of a Bayesian hierarchical model in the following form:

$$\begin{aligned} \Pr(\text{process}, \text{parameters} \mid \text{data}) &\propto \Pr(\text{data} \mid \text{process}, \text{parameters}) \\ &\times \Pr(\text{process} \mid \text{parameters}) \\ &\times \Pr(\text{parameters}) \end{aligned}$$

with the three components, $\Pr(\text{data}, \text{process} \mid \text{parameters})$, $\Pr(\text{process} \mid \text{parameters})$ and $\Pr(\text{parameters})$, denoting the data model, the process model and the parameter model, respectively. Give a proof of the decomposition of $\Pr(\text{process}, \text{parameters} \mid \text{data})$ using Bayes’ theorem.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>