Chapter 1

Introduction

Chapter 2

Bayesian Analysis of GL(M)Ms Using R/WinBUGS

A major theme of this book is that spatial capture-recapture models are, for the most part, just generalized linear models (GLMs) wherein the covariate, distance between trap and home range center, is partially or fully unobserved – and therefore regarded as a random effect. Such models are usually referred to as Generalized Linear Mixed Models (GLMMs) and, therefore, SCR models can be thought of as a specialized type of GLMM. Naturally then, we should consider analysis of these slightly simpler models in order to gain some experience and, hopefully, develop a better understanding of spatial capture-recapture models.

In this chapter, we consider classes of GLM models - Poisson and binomial (i.e., logistic regression) GLMs - that will prove to be enormously useful in the analysis of capture-recapture models of all kinds. Many readers are probably familiar with these models because they represent probably the most generally useful models in all of Ecology and, as such, have received considerable attention in many introductory and advanced texts. We focus on them here in order to introduce the readers to the analysis of such models in **R** and **WinBUGS**, which we will translate directly to the analysis of SCR models in subsequent chapters.

Bayesian analysis is convenient for analyzing GLMMs because it allows us to work directly with the conditional model – i.e., the model that is conditional on the random effects, using computational methods known as Markov chain Monte Carlo (MCMC). Learning how to do Bayesian analysis of GLMs and GLMMs in **WinBUGS** is, in part, the purpose of this chapter. While we use **WinBUGS** to do the Bayesian computations, we organize and summarize our data and execute **WinBUGS** from within **R** using the useful package `R2WinBUGS`

(Sturtz et al., 2005). Kéry (2010), and Kery and Schaub (2011) provide excellent introductions to the basics of Bayesian analysis and GLMs at an accessible level. We don't want to be too redundant with those books and so we avoid a detailed treatement of Bayesian methodology - instead just providing a cursory overview so that we can move on and attack the problems we're most interested in related to spatial capture-recapture. In addition, there are a number of texts that provide general introductions to Bayesian analysis, MCMC, and their applications in Ecology including McCarthy (2007), Kéry (2010), Link and Barker (2009),and King (2009).

While this chapter is about Bayesian analysis of GLMMs, such models are routinely analyzed using likelihood methods too, as discussed by Royle and Dorazio (2008), and Kéry (2010). Indeed, likelihood analysis of such models is the primary focus of many applied statistics texts, a good one being Zuur et al. (2009). Later in this book, we will use likelihood methods to analyze SCR models but, for now, we concentrate on providing a basic introduction to Bayesian analysis because that is the approach we will use in a majority of cases in later chapters.

2.1 Notation

We will sometimes use conventional "bracket notation" to refer to probability distributions. If y is a random variable the $[y]$ indicates its distribution or its probability density/mass function (pdf, pmf) depending on context. If x is another random variable then $[y|x]$ is the conditional distribution of y given x, and $[y, x]$ is the joint distribution of y and x. To differentiate specific distributions in some contexts we might label them $g(y)$, $g(y|\theta)$, $f(x)$, or similar. We will also write $y \sim \text{Normal}(\mu, \sigma^2)$ to indicate that y "is distributed as" a normal random variable with parameters μ and σ^2. The expected value or mean of a random variable is $E[y] = \mu$,and $Var[y] = \sigma^2$ is the variance of y. To indicate specific observations we'll use an index such as "i". So, y_i for $i = 1, 2, \ldots, n$ indicates observations for n individuals. Finally, we write $\Pr(y)$ to indicate specific probabilities, i.e., of events "y" or similar.

To illustrate these concepts and notation, suppose z is a binary outcome (e.g., species occurrence) and we might assume the model: $z \sim \text{Bern}(p)$ for observations. Under this model $\Pr(z = 1) = \psi$, which is also the expected value $E[z] = \psi$. The variance is $Var[z] = \psi * (1-\psi)$ and the probability mass function (pmf) is $[z] = \psi^z(1 - \psi)^{1-z}$. Sometimes we write $[z|\psi]$ when it is important to emphasize the conditional dependence of z on ψ. As another example, suppose y is a random variable denoting whether or not a species is detected if an occupied site is surveyed. In this case it might be natural to express the pmf of the observations y *conditional* on z. That is, $[y|z]$. In this case, $[y|z = 1]$ is the conditional pmf of y given that a site is occupied, and it is natural to assume that $[y|z = 1] = \text{Bern}(p)$ where p is the "detection probability" - the probability that we detect the species, given that it is present. The model for the observations y is completely specified once we describe the other conditional

pmf $[y|z = 0]$. For this conditional distribution it is sometimes reasonable to assume $\Pr(y = 1|z = 0) = 0$ (MacKenzie et al. (2002); see also Royle and Link (2006)). That is, if the species is absent, the probability of detection is 0. This implies that $\Pr(y = 0|z = 0) = 1$. To allow for situations in which the true state z is unobserved, we assume that $[z]$ is Bernoulli with parameter ψ. In this case, the marginal distribution of y is

$$[y] = [y|z = 1]Pr(z = 1) + [y|z = 0]Pr(z = 0)$$

because $[y|z = 0]$ is a point mass at $y = 0$, by assumption, the marginal probability that $y = 1$ is

$$\Pr(y = 1) = p\psi$$

and the marginal probability that $y = 0$ is

$$\Pr(y = 0) = (1 - p) * \psi + (1 - \psi)$$

2.2 GLMs and GLMMs

We have asserted already that SCR models work out most of the time to be variations of GLMs and GLMMs. Some of you might therefore ask: What are GLMs and GLMMs, anyhow? These models are covered extensively in many very good applied statistics books and we refer the reader elsewhere for a detailed introduction. We think Kéry (2010), Kery and Schaub (2011), and Zuur et al. (2009) are all accessible treatments of considerable merit. Here, we'll give the 1 minute treatment of GLMMs, not trying to be complete but rather only to preserve a coherent organization to the book.

The generalized linear model (GLM) is an extension of standard linear models by allowing the response variable to have some distribution from the exponential family of distributions (i.e., not just normal). This includes the normal distribution but also dozens of others such as the Poisson, binomial, gamma, exponential, and many more. In addition, GLMS allow the response variable to be related to the predictor variables (i.e., covariates) using a link function, which is usually nonlinear. Finally, GLMs typically accommodate a relationship between the mean and variance. The classical reference for GLMs is Nelder and Wedderburn (1972) and also McCullagh and Nelder (1989). The GLM consists of three components:

1. A probability distribution for the dependent variable y, from a class of probability distributions known as the exponential family.

2. A "linear predictor" $\eta = \mathbf{X}\boldsymbol{\beta}$.

3. A link function g that relates $E[y]$ to the linear predictor, $E[y] = \mu = g^{-1}(\eta)$. Therefore $g(E[y]) = \eta$.

The dependent variable y is assumed to be an outcome from a distribution of the exponential family which includes many common distributions including

the normal, gamma, Poisson, binomial, and many others. The mean of the distribution of y is assumed to depend on predictor variables x according to

$$g(E[y]) = \mathbf{x}'\boldsymbol{\beta}$$

where $E[y]$ is the expected value of y, and $\mathbf{x}'\boldsymbol{\beta}$ is termed the *linear predictor*, i.e., a linear function of the predictor variables with unknown parameters $\boldsymbol{\beta}$ to be estimated. The function g is the link function. In standard GLMs, the variance of y is a function V of the mean of y: $Var(y) = V(\mu)$ (see below for examples).

A Poisson GLM posits that $y \sim \text{Poisson}(\lambda)$ with $E[y] = \lambda$ and usually the model for the mean is specified using the *log link function* by

$$log(\lambda_i) = \beta_0 + \beta_1 * x_i$$

The variance function is $\text{V}(y_i) = \lambda_i$. The binomial GLM posits that $y_i \sim \text{Binomial}(K, p)$ where K is the fixed sample size parameter and $E[y_i] = K * p_i$. Usually the model for the mean is specified using the *logit link function* according to

$$logit(p_i) = \beta_0 + \beta_1 * x_i$$

Where $logit(u) = log(u/(1-u))$. The inverse-logit function, g^{-1} , is a function we will refer to as "expit", so that $expit(u) = exp(u)/(1 + exp(u))$.

A GLMM is the extension of GLMs to accommodate "random effects". Often this involves adding a normal random effect to the linear predictor, and so a simple example is:

$$\log(\lambda_i) = \alpha_i + \beta_1 * x_i$$

where

$$\alpha_i \sim \text{Normal}(\mu, \sigma^2)$$

2.3 Bayesian Analysis

Bayesian analysis is unfamiliar to many ecological researchers because older cohorts of ecologists were largely educated in the classical statistical paradigm of frequentist inference. But advances in technology and increasing exposure to benefits of Bayesian analysis are fast making Bayesians out of people or at least making Bayesian analysis an acceptable, general, alternative to classical, frequentist inference.

Conceptually, the main thing about Bayesian inference is that it uses probability directly to characterize uncertainty about things we don't know. "Things", in this case, are parameters of models and, just as it is natural to characterize uncertain outcomes of stochastic processes using probability, it seems natural also to characterize information about unknown "parameters" using probability. At least this seems natural to us and, we think, most ecologists either explicitly adopt that view or tend to fall into that point of view naturally. Conversely, frequentists use probability in many different ways, but never to characterize

uncertainty about parameters[1] Instead, frequentists use probability to characterize the behavior of *procedures* such as estimators or confidence intervals (see below), which can lead to some inelegant or unnatural interpretations of things. It is paradoxical that people readily adopt a philosophy of statistical inference in which the things you don't know (i.e., parameters) should *not* be regarded as random variables, so that, as a consequence, one cannot use probability to characterize ones state of knowledge about them.

2.3.1 Bayes Rule

As its name suggests, Bayesian analysis makes use of Bayes' rule in order to make direct probability statements about model parameters. Given two random variables z and y, Bayes rule relates the two conditional probability distributions $[z|y]$ and $[y|z]$ by the relationship:

$$[z|y] = [y|z][z]/[y]$$

Bayes' rule itself is a mathematical fact and there is no debate in the statistical community as to its validity and relevance to many problems. Generally speaking, these distributions are characterized as follows: $[y|z]$ is the conditional probability distribution of y *given* z, $[z]$ is the marginal distribution of z and $[y]$ is the marginal distribution of y. In the context of Bayesian inference we usually associate specific meanings in which $[y|z]$ is thought of as "the likelihood", $[z]$ as the "prior" and so on. We leave this for later because here the focus is on this expression of Bayes rule as a basic fact of probability.

As an example of a simple application of Bayes rule, consider the problem of determining species presence at a sample location based on imperfect survey information. Let z be a binary random variable that denotes species presence ($z = 1$) or absence ($z = 0$), let $\Pr(z = 1) = \psi$ where ψ is usually called occurrence probability, "occupancy" (MacKenzie et al., 2002) or "prevalence". Let y be the *observed* presence ($y = 1$) or absence ($y = 0$), and let p be the probability that a species is detected in a single survey at a site given that it is present. Thus, $\Pr(y = 1|z = 1) = p$. The interpretation of this is that, if the species is present, we will only observe presence with probability p. In addition, we assume here that $\Pr(y = 1|z = 0) = 0$. That is, the species cannot be detected if it is not present which is a conventional view adopted in most biological sampling problems (but see Royle and Link (2006)). If we survey a site T times but never detect the species, then this clearly does not imply that the species is not present ($z = 0$) at this site. Rather, our degree of belief in $z = 0$ should be made with a probabilistic statement $\Pr(z = 1|y_1 = 0, \ldots, y_T = 0)$. If the T surveys are independent so that we might regard y_t as *iid* Bernoulli trials, then the total number of detections, say y, is Binomial with probability p then we can use Bayes rule to compute the probability that it is present given that

[1] To hear this will be shocking to some readers perhaps.

it is not detected in T samples. In words, the expression we seek is:

$$\Pr(\text{present}|\text{not detected}) = \frac{\Pr(\text{not detected}|\text{present})\Pr(\text{present})}{\Pr(\text{detected})}$$

Mathematically, this is

$$\begin{aligned}\Pr(z=1|y=0) &= \Pr(y=0|z=1)\Pr(z=1)/\Pr(y=0)\\ &= [(1-p)^T\psi]/[(1-p)^T\psi+(1-\psi)].\end{aligned}$$

To apply this, suppose that $T=2$ surveys are done at a wetland for a species of frog, and the species is not detected there. Suppose further that $\psi=.8$ and $p=.5$ are obtained from a prior study. Then the probability that the species is present at this site is $.25*.8/(.25*.8+.2)=0.50$. That is, there seems to be about a 50/50 chance that the site is occupied despite the fact that the species wasn't observed there.

In summary, Bayes' rule provides a simple linkage between the conditional probabilities $[y|z]$ and $[z|y]$ which is useful whenever one needs to deduce one from the other. Bayes' rule as a basic fact of probability is not disputed.

2.3.2 Bayesian Inference

What is controversial to some is the scope and manner in which Bayes rule is applied by Bayesian analysts. Bayesian analysts assert that Bayes rule is relevant, in general, to all statistical problems by regarding all unknown quantities of a model as realizations of random variables - this includes "data", latent variables, and also "parameters". Classical (non-Bayesian) analysts sometimes object to regarding "parameters" as outcomes of random variables. Classically, parameters are thought of as "fixed but unknown" (using the terminology of classical statistics). Of course, in Bayesian analysis they are also unknown and, in fact, there is a single data-generating value and so they are also fixed. The difference is that this fixed but unknown value is regarded as having been generated from some probability distribution. Specification of that probability distribution is necessary to carryout Bayesian analysis, but it is not required in classical frequentist inference.

To see the general relevance of Bayes rule in the context of statistical inference, let y denote observations - i.e., "data" - and let $[y|\theta]$ be the observation model (often colloquially referred to as the "likelihood"). Suppose theta is a parameter of interest having (prior) probability distribution $[\theta]$. These are combined to obtain the posterior distribution using Bayes' rule, which is:

$$[\theta|y] = [y|\theta][\theta]/[y]$$

Asserting the general relevance of Bayes rule to all statistical problems, we can conclude that the two main features of Bayesian inference are that: (1) "parameters" θ are regarded as realizations of a random variable and, as a result, (2) inference is based on the probability distribution of the parameters

given the data, $[\theta|y]$, which is called the posterior distribution. This is the result of using Bayes rule to combine "the likelihood" and the prior distribution. The key concept is regarding parameters as realizations of a random variable because, once you admit this conceptual view, this leads directly to the posterior distribution, a very natural quantity upon which to base inference about things we don't know - including parameters of statistical models. In particular, $[\theta|y]$ is a probability distribution for θ and therefore we can make direct probability statements to characterize uncertainty about θ.

The denominator of our invocation of Bayes rule, $[y]$, is the marginal distribution of the data y. We note without further remark right now that, in many practical problems, this can be an enormous pain to compute. The main reason that the Bayesian paradigm has become so popular in the last 20 years or so is because methods have been developed for characterizing the posterior distribution that do not require that we possess a mathematical understanding of $[y]$, i.e., we never have to compute it or know what it looks like, or know anything specific about it.

A common misunderstanding on the distinction between Bayesian and frequentist inference goes something like this "in frequentist inference parameters are fixed but unknown but in a Bayesian analysis parameters are random." At best this is a sad caricature of the distinction and at worst it is downright wrong. What is true is that, to a Bayesian, parameters are random variables. However, a Bayesian assumes, just like a frequentist, that there was a single data-generating value of that parameter - a fixed, and unknown value that produced the given data set. The distinction between Bayesian and frequentist approaches is that Bayesians regard the parameter as a random variable, and its value as the outcome of a random value, on par with the observations. This allows Bayesians to use probability to make direct probability statements about parameters. Frequentist inference procedures do not permit direct probability statements to be made about parameter values – because parameters are not random variables!

While we can understand the conceptual basis of Bayesian inference merely by understanding Bayes rule – that's really all there is to it – it is not so easy to understand the basis of classical "frequentist" inference which is mostly like [2] a "basket of methods" with little coherent organization. What is mostly coherent in frequentist inference is the manner in which items in this basket of methods are evaluated – the performance of a given procedure is evaluated by "averaging over" hypothetical realizations of y, regarding the *estimator* as a random variable. For example, if $\hat{\theta}$ is an estimator of θ then the frequentist is interested in $E_y[\hat{\theta}|y]$ which is used to characterize bias. If the expected value of $\hat{\theta}$, when averaged over realizations of y, is equal to θ, then $\hat{\theta}$ is unbiased.

The view of parameters as fixed constants and estimators as random variables leads to interpretations that are not so straightforward. For example

[2] as characterized by Christopher A. Sims, winner of the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel in his Hotelling Lecture 6/29/2007 at Duke University - `http://sims.princeton.edu/yftp/EmetSoc607/AppliedBayes.pdf`: "Bayesian inference is a way of thinking, not a basket of 'Methods"'

confidence intervals having the interpretation "95% probability that the interval contains the true value" and p-values being "the probability of observing an outcome as extreme or more than the one observed." These are far from intuitive interpretations to most people. Moreover, this is conceptually problematic to some because the hypothetical realizations that characterize the performance of our procedure we will never get to observe.

While we do tend to favor Bayesian inference for the conceptual simplicity (parameters are random, posterior inference), we mostly advocate for a pragamatic non-partisian approach to inference because, frankly, some of these "bucket of methods" are actually very convenient in certain situations as we will see in later chapters.

2.3.3 Prior distributions

The prior distribution $[\theta]$ is an important feature of Bayesian inference. As a conceptual matter, the prior distribution characterizes "prior beliefs" or "prior information" about a parameter. Indeed, an oft-touted benefit of Bayesian analysis is the ease with which prior information can be included in an analysis. However, more commonly, the prior is chosen to express a lack of prior information, even if previous studies have been done and even if the investigator does in fact know quite a bit about a parameter. This is because the manner in which prior information is embodied in a prior (and the amount of information) is usually very subjective and thus the result can wind up being very contentious, e.g., different investigators might report different results based on subjective assessments of things. Thus it is usually better to "let the data speak" and use priors that reflect absence of information beyond the data set being analyzed.

But still the need occasionally arises to embody prior information or beliefs about a parameter formally into the estimation scheme. In SCR models we often have a parameter that is closely linked to "home range radius" and thus auxiliary information on the home range size of a species can be used as prior information (e.g., see Chandler and Royle (2012) ; also chapter XYZ).

Other times the situation arises where a prior can inadvertently impose substantial information on a parameter and that might not be desireable. For example, we use data augmentation to deal with the fact that N is an unknown parameter (**?**) which is equivalent to imposing a $Bin(M, \psi)$ prior on N for some integer M. One has to take care to make sure that M is sufficiently large so as to not affect the posterior distribution (see Section XYZ XYZ). Another situation that we have to be careful of is that a noninformative prior on one scale is informative on another scale. For example, if we have a flat prior on $logit(p)$ for some probability parameter p, this is very different from having a uniform(0,1) prior on p. We show an example where this makes a difference in Chapter 4 XXXXX. Reference to non-invariance of prior distributions to transformation..... XXXXXXXX

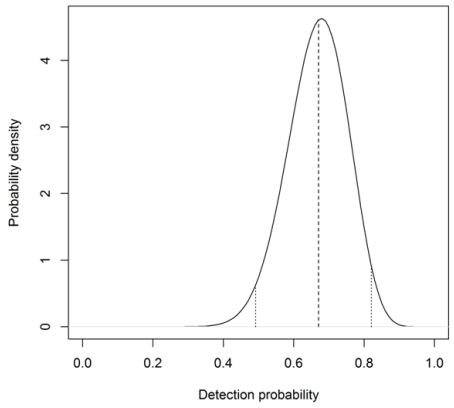

Figure 2.1: Probability density plot of a hypothetical posterior distribution of beta(20,10); dashed lines indicate mean and upper and lower 95% interval

2.3.4 Posterior Inference

In Bayesian inference, we are not focusing on estimating a single point or interval but rather on characterizing a whole distribution – the posterior distribution – from which one can report any summary of interest. A point estimate might be the posterior mean, median, mode, etc.. In many applications in this book, we will compute 95% Bayesian intervals using the 2.5% and 97.5% quantiles of the posterior distribution. For such intervals, it is correct to say $\Pr(L < \theta < U) = 0.95$. That is, "the probability that θ is between L and U is 0.95".

As an example, suppose we conducted a Bayesian analysis to estimate detection probability of some species at a study site (p), and we obtained a posterior distribution of beta(20,10) for the parameter p. The following **R** commands demonstrate how we make inferences based upon summaries of the posterior distribution. Fig. **??** shows the posterior along with the summary statistics.

```
> (post.median <- qbeta(0.5, 20, 10))
[1] 0.6704151
> (post.95ci <- qbeta(c(0.025, 0.975), 20, 10))
[1] 0.4916766 0.8206164
```

Thus, we can state that there is a 95% probability that θ lies between 0.49 and 0.82.

It is not a subtle thing that this cannot be said using frequentist methods - although people tend to say it anyway and not really understand why it is wrong or even that it is wrong. This is actually a failing of frequentist ideas and the inability of frequentists to get people to overcome their natural tendency to

use probability - which is something that, as a frequentist, you simply cannot do in the manner that you would like to.

As a conceptual matter, Bayesian inference based on the posterior allows us to make an inference conditional on the data that we actually observed - i.e., what we actually know. To us, this seems logical - to condition on what we know. Conversely, frequentist inference is based on considering average performance over hypothetical unobserved data sets (i.e., the "relative frequency" interpretation of probability). Frequentists know that their procedures work well when averaged over all hypothetical, unobserved, data sets but no one ever really knows how well they work for the specific data set analyzed. That seems like a relevant question to biologists who oftentimes only have their one, extremely valuable, data set.

2.3.5 Small sample inference

Using Bayesian inference, we obtain an estimate of the posterior distribution which is an exhaustive summary of the state-of-knowledge about an unknown quantity. It is the posterior distribution - not an estimate of that thing. It is also not, usually, an approximation except to within Monte Carlo error (in cases where we use simulation to calculate it). One of the great virtues of Bayesian analysis which is not really appreciated is that it is completely valid for any particular sample size. i.e., it is $[\theta|y]$, as precise as we claim it to be based on our ability to do calculations, for the particular sample size and observations that we have even if we have only a single datum y. The same cannot be said for almost all frequentist procedures in which estimates or variances are very often (almost always in practice) based on "asymptotic approximations" to the procedure which is actually being employed.

There seems to be a prevailing view in statistical ecology that classical likelihood-based procedures are virtuous because of the availability of simple formulas and procedures for carrying out inference, such as calculating standard errors, doing model selection by AIC, and assessing goodness-of-fit. In large samples, this may be an important practical benefit, but the theoretical validity of these procedures cannot be asserted in most situations involving small samples. This is not a minor issue because it is typical in many wildlife sampling problems - especially in surveys of carnivores or rare/endangered species - to wind up with a small, sometimes extremely small, data set (**?**). For example, a recent paper on the fossa (Cryptoprocta ferox), an endangered carnivore in Madagascar, estimated an adult density of 0.18 adults / km sq based on 20 animals captured over 3 years (Hawkins and Racey, 2005). A similar paper on the endangered southern river otter estimated a density of 0.25 animals per river km based on 12 individuals captured over 3 years (Sepúlveda et al., 2007). **?** analyzed data from a study of the Pampas cat, a species for which very little is known, wherein only 22 individual cats were captured .during the two year period. Trolle and Kéry (2005) reported only 9 individual ocelots captured and Jackson et al. (2006) captured 6 individual snow leopards using camera trapping. Thus, almost all likelihood-based analysis of data on rare and/or secretive

carnivores necessarily and flagrantly violate one of Le Cam's Basic Principles, that of "If you need to use asymptotic arguments, do not forget to let your number of observations tend to infinity."(Le Cam, 1990).

The biologist thus faces a dilemma with such data. On one hand, these datasets, and the resulting inference, are often criticized as being poor and unreliable. Or, even worse[3], "the data set is so small, this is a poor analysis." On the other hand, such data may be all that is available for species that are extraordinarily important for conservation and management. The Bayesian framework for inference provides a valid, rigorous, and flexible framework that is theoretically justifiable in arbitrary sample sizes. This is not to say that one will obtain precise estimates of density or other parameters, just that your inference is coherent and justifiable from a conceptual and technical statistical point of view. That is, we report the posterior probability $\Pr(D|data)$ which is easily interpretable and just what it is advertised to be and we don't need to do a simulation study to evaluate how well some approximate $\Pr(D|data)$ deviates from the actual $\Pr(D|data)$ because they are precisely the same quantity.

2.4 Characterizing posterior distributions by MCMC simulation

In practice, it is not really feasible to ever compute the marginal probability distribution $\Pr(y)$, the denominator resulting from application of Bayes' rule. For decades this impeded the adoption of Bayesian methods by practitioners. Or, the few Bayesian analyses done were based on asymptotic normal approximations to the posterior distribution. While this was useful stuff from a theoretical and technical standpoint and, practically, it allowed people to make the probability statements that they naturally would like to make, it was kind of a bad joke around the Bayesian water-cooler to, on one hand, criticize classical statistics for being, essentially, completely ad hoc in their approach to things but then, on the other hand, have to devise various approximations to what they were trying to characterize. The advent of Markov chain Monte Carlo (MCMC) methods has made it easier to calculate posterior distributions for just about any problem to arbitrary levels of precision.

Broadly speaking, MCMC is a class of methods for drawing random samples (i.e., simulating from or just "sampling") from the target posterior distribution. Thus, even though we might not recognize the posterior as a named distribution or be able to analyze its features analytically, e.g., devise mathematical expressions for the mean and variance, we can use these MCMC methods to obtain a large sample from the posterior and then use that sample to characterize features of the posterior. What we do with the sample depends on our intentions – typically we obtain the mean or median for use as a point estimate, and take a confidence interval based on Monte Carlo estimates of the quantiles.

[3]Actual quote from a referee

2.5 What Goes on Under the MCMC Hood

We will develop and apply MCMC methods in some detail for spatial capture-recapture models in chapter 7. Here we provide a simple illustration of some basic ideas related to the practice of MCMC.

A type of MCMC method relevant to most problems is Gibbs sampling (**?**) which we address in more detail in Chapt. 7. Gibbs sampling involves iterative simulation from the "full conditional" distributions (also called conditional posterior distributions). The full conditional distribution for an unknown quantity is the conditional distribution of that quantity given every other random variable in the model - the data and all other parameters. For example, for a normal regression model with $y \sim \text{Normal}(\alpha + \beta(x - \bar{x}), \sigma^2)$[4] where lets say σ^2 is known. The full conditionals are, in symbolic terms,

$$[\alpha|y, \beta]$$

and

$$[\beta|y, \alpha].$$

We might use our knowledge of probability to identify these mathematically. In particular, by Bayes' Rule, $[\alpha|y, \beta] = [y|\alpha, \beta][\alpha|\beta]/[y|\beta]$ and similarly for $[\beta|y, \alpha]$. For example, if we have priors for $[\alpha] = \text{Norm}(\mu_\alpha, \sigma^2_\alpha)$ and $[\beta] = \text{Norm}(\mu_\beta, \sigma^2_\beta)$ then some algebra reveals that

$$[\alpha|y, \beta] = \text{Norm}\left(w\bar{y} + (1 - w)\mu_\alpha, (\tau n + \tau_\alpha)^{-1}\right) \tag{2.1}$$

where $\tau = 1/\sigma^2$ and $\tau_\alpha = 1/\sigma^2_\alpha$ (the inverse of the variance is sometimes called *precision*), and $w = \tau n/(\tau n + \tau_\alpha)$. We see in this case that the posterior mean is a *precision-weighted* sum of the sample mean $\bar{y}$ and the prior mean μ_α, and the posterior *precision* is the sum of the precision of the likelihood and that of the prior. These results are typical of many classes of problems. In particular, note that as the prior precision tends to 0, i.e., $\tau_\alpha \to 0$, then the posterior of α tends to $\text{Norm}(\bar{y}, \sigma^2/n)$. We recognize the variance of this distribution as that of the variance of the sampling distribution of $\bar{y}$ and its mean is in fact the MLE of α for this model. The conditional posterior of β has a very similar form:

$$[\beta|y, \alpha] = \text{Norm}\left(\frac{\tau(\sum_i y_i(x_i - \bar{x})) + \tau_\beta\mu_\beta}{\tau\sum_i(x_i - \bar{x})^2 + \tau_\beta}, (\tau\sum_i(x_i - \bar{x})^2 + \tau_\beta)^{-2}\right) \tag{2.2}$$

which might look slightly unfamiliar, but note that if $\tau_\beta = 0$, then the mean of this distribution is the familiar $\hat{\beta}$, and the variance is, in fact, the sampling variance of $\hat{\beta}$.

The MCMC algorithm for this model has us simulate in succession, repeatedly, from those two distributions. See Gelman et al. (2004) for more examples of Gibbs sampling for the normal model, and we also provide another examplein

[4] We center the independent variable here so that things look more intuitive in the result

Chapt. 7. A conceptual representation of the MCMC algorithm for this simple model is therefore:

`Algorithm`: Gibbs Sampling for linear regression

`0. Initialize` α `and` β

`Repeat` {

`1. Draw a new value of` α `from Eq. ??`

`2. Draw a new value of` β `rom Eq. ??`

}

As we just saw for this simple "normal-normal" model it is sometimes possible to specify the full conditional distributions analytically. In general, when certain so-called conjugate prior distributions are chosen, the form of full conditional distributions is similar to that of the observation model. In this normal-normal case, the normal distribution for the mean parameters is the conjugate prior under the normal model, and thus the full-conditional distributions are also normal. This is convenient because, in such cases, we can simulate directly from them using standard methods (or **R** functions). But, in practice, we don't really ever need to know such things because most of the time we can get by using a simple algorithm, called the Metropolis-Hastings (henceforth "MH") algorithm, to obtain samples from these full conditional distributions without having to recognize them as specific, named, distributions. This gives us enormous freedom in developing models and analyzing them without having to resolve them mathematically because to implement the MH algorithm we need only identify the full conditional distribution up to a constant of proportionality, that being the marginal distribution in the denominator (e.g., $[y|\beta]$ above).

We will talk about the Metropolis-Hastings algorithm shortly, and we will use it extensively in the analysis of SCR models (e.g., chapter 7).

2.5.1 Rules for constructing full conditional distributions

The basic strategy for constructing full-conditional distributions for devising MCMC algorithms can be reduced conceptually to a couple of basic steps summarized as follows:

(step 1) Collect all stochastic components of the model;

(step 2) Recognize and express the full conditional in question as proportional to the product of all components;

(step 3) Remove the ones that don't have the focal parameter in them.

(step 4) Do some algebra on the result in order to identify the resulting pdf or pmf.

Of the 4 steps, the last of those is the main step that requires quite a bit of statistical experience and intuition because various algebraic tricks can be used to reshape the mess into something noticeable - i.e., a standard, named distribution.

But step 4 is not necessary if we decide instead to use the Metrpolis-Hastings algorithm as described below.

In the context of our simple linear regression model that we've been working with, to characterize $[\alpha|y,\beta]$ we first apply step 1 and identify the model components as: $[y|\alpha,\beta]$, $[\alpha]$ and $[\beta]$. Step 2 has us write $[\alpha|y,\beta] \propto [y|\alpha,\beta][\alpha][\beta]$. Step 3: We note that $[\beta]$ is not a function of α and therefore we remove it to obtain $[\alpha|y,\beta] \propto [y|\alpha,\beta][\alpha]$. Similarly we obtain $[\beta|y,\alpha] \propto [y|\alpha,\beta][\beta]$. We apply step 4 and manipulate these algebraically to arrive at the result (which we provided in Eqs. **??** and **??**) or, alternatively, we can sample them indirectly using the Metropolis-Hastings algorithm, which we discuss now.

2.5.2 Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm is a completely generic method for sampling from any distribution, say $f(\theta)$. In our applications, $f(\theta)$ will typically be the full conditional distribution of θ. While we sometimes use Gibbs sampling, we seldom use "pure" Gibbs sampling because we might use MH to sample from one or more of the full conditional distributions. When the MH algorithm is used to sample from full conditional distributions of a Gibbs sampler the resulting hybrid algorithm is called *Metrpolized Gibbs sampling* or more commonly *Metropolis-within-Gibbs*. Shortly we will actually construct such an algorithm for a simple class of models.

The MH algorithm generates candidates from some proposal or candidate-generating distribution, that may be conditional on the current value of the parameter, denoted by $h(\theta^*|\theta^t)$. Here, θ^* is the *candidate* or proposed value and θ^t is the current value, i.e., at iteration t of the MCMC algorithm. The proposed value is accepted with probability

$$r = \frac{f(\theta^*)h(\theta^t|\theta^*)}{f(\theta^t)h(\theta^*|\theta^t)}$$

which we call the MH acceptance probability. This ratio can sometimes be > 1 in which case we set it equal to 1. It is useful to note that $h()$ can be anything at all. No matter the choice of $h()$, we can evaluate this ratio numerically because the marginal $f(y)$ cancels from both the numerator and denominator, which is the magic of the MH algorithm.

2.6 Practical Bayesian Analysis and MCMC

There are a number of really important practical issues to be considered in any Bayesian analysis and we cover some of these briefly here.

2.6.1 Choice of prior distributions

Bayesian analysis requires that we choose prior distributions for all of the structural parameters of the model (we use the term structural parameter to mean

all parameters that aren't customary thought of as latent variables). We will strive to use priors that are meant to express little or no prior information - default or customary "non-informative" or diffuse priors. This will be Unif(a, b) priors for parameters that have a natural bounded support and, for parameters that live on the real line we use either (1) diffuse normal priors; (2) "improper" uniform priors or (3) sometimes even a bounded Unif(a, b) prior if that greatly improves the performance of **WinBUGS** or other software doing the MCMC for us. In **WinBUGS** a prior with low "precision", τ, where $\tau = 1/\sigma^2$, such as Norm$(0, .01)$ will typically be used. Of course $\tau = 0.01$ ($\sigma^2 = 100$) might be very informative for a regression parameter that has a high variance. Therefore, we recommend that predictor variables *always* be standardized. Clearly there are a lot of choices for ostensibly non-informative priors, and the degree of non-informativeness depends on the parameterization. For example, a natural non-informative prior for the intercept of a logistic regression

$$\text{logit}(p_i) = \alpha + \beta x_i$$

would be a very diffuse normal prior, $[\alpha] = \text{Norm}(0, \text{Large})$ or even $\alpha \sim \text{Unif}(-\text{Large}, \text{Large})$. However, we might also use a prior on the parameter $p_0 = logit^{-1}(\alpha)$, which is $\Pr(y = 1)$ for the value $x = 0$. Since p_0 is a probability a natural choice is $p_0 \sim \text{Unif}(0, 1)$. These priors are very different in their implications. For example, if we choose the normal prior for α with variance $\text{Large} = 5^2$ and look at the implied prior for p_0 we have the result shown in Fig. **??** which looks nothing like a Uniform(0,1) prior. These two priors can affect results (see Chapt. 3 sec. XXXXX), yet they are both sensible non-informative priors. Choice of priors and parameterization is very much problem-specific and often largely subjective. Moreover, it also affects the behavior of MCMC algorithms and therefore the analyst needs to pay some attention to this issue and possibly try different things out. Most standard Bayesian analysis books address issues related to specification and affect of prior distribution choice in some depth. Some good references include **?**, **?** and Link and Barker (2009).

2.6.2 Convergence and so-forth

Once we have carried-out an analysis by MCMC, there are many other practical issues that we have to confront. One of the most important is "have the chains converged?" Since we do not know what the stationary posterior distribution of our Markov chain should look like (this is the whole point of doing an MCMC approximation), we effectively have no means to assess whether it has truly converged to this desired distribution or not. Most MCMC algorithms only guarantee that, eventually, the samples being generated will be from the target posterior distribution, but no-one can tell us how long this will take. Also, you only now the part of your posterior distribution that the Markov chain has explored so far for all you know the chain could be stuck in a local maximum, while other maxima remain completely undiscovered. Acknowledging that there is truly nothing we can do to ever proof convergence of our MCMC chains,

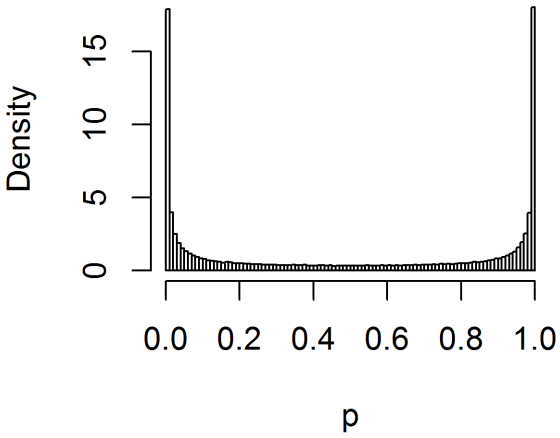

Figure 2.2: Implied prior for $p_0 = exp(\alpha)/(1 + exp(\alpha))$ if $\alpha \sim \text{Normal}(0, 5^2)$.

there are several things we can do to increase the degree of confidence we have about the convergence of our chains. Some problems are easily detected using simple plots. Typically a period of transience is observed in the early part of the MCMC algorithm, and this is usually discarded as the "burn-in" period. The quick diagnostic to whether convergence has been achieved is that your Markov chains look "grassy" – see Fig. 2.5 below. Another way to check convergence is to update the parameters some more and see if the posterior changes. Yet another option, and one generally implemented in **WinBUGS**, is to run several Markov chains and to start them off at different initial values that are overdispersed relative to the posterior distribution. Such initial values help to explore different areas of the parameter space simultaneously; if after a while all chains oscillate around the same average value, chances are good that they indeed converged to the posterior distribution. [5] Gelman and Rubin came

[5]Running several parallel chains is computationally expensive. But extra computational demands are not the only and by no means the major concern some people voice when it comes to running several parallel MCMC chains to assess convergence. Again, consider the fact that we do not know anything about the true form of the posterior distribution we are trying to approximate. How do we, then, know how to pick overdispersed initial values? We dont all we can do is pick overdispersed values relative to our expectations of what the posterior should look like. To use a quote from the home page of Charlie Geyer, a Bayesian statistician from the University of Minnesota, "If you don't know any good starting points [...], then restarting the sampler at many bad starting points is [...] part of the problem, not

up with the so-called"R-hat" statistic ($\hat{R}$) or Brooks-Gelman-Rubin statistic that essentially compares within-chain and between-chain variance to check for convergence of multiple chains ((Gelman et al., 1996)). $\hat{R}$ should be close to 1 if the Markov chains have converged and sufficient posterior samples have been obtained. In practice, $\hat{R} = 1.2$ is probably good enough for some problems. For some models you can't actually realize a low $\hat{R}$. E.g., if the posterior is a discrete mixture of distributions then you can be misled into thinking that your Markov chains have not converged when in fact the chains are just jumping back and forth in the posterior state-space. Another situation is when one of the parameters is on the boundary of the parameter space which might appear to be very poor mixing, but all within some extreme region of the parameter space.[6]. This kind of stuff is normally ok and you need to think really hard about the context of the model and the problem before you conclude that your MCMC algorithm is ill-behaved.

Some models exhibit "poor mixing" of the Markov chains or what people might also say "have not coverged" (or "slow convergence") which is a term we would disagree with because the samples might well be from the posterior (i.e., the Markov chains have converged to the proper stationary distribution) but simply mix around the posterior rather slowly. Anyway, poor mixing can happen for a huge number of reasons – when parameters are highly correlated (even confounded), or barely identified from the data, or the algorithms are very terrible and probably many other reasons. Slow mixing equates to high autocorrelation in the Markov chain - the successive draws are highly correlated, and thus we need to run the MCMC algorithm much longer to get an effective sample size that is sufficient for estimation - or to reduce the MC error to a tolerable level. A strategy often used to reduce autocorrelation is "thinning" - i.e., keep every m^{th} value of the Markov chain output. However, thinning is necessarily inefficient from the stand point of inference - you can always get more precise posterior estimates by using all of the MCMC output regardless of the level of autocorrelation (MacEachern and Berliner, 1994). Practical considerations might necessitate thinning, even though it is statistically inefficient. For example, in models with many parameters or other unknowns being tabulated, the output files might be enormous and unwieldy to work with. In such cases, thinning is perfectly reasonable. In many cases, how well the Markov chains mix is strongly influenced by parameterization, standardization of covariates, and the prior distributions being used. Some things work better than others,

part of the solution." (http://users.stat.umn.edu/ charlie/mcmc/diag.html). His suggestion is that your only chance to discover a potential problem with your MCMC sampler is to run it for a very long time. But again, there is no way of knowing how long is long enough. It is up to you to decide, which school of thoughts appeals more to you one long versus several parallel Markov chains. Irrespectively, part of developing an MCMC sampler should be to make sure (within reasonable limits) that you are not missing regions of high posterior density because of the way you specify your starting values. Once you have explored the behavior of your chain under a reasonable range of starting values, you may feel comfortable enough to run only one long chain.

[6]it would be nice if we could compile examples of this later in the book and reference back to this point

and the investigator should experiment with different settings and remain calm when things don't work out perfectly. MCMC is an art, and a science.

Is the posterior sample large enough? The subsequent samples generated from a Markov chain are not iid samples from the posterior distribution, due to the correlation amongst samples introduced by the Markov process and the sample size has to be adjusted to account for the autocorrelation in subsequent samples (see Chapt. 8 in **?** for more details). This adjusted sample size is referred to as the effective sample size. Checking the degree of autocorrelation in your Markov chains and estimating the effective sample size your chain has generated should be part of evaluating your model output. **WinBUGS** will automatically return the effective sample size for all monitored parameters. If you find that your supposedly long Markov chain has only generated a very short effective sample, you should consider a longer run. What exactly constitutes a reasonable effective sample size is hard to say. A more palpable measure of whether you've run your chain for enough iterations is the time-series or Monte Carlo error the 'noise' introduced into your samples by the stochastic MCMC process. The MC error is printed by default in summaries of **BUGS** output. You want that to be smallish relative to the magnitude of the parameter and this might depend on the purpose of the analysis. For a preliminary analysis you might settle for a few percent whereas for a final analysis then certainly less than 1% is called for, but you can run your MCMC algorithm as long as it takes. A consequence of the MC error is that even for the exact same model, results will always be different. Thus, as a good rule of thumb you should never report MCMC results to more than 2 decimal places.

2.6.3 Bayesian confidence intervals

The 95% Bayesian interval based on percentiles of the posterior is not a unique interval - there are many of them - and the so-called "highest posterior density" (HPD) interval is the narrowest interval. We might compute that frequently because it is easy to do with an integer parameter which N is (See the next chapter). The 95 % HPD is not often exactly 95% but usually slightly more conservative than nominal because it is the narrowest interval that contains at least 95% of the posterior mass.

2.6.4 Estimating functions of parameters

A benefit of analysis by MCMC is that we can seamlessly estimate functions of parameters by simply tabulating the desired function of the simulated posterior draws. For example, if θ is the parameter of interest and let $\theta^{(i)}$ for $i = 1, 2, \ldots, M$ be the posterior samples of θ. Let $\eta = exp(\theta)$, then a posterior sample of η can be obtained simply by computing $exp(\theta^{(i)})$ for $i = 1, 2, \ldots, M$. We give another example in section 2.7.2 below and throughout this book. Almost all SCR models in this book involve at least 1 derived parameter. For example, density D is a derived parameter, being a function of population size

N and the area A of the underlying state-space of the point process (see Chapt. 4).

2.7 Bayesian Analysis using WinBUGS

We won't be too concerned with devising our own MCMC algorithms for every analysis although we will do that a few times for fun. More often, we will rely on the freely available software package **WinBUGS** or **JAGS** for doing this. We will always execute these **BUGS** engines from within **R** using the `R2WinBUGS` (Sturtz et al., 2005) or `rjags` (**?**) pacages. **WinBUGS** and **JAGS** are MCMC black boxes that takes a pseudo-code description (i.e., the **BUGS** language) of all of the relevant stochastic and deterministic elements of a model and generates an MCMC algorithm for that model. But you never get to see the algorithm. Instead, **WinBUGS/JAGS** will run the algorithm and return the Markov chain output - the posterior samples of model parameters.

The great thing about using the **BUGS** language is that it forces you to become intimate with your statistical model - you have to write each element of the model down, admit (explicitly) all of the various assumptions, understand what the actual probability assumptions are and how data relate to latent variables and data and latent variables relate to parameters, and how parameters relate to one another.

While we normally use **WinBUGS** or **JAGS** in this book, we note that **OpenBUGS** is the current active development tree of the **BUGS** language. See Kéry (2010, ch.xyz) and Kery and Schaub (2011, appendix xyz) for more on practial analysis in **WinBUGS**. That book should also be consulted for a more comprehensive introduction to using **WinBUGS**. In this example, we're going to accelerate pretty fast.

2.7.1 Linear Regression in WinBUGS

We provide a brief introductory example of a normal regression model using a small simulated data set. The following commands are executed from within your **R** workspace. First, simulate a covariate x and observations y having prescribed intercept, slope and variance:

```
x<-rnorm(10)
mu<- -3.2+ 1.5*x
y<-rnorm(10,mu,sd=4)
```

The **BUGS** model specification for a normal regression model is written within **R** as a character string input to the command `cat()` and then dumped to a text file named `normal.txt`:

```
cat("
 model {
   for (i in 1:10){
```

```
      y[i]~dnorm(mu[i],tau)        # the "likelihood"
      mu[i]<- beta0 + beta1*x[i]   # the linear predictor
     }
   beta0~dnorm(0,.01)              # prior distributions
   beta1~dnorm(0,.01)
   sigma~dunif(0,100)
   tau<-1/(sigma*sigma)            # tau is a derived parameter
}
",file="normal.txt")
```

Alternatively, you can write the model specifications directly within a text file and save it in your current working directory, but we do not usually take that approach in this book.

Remarks: 1. The **BUGS** dialects parameterize the normal distribution in terms of the mean and inverse-variance, called the precision. Thus, `dnorm(0,.01)` implies a variance of 100; **2.** We typically use diffuse normal priors for mean parameters, β_0 and β_1 in this case, but sometimes we might use uniform priors with suitable bounds -B and +B. **3.** We typically use a Unif$(0, B)$ prior on standard deviation parameters (**?**). But sometimes we might use a gamma prior on the precision parameter τ. **4.** In a **BUGS** model file, every variable referenced in the model description has to be either data, which will be input (see below), a random variable which must have a probability distribution associated with it using the tilde character (aka "twiddle") "`~`", or it has to be a derived parameter connected to variables and data using "`<-`".

To fit the model, we need to describe various data objects to **WinBUGS**. In particular, we create an **R** list object called `data` which are the data objects identified in the BUGS model file. In the example, the data consist of two objects which exist as y and x in the **R** workspace and also in the **WinBUGS** model definition. We also have to create an **R** function that produces a list of starting values `inits` that get sent to **WinBUGS**. Finally, we identify the names of the parameters (labeled correspondingly in the **WinBUGS** model specification) that we want **WinBUGS** to save the MCMC output for. In this example, we will "monitor" the parameters β_0, β_1, σ and τ. **WinBUGS** is executed using the **R** command `bugs()`. We set the option `debug=TRUE` if we want the **WinBUGS** GUI to stay open (useful for analyzing MCMC output and looking at the **WinBUGS** error log). Also, we set `working.dir=getwd()` so that **WinBUGS** output files and the log file are saved in the current **R** working directory. All of these activities look like this:

```
library("R2WinBUGS")    # "attach" the R2WinBUGS library
data <- list ( "y","x")
inits <- function()
 list ( beta1=rnorm(1),beta0=rnorm(1),sigma=runif(1,0,2) )
parameters <- c("beta0","beta1","sigma","tau")
out<-bugs (data, inits, parameters, "normal.txt", n.thin=2, n.chains=2,
            n.burnin=2000, n.iter=6000, debug=TRUE,working.dir=getwd())
```

Remarks: A common question is "how should my data be formatted?" That depends on how you describe the model in the **BUGS** language, and how

your data are input into **R**. There is no unique way to describe any particular model and so you have some flexibility. We talk about data format further in the context of capture-recapture models and SCR models in Chapt. 4 and elsewhere. In general, starting values are optional but we recommend to always provide reasonable starting values for structural parameters, but are not always necessary for random effects. Note that the previously created objects defining data, initial values and parameters to monitor are passed to the function `bugs()`. In addition, various other things are declared: The number of Markov chains (`n.chains`), the thinning rate (`n.thin`), the number of burn-in iterations (`n.burnin`) and the total number of iterations (`n.iter`). To develop a detailed understanding of the various parameters and settings used for MCMC, consult a basic reference such as Kéry (2010).

You should execute all of the commands given above and then look at the resulting output. Kill the **WinBUGS** GUI and the data will be read back into **R** (or specify `debug=FALSE`). We don't want to give instructions on how to navigate and use the GUI - see XYZ XXX XXX REF (XYZ) for that. The object `out` prints important summaries by default (this is slightly edited):

```
> print(out,digits=2)
Inference for Bugs model at "normal.txt", fit using WinBUGS,
 2 chains, each with 6000 iterations (first 2000 discarded), n.thin = 2
 n.sims = 4000 iterations saved
          mean   sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
beta0    -2.43 1.84 -6.21 -3.50 -2.42 -1.34  1.27    1  4000
beta1     2.62 1.54 -0.42  1.68  2.62  3.57  5.67    1  4000
sigma     5.29 1.66  3.11  4.14  4.95  6.05  9.39    1  4000
tau       0.05 0.02  0.01  0.03  0.04  0.06  0.10    1  4000
deviance 59.85 3.24 56.18 57.47 59.00 61.37 68.32    1   840

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, pD = Dbar-Dhat)
pD = 2.6 and DIC = 62.4
```

Remarks: (1) convergence is assessed using the $\hat{R}$ statistic – which we might sometimes write "*Rhat*". A value of *Rhat* near 1 indicates convergence; (2) DIC is the "deviance information criterion" (Spiegelhalter et al., 2002) (see section 2.8) which some people use in a manner similar to AIC although it is recognized to have some problems in hierarchical models (Millar, 2009). We evaluate this in the context of SCR models in Chapt. 8.

2.7.2 Inference about functions of model parameters

Using the MCMC draws for a given model we can easily obtain the posterior distribution of any function of model parameters. We showed this in the above example by providing the posterior of τ when the model was parameterized in

terms of standar deviation σ. As another example, suppose that the normal regression model above had a quadratic response function of the form

$$E(y_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

Then the optimum value of x, i.e., that corresponding to the optimal expected response, can be found by setting the derivative of this function to 0 and solving for x. We find that

$$df/dx = \beta_1 + 2 * \beta_2 x = 0$$

yields that $x_{opt} = -\beta_1/(2*\beta_2)$. We can just take our posterior draws for β_1 and β_2 and obtain a posterior sample of x_{opt} by this simple calculation applied to the posterior output. As an exercise, take the normal model above and simulate a quadratic response and then describe the posterior distribution of x_{opt}.

2.8 Model Checking and Selection

In general terms model checking - or assessing the adequacy of the model - and model selection are quite thorny issues and, despite contrary and, sometimes, strongly held belief among practitioners, there are not really definitive, general solutions to either problem. We're against dogma on these issues and think people need to be open-minded about such things and recognize that models can be useful whether or not they pass certain statistical tests. Some models are intrinsically better than others because they make more biological sense or foster understanding or achieve some objective that some bootstrap or other goodness-of-fit test can't decide for you. That said, it gives you some confidence if your model seems adequate in a purely statistical sense and we try to provide some fit assessment in most real applications of SCR models We provide a very brief overview of concepts here, but provide more detailed coverage in chapter 8. See also Kéry (2010, ch. xyz) and Link and Barker (2009, ch. xyz) for specific context related to Bayesian model checking and selection.

2.8.1 Goodness-of-fit

Goodness-of-fit testing is an important element of any analysis because our model represents a general set of hypotheses about the ecological and observation processes that generated our data. Thus, if our model "fits" in some statistical or scientific sense, then we believe it to be consistent with the hypotheses that went into the model. More formally, we would conclude that the data are *not inconsistent* with the hypotheses, or that the model appears adequate. If we have enough data, then of course we will reject any set of statistical hypotheses. Conversely, we can always come up with a model that fits by making the model extremely complex. Despite this paradox, it seems to us that simple models that you can understand should usually be preferred even if they don't fit, for example if they embody essential mechanisms central to our understanding of things, or if we think that some contributing factors to

lack-of-fit are minor or irrelevant to the scientific context and intended use of the model. In other words, models can be useful irrespective of whether they fit according to some formal statistical test of fit. Yet the tension is there to obtain fitting models, and this comes naturally at the expense of models that can be easily interpreted and studied and effectively used. Unfortunately, conducting a goodness-of-fit test is not always so easy to do. And, moreover, it is never really easy (or especially convenient) to decide if your goodness-of-fit test is worth anything. It might have 0 power! Despite this, we recommend attempting to assess model fit in real applications, as a general rule, and we provide some basic guidance here and some more specific to SCR models in chapter 8.

To evaluate goodness-of-fit in Bayesian analyses, we will most often use the Bayesian p-value (Gelman et al., 1996). The basic idea is to define a fit statistic or "discrepancy measure" and compare the posterior distribution of that statistic to the posterior predictive distribution of that statistic for hypothetical perfect data sets for which the model is known to be correct. For example, with count frequency data, a standard measure of fit is the sum of squares of the "Pearson residuals",

$$D(y_i, \theta) = \frac{(y_i - E(y_i))^2}{Var(y_i)}$$

The fit statistic based on the squared residuals is

$$FIT = \sum_i D(y_i, \theta)^2$$

which can be computed at each iteration of a MCMC algorithm given the current values of parameters that determine the response distribution. At the same time (i.e., at each MCMC iteration), the equivalent statistic is computed for a "new" data set, simulated using the current parameter values. The Bayesian p-value is simply the posterior probability $\Pr(\text{Fit} > \text{Fit}_{new})$[7] which should be close to 0.50 for a good model – one that "fits" in the sense that the observed data set is consistent with realizations simulated under the model being fitted to the observed data. In practice we judge "close to 0.50" as being "not too close to 0 or 1" and, as always, closeness is somewhat subjective. We're happy with anything $> .1$ and $< .9$ but might settle for $> .05$ and < 0.95. Another useful fit statistic is the Freeman-Tukey statistic[8], in which

$$D(\mathbf{y}, \theta) = \sum_i (\sqrt{y_i} - \sqrt{e_i})^2$$

(Brooks et al., 2000), where y_i is the observed value of observation i and e_i its expected value. In contrast to a Chi-square discrepancy, the Freeman-Tukey statistic removes the need to pool cells with small expected values. In summary, the Bayesian p-value is easy to compute, and widely used as a result.

[7] Check this definition!

[8] Ref for this?

2.8.2 Model Selection

In ecology, scientific hypotheses are often manifest as different models or parameters of a model, and so evaluating the importance of different models is fundamental to many ecological studies. For model selection we typically use three different methods: First is, let's say, common sense. If a parameter has posterior mass concentrated away from 0 then it seems like it should be regarded as important - that is, it is "significant." This approach seems to have fallen out of favor with all of the interest over the last 10 or 15 years on model selection in ecology. It seems reasonable to us.

For regression problems we sometimes use the factor weighting idea which is to introduce a set of binary variables w_k for variable k, and express the model as, e.g., for a single covariate model:

$$E(y_i) = \alpha + w\beta x_i$$

where w is given a Bernoulli prior distribution with some prescribed probability. E.g., $w \sim Bern(0.50)$ to provide a prior probability of 0.50 that variable x should be an element of the linear predictor. The posterior probability of the event $w = 1$ is a gauge of the importance of the variable x. i.e., high values of $\Pr(w = 1)$ indicate stronger evidence to support that "x is in the model" whereas values of $\Pr(w = 1)$ close to 0 suggest that x is less important.

This idea seems to be due to Kuo and Mallick (1998)[9] and see Royle and Dorazio (2008, Chapt. 3) for an example in the context of logistic regression. This approach seems to even work sometimes with fairly complex hierarchical models of a certain form. E.g., Royle (2008) applied it to a random effects model to evaluate the importance of the random effect component of the model. The main problem with this approach is that its effectiveness and results will typically be highly sensitive to the prior distribution on the structural parameters (e.g., see Royle and Dorazio (2008, table xyz)). The reason for this is obvious: If $w = 0$ for the current iteration of the MCMC algorithm, so that β is sampled from the prior distribution, and the prior distribution is very diffuse, then extreme values of β are likely. Consequently, when the current value of β is far away from the mass of the posterior when $w = 1$, then the Markov chain may only jump from $w = 0$ to $w = 1$ infrequently. One seemingly reasonable solution to this problem (**?**) is to fit the full model to obtain posterior distributions for all parameters, and then use those as prior distributions in a "model selection" run of the MCMC algorithm. This seems preferable to more-or-less arbitrary restriction of the prior support to improve the performance of the MCMC algorithm.

A third method that that we advocate is subject-matter context. It seems that there are some situations – some models – where one should not have to do model selection because it is necessitated by the specific context of the problem, thus rendering a formal hypotesis test pointless (Johnson, 1999). SCR models are such an example. In SCR models, we will see that "spatial location"

[9] Is this also what people call Zellner's G-priors?

of individuals is an element of the model. The simpler, reduced, model is an ordinary capture-recapture model which is not spatially explicit (i.e., chapter 3), but it seems silly and pointless to think about actually using the reduced model even if we could concoct some statistical test to refute the more complex model. The simpler model is manifestly wrong but, more importantly, not even a plausible data-generating model! Other examples are when effort, area or sample rate is used as a covariate. One might prefer to have such things in models regardless of whether or not they pass some statistical litmus test (although one can always find referees to argue for pedantic procedure over thinking).

Many problems can be approached using one of these methods but there are also broad classes of problems that can't and, for those, you're on your own. In later chapters we will address model selection in specific contexts and we hope those will prove useful for a majority of the situations you encounter.

2.9 Poisson GLMs

The Poisson GLM (also known as "Poisson regression") is probably the most relevant and important class of models in all of ecology. The basic model assumes observations $y_i; i = 1, 2, ..., n$ follow a Poisson distribution with mean λ which we write

$$y_i \sim \text{Poisson}(\lambda)$$

Commonly y_i is a count of animals or plants at some point in space and lambda might depend on i. For example, i might index point count locations in a forest, BBS route centers, or sample quadrats, or similar. If covariates are available it is typical to model them as linear effects on the log mean. If $x(i)$ is some measured covariate associated with observation i. Then,

$$log(x(i)) = \alpha + \beta * x(i)$$

While we only specify the mean of the Poisson model directly, the Poisson model (and all GLMs) has a "built-in" variance which is directly related to the mean. In this case, $\text{Var}(y) = \text{E}(y) = \lambda$. Thus the model accommodates a linear increase in variance with the mean.

2.9.1 Important properties of the Poisson distribution

There are two properties of the Poisson distribution that make it extremely useful in ecology. First is the property of *compound additivity*. If y_1 and y_2 are Poisson random variables with means λ_1 and λ_2, then their sum $N = y_1 + y_2$ is Poisson with mean $\lambda_1 + \lambda_2$. Thus, if the observations can be viewed as an aggregate of counts over some finer unit of measurement, then the Poisson mean aggregates in a corresponding manner. This comes in hand in some ecological applications where we have counts, y_i, made on sample units with different but known areas A_i. Then, we might assume the counts have a Poisson distribution

with mean $A_i\lambda$. On the log-scale we see that $log(A_i)$ enters as an additive constant – usually referred to as the "offset" in GLM lingo. A second useful property of the Poisson distribution is its direct relationship to the multinomial. If y_1 and y_2 are *iid* Poisson then, conditional on their sum $N = y_1+y_2$, their joint distribution is multinomial with sample size N and cell probabilities $\lambda_1/(\lambda_1+\lambda_2)$ and $\lambda_2/(\lambda_1+\lambda_2)$. As a result of this, most multinomial models can be analyzed as a Poisson GLM and *vice versa*.

2.9.2 Example: Breeding Bird Survey Data

As an example we consider a classical situation in ecology where counts of an organism are made at a collection of spatial locations. In this particular example, we have mourning dove counts made along North American Breeding Bird Survey (BBS) routes in Pennsylvania, USA. A route consists of 50 stops separated by 0.5 mile. For the purposes here we are defining y_i = route total count and he sample location will be marked by the center point of the BBS route. The survey is run annually and the data set we have is 1966-1998. BBS data can be obtained online at `http://www.pwrc.usgs.gov/bbs/`. We will make use of the whole data set shortly but for now we're going to focus on a specific year of counts – 1990 – for the sake of building a simple model. For 1990 there were 77 active routes. We have the data stored in a `.csv` file[10] where rows index the unique route, column 1 is the route ID, columns 2-3 are the route coordinates (longitude/latitude), column 4 is a habitat covariate "forest cover" (standardized, see below) and the remaining columns are the yearly counts. Years for which a route was not run are coded as "`NA`" in the data matrix. We imagine that this will be a typical format for many ecological studies, perhaps with more columns representing covariates. To read in the data and display the first few elements of this matrix, do this:

```
> a<-read.csv("pa-bbsdovedata-all.csv")
> data[1:2,1:6]
      X     lon    lat    habitat X66 X67
1 72002 -80.445 41.501 -0.3871372  NA  24
2 72003 -80.347 41.214 -1.0171629  NA  NA
```

It is useful to display the spatial pattern in the observed counts. For that we use a spatial dot plot - where we plot the coordinates of the observations and mark the color of the plotting symbol based on the magnitude of the count. We have a special plotting function for that which is called `spatial.plot()` and it is available with the supplemental **R** package. Actually, what we want to do here is plot the log-count (+1 of course) which (Fig. 2.1) displays a notable pattern that could be related to something. The **R** commands for obtaining this figure are:

```
data<-read.csv("pa-bbsdovedata-all.csv")
y<-data[,29]  # pick out 1990
notna<-!is.na(y)
y<-y[notna]
spatial.plot(data[notna,2:3],y)
```

[10] check this data format

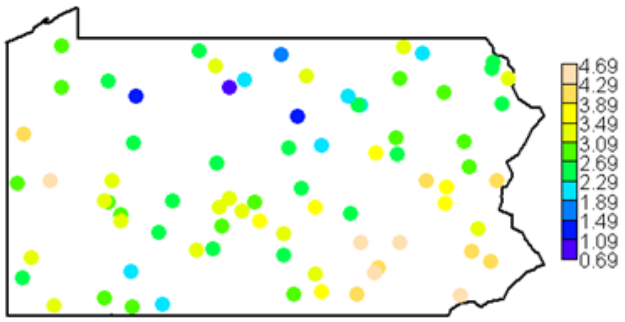

Figure 2.3: Needs a caption

We can ponder the potential effects that might lead to dove counts being high....corn fields, telephone wires, barn roofs along with misidentification of pigeons, these could all correlated reasonably well with the observed count of mourning doves. Unfortunately we don't have any of that information.

We do have a measure of forest cover in the vicinity of each point which is contained in the data set (variable "habitat"). This was derived from a larger GIS coverage of the state (provided in the data file "`pahabdata.csv`") which can be plotted using the `spatial.plot` function using the following commands

```
> map('state',regions="penn",lwd=2)
> spatial.plot(pahabdata[,2:3],pahabdata[,"dfor"],cx=2)
> map('state',regions="penn",lwd=2,add=TRUE)
```

where the result appears in Fig. 2.2. We see a prominent pattern that indicates high forest coverage in the central part of the state and low forest cover in the SE. Inspecting the previous figure of log-counts suggests a relationship between counts and forest cover which is perhaps not surprising.

2.9.3 Doing it in WinBUGS

Here we demonstrate how to fit a Poisson GLM in **WinBUGS** using the covariate x_i = forest cover. It is advisable that x_i be standardized in most cases as this will improve mixing of the Markov chains. Recall that the data we have stored include a standardized covariate (forest cover) and so we don't have to worry about that here. To read the BBS data into **R** and get things set up for **WinBUGS** we issue the following commands:

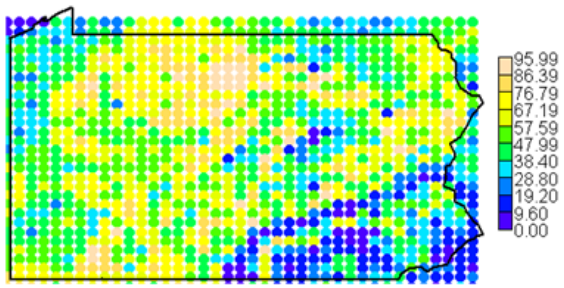

Figure 2.4: Needs a caption

```
data<-read.csv("pa-bbsdovedata-all.csv")
y<-data[,29]                          # pick out 1990
notna<-!is.na(y)
y<-y[notna]                           # discard missing
habitat<-data[notna,4]                # get habitat data
library("R2WinBUGS")                  # load R2WinBUGS
data <- list ( "y","M","habitat")     # bundle data for WinBUGS
```

Now we write out the Poisson model specification in **WinBUGS** pseudo-code, provide initial values, identify parameters to be monitored and then execute **WinBUGS**:

```
cat("
model {
    for (i in 1:M){
      y[i]~dpois(lam[i])
      log(lam[i])<- beta0+beta1*habitat[i]
     }
 beta0~dunif(-5,5)
 beta1~dunif(-5,5)
}
",file="PoissonGLM.txt")

inits <- function()  list ( beta0=rnorm(1),beta1=rnorm(1))
parameters <- c("beta0","beta1")
out<-bugs (data, inits, parameters, "PoissonGLM.txt", n.thin=2,n.chains=2,
                n.burnin=2000,n.iter=6000,debug=TRUE,working.dir=getwd())
```

Remarks: (1) Note the close correspondence in how the model is specified here compared with the normal regression model previously. As an exercise you should discuss the specific differences between the **BUGS** model specifications for the normal and Poisson models.

```
> print(out,digits=3)
Inference for Bugs model at
``PoissonGLM.txt'', fit using WinBUGS,
 2 chains, each with 4000 iterations (first 1000 discarded), n.thin = 2
 n.sims = 3000 iterations saved
             mean     sd     2.5%      25%      50%      75%    97.5%  Rhat n.eff
beta0       3.151  0.025    3.102    3.135    3.151    3.168    3.199 1.001  2300
beta1      -0.498  0.021   -0.539   -0.512   -0.498   -0.484   -0.457 1.001  3000
fit       869.930 19.856  835.500  855.700  868.600  881.900  913.602 1.002  1600
fitnew     76.709 12.519   54.098   68.107   76.215   84.510  102.602 1.001  3000
deviance 1116.605  2.014 1115.000 1115.000 1116.000 1117.000 1122.000 1.001  3000
```

We might wonder whether this model provides an adequate fit to our data. To evaluate that, we used a Bayesian p-value analysis with fit statistic based on the Freeman-Tukey residual by replacing the model specification above with this:

```
cat("
model {
    for (i in 1:M){
      y[i]~dpois(lam[i])
      log(lam[i])<- beta0+beta1*habitat[i]
      d[i]<-  pow(pow(y[i],0.5)-pow(lam[i],0.5),2)   #

      ynew[i]~dpois(lam[i])
      dnew[i]<-pow( pow(ynew[i],0.5)-pow(lam[i],0.5),2)

     }
 fit<-sum(d[])
 fitnew<-sum(dnew[])
 beta0~dunif(-5,5)
 beta1~dunif(-5,5)
}
",file="PoissonGLM.txt")
```

The Bayesian p-value is the proportion of times $fitnew > fit$ which, for this data set, is 0. This suggests that the basic Poisson model does not fit well.

2.9.4 Constructing your own MCMC algorithm

At this point it might be helpful to suffer through an example building a custom MCMC algorithm. Here, we develop an MCMC algorithm for the Poisson regression model, using a Metropolis-within-Gibbs sampling framework. Building MCMC algorithms is covered in more detail in Chapt. 7 where you can also find step-by-step instructions for Metropolis-within-Gibbs samplers, should the following section move through all this stuff too quickly.

We will assume that the two parameters have diffuse normal priors, say $[\alpha] = \text{Norm}(0, 100)$ and $[\beta] = \text{Norm}(0, 100)$ where each has *standard deviation* 100 (recall that **WinBUGS** parameterizes the normal in terms of $1/\sigma^2$). We need to assemble the relevant elements of the model which are these two prior distributions and the likelihood $[\mathbf{y}|\alpha, \beta] = \prod_i [y_i|\alpha\beta]$ which is, mathematically, the product of the Poisson pmf evaluated at each y_i, given particular values of α and β. Next, we need to identify the full conditionals $[\alpha|\beta, \mathbf{y}]$ and $[\beta|\alpha, \mathbf{y}]$. We use the all-purpose rule for constructing full conditionals (section 2.5.1) to discover that:

$$[\alpha|\beta, \mathbf{y}] \propto \left\{ \prod_i [y_i|\alpha, \beta] \right\} [\alpha]$$

and

$$[\beta|\alpha, \mathbf{y}] \propto \left\{ \prod_i [y_i|\alpha, \beta] \right\} [\beta]$$

Remember, we could replace the "$\propto$" with "=" if we put $[y|\beta]$ or $[y|\alpha]$ in the denominator. But, in general, $[y|\alpha]$ or $[y|\beta]$ will be quite a pain to compute and, more importantly, it is a constant as far as the operative parameters (α or β, respectively) are concerned. Therefore, the MH acceptance probability will be the ratio of the full-conditional evaluated at a candidate draw to that evaluated at the current value, and so the denominator required to change $\propto$ to $=$ winds up canceling from the MH acceptance probability. Here we will use the so-called random walk candidate generator, which is a Normal proposal distribution, so that, for example, $\alpha^* \sim \text{Normal}(\alpha^t, \delta)$ where δ is the standard-deviation of the proposal distribution, which is just a tuning parameter that is set by the user and adjusted to achieve efficient mixing of chains (see Section XX in Chapt. 7). We remark also that calculations are often done on the log-scale to preserve numerical integrity of things when quantities evaluate to small or large numbers, so keep in mind, for example, $a * b = exp(log(a) + log(b))$. The "Metropolis within Gibbs" algorithm for a Poisson regression turns out to be remarkably simple:

```
set.seed(2013)  # so we all get the same result

out<-matrix(NA,nrow=1000,ncol=2)   # matrix to store the output
alpha<- -1                         # starting values
beta <- -.8

# begin the MCMC loop ; do 1000 iterations
for(i in 1:1000){

# update the alpha parameter
lambda<- exp(alpha+beta*habitat)
lik.curr<- sum(log(dpois(y,lambda)))
prior.curr<- log(dnorm(alpha,0,100))
alpha.cand<-rnorm(1,alpha,.05)         # generate candidate
lambda.cand<- exp(alpha.cand + beta*habitat)
lik.cand<- sum(log(dpois(y,lambda.cand)))
```

```
prior.cand<- log(dnorm(alpha.cand,0,100))
mhratio<- exp(lik.cand +prior.cand - lik.curr-prior.curr)
if(runif(1)< mhratio)
     alpha<-alpha.cand

# update the beta parameter
lik.curr<- sum(log(dpois(y,exp(alpha+beta*habitat))))
prior.curr<- log(dnorm(beta,0,100))
beta.cand<-rnorm(1,beta,.25)
lambda.cand<- exp(alpha+beta.cand*habitat)
lik.cand<- sum(log(dpois(y,lambda.cand)))
prior.cand<- log(dnorm(beta.cand,0,100))
mhratio<- exp(lik.cand + prior.cand - lik.curr - prior.curr)
if(runif(1)< mhratio)
     beta<-beta.cand

out[i,]<-c(alpha,beta)             # save the current values
}


plot(out[,1],ylim=c(-1.5,3.3),type="l",lwd=2,ylab="parameter value",
     xlab="MCMC iteration")
lines(out[,2],lwd=2,col="red")
```

The first 300 iterations of the MCMC history of each parameter are shown in Fig. 2.4. These chains are not very appealing but a couple of things are evident: We see that the burn-in takes about 250 iterations and that after that chains seem to mix reasonably well, although this is not so clear given the scale of the y-axis. We generated 10,000 posterior samples, discarding the first 500 as burn-in, and the result is shown in Fig. 2.5, this time seperate panels for each parameter. The "grassy" look of the MCMC history is diagnostic of Markov chains that are well-mixing and we would generally be very satisfied with results that look like this.

Remarks: (1) We used a specific set of starting values for these simulations. It should be clear that starting values closer to the mass of the posterior distribution might cause burn-in to occur faster. As an exercise, evaluate that. (2) For the flat normal prior distributions here we could leave the prior contribution out of the full conditional evaluation since it is locally constant, i.e., constant in the vicinity of the posterior mass, and thus has no practical effect. Removing the prior contribution from the MH acceptance probability is equivalent to saying that the parameters have an improper uniform prior, i.e., $\alpha \sim$ const, which is commonly used for mean parameters in practice. Note also that we have used a different prior than in our **WinBUGS** model specification given previously. As an exercise, evaluate whether this seems to affect the result.

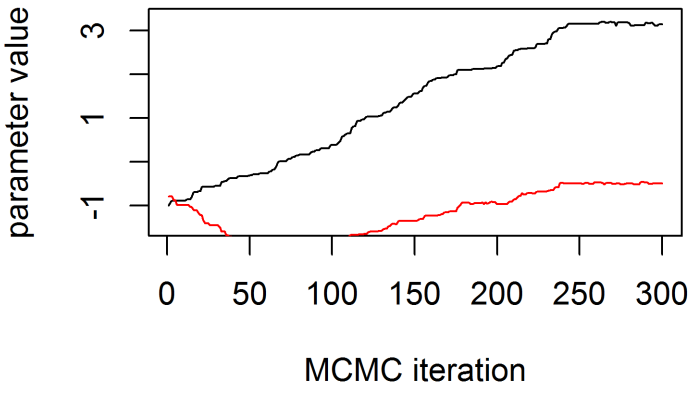

Figure 2.5: First 300 MCMC iterations for the Poisson GLM model parameters using a Metropolis-Hastings tuning parameter of $\delta = 0.05$.

2.10 Poisson GLM with Random Effects

What we will be doing in most of this book is dealing with random effects in GLM-like models - similar to what are usually referred to as generalized linear mixed models (GLMMs). We provide a brief introduction by way of example, extending our Poisson regression model to include a random effect.

The Log-Normal mixture: The classical situation involves a GLM with a normally distributed random effect that is additive on the linear predictor. For the Poisson case, we have:

$$log(\lambda_i) = \alpha + \beta x_i + \eta_i$$

where $\eta_i \sim \text{Normal}(0, \sigma^2)$. A natural alternative is to have multiplicative gamma-distributed noise, $exp(\eta_i) \sim \text{Gamma}(a, b)$ which would correspond to a negative binomial kind of over-dispersion, implying a different mean/variance relationship to the log-normal mixture (the interested reader should work that out). Choosing between such possibilities is not a topic we will get into here because it doesn't seem possible to provide general guidance on it. For this model we carried-out a goodness-of-fit evaluation using the Bayesian p-value based on a Pearson residual statistic. See also (Kéry, 2010, ch. 18) for an example involving a binomial mixed model[11]. Anyhow, it is really amazingly simple to

[11]Kery has noticed that such tests probably have 0 power. Should use the marginal frequency of the data

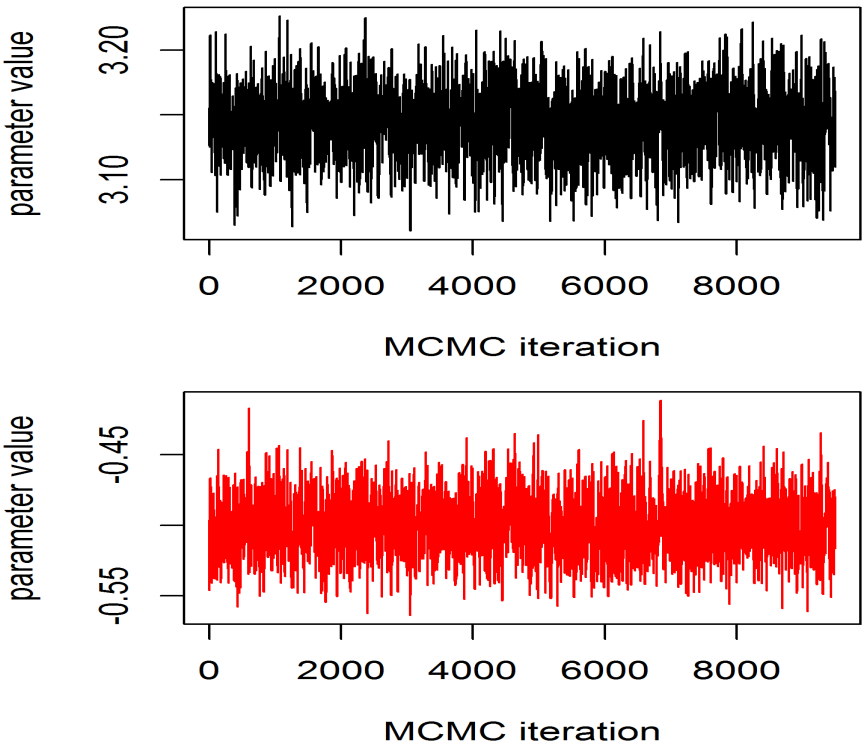

Figure 2.6: nice grassy mcmc output, longer run of previous with $\delta = 0.05$.

express this model in **WinBUGS** and have **WinBUGS** draw samples from the posterior distribution using the following code for the BBS dove counts:

```
data<-read.csv("pa-bbsdovedata-all.csv")
locs<-data[,2:3]
habitat<-data[,4]
y<-data[,29]     # grab year 1990
M<-length(y)

set.seed(2013)

cat("
model {
  for (i in 1:M){
     y[i]~dpois(lam[i])
     log(lam[i])<- alpha+ beta*habitat[i] + eta[i]
     frog[i]<-beta*habitat[i] + eta[i]
     eta[i] ~ dnorm(0,tau)
     d[i]<-  pow(pow(y[i],0.5)-pow(lam[i],0.5),2)

     ynew[i]~dpois(lam[i])
     dnew[i]<- pow(pow(ynew[i],0.5)-pow(lam[i],0.5),2)
   }
 fit<-sum(d[])
 fitnew<-sum(dnew[])

 alpha~dunif(-5,5)
 beta~dunif(-5,5)
 sigma~dunif(0,10)
 tau<-1/(sigma*sigma)
}

",file="model.txt")
data <- list ( "y","M","habitat")
inits <- function()
  list ( alpha=rnorm(1),beta=rnorm(1),sigma=runif(1,0,4))
parameters <- c("alpha","beta","sigma","tau","fit","fitnew")
library("R2WinBUGS")

out<-bugs (data, inits, parameters, "model.txt", n.thin=2,n.chains=2,
 n.burnin=1000,n.iter=5000,debug=TRUE)
```

This produces the following posterior summary statistics:

```
> print(out,digits=2)
Inference for Bugs model at "model.txt", fit using WinBUGS,
 2 chains, each with 5000 iterations (first 1000 discarded), n.thin = 2
 n.sims = 4000 iterations saved
           mean    sd   2.5%    25%    50%    75%  97.5% Rhat n.eff
alpha      2.98  0.08   2.82   2.93   2.98   3.03   3.12 1.00  1400
beta      -0.53  0.07  -0.68  -0.58  -0.53  -0.49  -0.38 1.01   350
```

```
sigma      0.60  0.06   0.49   0.56   0.59   0.64   0.73 1.00  2000
tau        2.88  0.57   1.88   2.47   2.86   3.24   4.12 1.00  2000
fit       26.58  3.72  19.87  23.96  26.37  29.01  34.46 1.00  4000
fitnew    26.83  3.90  19.60  24.12  26.68  29.36  35.04 1.00  4000
deviance 445.94 12.18 424.00 437.40 445.20 453.90 471.50 1.00  4000

[... some output deleted ...]
```

The Bayesian p-value for this model is

```
> mean(out$sims.list$fit>out$sims.list$fitnew)
[1] 0.4815
```

indicating a pretty good fit. Given the site-level random effect, it would be surprising for this model to not fit! One thing we notice is that the posterior standard deviations of the regression parameters are much higher, a result of the excess variation. Wwe would also notice much less precise predictions of hypothetical new observations.

2.11 Binomial GLMs

Another extremely important class of models in ecology are binomial models. We use binomial models for count data whenever the observations are counts or frequencies and it is natural to condition on a "sample size", say K, the maximum frequency possible in a sample. The random variable, $y \leq K$, is then the frequency of occurrences out of K "trials". The parameter of the binomial models is p, often called "success probability" which is related to the expected value of y by $E(y) = pK$. Usually we are interested in modeling covariates that affect the parameter p, and such models are called binomial GLMs , binomial regression models or logistic regression, although logistic regression really only applies when the logistic link is used to model the relationship between p and covariates (see below).

One of the most typical binomial GLMs occurs when the sample size equals 1 and the outcome, y, is "presence" ($y = 1$) or "absence" ($y = 0$) of a species. This is a classical "species distribution" modeling situation. A special situation occurs when presence/absence is observed with error (MacKenzie et al., 2002; Tyre et al., 2003). In that case, $K > 1$ samples are usually needed for effective estimation of model parameters.

In standard binomial regression problems the sample size is fixed by design but interesting models also arise when the sample size is itself a random variable. These are the N-mixture models (Royle, 2004; Kéry et al., 2005; Royle and Dorazio, 2008; Kéry, 2010) and related models (in this case, N being the sample size, which we labeled K above)[12]. Another situation in which the binomial sample size is "fixed" is closed population capture-recapture models in which

[12]Some of the jargon is actually a little bit confusing here because the binomial index is customarily referred to as "sample size" but in the context of N-mixture models N is actually the "population size"

a population of individuals is sampled K times. The number of times each individual is encountered is a binomial outcome with parameter - encounter probability – p, based on a sample of size K. In addition, the total number of unique individuals observed, n, is also a binomial random variable based onpopulation size N. We consider such models in the chapter 3.

2.11.1 Binomial regression

In binomial models, covariates are modeled on a suitable transformation (the link function) of the binomial success probability, p. Let x_i denote some measured covariate for sample unit i and let p_i be the success probability for unit i. The standard choice is the "logit" link function which is:

$$log(p_i/(1-p_i)) = \alpha + \beta * x_i.$$

The inverse-logit (or "expit") is

$$p_i = \text{expit}(\alpha + \beta * x_i) = \frac{\exp(\alpha + \beta * x_i)}{1 + \exp(\alpha + \beta * x_i)}$$

There are many other possible link functions. However, ecologists seem to adopt the logit link function without question in most applications[13]. We sometimes use the "complementary log-log" (= "cloglog") link function in ecological applications because it arises naturally in many situations (Royle and Dorazio, 2008, p. 150). For example, consider the "probability of observing a count greater than 0" under a Poisson model: $\Pr(y > 0) = 1 - exp(-\lambda)$. In that case,

$$cloglog(p) = log(-log(1-p)) = log(\lambda)$$

So that if you have covariates in your linear predictor for $E(y)$ under a Poisson model then they are linear on the complementary log-log link of p. In models of species occurrence it seems natural to view occupancy as being derived from local abundance N (Royle and Nichols, 2003; Royle and Dorazio, 2006; Dorazio, 2007). Therefore, models of local abundance in which $N \sim \text{Poisson}(A\lambda)$ for a habitat patch of area A implies a model for occupancy ψ of the form

$$cloglog(\psi) = log(A) + log(\lambda).$$

We will use the cloglog link in some analyses of SCR models in chapter 4 and elsewhere.

2.11.2 Example: Waterfowl Banding Data

It would be easy to consider a standard "distribution modeling" application where $K = 1$ and the outcome is occurrence ($y = 1$) or not ($y = 0$) of some species. Such examples abound in books (e.g., Royle and Dorazio (2008, ch. 3);

[13] a notable exception is distance sampling, which is all about choosing among link functions

Kéry (2010, ch. 21); Kery and Schaub (2011, ch. 13)) and in the literature. Instead, we will consider an example involving band returns of waterfowl which were analyzed by Royle and Dubovsky (2001)[14].

For these data, y_i is the number of waterfowl bands recovered out of B_i birds banded at some location $\mathbf{s}_i$. In this case B_i is fixed. Thinking about recovery rate as being proportional to harvest rate, we use these data to explore geographic gradients in recovery rate resulting from variability in harvest pressure experienced by populations depending on their migration ecology. As such, we fit a basic binomial GLM with a linear response to geographic coordinates (including an interaction term). The data are provided with the **R** package `scrbook`. Here we provide the part of the script for creating the model and fitting the model in **WinBUGS** using the `bugs` function. There are few structural differences between this model and the Poisson GLM fitted previously. The main things are due to the data structure (we have a matrix here instead of a vector) and otherwise we change the main distributional assumption to binomial (specified with `dbin`) and then use the `logit` function to relate the parameter p_{it} to the covariates. Here is the script:

```
load("mallarddata")  # not sure how this will look

sink("model.txt")
cat("
model {
 for(t in 1:5){
    for (i in 1:nobs){
       y[i,t] ~ dbin(p[i,t], B[i,t])
       logit(p[i,t]) <- alpha0[t] + alpha1*X[i,1] + alpha2*X[i,2] + alpha3*X[i,1]*X[i,2]
     }
}
alpha1~dnorm(0,.001)
alpha2~dnorm(0,.001)
alpha3~dnorm(0,.001)
for(t in 1:5){
  alpha0[t] ~ dnorm(0,.001)
 }
}
",fill=TRUE)
sink()

data  <- list(B=mallard.bandings, y=mallard.recoveries,
             nobs=nrow(banding.locs),X=banding.locs)
inits <- function(){
      list(alpha0=rnorm(5),alpha1=0,alpha2=0,alpha3=0) }
parms <- list('alpha0','alpha1','alpha2','alpha3')
out   <- bugs(data,inits, parms,"model.txt",n.chains=3,
  n.iter=2000,n.burnin=1000, n.thin=2,debug=TRUE)
```

Posterior summaries of model parameters are as follows:

[14] I hate this example. Anyone got a better one thats not distribution modeling?

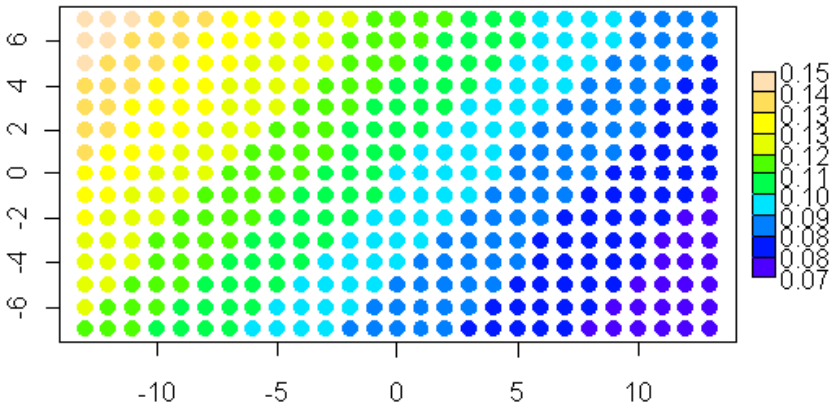

Figure 2.7: Predicted recovery rate of bands.

```
> print(out,digits=3)
Inference for Bugs model at "model.txt", fit using WinBUGS,
 3 chains, each with 2000 iterations (first 1000 discarded), n.thin = 2
 n.sims = 1500 iterations saved
              mean    sd     2.5%      25%      50%      75%    97.5%  Rhat n.eff
alpha0[1]   -2.346 0.036   -2.417   -2.370   -2.346   -2.323   -2.277 1.001  1500
alpha0[2]   -2.356 0.032   -2.420   -2.379   -2.356   -2.335   -2.292 1.001  1500
alpha0[3]   -2.220 0.035   -2.291   -2.244   -2.219   -2.197   -2.153 1.001  1500
alpha0[4]   -2.144 0.039   -2.225   -2.169   -2.143   -2.116   -2.068 1.000  1500
alpha0[5]   -1.925 0.034   -1.990   -1.949   -1.924   -1.901   -1.856 1.004   570
alpha1      -0.023 0.003   -0.028   -0.025   -0.023   -0.022   -0.018 1.001  1500
alpha2       0.020 0.006    0.009    0.016    0.020    0.024    0.031 1.001  1500
alpha3       0.000 0.001   -0.002   -0.001    0.000    0.000    0.002 1.001  1500
deviance  1716.001 4.091 1710.000 1713.000 1715.000 1718.000 1726.000 1.001  1500

[... some output deleted ...]
```

The basic result suggests a negative east-west gradient and a positive south to north gradient but no interaction. A map of the response surface is shown in Fig. 2.6. We did an additional MCMC run where we saved the binomial parameter p and computed the Bayesian p-value (double use of "p" here is confusing, but I guess that happens sometimes!) using a fit statistic based on the Freeman-Tukey statistic (see Sec. **??** above). The result indicates that the linear response surface model does not provide an adequate fit of the data. The reader should contemplate whether this invalidates the basic interpretation of the result.

2.12 Summary and Outlook

GLMs and GLMMs are the most useful statistical methods in all of ecology. The principles and procedures underlying these methods are relevant to nearly all modeling and analysis problems in every branch of ecology. Moreover, understanding how to analyze these models is crucial in a huge number of diverse problems. If you understand and can conduct classical likelihood and Bayesian analysis of Poisson and binomial GLM(M)s, then you will be successful analyzing and understanding more complex classes of models that arise. We will see shortly that spatial capture-recapture models are a type of GLMM and thus having a basic understanding of the conceptual origins and formulation of GLM(M)s and their analysis is extremely useful.

We note that GLM(M)s are routinely analyzed by likelihood methods but we have focused on Bayesian analysis here in order to develop the tools that are less familiar to most ecologists. In particular, Bayesian analysis of models with random effects is relatively straightforward because the models are easy to analyze conditional on the random effect, using methods of MCMC. Thus, we will often analyze SCR models in later chapters by MCMC, explicitly adopting a Bayesian inference framework. In that regard, the various **BUGS** engines (**WinBUGS**, **OpenBUGS**, **JAGS**) are enormously useful because they provide an accessible platform for carrying out analyses by MCMC by just describing the model, and not having to worry about how to actually build MCMC algorithms. That said, the **BUGS** language is more important than just to the extent that it enables one to do MCMC - it is useful as a modeling tool because it fosters understanding, in the sense that it forces you to become intimate with your model. You have to write down all of the probability assumptions, the relationships between observations and latent variables and parameters. This is really a great learning paradigm that you can grow with.

While we have emphasized Bayesian analysis in this chapter, and make primary use of it through the book, we we will provide an introduction to likelihood analysis in Chapt. 6 and use those methods also from time to time. Before getting to that, however, it will be useful to talk about more basic, conventional closed population capture-recapture models and these are the topic of the next chapter.

Chapter 3

Closed population models

Chapter 4

Fully Spatial Capture-Recapture Models

Chapter 5

Other observation models

Chapter 6

Maximum likelihood estimation

Chapter 7

MCMC details

Chapter 8

Goodness of Fit and stuff

Chapter 9

Covariate models

Chapter 10

Inhomogeneous Point Process

Chapter 11

Open models

Bibliography

Aitkin, M. (1991), "Posterior bayes factors," *Journal of the Royal Statistical Society. Series B (Methodological)*, 111–142.

Brooks, S. P., Catchpole, E. A., and Morgan, B. J. T. (2000), "Bayesian Animal Survival Estimation," *Statistical Science*, 15, 357–376.

Chandler, R. B. and Royle, J. A. (2012), "Spatially-explicit models for inference about density in unmarked populations," *Biometrics (in review)*.

Dorazio, R. M. (2007), "On the choice of statistical models for estimating occurrence and extinction from animal surveys," *Ecology*, 88, 2773–2782.

Foster, R. J. and Harmsen, B. J. (2012), "A critique of density estimation from camera-trap data," *The Journal of Wildlife Management*, 76, 224–236.

Gardner, B., Reppucci, J., Lucherini, M., and Royle, J. (2010), "Spatially explicit inference for open populations: estimating demographic parameters from camera-trap studies," *Ecology*, 91, 3376–3383.

Gelman, A. (2006), "Prior distributions for variance parameters in hierarchical models," *Bayesian analysis*, 1, 515–533.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian data analysis, second edition.*, Bocan Raton, Florida, USA: CRC/Chapman & Hall.

Gelman, A., Meng, X. L., and Stern, H. (1996), "Posterior predictive assessment of model fitness via realized discrepancies," *Statistica Sinica*, 6, 733–759.

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6, 721–741.

Hawkins, C. E. and Racey, P. A. (2005), "Low population density of a tropical forest carnivore, Cryptoprocta ferox: implications for protected area management," *Oryx*, 39, 35–43.

Jackson, R., Roe, J., Wangchuk, R., and Hunter, D. (2006), "Estimating Snow Leopard Population Abundance Using Photography and Capture-Recapture Techniques," *Wildlife Society Bulletin*, 34, 772–781.

Johnson, D. H. (1999), "The insignificance of statistical significance testing," *The journal of wildlife management*, 763–772.

Kass, R. and Wasserman, L. (1996), "The selection of prior distributions by formal rules," *Journal of the American Statistical Association*, 1343–1370.

Kéry, M. (2010), *Introduction to WinBUGS for Ecologists: Bayesian Approach to Regression, ANOVA, Mixed Models and Related Analyses*, Academic Press.

Kéry, M., Royle, J. A., and Schmid, H. (2005), "Modeling avian abundance from replicated counts using binomial mixture models," *Ecological Applications*, 15, 1450–1461.

Kery, M. and Schaub, M. (2011), *Bayesian Population Analysis Using WinBugs*, Academic Press.

King, R. (2009), "Missing," *missing*, Missing.

Kuo, L. and Mallick, B. (1998), "Variable selection for regression models," *Sankhy{\=a}: The Indian Journal of Statistics, Series B*, 65–81.

Le Cam, L. (1990), "Maximum likelihood: an introduction," *International Statistical Review/Revue Internationale de Statistique*, 153–171.

Link, W. A. and Barker, R. J. (2009), *Bayesian Inference: With Ecological Applications*, London, UK: Academic Press.

MacEachern, S. N. and Berliner, L. M. (1994), "Subsampling the Gibbs sampler," *American Statistician*, 188–190.

MacKenzie, D. I., Nichols, J. D., Lachman, G. B., Droege, S., Royle, J. A., and Langtimm, C. A. (2002), "Estimating site occupancy rates when detection probabilities are less than one," *Ecology*, 83, 2248–2255.

McCarthy, M. A. (2007), *Bayesian Methods for Ecology*, Cambridge: Cambridge University Press.

McCullagh, P. and Nelder, J. A. (1989), *Generalized linear models*, Chapman & Hall/CRC.

Millar, R. B. (2009), "Comparison of hierarchical Bayesian models for overdispersed count data using DIC and Bayes' Factors," *Biometrics*, 65, 962–969.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized linear models," *Journal of the Royal Statistical Society. Series A (General)*, 370–384.

Plummer, M. (2009), "rjags: Bayesian graphical models using mcmc. R package version 1.0. 3-12," .

Robert, C. P. and Casella, G. (2010), *Introducing Monte Carlo Methods with R*, New York, USA: Springer.

Royle, J. A. (2004), "*N*-mixture models for estimating population size from spatially replicated counts," *Biometrics*, 60, 108–115.

— (2008), "Modeling individual effects in the Cormack–Jolly–Seber model: a state–space formulation," *Biometrics*, 64, 364–370.

Royle, J. A. and Dorazio, R. M. (2006), "Hierarchical models of animal abundance and occurrence," *Journal of Agricultural, Biological, and Environmental Statistics*, 11, 249–263.

— (2008), *Hierarchical modeling and inference in ecology: the analysis of data from populations, metapopulations and communities*, Academic Press.

Royle, J. A., Dorazio, R. M., and Link, W. A. (2007), "Analysis of multinomial models with unknown index using data augmentation," *Journal of Computational and Graphical Statistics*, 16, 67–85.

Royle, J. A. and Dubovsky, J. A. (2001), "Modeling spatial variation in waterfowl band-recovery data," *The Journal of wildlife management*, 726–737.

Royle, J. A. and Link, W. A. (2006), "Generalized site occupancy models allowing for false positive and false negative errors," *Ecology*, 87, 835–841.

Royle, J. A. and Nichols, J. D. (2003), "Estimating abundance from repeated presence-absence data or point counts," *Ecology*, 84, 777–790.

Sepúlveda, M. A., Bartheld, J. L., Monsalve, R., Gómez, V., and Medina-Vogel, G. (2007), "Habitat use and spatial behaviour of the endangered Southern river otter (Lontra provocax) in riparian habitats of Chile: conservation implications," *Biological Conservation*, 140, 329–338.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van Der Linde, A. (2002), "Bayesian measures of model complexity and fit," *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 583–639.

Sturtz, S., Ligges, U., and Gelman, A. (2005), "R2WinBUGS: A Package for Running WinBUGS from R," *Journal of Statistical Software*, 12, 1–16.

Trolle, M. and Kéry, M. (2005), "Camera-trap study of ocelot and other secretive mammals in the northern Pantanal," *Mammalia*, 69, 409–416.

Tyre, A. J., Tenhumberg, B., Field, S. A., Niejalke, D., Parris, K., and Possingham, H. P. (2003), "Improving precision and reducing bias in biological surveys: estimating false-negative error rates," *Ecological Applications*, 13, 1790–1801.

Zuur, A. F., Ieno, E. N., Walker, N. J., Saveliev, A. A., and Smith, G. M. (2009), *Mixed effects models and extensions in ecology with R*, Springer Verlag.