

Estimation of the Size of a Closed Population from Capture–Recapture Data

OUTLINE

6.1	Introduction	134
6.2	Generation and Analysis of Simulated Data with Data Augmentation	139
6.2.1	<i>Introduction to Data Augmentation for the Simplest Case: Model M_0</i>	139
6.2.2	<i>Time Effects: Model M_t</i>	145
6.2.3	<i>Behavioral or Memory Effects: Model M_b</i>	148
6.2.4	<i>Individual (Random) Effects: The Heterogeneity Model M_h</i>	150
6.2.5	<i>Combined Effects: Model M_{th}</i>	154
6.3	Analysis of a Real Data Set: Model M_{tth} for Species Richness Estimation	157
6.4	Capture–Recapture Models with Individual Covariates: Model M_{t+X}	162
6.4.1	<i>Individual Covariate Model for Species Richness Estimation</i>	163
6.4.2	<i>Individual Covariate Model for Population Size Estimation</i>	166
6.5	Summary and Outlook	169
6.6	Exercises	170

6.1 INTRODUCTION

In Chapter 5 (and partly in Chapter 4; see Section 4.4), we met models that attempt to distinguish between an ecological process and an observation process. This is important, since it allows for a more proper accounting of the uncertainty in a modeled system (Cressie et al., 2009) and is required to avoid bias in some important system descriptors, such as density dependence (Freckleton et al., 2006). However, at a closer look, these models do not fully deliver what they promise at first sight: after all, they typically do not have parameters with a clear ecological meaning. Or, put in another way, what exactly is the ecological relevance of an “expected count” (Bob Dorazio, pers. comm.)? After all, we would like to describe a population in terms of its size, not in terms of the field method that we apply to produce a count of animals or plants.

In this Chapter, we introduce a different class of models that achieve a proper accounting for the processes that give rise to observed data, in terms of an ecological and of an observation process: closed-population capture–recapture models. Capture–recapture models in general are a very large class of models that have become increasingly used in ecological applications of statistical modeling (Seber, 1982; Borchers et al., 2002; Williams et al., 2002; Royle and Dorazio, 2008; King et al., 2010). In principle, they all boil down to estimation of detection probability, which provides the link between what we observe and the true population parameters, such as population size or survival probability. Detection probability can be estimated from repeated encounters of individually marked animals or plants. Historically, encountering meant capturing an animal and hence, the associated statistical methods have come to be called capture–recapture, or mark-recapture, models. However, the essential feature of these models is that they require as data repeated *observations of individually recognizable units*, such as individuals, occupied sites, or species, to estimate population size, number of occupied patches, or species richness, respectively. Individual recognition may well be possible without physical capture.

Almost all of the remainder of this book will be dedicated to models for inference about populations which contain an explicit parameter representing detection probability. These models describe the observation process explicitly and include capture–recapture models for marked individuals and other methods for unmarked “individuals” as well; see the metapopulation estimation models in Chapters 12 and 13. There is a single exception in Chapter 11, where we partly fall back on the modeling of Chapters 4 and 5. The models in Chapter 6 deal with repeated surveys from a single population. Surveys are conducted over a period that is sufficiently short so that the population can be assumed “closed” to

numerical changes, that is, no recruitment, mortality, and dispersal occur. Most of the methods in later chapters relax that assumption, at the expense of not being able to estimate abundance (but see Chapter 10 and also Chapters 12 and 13).

Assessing the size of a closed population (N), or analogously, its density (D), appears simple at first sight; simply tally up all individuals (N) that live on a piece of land (A). However, in reality, assessing N is fairly challenging because of at least two reasons: imperfect detection (p) and the problem of delimiting the size of the study area (A).

First, imperfect detection is a hallmark of all studies of wild organisms, be they plants (Kéry and Gregg, 2003; Kéry, 2004; Kéry et al., 2005a) or animals (Williams et al., 2002; Nichols et al., 2009). In most cases and under most situations, not every individual present in a study area will be seen and hence, counts (C) will typically be an underestimate of true population size (N). Over several decades, a huge number of protocols and associated models have been developed to cope with this challenge and to “correct” the observed counts for detection error induced by $p < 1$. Conceptually, all derive some estimates of the probability to detect a member of N , that is, of detection probability p , and then apply the commonsense or canonical estimator for N , which is $\hat{N} = C/\hat{p}$, where hats denote estimates.

One of the main differences between these methods is in how they derive their estimate of p . Classical capture–recapture models use repeated observations of recognizable individuals and derive information about p from the pattern of detection and nondetection of each marked individual. Capture–recapture methods are huge in scope, and there is a tremendous variety of protocols and associated models (Borchers et al., 2002; Williams et al., 2002; Royle and Dorazio, 2008). Distance sampling methods derive information about p from the distribution of the distances between an observer (or some detection device) and the detected animals or objects sampled (Buckland et al., 2001). Repeated counts methods (Royle, 2004c) do not require individual identification of the objects counted; they are a third and very useful protocol (see Chapter 12).

The second challenge in assessing N is that we need to decide how to delimit the study area or put another way, how should we define the area with which our population is associated or simply, what is A ? Obviously, this is trivial for actual islands or islands of habitat such as mountain tops or desert oases. However, in most cases, objectively delimiting the area that is used by a population is difficult or may be downright impossible without making some arbitrary decisions.

One reason it may be difficult to determine the area on which a population lives is movement of animals within their home range. Any defined area will then sometimes contain individuals who have the center of gravity of their entire activity range outside of that perimeter, and who, therefore,

in a sense, do not belong to the focal population but can nevertheless be seen within a study area. Similarly, some individuals at the edge of the study area, who do have the majority of their activity range within the nominal area associated with the desired population, may sometimes move beyond that perimeter and therefore be absent and undetectable within the perimeter. But how should an individual at the border of the study area be counted? As a fractional individual only? Or full, if its home-range center is inside and not at all, when it is outside the boundary of the nominal study area? Dealing with this sort of *temporary emigration* (as it is called in the capture–recapture literature; Kendall et al., 1997; Kendall, 1999) is almost always a challenge in population size estimation (although neither in the case of distance sampling nor in spatially explicit capture–recapture models). Conceptually, it is related to the problem of separating permanent emigration from mortality probability in the Cormack–Jolly–Seber model (Chapter 7). Spatially explicit capture–recapture (SECR) models (e.g., Efford, 2004; Borchers and Efford, 2008; Royle and Young, 2008) should be applied in this context because they effectively deal with the challenge of determining the size of A .

In this chapter, we do not describe spatial capture–recapture models but instead assume that we are able to delimit a piece of land with which some unknown number N of animals or plants are associated, and we want to estimate that number N by “correcting” our count by our estimate of detection probability p . One important further assumption is that there are no misclassifications, that is, no individual is mistaken for another. This is the classical setting of a capture–recapture study, where we need repeated surveys to obtain so-called detection histories, strings of ones and zeroes, that denote whether an individual was detected or not during any given repeat survey (also called an occasion). Classical capture–recapture models consist of modeling patterns in p and obtain an estimate of N through the canonical estimator C/\hat{p} .

Otis et al. (1978) provided a useful catalog of possible patterns in detection probability in the context of closed-population size estimation. They distinguish between three effects: individual effects, time effects, and behavioral response effects. Individual and time effects are the differences among individuals and those among capture occasions. Behavioral response denotes a situation wherein detection probability at an occasion depends on whether an individual was detected in a previous occasion. The term comes from live trapping of animals, where such a nonindependence of detection probability within an individual detection history is due to the individual changing its behavior in response to a capture event. If trapping leads to an increased detection probability at a later occasion, one talks about “trap-happiness”; the converse is called “trap-shyness”. However, patterns akin to a trap response may arise also as a result of quite different mechanisms, for instance, if an observer

TABLE 6.1 Parameters for Detection Probability p_{it} in a Hypothetical Study with Three Individuals and Four Capture Occasions.

Individual	Occasion 1	Occasion 2	Occasion 3	Occasion 4
1	$p_{1,1}$	$p_{1,2}$	$p_{1,3}$	$p_{1,4}$
2	$p_{2,1}$	$p_{2,2}$	$p_{2,3}$	$p_{2,4}$
3	$p_{3,1}$	$p_{3,2}$	$p_{3,3}$	$p_{3,4}$

Note: The models M_0 , M_t , M_b , M_h (see text) differ in the assumptions that they make about how cells are related among rows, columns, and within a row.

remembers the location of a species in a study of species richness (e.g., an owl in a hollow tree) and is more likely to detect a species that he/she has detected previously. In addition, individual heterogeneity in detection probability may appear like trap-happiness (Kéry et al., 2006).

The three main classes of models in Otis et al. (1978) are known as models M_h (h for individual heterogeneity), M_t (t for time), and M_b (b for behavioral response). These effects can be visualized in a cross-classification of individuals and occasion such as in Table 6.1 for a toy study of three individuals and four occasions. Model M_h accounts for row effects (i.e., individuals differ), model M_t accounts for column effects (occasions differ), and model M_b accounts for within-row effects, that is, within-individual capture-history dependence.

Otis et al. (1978) defined eight models using these three effects: Model M_0 , which assumes that p is constant across all individuals and times; models M_h , M_t , and M_b as defined earlier; and four models with two-way and three-way combinations of effects: M_{th} , M_{bh} , M_{tb} , and M_{tbh} . Estimators for most of these models were implemented in the grandfather of population size estimation software, CAPTURE (Rexstad and Burnham, 1991). Many of these methods have now been superseded by more efficient (e.g., likelihood-based) estimators, and many of those are available in program MARK (White and Burnham, 1999). Indeed, over the last 1–2 decades there has been a proliferation of new methods for population size estimation, see Borchers et al. (2002), Williams et al. (2002) and also Chapter 14 by Paul Lukacs in the *Gentle Introduction to program MARK* (<http://www.phidot.org/software/mark/docs/book/>). The Otis et al. (1978) catalog of effects remains useful because it clarifies our thinking about possible patterns in such collections of parameters in a cross-classification of individual by time.

Three things must be remembered however. First, what used to be called model t, b, or h is in fact not a single model, but in reality a whole family of models: there are various ways in which temporal, behavioral, or individual effects may be modeled. For instance, we may want to model a permanent trap response, such as is the tradition in population

size estimation, or we may want to model an immediate trap response, as is customary in Cormack–Jolly–Seber models (see Section 7.8; Appendix 2.2; Pradel, 1993). Moreover, there are many different ways to model individual heterogeneity, for instance, by finite or continuous mixture distributions, and for continuous mixtures, by different parametric forms (Pledger, 2000; Dorazio and Royle, 2003; Link, 2003). So, it is really not enough to say “we used model M_b ” or “we used model M_h ”; rather, we need to say (and know) which model M_b or which model M_h we are using.

Second, as there is not a single model M_b for instance, there is not a single model M_{tb} either. When two effects are present in a model, we can combine them in an additive or in an interactive way. So, now that we are not constrained anymore to the rigid set of models for population size estimation from the old CAPTURE days, we must also say (and know) how to combine different effects in a model.

Third, all models in the Otis et al. (1978) classification represent “group” effects, where we batch the parameters in Table 6.1 in some way and estimate differences among these batches. However, once we recognize how these models can be specified as GLMs or GLMMs, we immediately have much more modeling freedom. We can do all this modeling on the scale of the logit-transformed detection probability (p) and then combine group effects and continuous covariate effects in a linear model of the usual form, for instance:

$$\text{logit}(p_{ij}) = \alpha_i + \beta_j + \delta * F_{ij} + \gamma * X_{ij},$$

$$\text{with } \alpha_i \sim \text{Normal}(\mu_\alpha, \sigma_\alpha^2).$$

For illustration, in this model for detection probability of individual i during occasion j , we specify random individual effects α_i , fixed time effects β_j , an effect δ that depends on whether an animal was captured before or not (this information is contained in the indicator variable F), and an effect γ of another covariate X . In any practical application, we may be far from being able to estimate all these quantities, there may be confounding and not enough data etc. Also, one of the fixed time effects (β_j) would have to be constrained to zero to avoid overparameterization.

So, it is useful to think about all these models in a GLM way. But in addition, we find it useful to be able to relate a model that is written in this way back to the Otis et al. (1978) classification. For instance, our model above would represent a model M_{tbh} with an added covariate X and where all effects are additive rather than interactive (so, we may also want to write that model as $M_{t+b+h+X}$). WinBUGS gives us full modeling freedom to fit such custom models because its model definition language is so apt to free the modeler in us (Kéry, 2010).

It is worth noting that the linear models concept (i.e., the design matrix) underlying capture–recapture models when viewed as GLM are not only

important for population size estimation. Far from that: the modeling of time effects (e.g., fixed or random or as a function of covariates), the modeling of “behavioral” effects and the modeling of individual effects are elements that we will see as part of quite different models over and over again. Hence, a full understanding of the material in this chapter will be a great help for your modeling in WinBUGS for a vast range of models. Essentially the same topics arise in other such collections of parameters, for instance, in the modeling of survival probability in a cross-classification of individual and time in the Cormack–Jolly–Seber model (Chapter 7) or of detection probability in a cross-classification of site and time in a metapopulation model for abundance (Chapter 12) or occurrence (Chapter 13). We believe that seeing these connections will be very helpful for obtaining a more synthetic understanding of capture–recapture models. Hence, the way in which we structure such tables of parameters, which is implied by the cross-classification of individuals and time in this chapter is relevant for the modeling of tables of parameters in many other inferential situations as well.

This chapter is the place to describe for the first time data augmentation (Tanner and Wong, 1987), introduced in the context of capture–recapture models by (Royle et al., 2007a); see also Royle and Dorazio (2011) and Section 10.3. Data augmentation greatly simplifies inference for a vast range of models and makes them amenable to simple fitting in WinBUGS. In particular, data augmentation offers an extremely flexible way to model patterns of detection probability in closed populations. We first introduce data augmentation with the simplest possible closed-capture model, M_0 . Then, we will look into a few simple examples of closed-capture models with simulated data: M_t , M_b , M_h , and M_{th} . With model M_t , we will see again how one simply changes from fixed effects to random effects, and how transparent this is when the model is described in the BUGS language. Finally, we will analyze a real data set under models M_{tbh} and M_{tbh+X} , where “population size” is represented by species richness. In the latter model, we provide an illustration of individual covariate modeling, which is greatly simplified when using data augmentation (Royle, 2009).

6.2 GENERATION AND ANALYSIS OF SIMULATED DATA WITH DATA AUGMENTATION

6.2.1 Introduction to Data Augmentation for the Simplest Case: Model M_0

The simplest possible model for inference about the size of a single population is model M_0 (Otis et al., 1978), where detection probability is assumed constant over the two dimensions of the detection parameter

table, that is, over individuals and over time periods (Table 6.1). First, we write a function that simulates capture data under model M_0 . The three arguments are N (population size), p (detection probability), and T (the number of sampling occasions). T is assumed constant for all individuals, but this is for pure convenience. In the analysis of the real data, we will see how we deal with the case where T is not the same across all individuals.

```
# Define function to simulate data under M0
data.fn <- function(N=100, p=0.5, T=3) {
  yfull <- yobs <- array(NA, dim=c(N, T))
  for (j in 1:T) {
    yfull[,j] <- rbinom(n=N, size=1, prob=p)
  }
  ever.detected <- apply(yfull, 1, max)
  C <- sum(ever.detected)
  yobs <- yfull[ever.detected == 1,]
  cat(C, "out of", N, "animals present were detected.\n")
  return(list(N=N, p=p, C=C, T=T, yfull=yfull, yobs=yobs))
}
```

We create one realization of the stochastic process represented by sampling a population of size N . This gives us our data set.

```
data <- data.fn()
```

On execution of that function, we get a list of R objects. We can have a look at them by typing the name of the object, `data`, or we can get a summary of them by doing this:

```
str(data)
> str(data)
List of 6
 $ N      : num 100
 $ p      : num 0.5
 $ C      : num 87
 $ T      : num 3
 $ yfull: num [1:100, 1:3] 0 0 0 1 1 0 0 1 1 0 ...
 $ yobs  : num [1:87, 1:3] 0 0 1 1 0 0 1 1 0 1 ...
```

Here, `yfull` is the full capture-history matrix of all N animals, including those that were never captured and which have an all-zero capture-history. Evidently, this is not what we ever get to observe. The observed data are called `yobs`, and they have C rows only ($C \leq N$), corresponding to the observed count of animals. We will apply a model to use these data to make an inference based on the rules of probability, of how great N might likely be.

When modeling population size or other parameters in a model that contains population size using Bayesian MCMC techniques, one formidable

technical challenge is that the dimension of the parameter vector for N may change at every iteration of the MCMC algorithm. An ingenious solution is a technique called *data augmentation* (Tanner and Wong, 1987), introduced in the context of capture–recapture models by Royle et al. (2007a); see also Royle and Dorazio (2011) and Section 10.3. Parameter-expanded data augmentation (PX-DA), as it is more accurately called, consists of two things: (1) adding an arbitrary number of zeroes to the data set and (2) analyzing a reparameterized version of the original model. Essentially, it converts the closed-population model into an occupancy model (see Chapter 13) and turns the problem of estimating abundance N into that of estimating occupancy (ψ). Data augmentation may be simple to do in practice, but it has far-reaching consequences: it greatly simplifies the fitting of a large range of capture–recapture type models (Royle et al., 2007b; Royle and Dorazio, 2008).

Data augmentation consists of augmenting a data set by adding a large number of “potential”, unobserved individuals, with all zero-only encounter histories. The augmented data set has dimension M by T , where $M \gg N$ (remember, N is the unknown population size, and T is the number of sampling occasions). To this augmented data set we fit a zero-inflated version of the model we would fit if N were known. To do this, we add to the model a binary indicator variable, say z , which is an indicator for whether a row in the augmented data matrix represents a “real” individual, or one that does not exist in practice. These indicators are given a Bernoulli prior distribution, and the parameter of that distribution, say Ω , is estimable from the data. Ω may be called the inclusion probability, since it is the probability with which a member of the augmented data set, M , is included in the population of size N .

Data augmentation translates the problem of estimating N into the equivalent problem of estimating Ω , since the expectation of N is equal to $M\Omega$. Formally, data augmentation induces for N a discrete uniform prior on the interval $(0, M)$. In words, we make our analysis under the formal prior assumption “We do not know how big N is, but it could be any integer number between 0 and M , the size of the augmented data set, with equal probability”.

Although simple in reality, our experience suggests that the idea of data augmentation takes some time to sink in. So, here is a numerical example that summarizes what we have just said. Assume that we want to estimate the size of a population with $N = 100$. Of course, that there are 100 individuals is unknown to us because we observe only $C = 87$ of them. So, instead of analyzing the observed data set of size 87 and estimating detection probability p and population size through some variant of the canonical estimator $\hat{N} = C/\hat{p}$, we add to the observed detection history matrix, say, 150 rows consisting of all zeroes. This brings the size of the augmented data set to $M = 237$. This is equivalent to us saying that N could really be anywhere between 0 and 237. We then fit an occupancy model (see Chapter 13) to

these data, where we estimate detection probability p and the inclusion probability Ω and obtain the estimate of N as a derived parameter by summing up the latent indicators z (the expectation of N is $M\Omega$). So, let us see now how this works in practice for the data set previously generated.

```
# Augment data set by 150 potential individuals
nz <- 150
yaug <- rbind(data$yobs, array(0, dim=c(nz, data$T)))

# Specify model in BUGS language
sink("model.txt")
cat("
model {

# Priors
omega ~ dunif(0, 1)
p ~ dunif(0, 1)

# Likelihood
for (i in 1:M) {
  z[i] ~ dbern(omega)                # Inclusion indicators
  for (j in 1:T) {
    yaug[i,j] ~ dbern(p.eff[i,j])
    p.eff[i,j] <- z[i] * p          # Can only be detected if z=1
  } #j
} #i

# Derived quantities
N <- sum(z[])
}
", fill=TRUE)
sink()
```

We started this chapter claiming that capture–recapture models achieve a proper separation of the ecological and the observation process, so where can we see this in the model? In fact, the first Bernoulli distribution governing the inclusion probabilities is equivalent to the ecological process (which creates “real” and nonexisting individuals), whereas the second represents the observation process.

We present the analysis in the usual format by preparing all the ingredients that `bugs()` require to instruct WinBUGS and then letting WinBUGS run. We estimate N at 101.8 with 95% CRI of 93–114 (Fig. 6.1), which comfortably includes the truth of 100.

```
# Bundle data
win.data <- list(yaug = yaug, M = nrow(yaug), T = ncol(yaug))

# Initial values
inits <- function() list(z = rep(1, nrow(yaug)), p = runif(1, 0, 1))

# Parameters monitored
params <- c("N", "p", "omega")
```

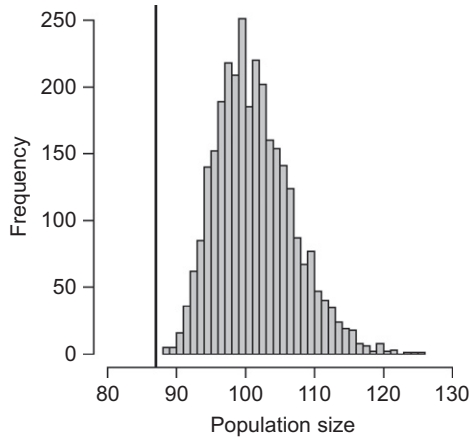


FIGURE 6.1 Posterior distribution of population size N (black line: observed number of animals).

MCMC settings

```
ni <- 2500
nt <- 2
nb <- 500
nc <- 3
```

Call WinBUGS from R (BRT <1 min)

```
out <- bugs(win.data, inits, params, "model.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=TRUE, bugs.directory=
  bugs.dir, working.directory=getwd())
```

Summarize posteriors

```
print(out, dig=3)
hist(out$sims.list$N, nclass=50, col="gray", main="", xlab=
  "Population size", las=1, xlim=c(80, 150))
abline(v=data$C, lwd=3)
```

```
[...]
      mean    sd  2.5%  25%   50%   75%  97.5%  Rhat n.eff
N      101.803 5.547 93.000 98.000 101.000 105.000 114.000 1.003   940
p       0.479 0.038  0.404  0.453  0.480  0.505  0.555 1.002  1200
omega   0.430 0.039  0.356  0.404  0.428  0.456  0.507 1.001  3000
[...]
```

No doubt some of you will think that data augmentation is some sort of voodoo. Yet, it is not. To convince yourself that the number of zeroes added does not affect the estimates of detection probability (p) and population size (N), we repeat the analysis with the following numbers of all-zero detection histories added (i.e., values of nz): 5, 150, and 1500. For each case, we will inspect the estimates for N , p , and Ω , along with their

uncertainty. To help understand data augmentation and this comparison, we also plot the posterior for N each time:

```

nz <- 5
yaug <- rbind(data$yobs, array(0, dim=c(nz, data$T)))
win.data <- list(yaug=yaug, M=dim(yaug)[1], T=dim(yaug)[2])
out5 <- bugs(win.data, inits, params, "model.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=FALSE, bugs.directory=
  bugs.dir, working.directory=getwd())
print(out5, dig=3)
par(mfrow=c(3, 1))
hist(out5$sims.list$N, nclass=30, col="gray", main="Augmentation
  by 5", xlab="Population size", las=1, xlim=c(80, 140))
abline(v=data$C, col="black", lwd=5)
abline(v=mean(out5$sims.list$N), col="blue", lwd=5)

nz <- 150
yaug <- rbind(data$yobs, array(0, dim=c(nz, data$T)))
win.data <- list(yaug=yaug, M=dim(yaug)[1], T=dim(yaug)[2])
out150 <- bugs(win.data, inits, params, "model.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=FALSE, bugs.directory=
  bugs.dir, working.directory=getwd())
print(out150, dig=3)
hist(out150$sims.list$N, nclass=30, col="gray", main="Augmentation by
  150", xlab="Population size", las=1, xlim=c(80, 140))
abline(v=data$C, col="black", lwd=5)
abline(v=mean(out150$sims.list$N), col="blue", lwd=5)

nz <- 1500
yaug <- rbind(data$yobs, array(0, dim=c(nz, data$T)))
win.data <- list(yaug=yaug, M=dim(yaug)[1], T=dim(yaug)[2])
out1500 <- bugs(win.data, inits, params, "model.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=FALSE, bugs.directory=
  bugs.dir, working.directory=getwd())
print(out1500, dig=3)
hist(out1500$sims.list$N, nclass=30, col="gray", main=
  "Augmentation by 1500", xlab="Population size", las=1, xlim=
  c(80, 140))
abline(v=data$C, col="black", lwd=5)
abline(v=mean(out1500$sims.list$N), col="blue", lwd=5)

```

This exercise shows that provided you have augmented the data set enough (Fig. 6.2, central and bottom panels), data augmentation does not have any effect on the estimates, but simply on efficiency: more data augmentation is more costly in terms of computation time. But up to Monte Carlo error, the estimates and their standard errors remain the same (up to Monte Carlo error, as you can see in the posterior summaries). And how do you diagnose not enough data augmentation? Simple: just look at the posterior distribution of N : if it is truncated on the right by your choice of M (as it is in the top panel), you need to repeat the analysis with more zeroes added, that is, larger M .

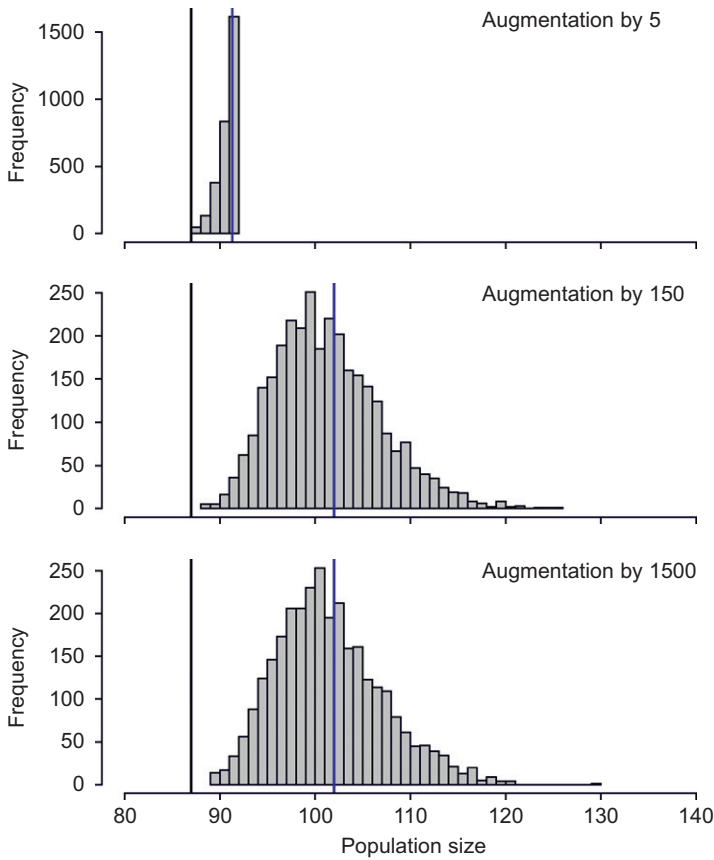


FIGURE 6.2 Posterior distributions of population size N for the same data set but under different degrees of data augmentation (black line: observed number of animals, blue line: posterior mean).

6.2.2 Time Effects: Model M_t

Another model assumes that detection probability p varies by occasion, perhaps because of weather conditions or because different traps or detection devices were used. The simultaneous use of different detection methods (e.g., trap types or observers) during a single occasion can be treated exactly as time effects in capture–recapture modeling. This model is called model M_t by Otis et al. (1978).

Define function to simulate data under M_t

```
data.fn <- function(N=100, mean.p=0.5, T=3, time.eff=runif(T, -2,
2)) {
  yfull <- yobs <- array(NA, dim=c(N, T) )
  p.vec <- array(NA, dim=T)
```

```

for (j in 1:T) {
  p <- plogis(log(mean.p / (1-mean.p)) + time.eff[j])
  yfull[,j] <- rbinom(n=N, size=1, prob=p)
  p.vec[j] <- p
}
ever.detected <- apply(yfull, 1, max)
C <- sum(ever.detected)
yobs <- yfull[ever.detected == 1,]
cat(C, "out of", N, "animals present were detected.\n")
return(list(N=N, p.vec=p.vec, C=C, T=T, yfull=yfull, yobs=
yobs))
}

```

We create one data set. Again, note that each time you do this you will get a slightly different realization of the same stochastic process. It may even (rarely) happen to have all $N = 100$ individuals captured.

```
data <- data.fn()
```

We do the same kind of analysis as before but just have to add the time effects in the BUGS code.

```

# Augment data set
nz <- 150
yaug <- rbind(data$yobs, array(0, dim=c(nz, data$T)))

# Specify model in BUGS language
sink("model.txt")
cat("
model {

# Priors
omega ~ dunif(0, 1)
for (i in 1:T){
  p[i] ~ dunif(0, 1)
}

# Likelihood
for (i in 1:M){
  z[i] ~ dbern(omega)
  for (j in 1:T){
    yaug[i,j] ~ dbern(p.eff[i,j])
    p.eff[i,j] <- z[i] * p[j]
  } #j
} #i

# Derived quantities
N <- sum(z[])
}
", fill=TRUE)
sink()

# Bundle data
win.data <- list(yaug=yaug, M=nrow(yaug), T=ncol(yaug))

```

```

# Initial values
inits <- function() list(z = rep(1, nrow(yaug)), p = runif(data$T, 0, 1))

# Parameters monitored
params <- c("N", "p", "omega")

# MCMC settings
ni <- 2500
nt <- 2
nb <- 500
nc <- 3

# Call WinBUGS from R (BRT <1 min)
out <- bugs(win.data, inits, params, "model.txt", n.chains = nc,
  n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
  bugs.dir, working.directory = getwd())

# Summarize posteriors
print(out, dig = 3)
hist(out$sims.list$N, nclass = 40, col = "gray", main = "", xlab =
  "Population size", las = 1, xlim = c(70, 150))
abline(v = data$C, col = "black", lwd = 3)
[...]
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
N	99.854	9.321	85.000	93.000	99.000	105.000	122.000	1.002	1500
p[1]	0.454	0.063	0.333	0.411	0.453	0.497	0.580	1.002	1900
p[2]	0.346	0.055	0.246	0.307	0.343	0.382	0.464	1.001	3000
p[3]	0.326	0.055	0.224	0.288	0.325	0.362	0.439	1.001	3000
omega	0.445	0.052	0.354	0.408	0.441	0.474	0.565	1.001	2000

```

[...]
```

So, all is similar as before. Note that an important contribution to the uncertainty about population size N comes from the uncertainty about detection probability p . The more we need to estimate for p , that is, the more parameters we need to describe the patterns in p , the less precise will be our estimate of N . You can fit the model M_0 to this same data set to confirm this. Therefore, when designing a study, it is always good to try and eliminate as many factors that will introduce variance in p as possible. Of course, we can always model such factors, but we pay with reduced precision.

In the above model, we have assumed that the time effects are fixed (although for convenience we simulated detection probabilities from a uniform random number generator). It would be straightforward to fit random instead of fixed time effects: just add a common prior distribution to the set of detection probabilities and then estimate the hyperparameters of that prior distribution. It is customary to model detection probabilities on a transformed scale and assume a normal distribution for the random time effects. The usual transformation is the logit, that is, $\log[p/(1-p)]$. We leave this for one of the exercises for you to try out; see also Section 7.4.2.

6.2.3 Behavioral or Memory Effects: Model M_b

Another possible effect is within-capture-history dependence of detection probability p . This model is called model M_b by Otis et al. (1978). It fits a different parameter for p depending on whether an animal was caught before or not. We need to distinguish between immediate (or ephemeral) trap response and permanent trap response. We might also envision something in between and make p a function of the number of times that an animal was caught over the last, say, 3 or 5 or 10 occasions. In this example, we model immediate trap response, that is, when an individual is captured, it has a different detection probability only on the immediately following occasion, but not thereafter, unless it is captured again.

In the following function, we simulate trap response on the probability scale. We could also do this on the logit scale, which would be more natural if we were to combine trap response with other effects. We denote the capture probability as c or p depending on whether an individual has or has not been captured during the preceding occasion.

```
# Define function to simulate data under Mb
data.fn <- function(N=200, T=5, p=0.3, c=0.4) {
  yfull <- yobs <- array(NA, dim=c(N, T) )
  p.eff <- array(NA, dim=N)

  # First capture occasion
  yfull[,1] <- rbinom(n=N, size=1, prob=p)

  # Later capture occasions
  for (j in 2:T) {
    p.eff <- (1 - yfull[, (j-1)]) * p + yfull[, (j-1)] * c
    yfull[,j] <- rbinom(n=N, size=1, prob=p.eff)
  }

  ever.detected <- apply(yfull, 1, max)
  C <- sum(ever.detected)
  yobs <- yfull[ever.detected == 1,]
  cat(C, "out of", N, "animals present were detected.\n")
  return(list(N=N, p=p, c=c, C=C, T=T, yfull=yfull, yobs=yobs))
}
```

We create one data set with trap-happiness ($p < c$). You may want to look at the data set created and try to see how trap response induces serial autocorrelation in the individual detection histories. This needs to be accounted for in the analysis.

```
data <- data.fn(N=200)

# Augment data set
nz <- 150
yaug <- rbind(data$yobs, array(0, dim=c(nz, data$T)))

# Specify model in BUGS language
sink("model.txt")
cat("
model {
```



```

# Priors
omega ~ dunif(0, 1)
p ~ dunif(0, 1)      # Cap prob when not caught at t-1
c ~ dunif(0, 1)      # Cap prob when caught at t-1

# Likelihood
for (i in 1:M) {
  z[i] ~ dbern(omega)

  # First occasion
  yaug[i,1] ~ dbern(p.eff[i,1])
  p.eff[i,1] <- z[i] * p

  # All subsequent occasions
  for (j in 2:T){
    yaug[i,j] ~ dbern(p.eff[i,j])
    p.eff[i,j] <- z[i] * ( (1-yaug[i,(j-1)]) * p + yaug[i,(j-1)] * c )
  } #j
} #i

# Derived quantities
N <- sum(z[])
trap.response <- c - p
}
",fill=TRUE)
sink()

# Bundle data
win.data <- list(yaug = yaug, M = nrow(yaug), T = ncol(yaug))

# Initial values
inits <- function() list(z = rep(1, nrow(yaug)), p = runif(1, 0, 1))

# Parameters monitored
params <- c("N", "p", "c", "trap.response", "omega")

# MCMC settings
ni <- 2500
nt <- 2
nb <- 500
nc <- 3

# Call WinBUGS from R (BRT <1 min)
out <- bugs(win.data, inits, params, "model.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=TRUE, bugs.directory=
  bugs.dir, working.directory=getwd())

# Summarize posteriors
print(out, dig=3)
[...]
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
N	203.129	11.878	183.000	195.000	202.000	210.000	230.000	1.010	230
p	0.273	0.026	0.223	0.255	0.273	0.291	0.325	1.008	280
c	0.416	0.032	0.356	0.395	0.417	0.438	0.477	1.002	1300
trap.response	0.143	0.041	0.065	0.115	0.144	0.170	0.225	1.003	830
omega	0.652	0.046	0.567	0.620	0.650	0.682	0.745	1.007	350

```

[...]
```

```
hist(out$sims.list$N, nclass=40, col="gray", main="", xlab=
  "Population size", las=1, xlim=c(150, 300))
abline(v= data$C, col="black", lwd=3)
```

6.2.4 Individual (Random) Effects: The Heterogeneity Model M_h

The third model in the Otis et al. (1978) catalog is often called the “heterogeneity model”, or M_h for short. This term is misleading, since it suggests a single model. However, a multitude of potential models can all be subsumed under the term M_h . What M_h means is that we assume that each individual has its own detection probability, and that this heterogeneity cannot be described by known and measured covariates; instead, we assume that there are individual latent (random) effects in detection probability.

There are various possible statistical descriptions of this “diffuse” heterogeneity, for instance, finite mixtures (Pledger, 2000) and beta binomial or logistic-normal continuous mixtures (Coull and Agresti, 1999; Dorazio and Royle, 2003). However, Link (2003) has shown that population size N is unfortunately not an identifiable parameter *across different classes of models* for individual heterogeneity in p , such as finite mixtures, beta binomial, or logistic-normal continuous mixtures. This means that models with a different specification of individual heterogeneity may well give very different answers about N , and yet, we have no data-based criterion to choose among them, such as likelihood ratio tests or AIC. However, we believe that simply ignoring individual heterogeneity would mean to throw out the baby with the bathtub, since not specifying effect h would lead to underestimated population size (Williams et al., 2002) and thus probably to worse inference, on average, than specifying the “wrong” mixture distribution (Kéry, 2011a).

Here, we consider one such mixture model, the logistic-normal (Coull and Agresti, 1999), which is a continuous mixture model that allows flexible modeling of individual effects along with others, such as time or behavior effects (see [Section 6.3](#)). In this model, individual heterogeneity is modeled as random noise around some mean on a logit-transformed scale, and the model for the noise is the normal distribution. Therefore, it is called logistic normal. OpenBUGS contains another example of model M_h with data augmentation (Examples > Ecology examples > Birds).

In our example, we aggregate the capture-histories to capture frequencies, that is, counts of the number of times each individual was encountered. In the absence of factors that induce time- or trap-dependent effects on detection, we do not need to model the binary detection histories and can parsimoniously model capture frequencies. In our code, `mean.p` is the average detection probability of individuals in the population studied, and `sd` is

the standard deviation of the normal distribution which describes the heterogeneity in the individual detection probability on the logit scale. That is, we will estimate a normal distribution for individual detection probability with mean equal to $\text{logit}(\text{mean.p})$ and standard deviation equal to sd .

Define function to simulate data under Mh

```
data.fn <- function(N=100, mean.p=0.4, T=5, sd=1) {
  yfull <- yobs <- array(NA, dim=c(N, T) )
  mean.lp <- log(mean.p / (1-mean.p))
  p.vec <- plogis(mean.lp+ rnorm(N, 0, sd))

  for (i in 1:N){
    yfull[i,] <- rbinom(n=T, size=1, prob=p.vec[i])
  }

  ever.detected <- apply(yfull, 1, max)
  C <- sum(ever.detected)
  yobs <- yfull[ever.detected == 1,]
  cat(C, "out of", N, "animals present were detected.\n")
  hist(p.vec, xlim=c(0,1), nclass=20, col="gray", main="", xlab =
    "Detection probability", las=1)
  return(list(N=N, p.vec=p.vec, mean.lp=mean.lp, C=C, T=T, yfull =
    yfull, yobs=yobs))
}
```

We create one data set and get a summary of how many individuals were ever detected and a histogram of the individual detection probabilities (Fig. 6.3).

```
data <- data.fn()
84 out of 100 animals present were detected.
```

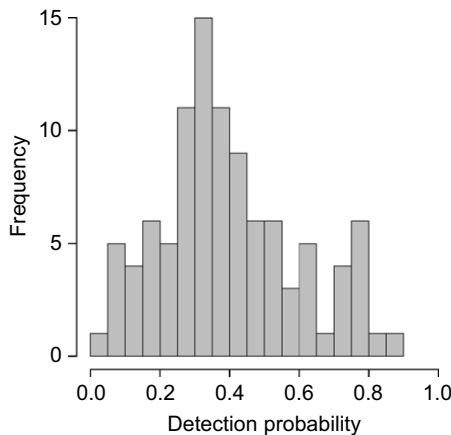


FIGURE 6.3 Distribution of individual detection probability for 100 simulated individuals.

We aggregate the capture-histories to capture frequencies and sort them for convenience. Then, we run WinBUGS on the model. As usual, random-effects models require longer Markov chains to achieve convergence. They also require more data augmentation: the posterior for N is more drawn out (there is more uncertainty about N) and so to avoid truncation of the posterior, we have to allow for more potential individuals by increasing the size of M .

```
# Aggregate capture histories and augment data set
y <- sort(apply(data$yobs, 1, sum), decreasing = TRUE)
nz <- 300
yaug <- c(y, rep(0, nz))
yaug
[1] 5 5 5 5 5 5 5 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3
[32] 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[63] 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
[... ]
[373] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

# Specify model in BUGS language
sink("model.txt")
cat("
model {

# Priors
omega ~ dunif(0, 1)
mean.lp <- logit(mean.p)
mean.p ~ dunif(0, 1)
tau <- 1 / (sd * sd)
sd ~ dunif(0, 5)

# Likelihood
for (i in 1:M){
  z[i] ~ dbern(omega)
  logit(p[i]) <- eps[i]
  eps[i] ~ dnorm(mean.lp, tau)I(-16, 16)      # See web appendix A in
                                              Royle (2009)

  p.eff[i] <- z[i] * p[i]
  y[i] ~ dbin(p.eff[i], T)
}

# Derived quantities
N <- sum(z[])
}
", fill = TRUE)
sink()

# Bundle data
win.data <- list(y = yaug, M = length(yaug), T = ncol(data$yobs))

# Initial values
inits <- function() list(z = rep(1, length(yaug)), sd = runif(1, 0.1, 0.9))

# Parameters monitored
params <- c("N", "mean.p", "sd", "omega")
```

```
# MCMC settings
ni <- 25000
nt <- 2
nb <- 5000
nc <- 3

# Call WinBUGS from R (BRT 6 min)
out <- bugs(win.data, inits, params, "model.txt", n.chains=nc,
            n.thin=nt, n.iter=ni, n.burnin=nb, debug=TRUE, bugs.directory=
            bugs.dir, working.directory=getwd())

# Summarize posteriors
print(out, dig=3)
hist(out$sims.list$N, nclass=50, col="gray", main="", xlab =
      "Population size", las=1, xlim=c(80, 250))
abline(v=data$C, col="black", lwd=3)
[...]
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
N	101.584	12.927	87.000	93.000	98.000	106.000	134.000	1.015	590
mean.p	0.405	0.074	0.225	0.364	0.416	0.457	0.517	1.023	830
sd	1.091	0.355	0.510	0.845	1.050	1.289	1.904	1.017	260
omega	0.266	0.040	0.205	0.239	0.261	0.286	0.358	1.007	950

```
[...]
```

We see that the precision for N is lower compared with that under the previous models. Estimation in individual-effects models is notoriously more challenging than in models without such random effects. We see this in longer run time to get an adequate posterior sample as well as in the drawn-out posterior distribution of the key parameter N (Fig. 6.4).

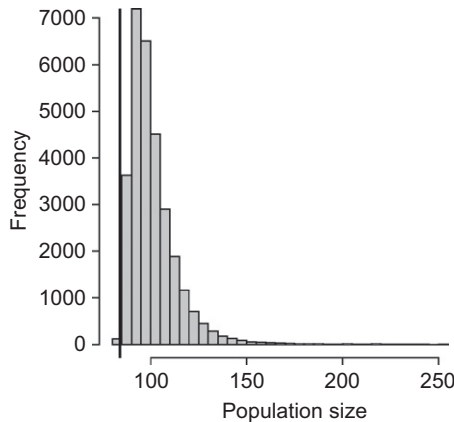


FIGURE 6.4 Posterior distribution of population size N under the heterogeneity model M_h (black line: observed number of individuals). The posterior mass does not extend below the observed number of individuals (84).

6.2.5 Combined Effects: Model M_{th}

Time, behavioral, and individual effects may also be combined pairwise or all three, giving rise to what Otis et al. (1978) have called models M_{th} and so forth. Again, this terminology is useful, but may also be misleading, since it may appear as if there was a single such model. Instead, all that a term such as M_{th} means is that a model includes the effects of both time and individual heterogeneity. This term does not tell us about how time and individual heterogeneity is parameterized, for instance, whether time effects are fixed or random or whether a finite or a continuous mixture is assumed for individual latent effects. It also does not specify whether the two effects are assumed to be independent or to act in a combined way (i.e., whether time and heterogeneity act as main effects only or with an interaction).

Here, we give one example of a model M_{th} , which may be more precisely described as having additive (i.e., main) fixed effects of time and random (logistic normal) effects of individuals. This is a useful model in many circumstances in the absence of behavioral effects. Now, we consider effects along both dimensions of the capture-history matrix (corresponding to Table 6.1, with columns representing time, and rows representing individuals). so, we model the unaggregated data. Furthermore, we will model all effects on a logistic scale.

```
# Define function to simulate data under Mth
data.fn <- function(N=100, T=5, mean.p=0.4, time.effects=runif(T,
-1, 1), sd=1){
  yfull <- yobs <- p <- array(NA, dim=c(N, T) )
  mean.lp <- log(mean.p / (1-mean.p))      # mean p on logit scale
  eps <- rnorm(N, 0, sd)                   # Individual effects

  for (j in 1:T){
    pp <- p[,j] <- plogis(mean.lp + time.effects[j] + eps)
    yfull[,j] <- rbinom(n=N, size=1, prob=pp)
  }

  ever.detected <- apply(yfull, 1, max)
  C <- sum(ever.detected)
  yobs <- yfull[ever.detected == 1,]
  cat(C, "out of", N, "animals present were detected.\n")
  cat("Mean p per occasion:", round(apply(p, 2, mean), 2), "\n")
  par(mfrow=c(2,1))
  plot(plogis(mean.lp + time.effects), xlab="Occasion", type="b",
       main="Approx. mean p at each occasion", ylim=c(0, 1))
  hist(plogis(mean.lp + eps), xlim=c(0, 1), col="gray", main=
       "Approx. distribution of p at average occasion")
  return(list(N=N, mean.lp=mean.lp, time.effects=time.effects,
             sd=sd, eps=eps, C=C, T=T, yfull=yfull, yobs=yobs))
}
```

As shown below, this function can also be used to generate data sets under models M_0 , M_t , and M_h by writing as one or two arguments

`time.effects = runif(5, 0, 0)` and `sd = 0` (you have to adjust T manually). Here, we create one data set under the model with both effects present and analyze that. We avoid to use the WinBUGS logit function and instead define the logit explicitly, which may sometimes avoid numerical problems.

```
data <- data.fn()
85 out of 100 animals present were detected.
Mean p per occasion: 0.35 0.41 0.47 0.61 0.45

# data<-data.fn(T=10, mean.p=0.2, time.effects=runif(10, 0, 0),
# sd=0) # M0
# data<-data.fn(T=10, mean.p=0.5, time.effects=runif(10, 0, 0),
# sd=1) # Mh
# data <- data.fn(T=10, sd=0) # Mt

# Augment data set
nz <- 300
yaug <- rbind(data$yobs, array(0, dim=c(nz, data$T)))

# Specify model in BUGS language
sink("model.txt")
cat("
model {

# Priors
omega ~ dunif(0, 1)
for (j in 1:T){
  mean.lp[j] <- log(mean.p[j] / (1 - mean.p[j])) # Define logit
  mean.p[j] ~ dunif(0, 1)
}
tau <- 1 / (sd * sd)
sd ~ dunif(0,5)

# Likelihood
for (i in 1:M){
  z[i] ~ dbern(omega)
  eps[i] ~ dnorm(0, tau) I(-16, 16) # See web appendix A in Royle (2009)
  for (j in 1:T){
    lp[i,j] <- mean.lp[j] + eps[i]
    p[i,j] <- 1 / (1 + exp(-lp[i,j])) # Define logit
    p.eff[i,j] <- z[i] * p[i,j]
    y[i,j] ~ dbern(p.eff[i,j])
  } #j
} #i

# Derived quantities
N <- sum(z[])
}
", fill=TRUE)
sink()

# Bundle data
win.data <- list(y=yaug, M=nrow(yaug), T=ncol(yaug))

# Initial values
inits <- function() list(z=rep(1, nrow(yaug)), sd=runif(1, 0.1, 0.9))
```

```

# Parameters monitored
params <- c("N", "mean.p", "mean.lp", "sd", "omega")

# MCMC settings
ni <- 25000
nt <- 2
nb <- 5000
nc <- 3

# Call WinBUGS from R (BRT 47 min)
out <- bugs(win.data, inits, params, "model.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=TRUE, bugs.directory=
  bugs.dir, working.directory=getwd())

# Summarize posteriors
print(out, dig=3)
hist(out$sims.list$N, nclass=50, col="gray", main="", xlab=
  "Population size", las=1, xlim=c(80, 200))
abline(v=data$C, col="black", lwd=3)
[...]
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
N	103.665	10.416	89.000	96.000	102.000	109.000	129.000	1.001	6900
mean.p[1]	0.381	0.083	0.214	0.325	0.383	0.439	0.540	1.001	13000
mean.p[2]	0.419	0.085	0.244	0.363	0.422	0.478	0.578	1.001	30000
mean.p[3]	0.419	0.086	0.243	0.361	0.422	0.479	0.578	1.001	12000
mean.p[4]	0.582	0.089	0.388	0.527	0.590	0.645	0.736	1.001	9000
mean.p[5]	0.322	0.078	0.169	0.269	0.322	0.374	0.474	1.001	21000
mean.lp[1]	-0.500	0.371	-1.301	-0.731	-0.476	-0.244	0.162	1.001	10000
mean.lp[2]	-0.338	0.365	-1.132	-0.562	-0.314	-0.086	0.313	1.001	20000
mean.lp[3]	-0.340	0.369	-1.138	-0.570	-0.314	-0.083	0.315	1.001	9200
mean.lp[4]	0.342	0.374	-0.454	0.109	0.362	0.597	1.026	1.001	6600
mean.lp[5]	-0.772	0.376	-1.590	-1.002	-0.746	-0.514	-0.104	1.001	18000
sd	1.411	0.347	0.795	1.173	1.386	1.624	2.173	1.003	2700
omega	0.270	0.035	0.211	0.246	0.267	0.291	0.348	1.001	19000

```

[...]
```

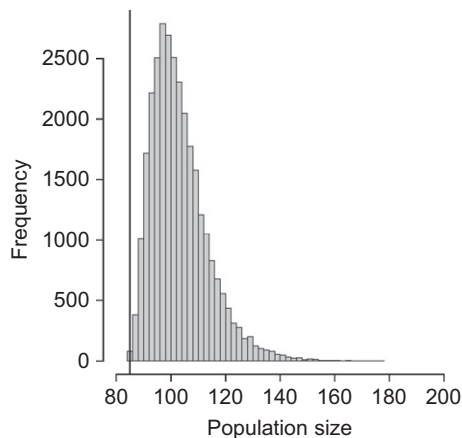


FIGURE 6.5 Posterior distribution of population size N under model M_{th} (black line: observed number of individuals).

We obtain decent parameter estimates (see Fig. 6.5 and preceding posterior summary), judging by their resemblance to the parameter values with which we generated our data set.

6.3 ANALYSIS OF A REAL DATA SET: MODEL M_{tbh} FOR SPECIES RICHNESS ESTIMATION

To illustrate estimation of the size of a real population, we will look at the size of a bird community. It has been pointed out repeatedly that the number of species in a community is analogous to the number of individuals in a population (e.g., Boulinier et al., 1998; Nichols et al., 1998a). Therefore, the same inferential framework, closed-population models (though not, for instance, distance sampling), can be used for both (Kéry, 2011a). One important consideration specific to species richness estimation is that a model that does not allow for individual heterogeneity will probably not be adequate: species in a community usually differ by many factors that determine their detection probability, most of all perhaps in their abundance. Thus, detection probability is likely to vary a great deal more among the species in a community than among the individuals in a population. Since not allowing for heterogeneity in p yields severe underestimates of N , species richness estimation with closed capture–recapture models should always be based on a heterogeneity model.

We will use bird point count data from the Czech republic (data courtesy of Jiri Reif). With some colleagues, Jiri cycled an East–West transect spanning almost the whole country and conducted a 5 min point count after every 500 m in 2004–2005 (768 points in total). They repeated this five times within the same breeding season. A total of 146 species were detected (Reif et al., 2008), among them the wryneck (Fig. 6.6). We will use their data from one such point: point count number 610. For each species and occasion, the data contain the number of individuals counted. For the analysis, we first reduce these data to simple detection/nondetection data.

```
# Read in data and look at them
p610 <- read.table("p610.txt", header = TRUE)
y <- p610[,5:9]                                # Grab counts
y[y > 1] <- 1                                    # Counts to det-nondetections
C <- sum(apply(y, 1, max)) ; print(C)             # Number of observed species
table(apply(y, 1, sum))                          # Capture-frequencies
```

The data contain the detection histories not only of the 31 species detected at this particular point but also those of all species detected anywhere along the full national transect. Hence, this data set is “naturally data augmented”. We will not add any more zeroes because 115 zeroes corresponding to the species not detected at point 610, but detected somewhere else in the Czech



FIGURE 6.6 Wryneck (*Jynx torquilla*), Latvia, 2004 (Photograph by T. Muukkonen).

Republic, are probably enough. But we can easily check that by inspecting the posterior distribution for N .

All models in this chapter assume closure (here: community closure). In the present context, this means that each species that is part of the sampled community at a given point must be available for detection during all replicate surveys. This may well not be true. For instance, many bird species are migrants, and some may not yet have arrived during early surveys. If typical arrival dates of each migrant species are known, then this problem could be dealt with by simply turning the corresponding data into missing values. Our data set would then no longer be balanced, but this in general poses no problem to the estimation framework, as long as some replicate surveys are available for some species at least.

Here, we will ignore this possibility and assume closure. We adopt a model M_{tbh} which has additive effects of time, behavior, and individual heterogeneity on the detection probability of a species. The test for behavioral effects is particularly interesting, since it has been suggested that in monitoring programs, people may detect species detected previously more or less easily thereafter (Riddle et al., 2010).

```
# Specify model in BUGS language
sink("M_tbh.txt")
cat("
model {
```

```

# Priors
omega ~ dunif(0, 1)
for (j in 1:T){
  alpha[j] <- log(mean.p[j] / (1-mean.p[j])) # Define logit
  mean.p[j] ~ dunif(0, 1) # Detection intercepts
}
gamma ~ dnorm(0, 0.01)
tau <- 1 / (sd * sd)
sd ~ dunif(0, 3)

# Likelihood
for (i in 1:M){
  z[i] ~ dbern(omega)
  eps[i] ~ dnorm(0, tau)I(-16, 16)

  # First occasion: no term for recapture (gamma)
  y[i,1] ~ dbern(p.eff[i,1])
  p.eff[i,1] <- z[i] * p[i,1]
  p[i,1] <- 1 / (1 + exp(-lp[i,1]))
  lp[i,1] <- alpha[1] + eps[i]

  # All subsequent occasions: includes recapture term (gamma)
  for (j in 2:T){
    y[i,j] ~ dbern(p.eff[i,j])
    p.eff[i,j] <- z[i] * p[i,j]
    p[i,j] <- 1 / (1 + exp(-lp[i,j]))
    lp[i,j] <- alpha[j] + eps[i] + gamma * y[i, (j-1)]
  } #j
} #i

# Derived quantities
N <- sum(z[])
}
", fill=TRUE)
sink()

# Bundle data
win.data <- list(y=as.matrix(y), M=nrow(y), T=ncol(y))

# Initial values
inits <- function() list(z=rep(1, nrow(y)), sd=runif(1, 0.1, 0.9))

# Parameters monitored
params <- c("N", "mean.p", "gamma", "sd", "omega")

# MCMC settings
ni <- 50000
nt <- 4
nb <- 10000
nc <- 3

# Call WinBUGS from R (BRT 24 min)
out <- bugs(win.data, inits, params, "M_tbh.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=TRUE, bugs.directory=
  bugs.dir, working.directory=getwd())

```

```
# Summarize posteriors and plot posterior for N
print(out, dig=3)
par(mfrow=c(1,2))
hist(out$sims.list$N, breaks=35, col="gray", main="", xlab=
      "Community size", las=1, xlim=c(30, 100), freq=FALSE)
abline(v=C, col="black", lwd=3)
[...]
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
N	42.050	7.444	33.000	37.000	40.000	45.000	61.000	1.002	2300
mean.p[1]	0.241	0.084	0.096	0.181	0.235	0.294	0.420	1.001	13000
mean.p[2]	0.294	0.096	0.126	0.224	0.288	0.356	0.495	1.002	3100
mean.p[3]	0.295	0.097	0.125	0.225	0.289	0.358	0.502	1.002	2200
mean.p[4]	0.222	0.084	0.083	0.161	0.214	0.274	0.408	1.002	1600
mean.p[5]	0.319	0.099	0.140	0.248	0.315	0.385	0.522	1.002	3200
gamma	-0.080	0.496	-1.062	-0.409	-0.099	0.268	0.892	1.010	300
sd	0.709	0.420	0.044	0.387	0.676	0.983	1.622	1.005	11000
omega	0.291	0.062	0.193	0.248	0.283	0.325	0.435	1.002	3100

```
[...]
```

After 24 min worth of sampling the joint posterior distribution of parameters, estimates look decent, and so we make a first interpretation of these results. With 31 species detected, we estimate there were in fact 42, therefore, only $31/42 = 74\%$ were ever detected (Fig. 6.7a). Given the short survey duration, this appears reasonable. The average detection probability of a species in the community was between 0.22 and 0.32 and did not appear to vary much (in view of the wide uncertainty around these estimates). No behavioral (“memory”) effect on detection probability was discernible (95% CRI of gamma: -1.06 to 0.89). Finally, the standard deviation of the heterogeneity distribution was estimated at 0.71, which, not surprisingly, is considerable; species differ greatly.

It appears likely that whether or not “memory” exists should partly be related to scale. If a single person goes out to a small area and surveys on consecutive days then maybe he remembers what he saw. But, over large scales, over long time, and sampling many species, it is probably less important (Royle, pers. comm.). Since behavioral effects appeared to be absent, we might simplify the model and fit a time and heterogeneity model (i.e., M_{th}).

We have mentioned that inference under model M_0 will often lead to a serious negative bias in the estimate of N when there is heterogeneity among species in p . We illustrate this next by fitting model M_0 .

```
# Define model
sink("M0.txt")
cat("
model {

# Priors
omega ~ dunif(0, 1)
p ~ dunif(0, 1)
```

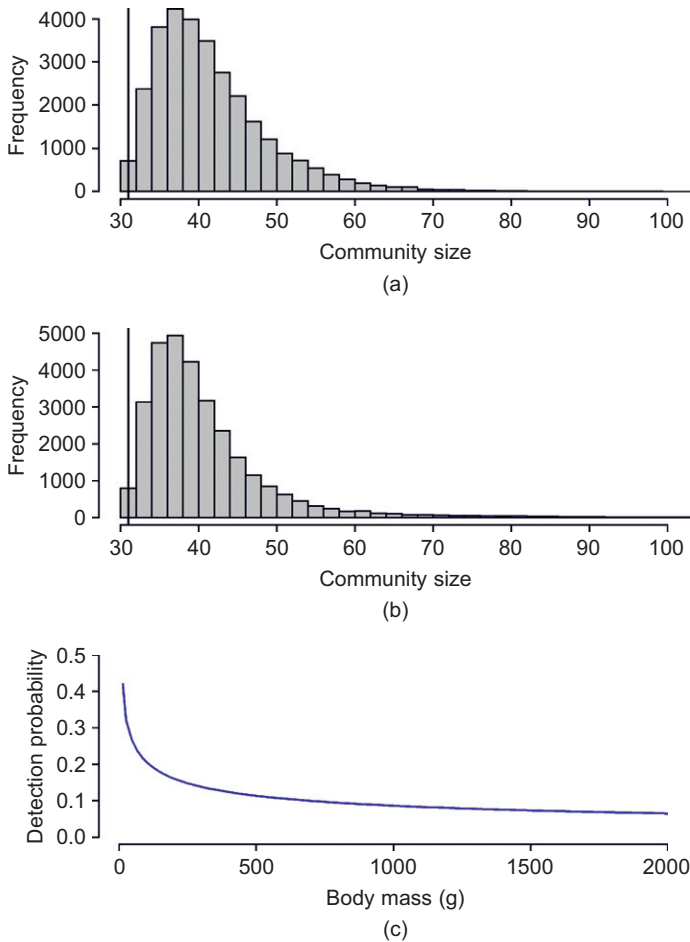


FIGURE 6.7 Analysis of community size (species richness, N) at point 610 in the Czech transect data (courtesy of Jiri Reif). Posterior distribution of N (a) under a model with time, behavioral, and heterogeneity effects (M_{tbh}) and (b) under a model with effects of time and the individual covariate body mass (M_{t+X}). The black line denotes the observed number of species. (c) Effect of body mass (g) on detection probability.

```
# Likelihood
for (i in 1:M){
  z[i] ~ dbern(omega)
  for (j in 1:T){
    y[i,j] ~ dbern(p.eff[i,j])
    p.eff[i,j] <- z[i] * p
  } #j
} #i
```

```

# Derived quantities
N <- sum(z[])
} # end model
", fill = TRUE)
sink()

# Initial values
inits <- function() list(z = rep(1, nrow(y)))

# Define parameters to be monitored
params <- c("N", "p", "omega")

# MCMC settings
ni <- 50000
nt <- 4
nb <- 10000
nc <- 3

# Call WinBUGS from R (BRT 1 min)
out0 <- bugs(win.data, inits, params, "M0.txt", n.chains = nc,
  n.thin = nt, n.iter = ni, n.burnin = nb, debug = FALSE, bugs.directory =
  bugs.dir, working.directory = getwd())

# Inspect output
print(out0, dig = 3)
[...]
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
N	37.877	3.970	32.000	35.000	37.00	40.000	47.000	1.001	21000
p	0.301	0.044	0.216	0.271	0.30	0.331	0.390	1.001	16000
omega	0.263	0.045	0.182	0.231	0.26	0.291	0.358	1.001	30000

```

[...]
```

Under model M_0 , we estimate 38 instead of 42 species. This illustrates the point that ignoring individual heterogeneity in detection probability produces underestimates of, and too short standard errors for, population size N .

6.4 CAPTURE–RECAPTURE MODELS WITH INDIVIDUAL COVARIATES: MODEL M_{t+X}

The old classification of Otis et al. (1978) of closed-population models does not include individual covariates. However, continuous detection covariates may often be available, and their effects could be modeled. The challenge is that the covariate values for undetected individuals are not known. Next, we illustrate a model described by Royle (2009), who uses data augmentation to analyze the joint likelihood of the capture-histories and the covariate values. In this model, we describe the distribution of the covariate values in the statistical population in addition to the probabilistic description of the capture-histories. We will give two illustrations for this model: first, in the context of species

richness estimation for the Czech transect data and second, in the context of population size estimation in a mussel.

6.4.1 Individual Covariate Model for Species Richness Estimation

In our first example, we model the effect of body mass on detection probability in the Czech bird data. Body mass could affect detection probability p in various ways. First, everything else equal, larger birds are rarer than smaller birds; hence, we would expect p to be lower than for smaller birds. Second, larger birds have larger territories, so their p will contain an important availability component (Kéry and Schmidt, 2008): when they are in a part of their territory that is not sampled, they cannot be detected, and this will lower their p . In contrast, and third, a larger bird may intrinsically be more visible than a smaller bird. Thus, we might expect an effect of body size on p , though its direction may not be obvious.

In the Czech bird example, the total list of species that could occur at point 610 is assumed to be known; it is those 146 species that were ever detected anywhere in the Czech transect study (Reif et al., 2008). Hence, the covariate values for all “individuals” are known, and we could proceed with the modeling without assuming a prior distribution for the covariate. However, more typically, when estimating population size, the covariate values are *not* known for individuals not encountered and hence, to estimate these latent covariate values, we must assume a prior distribution for them and estimate the hyperparameters of the latter. To mimic this situation, we discard all body mass data of the $146 - 31 = 115$ species that were not detected at point 610.

```
p610 <- read.table("p610.txt", header = TRUE)
y <- p610[,5:9]                                # Grab counts
y[y > 1] <- 1                                    # Convert to det-nondetections
ever.observed <- apply(y, 1, max)
wt <- p610$bm[ever.observed == 1]                # Body mass
yy <- as.matrix(y[ever.observed == 1,])         # Detection histories
dimnames(yy) <- NULL
```

To determine a distribution suitable to describe the species-specific body mass, we cheat a little and inspect the data for all 146 species. Body mass is expected to be proportional to the cube of length, and length measurements are often approximately normally distributed. Therefore, we take the cubic root of body mass. In addition, we take the log. It is obvious that the lognormal is not a fantastic description of the cubic root of body mass, but here, we assume it is adequate, at least for our illustrative purposes.

```

mlog <- mean(log(p610$bm^(1/3)))
sdlog <- sd(log(p610$bm^(1/3)))
hist(p610$bm^(1/3), xlim=c(0, 30), nclass=25, freq=FALSE,
     col="gray")
lines(density(rlnorm(n=10^6, meanlog=mlog, sdlog=sdlog)),
     col="blue", lwd=3)

```

Since we discarded the data on undetected species, we now need to actively augment the data set and add 150 potential species. This yields a detection data set with $M = 181$ rows. We also need to augment the individual covariate with NAs for individuals 32–181.

```

# Augment both data sets
nz = 150
yaug <- rbind(yy, array(0, dim=c(nz, ncol(yy))))
logwt3 <- c(log(wt^(1/3)), rep(NA, nz))

```

It should in theory be possible to fit the same model as in [Section 6.3](#) with the body mass covariate added. However, trying to squeeze out so many parameter estimates from so little data (31 species) may be asking for too much, and indeed, we failed at getting convergence for this model. Hence, we fit a simpler model with effects of time and the covariate, thus dropping the individual and trap response effects. We center the covariate to facilitate interpretation of the parameters and improve convergence. We also provide part of the prior definition in the BUGS model as data (`prior.sd.upper`), which makes the code more flexible.

```

# Specify model in BUGS language
sink("M_t+X.txt")
cat("
model {

# Priors
omega ~ dunif(0, 1)
for (j in 1:T){
  alpha[j] <- log(mean.p[j] / (1-mean.p[j]))
  mean.p[j] ~ dunif(0, 1)
}
beta ~ dnorm(0, 0.01)
mu.size ~ dnorm(0, 0.01)
tau.size <- 1 / pow(sd.size, 2)
sd.size ~ dunif(0, prior.sd.upper)      # Provide upper bound as data

# Likelihood
for (i in 1:M){      # Loop over individuals
  z[i] ~ dbern(omega)
  size[i] ~ dnorm(mu.size, tau.size)I(-6, 6)
  for (j in 1:T){ # Loop over occasions
    y[i,j] ~ dbern(p.eff[i,j])
    p.eff[i,j] <- z[i] * p[i,j]

```



```

      p[i,j] <- 1 / (1 + exp(-lp[i,j]))
      lp[i,j] <- alpha[j] + beta * size[i]
    } #j
  } #i

# Derived quantities
N <- sum(z[])
}
", fill=TRUE)
sink()

# Bundle data
win.data <- list(y=yaug, size=logwt3 - mean(logwt3, na.rm=TRUE), M=
  nrow(yaug), T=ncol(yaug), prior.sd.upper=3)

# Initial values
inits <- function() list(z=rep(1, nrow(yaug)), beta=runif(1, 0, 1),
  mu.size=rnorm(1, 0, 1))

# Parameters monitored
params <- c("N", "mean.p", "beta", "omega", "mu.size", "sd.size")

# MCMC settings
ni <- 50000
nt <- 4
nb <- 10000
nc <- 3

# Call WinBUGS from R (BRT 19 min)
outX <- bugs(win.data, inits, params, "M_t+X.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=TRUE, bugs.directory=
  bugs.dir, working.directory=getwd())

# Summarize posteriors and plot posterior for N
print(outX, dig=3)

```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
N	41.638	10.384	32.000	36.000	39.000	44.000	68.000	1.099	65
mean.p[1]	0.270	0.075	0.138	0.216	0.264	0.318	0.430	1.003	1100
mean.p[2]	0.320	0.080	0.176	0.263	0.316	0.373	0.488	1.003	1200
mean.p[3]	0.321	0.080	0.175	0.263	0.317	0.374	0.487	1.003	1100
mean.p[4]	0.244	0.072	0.120	0.192	0.239	0.290	0.399	1.003	1200
mean.p[5]	0.345	0.083	0.197	0.287	0.341	0.400	0.518	1.003	1000
beta	-1.313	0.875	-3.143	-1.873	-1.308	-0.725	0.346	1.061	40
omega	0.233	0.064	0.152	0.195	0.222	0.256	0.387	1.049	100
mu.size	0.070	0.111	-0.092	0.002	0.055	0.116	0.342	1.083	63
sd.size	0.366	0.065	0.272	0.323	0.356	0.398	0.520	1.019	230
[...]									

```

hist(outX$sims.list$N, breaks=100, col="gray", main="", xlab=
  "Community size", las=1, xlim=c(30, 100), freq=FALSE)
abline(v=31, col="black", lwd=3)

```

We get a very similar estimate of species richness (N) as under model M_{tbh} , though the uncertainty is now greater (Fig. 6.7b). This might suggest that much of the heterogeneity among species in p may be explained by body size and its correlates. What about the relationship between p and

body mass? The latter varies about 5–10,500 g, and we produce predictions of p in relation to mass (up to 2000 g), averaging over time effects. We find a strong negative effect of body mass on detection probability (Fig. 6.7c).

```
pred.wt <- seq(5, 2000, length.out = 100) # Cov. vals for prediction
pred.wt.st <- log(pred.wt^(1/3)) - mean(logwt3, na.rm = TRUE)
# Transform them in the same way as in the analysis
pred.p <- plogis(log(mean(outX$mean$mean.p) /
  (1 - mean(outX$mean$mean.p))) + outX$mean$beta * pred.wt.st)
# Compute predicted response
plot(pred.wt, pred.p, type = "l", lwd = 3, col = "blue", las = 1,
  frame.plot = FALSE, ylim = c(0, 0.5))
```

The individual covariate model of Royle (2009) is an interesting and flexible model. As usual, the motivation for the introduction of covariates may be to eliminate unexplained heterogeneity or to explore the covariate relationships. Casting the closed-population model as an occupancy model with or without partly unobserved covariate values gives us much flexibility for either. Of course, the distributional assumption for the covariate is very much part of the model. The dependence of the inference upon this assumption should be tested, as did Royle (2009).

6.4.2 Individual Covariate Model for Population Size Estimation

In our second example, we analyze data from a population study of the pen shell (Fig. 6.8), conducted by Iris Hendriks and her colleagues in the Balearic islands in 2007 and 2010 (Hendriks et al., in preparation). The pen shell is a large bivalve living in *Posidonia* meadows in the Mediterranean. Its habitat is increasingly affected by anchors of leisure boats. Hendriks et al. had two teams of divers who each conducted an independent survey of a number of transects and recorded and measured the size (shell width in cm) of each *Pinna* individual encountered. The resulting data consisted of the detection history of each shell (e.g., 1 0, for a shell detected by the first team and missed by the second) along with its width. We expected a positive effect of mussel size on detection probability. We ignored several other potential covariates, such as site and vegetation density, and restricted our analysis to the 143 shells encountered in 2010.

```
# Read in data and look at shell width distribution
pinna <- read.table("pinna.txt", header = TRUE)
y <- cbind(pinna$d1, pinna$d2)
size <- pinna$width
hist(size, col = "gray", nclass = 50, xlim = c(0, 30), freq = FALSE)
lines(density(rnorm(10^6, mean = mean(size), sd = sd(size))),
  col = "blue", lwd = 3)
```



FIGURE 6.8 Pen shell (*Pinna nobilis*), Spain, 2009 (Photograph by I. Hendriks).

We recycle most code from the Czech bird data analysis in the previous section.

```
# Augment both data sets
nz = 150
yaug <- rbind(y, array(0, dim=c(nz, ncol(y))))
size <- c(size, rep(NA, nz))

# Bundle data
win.data <- list(y=yaug, size=size - mean(size, na.rm=TRUE),
  M=nrow(yaug), T=ncol(yaug), prior.sd.upper=5)

# MCMC settings
ni <- 2500
nt <- 2
nb <- 500
nc <- 3

# Call WinBUGS from R (BRT 1 min)
outXX <- bugs(win.data, inits, params, "M_t+X.txt", n.chains=nc,
  n.thin=nt, n.iter=ni, n.burnin=nb, debug=TRUE, bugs.directory=
  bugs.dir, working.directory=getwd())

# Summarize posteriors
print(outXX, dig=2)
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
N	172.04	9.45	156.00	165.00	171.00	178.00	193.00	1.00	450
mean.p[1]	0.67	0.05	0.57	0.64	0.68	0.71	0.77	1.00	1200
mean.p[2]	0.53	0.05	0.43	0.49	0.53	0.56	0.61	1.00	860
beta	0.11	0.04	0.04	0.09	0.11	0.14	0.18	1.09	50
omega	0.59	0.04	0.50	0.56	0.59	0.61	0.68	1.00	570
mu.size	-0.15	0.26	-0.67	-0.33	-0.16	0.02	0.35	1.01	300
sd.size	3.51	0.15	3.22	3.41	3.51	3.61	3.83	1.00	3000

We estimate that 172 instead of 143 pen shells were available for detection along the surveyed transects. This means that 29 (95% CRI 13–50) were missed by both teams of divers. Detection probability was higher for the first teams, and as expected, there was a positive relationship with shell width (Fig. 6.9).

Plot posterior for N and prediction of p

```
par(mfrow=c(1,2), mar=c(4.5, 4, 2, 1))
hist(outXX$sims.list$N, breaks=30, col="gray", main="", xlab=
  "Population size", las=1, xlim=c(143, 220), freq=FALSE)
abline(v=143, col="black", lwd=3)

pred.size <- seq(0, 30, length.out=1000) # Cov. vals for prediction
pred.size.st <- pred.size - mean(size, na.rm=TRUE) # Transform them
pred.p <- plogis(log(mean(outXX$mean$mean.p) /
  (1 - mean(outXX$mean$mean.p))) + outXX$mean$beta * pred.size.st)
# Compute predicted detection probab.
plot(pred.size, pred.p, type="l", lwd=3, col="blue", las=1,
  frame.plot=FALSE, ylim=c(0, 1), xlab="Shell width (cm)", ylab=
  "Predicted detection probability")
```

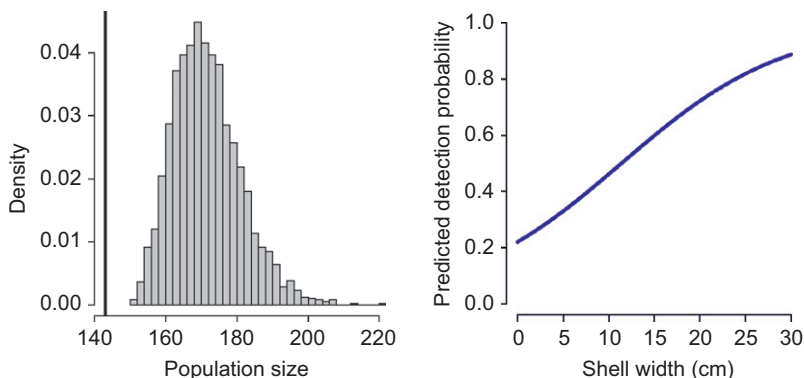


FIGURE 6.9 Analysis of population size (N) of pen shells. Left: Posterior distribution of N (black line: observed number of individuals); right: predicted detection probability in relation to size.

6.5 SUMMARY AND OUTLOOK

The population is a central concept in ecology and its size N perhaps its key descriptor. Because of detection error, N is usually not directly observable. Capture–recapture methods obtain information about detection probability p from the repeated encounters of uniquely identifiable individuals and from this derive an estimate of N . In practice, estimating N means estimating p and modeling the key patterns in p . The same concepts and methods can be used for vastly different kinds of “populations”, including populations of species, that is, communities, where size is equivalent to species richness. In this chapter, we have encountered a catalog of effects (Null, time, behavior, individual heterogeneity) that may be present in p when estimating N , including the effects of individual covariates. These concepts are very general, and we will find them in different situations for parameters other than detection probability later in this book. We have also introduced data augmentation (DA) and fitted all models in this chapter using this ingenious but simple technique. DA is an important concept and helps to unify a vast number of capture–recapture type models, which were hitherto regarded as technically distinct procedures. See Royle and Dorazio (2008) to get a flavor of the power of DA.

Obviously, we have only given a very selective view of the large field of population size estimation, and interested readers will have to refer to the many standard references now available. We also briefly note that we do not deal with one important class of models that also achieve a clean accounting of the ecological and the observation processes in estimating density or abundance: distance sampling (Buckland et al., 2001; Williams et al., 2002; Buckland et al., 2004; Royle and Dorazio, 2008). These models are the focus of an active branch of ecological statistics and have much relevance for the analysis of populations. Distance sampling can be implemented in WinBUGS; see Royle and Dorazio (2008). OpenBUGS contains a distance sampling example (Examples > Ecology examples > Impala).

We have assumed that the area with which a population is associated may be delimited unambiguously. This is often not the case and to solve this problem, spatial capture–recapture methods are required, which represent a merging of the capture–recapture class of models with those of the distance sampling-type (Efford, 2004; Borchers and Efford, 2008; Royle and Young, 2008; Gardner et al., 2009; Efford et al., 2009a, 2009b; Royle et al., 2009a, 2009b). These models are very powerful and flexible. One of their advantages is that they can estimate detection probability, and therefore, can estimate abundance from the spatial pattern of multiple detections of individuals alone (Efford et al., 2009b); no temporal replicate surveys are required. This is a huge design advantage! It seems likely that

in the near future we will see much development along this branch of the population assessment tree in ecological statistics. Again, most spatial capture–recapture models can be implemented in WinBUGS in a transparent way. Likelihood inference can be obtained with user-friendly packages such as program Density (www.otago.ac.nz/density/) or the recent R package secr developed by Murray Efford.

All models in the current chapter have assumed closed populations, but we may also want to estimate N in open populations. The Jolly–Seber model (Chapter 10) models population dynamics, that is, survival and recruitment rates and population size, whereas the N -mixture and site-occupancy models (Chapters 12 and 13) may be used to model N in collections of open populations.

6.6 EXERCISES

1. Rewrite the model M_h in Section 6.2.4 for encounter history instead of capture frequency data and fit it. (The response will be Bernoulli instead of Binomial.)
2. Try to fit the model with permanent trap response to the bird survey data. Imagine a biological situation that might represent this model, on the part of the observer? On the part of the animal?
3. Check out the behavior of estimators in small sample situations, example, the heterogeneity model with 20 individuals and heterogeneity. Does this work?
4. Generate data with individual heterogeneity in p and fit model M_0 . See how well N is estimated.
5. Find out whether a model with trap response and time effects is estimable with $T = 2$.
6. And what about pure model M_b with $T = 2$?
7. In M_t , adapt both the data generation and the model fitting code to random instead of fixed time effects.
8. Check the effects of assumption violations. Fit a model to a data set that was not generated under the same model. For instance, generate data under model M_t and analyze the resulting data set under M_0 to see what happens to your estimates of N and p when you ignore time variation in p . Do similar things to other pairs of models.
9. Use the Czech point count data and estimate species richness, where detection is a function of body mass, similar as in Section 6.4.1. But this time, include the body mass of all unobserved species. Hint: you then no longer have to give a prior for body mass. Does the estimate of population size and of detection probability become more precise?