CHAPTER

13

Estimation of Occupancy and Species Distributions from Detection/Nondetection Data in Metapopulation Designs Using Site-Occupancy Models

OUTLINE

Bayesian Population Analysis using WinBUGS
DOI: 10.1016/B978-0-12-387020-9.00013-4

13.1 INTRODUCTION

In much of ecology, abundance (N) is the most interesting state variable when analyzing a population. Abundance is usually estimated from capture–recapture data or counts using the methods in Chapters 6, 10 or 12, or else strong assumptions are made about the count-abundance relationship. However, sometimes we do not have counts but only less information-rich data of the detection/nondetection kind (also misleadingly called presence/absence data). These are binary data indicating whether a species is detected (1) or not (0) at a site. We may then want to characterize one or several sites using occupancy: the probability that a site is occupied, that is, that local abundance is greater than zero. Often, occupancy is not of direct interest and merely a proxy for abundance, in which one is really interested. Indeed, it is often hard to think about occupancy separately from the abundance at the occupied sites.

However, there are also important fields in ecology that do focus on occupancy rather than abundance. Outstanding examples include metapopulation ecology (Hanski, 1994, 1998), niche and species distribution (Guisan and Thuiller, 2005), and disease modeling (Thompson, 2007; McClintock et al., 2010). In addition, there is a sense in which, at a small spatial scale, occupancy and abundance coincide; when a site is chosen so small that at most one individual or pair can occupy it. The spotted owl data set in MacKenzie et al. (2003) and our Section 13.5.1 provide examples for this. A similar example is given by Bled et al. (2011a), who studied the habitat selection of kittiwakes in breeding cliffs. Here, a potential nest site is a straightforward site definition, and it can be occupied by two birds at most.

This chapter deals with a class of hierarchical models known as "site-occupancy models". In the statistical literature, these models are also called zero-inflated binomial models. In the context of distribution modeling in ecology, they have been introduced independently by MacKenzie et al. (2003) and Tyre et al. (2003), though they have important roots in earlier approaches as summarized in MacKenzie et al. (2006). "Site-occupancy model" is a fairly uninformative name for this extremely flexible modeling framework. We believe that this has helped to hide its usefulness for inference about any kind of occurrence ("presence/absence") data at discrete sites. Essentially, site-occupancy models are hierarchical logistic regression models that jointly model the probability of occupancy and detection in animals or plants.

As usual, we believe that a hierarchical view of occurrence data is important to properly separate the ecological component and the observation component that combine to produce the observed data. However, this has not been a widely-held opinion in ecology so far. For instance, in classical species distribution modeling (e.g., Guisan and Thuiller, 2005), it is typically ignored what is actually being modeled: it is *not* the distribution of a species. Rather, it is the *apparent species distribution* (unless detection probability is estimated). The apparent distribution is a function of both the true species distribution and of the detection probability of the species (Kéry and Schmidt, 2008; Kéry et al., 2010a; Kéry, 2011b).

There are three concerns when apparent instead of true distribution is modeled:

1. The extent of species distributions will be underestimated when $p < 1$,
2. Estimates of covariate relationships will be biased towards zero when $p < 1$,
3. Factors that affect the difficulty with which a species is found may end up in predictive models of species occurrence or may mask factors that do affect species occurrence.

The first is intuitively clear: if a species is not found at all sites where it occurs, the perceived range will be smaller than the actual range. However, the second is not so intuitive, especially perhaps, because it seems to be different from the modeling of abundance when detection probability is ignored. Yet, this effect has been demonstrated very clearly by Tyre et al. (2003) and in the next section, we conduct a little simulation to illustrate it. Finally, as an example of the third effect, assume that a species is more detectable in habitat A than in habitat B, for instance, because habitat A is more open and B is more wooded. In this case, open habitat may be identified as a factor that positively affects the occupancy probability/distribution of the species. For an example of the converse, see Section 13.3.2.

As always, to account for imperfect detection, extra data about the observation process are required. This means temporally replicated "presence/absence" observations, where the pattern of detection/nondetection at a site contains the information about the observation process. We note that spatial replication at a small scale is informative about detection probability as well (Nichols et al., 1998a, 1998b; Kendall and White, 2009; Hines et al., 2010), but we focus on temporal replication here. Site-occupancy models require data collected in a metapopulation design (Royle, 2004c; Kéry and Royle, 2010), where (temporally or small-scale spatially) replicated detection/nondetection observations are available for a number of spatial replicates (for instance, > 20). As in Chapter 12, analyzing such a data set does not mean to imply that it represents a metapopulation in the ecological sense of the term.

In the simplest case, we consider detection/nondetection observation $y_{i,j}$ at site i during survey j: $y_{i,j}$ takes on a value of 1 when a species is detected at site i on survey j and value of 0 when it is *not* detected.

It is useful to consider the genesis of all species distribution or metapopulation data as a combination of two processes: one (ecological) process determines whether a site is occupied or not and the other (observation) process determines whether the species is found or not, given that a site is occupied. Correspondingly, in a site-occupancy model, we formally distinguish between a first submodel for the partly observed true state (occurrence, the result of the ecological process) and second submodel for the actual observations. The actual observations result from both the particular realization of the ecological process and of the observation process.

$z_i \sim \text{Bernoulli}(\psi)$	1. Ecological process yields true state
$y_{i,j} \mid z_i \sim \text{Bernoulli}(z_i p)$	2. Observation process yields observations

We naturally model true occurrence z_i ($z_i = 1$, if site i is occupied; $z_i = 0$ if site i is not occupied) as a Bernoulli random variable governed by the parameter ψ (occupancy probability); ψ is the parameter that distribution modelers would wish they were modeling but only do so when detection is perfect or detection probability can be estimated. (Note that we denote probability of occupancy by ψ and the latent occurrence state of a site as z.) However, z_i is not what we usually get to see; instead, our actual observations, $y_{i,j}$, detection or not at site i during survey j (or "presence/absence" datum $y_{i,j}$), are another Bernoulli random variable with a success rate that is the product of the actual occurrence at site i, z_i, and detection probability p at site i during survey j. At a site where a study species does not occur, z equals 0, and y must be 0, unless there are false-positive errors. Conversely, at an occupied site, we have $z = 1$, and the species is detected with probability p. That is, in the site-occupancy model, detection probability is expressed *conditional on actual occurrence*, and the two parameters ψ and p are separately estimable if replicate visits are available. We could call this model a Bernoulli-Bernoulli mixture model. Moreover, recognizing that the modeling of the latent occurrence (z) in the first level of the hierarchy accommodates additional zeroes in the data set (beyond those coming from the Bernoulli observation process), we see that it is also zero-inflated binomial (ZIB) model.

We have claimed that the term "presence/absence" for data $y_{i,j}$ is misleading. The preceding equations clarify why this is so: $y_{i,j}$ is a function of two processes, and only one of them has to do with occurrence and the other one is a nuisance process owing to the imperfect nature of the observation process. The true presence/absence data are the z_i, and they are only imperfectly observed and therefore latent: $z = 1$ can be observed as $y = 0$ or as $y = 1$. Site-occupancy models allow one to make a formal distinction between the two latter cases.

Two important assumptions of the model are closure and lack of false-positive errors. Closure in the context of the site-occupancy model means that over the duration of surveys, the occurrence state of a site must not change. Each site is either occupied or it is not, but there is no extinction or colonization. This sounds like a rather strong assumption; however, it is not always that problematic. Lack of closure is akin to temporary emigration (see Chapter 9), so if temporary emigration is random, it will be confounded with detection probability. This means that temporary (but not permanent) absence of a species from a site will be one component of imperfect detection. Consequently, the estimate of the occupancy parameter will describe the proportion of sites ever occupied or used during the study period, rather than of sites that are permanently occupied, as it would in the absence of temporary emigration. If there is colonization/extinction, for instance when surveys are spread over several years, we could simply model occupancy separately for each period of closure, as we did for the open-population binomial mixture model in Section 12.3. Alternatively, we can use the dynamic occupancy model described in Section 13.5, which expresses changes in occurrence over multiple "seasons" as a function of colonization and extinction.

Absence of false positives means that no other species must be mistakenly identified as our focal species, or more generally, we must be sure that a 1 really means that our focal species was present. False positives can seriously bias occupancy estimates (Royle and Link, 2006); hence, they should be avoided for instance by good training of field personnel or by discarding doubtful records. If we discard doubtful sightings that in reality refer to our focal species, we simply lower detection probability but do not incur biased estimators. However, our models are able to deal with imperfect detection very well. When different kinds of occupancy data are available and false positives can be excluded for at least one of them, multistate occupancy models (see Section 13.6) can be used to account for both false negatives and for false positives (Miller et al., 2011).

One way to look at site-occupancy models is as a hierarchical, coupled logistic regression. One logistic regression describes true occurrence, and the other describes detection, given that the species occurs. Remember that conventional methods for distribution modeling (GLM, GAM, boosted regression trees: Elith et al., 2008; Maxent: Phillips and Dudik, 2008) would pool the temporal replicates j and model the maximal observation, that is, site i will get a value of 1 if the species was ever detected there. Those approaches discard the information available about the observation process and thus in principle cannot model true, but only apparent species distributions (Kéry et al., 2010a). In contrast, site-occupancy models exploit all the available information about both ecological and observation process contained in detection/nondetection data.

The two Bernoulli distributions above describe the simplest possible site-occupancy model, where both occupancy (ψ) and detection probability (p)

are constant (see Section 13.3.1). This simple model can be extended in many ways. Most importantly, we need to be able to model the effects of measured covariates on one or both parameter(s). Both the ecological and the observation processes represent a logistic regression (with an intercept only so far), so it is natural to include covariate effects via a logit link function. Hence, we can add statements of the following kind to the model description

$$\text{logit}(\psi_i) = \alpha + \beta * x_i.$$

Here, x_i is the value of some occurrence-relevant covariate measured at site i, and α and β are the intercept and slope parameters of this logit-linear regression. We can do the same for the observation model, where we distinguish between "site covariates" and "sampling covariates". Site covariates vary among sites only and are constant across repeated surveys to a site, that is, they will be indexed by i only. In contrast, survey covariates vary by site *and* by survey; hence, they will be indexed by i and j. This is a minor distinction but in practice, the modeling of sampling covariates requires a little more book-keeping effort. Explicitly couching site-occupancy models within the GLM framework makes it clear that other GLM extensions might be applied, too. For instance, overdispersion in detection probability could be modeled by the introduction of random site effects (Royle, 2006). Of course, we could model the effects of many explanatory variables, of polynomial terms, or of splines (Gimenez et al., 2006a, b; Collier et al., 2011).

In Section 13.2, we conduct a simulation to understand what happens to the estimates of regression coefficients in conventional species distribution models when detection probability is not perfect. In Section 13.3, we analyze simulated data sets and in Section 13.4 a real data set using single-season site-occupancy models. In Section 13.5, we extend the model to multiple "seasons" and thus arrive at an extended metapopulation model. In Section 13.6, we extend the single-season model to multiple states of occurrence, which in our example are owl territories occupied with or without reproduction.

We emphasize that we will not conduct any goodness-of-fit assessments based on posterior predictive checks in this chapter. The reason for this is that with a binary response, the deviance or other discrepancy measures based directly on the response are uninformative about the fit of a model (McCullagh and Nelder, 1989). Kéry (2010) erroneously showed such posterior predictive checks for site-occupancy models. These checks are meaningless because regardless of the model structure, they will always and thus sometimes spuriously indicate a fitting model. To do a goodness-of-fit test, the binary responses have to be aggregated.

13.2 WHAT HAPPENS WHEN $p < 1$ AND CONSTANT AND p IS NOT ACCOUNTED FOR IN A SPECIES DISTRIBUTION MODEL?

We use simulation to understand what happens when there is a constant degree of imperfect detection and this is not accounted for in an analysis. We simulate 100,000 data sets from 250 sites, with a constant $p < 1$ (here, $p = 0.60$), and analyze them with a conventional species distribution model (here, a nonhierarchical logistic regression). We have a single explanatory variable (think of it as a habitat or environmental covariate) that links the habitat to occurrence probability on the logit-linear scale with intercept −3 and slope 1. (You may want to change `nreps` in the code to 1000.)

```
nreps <- 10^5                                    # No. replicates
estimates <- array(NA, dim = c(nreps, 2))        # Array to contain the
                                                   estimates
R <- 250                                         # No. sites

for (i in 1:nreps) {
    cat(i, "\n"); flush.console()
    x <- runif(R, 0, 10) # choose covariate values
    state<-rbinom(n = R, size = 1, prob = plogis(-3 + 1 * x)) # Occ. state
    obs <- rbinom(n = R, size = 1, prob = 0.6) * state       # Observations
    fm <- glm(obs~x, family = binomial)
    estimates[i,] <- fm$coef
    }

par(mfrow = c(3, 1))
hist(estimates[,1], col = "gray", nclass = 50, main = "",
   xlab = "Intercept estimates", las = 1, ylab = "", freq = FALSE)
abline(v = -3, col = "red", lwd = 3)            # Truth
hist(estimates[,2], col = "gray", nclass = 50, main = "", xlab = "Slope
   estimates", xlim = c(0,1), las = 1, ylab = "", freq = FALSE)
abline(v = 1, col = "red", lwd = 3)             # Truth

plot(1:10, plogis(estimates[1,1] + estimates[1,2] * (1:10)), col =
   "gray", lwd = 1, ylab = "Occupancy probability", xlab = "Covariate
   value", type = "l", ylim = c(0, 1), frame.plot = FALSE, las = 1)
samp <- sample(1:nreps, 1000)
for (i in samp){
    lines(1:10, plogis(estimates[i,1] + estimates[i,2] * (1:10)),
       col = "gray", lwd = 1, type = "l")
    }
lines(1:10, plogis(-3 + 1 * (1:10)), col = "red", lwd = 3, type = "l")
```

When failing to account for a constant nondetection error, slope estimates of a covariate are biased towards zero (Fig. 13.1, middle panel). The intercept (Fig. 13.1, top panel) is not necessarily estimated too low; rather, here, it is overestimated. However, the combined effect is such

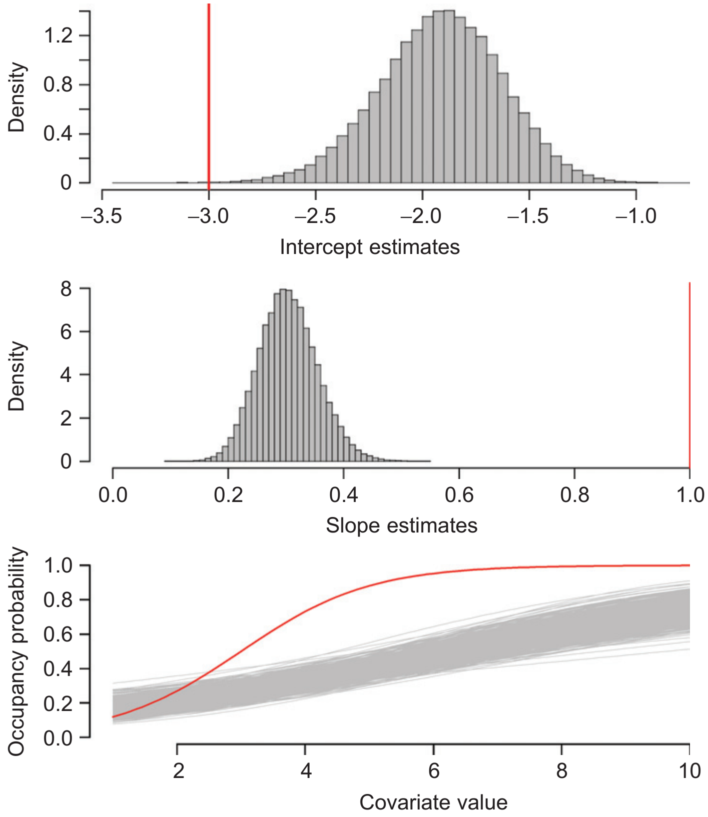

FIGURE 13.1 Effect of imperfect detection on a conventional species distribution model: slope estimates become biased low with imperfect detection even if detection probability is constant (here, 0.60). In the bottom panel, the red lines show the truth and the gray lines show a random sample of 1000 estimated regression lines: the extent of the distribution is always underestimated. See also Tyre et al. (2003).

that the total extent of a distribution is underestimated. The latter is represented by the area under the red curve in the bottom panel. The area under the gray curves (the estimated distribution) is always less than the area under the red curve (true distribution).

13.3 GENERATION AND ANALYSIS OF SIMULATED DATA FOR SINGLE-SEASON OCCUPANCY

13.3.1 The Simplest Possible Site-Occupancy Model

To fully grasp how the site-occupancy model "works", we first look at the simplest possible case: both the ecological and the observation process are described by an intercept only. To generate detection/nondetection data $y_{i,j}$ under this Null model for $R = 200$ spatial replicates (sites) and $T = 3$ temporal replicates, we simply do this.

```
# Select sample sizes (spatial and temporal replication)
R <- 200
T <- 3

# Determine process parameters
psi <- 0.8      # Occupancy probability
p <- 0.5        # Detection probability

# Create structure to contain counts
y <- matrix(NA, nrow = R, ncol = T)

# Ecological process: Sample true occurrence (z, yes/no) from a
  Bernoulli (occurrence probability = psi)
z <- rbinom(n = R, size = 1, prob = psi) # Latent occurrence state

# Observation process: Sample detection/nondetection observations
  from a Bernoulli (with p) if z=1
for (j in 1:T){
    y[,j] <- rbinom(n = R, size = 1, prob = z * p)
    }

# Look at truth and at our imperfect observations
sum(z)                 # Realized occupancy among 200 surveyed sites
[1] 169
sum(apply(y, 1, max)) # Observed occupancy
[1] 151
```

Note that in the simulation of the observation process, we have multiplied the Bernoulli draw with z. This means that the result will be zero whenever $z = 0$, that is, whenever the species does not occur. Next, we analyze this data set.

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

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

# Likelihood
# Ecological model for true occurrence
for (i in 1:R) {
    z[i] ~ dbern(psi)
    p.eff[i] <- z[i] * p

    # Observation model for replicated detection/nondetection
      observations
    for (j in 1:T) {
        y[i,j] ~ dbern(p.eff[i])
      } #j
   } #i

# Derived quantities
occ.fs <- sum(z[])       # Number of occupied sites among the 200
}
",fill = TRUE)
sink()

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

# Initial values
zst <- apply(y, 1, max)    # Observed occurrence as starting values for z
inits <- function() list(z = zst)

# Parameters monitored
params <- c("psi", "p", "occ.fs")

# MCMC settings
ni <- 1200
nt <- 2
nb <- 200
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 = 2)
[...]
           mean    sd   2.5%    25%    50%    75%  97.5% Rhat n.eff
psi        0.89  0.04   0.80   0.86   0.89   0.92   0.97 1.01   340
p          0.47  0.03   0.41   0.45   0.47   0.49   0.53 1.00   870
occ.fs   178.29  7.59 165.00 173.00 178.00 183.00 195.00 1.01   510
deviance 739.37 28.50 686.50 719.00 738.20 758.05 798.75 1.01   470
[...]
```

This looks good. You will note quite a bit of sampling variability in this system. The estimates may be fairly different among repeated generations of the data set or among the replicate data sets of different people. This basic model is a good starting point for running simulation exercises to find out about how good inferences can be in marginal data situations; see exercises and Guillera-Arroita et al. (2010).

13.3.2 Site-Occupancy Models with Covariates

Next, we look into the case where covariates affect the ecological and the observation process. We model covariate effects on a parameter θ through the canonical GLM link function, the logit = $\log(\theta/(1-\theta))$. As in the previous chapter, we will look at a worst-case scenario for a species distribution model, where opposing effects of a single covariate on the two processes generating the observed data effectively cancel each other out in the observations. The result will be that in a conventional species distribution model, the effect of this covariate on the species distribution will not be identified correctly. We next define a function that creates species distribution data (detection/nondetection data) for us.

```
# Define function for generating species distribution data
data.fn <- function(R = 200, T = 3, xmin = -1, xmax = 1, alpha.psi = -1,
  beta.psi = 3, alpha.p = 1, beta.p = -3) {

   y <- array(dim = c(R, T)) # Array for counts

   # Ecological process
   # Covariate values
   X <- sort(runif(n = R, min = xmin, max = xmax))

   # Relationship expected occurrence - covariate
   psi <- plogis(alpha.psi + beta.psi * X)     # Apply inverse logit

   # Add Bernoulli noise: draw occurrence indicator z from
     Bernoulli(psi)
   z <- rbinom(n = R, size = 1, prob = psi)
   occ.fs <- sum(z)     # Finite-sample occupancy (see
                          Royle and Kéry, 2007)

   # Observation process
   # Relationship detection prob - covariate
   p <- plogis(alpha.p + beta.p * X)

   # Make a 'census'
   p.eff <- z * p
   for (i in 1:T){
      y[,i] <- rbinom(n = R, size = 1, prob = p.eff)
      }

   # Naïve regression
   naive.pred <- plogis(predict(glm(apply(y, 1, max) ~ X + I(X^2),
     family = binomial)))

   # Plot features of the simulated system
   par(mfrow = c(2, 2))
   plot(X, psi, main = "Expected occurrence", xlab = "Covariate",
     ylab = "Occupancy probability", las = 1, type = "l", col = "red",
     lwd = 3, frame.plot = FALSE)
   plot(X, z, main = "Realised (true) occurrence", xlab = "Covariate",
     ylab = "Occurrence", las = 1, frame.plot = FALSE, col = "red",)
   plot(X, p, ylim = c(0,1), main = "Detection probability",
     xlab = "Covariate", ylab = "p", type = "l", lwd = 3, col = "red",
     las = 1, frame.plot = FALSE)
   plot(X, naive.pred, main = "Detection/nondetection observations \n
     and conventional SDM", xlab = "Covariate", ylab = "Apparent
     occupancy", ylim = c(min(y), max(y)), type = "l", lwd = 3, lty = 2,
     col = "blue", las = 1, frame.plot = FALSE)
   points(rep(X, T), y)
```

```
    # Return stuff
    return(list(R = R, T = T, X = X, alpha.psi = alpha.psi, beta.psi =
      beta.psi, alpha.p = alpha.p , beta.p = beta.p, psi = psi, z = z,
      occ.fs = occ.fs, p = p, y = y))
    }
```

We obtain one realization from the stochastic system just defined and conduct a conventional species distribution model (Fig. 13.2):

```
sodata <- data.fn()
str(sodata)                  # Look at data

summary(glm(apply(y, 1, max) ~ X + I(X^2), family = binomial,
   data = sodata))

Call:
glm(formula = apply(y, 1, max) ~ X + I(X^2), family = binomial,
    data = sodata)

Deviance Residuals:
        Min          1Q      Median          3Q         Max
  -1.10984   -0.83363   -0.28985   -0.04219     2.45653

Coefficients:
                 Estimate  Std. Error   z value    Pr(>|z|)
(Intercept)      -1.0439      0.2624    -3.978    6.95e-05 ***
X                 3.3989      0.8348     4.072    4.67e-05 ***
I(X^2)           -3.2680      1.1757    -2.780    0.00544 **
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 213.27 on 199 degrees of freedom
Residual deviance: 170.95 on 197 degrees of freedom
AIC: 176.95

Number of Fisher Scoring iterations: 6
```

Hence, in this simulated data set and with a conventional species distribution model, we identify an optimum value of the covariate for the occupancy probability of the study species (see blue curve in bottom right panel of Fig. 13.2). Let us see what a site-occupancy model can do.

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

# Priors
alpha.occ ~ dunif(-10, 10)
beta.occ ~ dunif(-10, 10)
alpha.p ~ dunif(-10, 10)
beta.p ~ dunif(-10, 10)

# Likelihood
for (i in 1:R) {
```

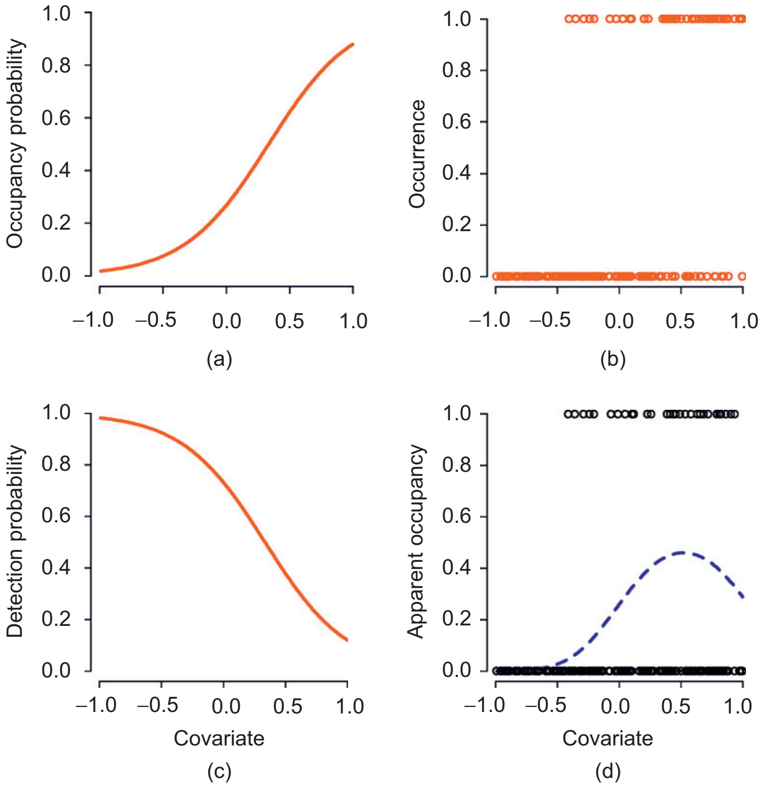

FIGURE 13.2 Features of the simulated data set, and truth behind it, and inference about the system based on a conventional species distribution model (blue line in bottom right panel). The truth is shown in red and observed data in black. (a) Occupancy probability, (b) realized (true) occurrence, (c) detection probability, and (d) detection/nondetection ("presence/absence") observations and estimated occupancy probability under a conventional species distribution model.

```
        # True state model for the partially observed true state
        z[i] ~ dbern(psi[i])          # True occupancy z at site i
        logit(psi[i]) <- alpha.occ + beta.occ * X[i]

        for (j in 1:T) {

            # Observation model for the actual observations
            y[i,j] ~ dbern(p.eff[i,j]) # Detection-nondetection at i and j
            p.eff[i,j] <- z[i] * p[i,j]
            logit(p[i,j]) <- alpha.p + beta.p * X[i]
            } #j
        } #i

    # Derived quantities
    occ.fs <- sum(z[])# Number of occupied sites among those studied
    }
    ",fill = TRUE)
    sink()
```

```
# Bundle data
win.data <- list(y = sodata$y, X = sodata$X, R = nrow(sodata$y),
   T = ncol(sodata$y))

# Initial values
zst <- apply(sodata$y, 1, max) # Good inits for latent states essential
inits <- function(){list(z = zst, alpha.occ = runif(1, -3, 3),
   beta.occ = runif(1, -3, 3), alpha.p = runif(1, -3, 3), beta.p = runif
   (1, -3, 3))}

# Parameters monitored
params <- c("alpha.occ", "beta.occ", "alpha.p", "beta.p", "occ.fs")

# MCMC settings
ni <- 10000
nt <- 8
nb <- 2000
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())
```

We compare the known truth in the data-generating mechanism with our estimates of truth under the site-occupancy species distribution model. We find that the model does a decent job at recovering the parameters for the habitat relationships of the probability of occupancy (`alpha.occ` and `beta.occ`) and of detection (`alpha.p` and `beta.p`), but that the estimates are much more precise for the relationship with detection. This makes sense because there is more data ($n = 600$ instead of $n = 200$) from which to estimate those regression parameters. A total of 59 sites were occupied in our simulated data set, and at 45 of those, the study species was discovered. Our model estimated 67 occurrences (95% CRI 57–78). This number, finite-sample occurrence, is not a function of population occupancy probability, but of the latent occurrence states z, which we can easily estimate in an MCMC-based analysis (Royle and Kéry, 2007).

```
TRUTH <- c(sodata$alpha.psi, sodata$beta.psi, sodata$alpha.p, sodata
   $beta.p, sum(sodata$z))
print(cbind(TRUTH, out$summary[1:5, c(1,2,3,7)]), dig = 3)
            TRUTH    mean     sd   2.5%   97.5%
alpha.occ      -1  -1.269  0.274  -1.81  -0.738
beta.occ        3   4.084  0.854   2.58   5.939
alpha.p         1   0.925  0.330   0.28   1.584
beta.p         -3  -2.942  0.546  -4.00  -1.865
occ.fs         59  67.335  5.291  57.00  78.000

sum(apply(sodata$y, 1, sum) > 0)# Apparent number of occupied sites
[1] 45
```

We graphically compare the conclusions from the two species distribution models (Fig. 13.3). We see again that the conventional approach,

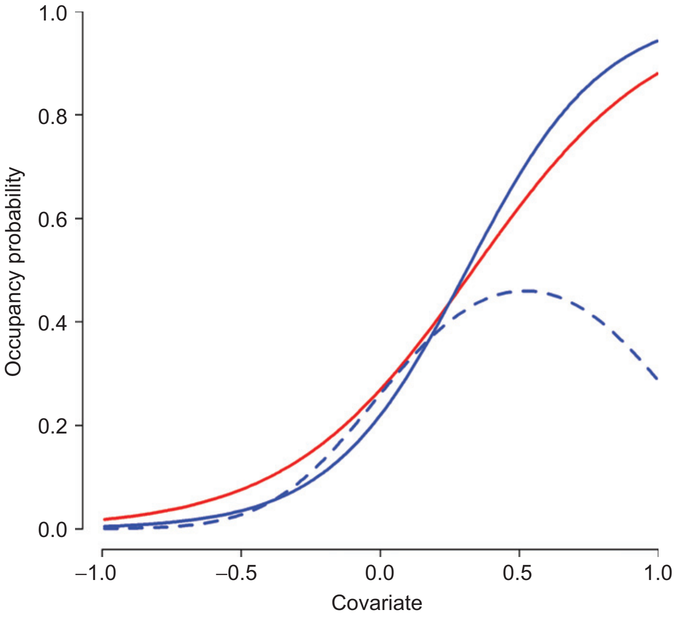

FIGURE 13.3 Comparison of true and estimated relationship between occupancy probability and an environmental covariate under a site-occupancy model (solid blue) and under the conventional approach that ignores detection probability (dashed blue). Truth is shown in red.

which ignores the effects of the observation process in the generation of detection/nondetection data, models apparent rather than true species distributions only (Kéry, 2011b).

```
naive.pred <- plogis(predict(glm(apply(sodata$y, 1, max) ~ X + I(X^2),
   family = binomial, data = sodata)))
lin.pred2 <- out$mean$alpha.occ + out$mean$beta.occ * sodata$X

plot(sodata$X, sodata$psi, ylim = c(0, 1), main = "", ylab = "Occupancy
   probability", xlab = "Covariate", type = "l", lwd = 3, col = "red",
   las = 1, frame.plot = FALSE)
lines(sodata$X, naive.pred, ylim = c(0 ,1), type = "l", lty = 2, lwd = 3,
   col = "blue")
lines(sodata$X, plogis(lin.pred2), ylim = c(0, 1), type = "l", lty = 1,
   lwd = 2, col = "blue")
```

13.4 ANALYSIS OF REAL DATA SET: SINGLE-SEASON OCCUPANCY MODEL

We will next analyze a small, but typical real-world occurrence data set: surveys to breeding sites of the endangered beetle *Rosalia alpina* (Fig. 13.4; see also the cover of Kéry, 2010) during a single flight period (July–August

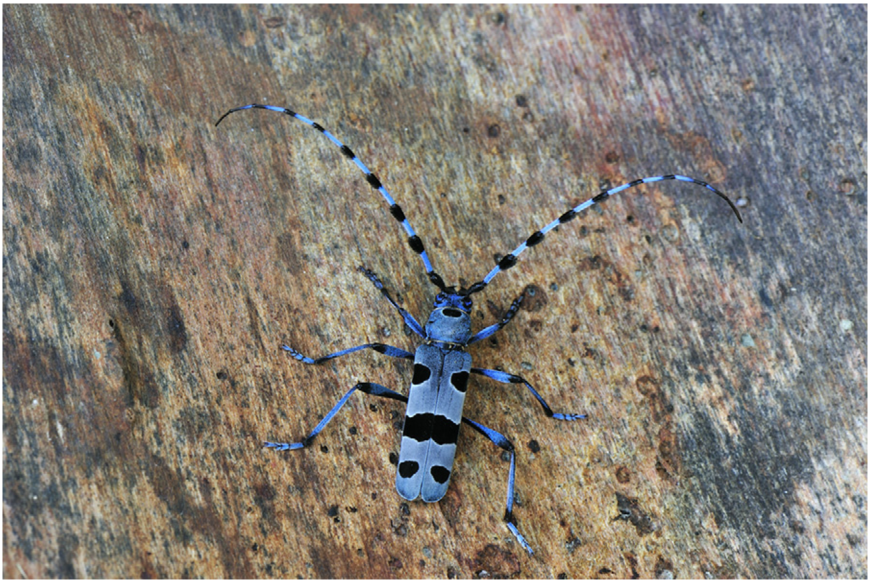

FIGURE 13.4 The remarkable "blue bug", the cerambycid beetle *Rosalia alpina*, Switzerland, 2009 (Photograph by T. Marent).

2009). In Switzerland, this striking blue bug lays its eggs into the wood of dead beech trees *Fagus sylvatica*, preferentially in tall and old logs, but unfortunately also in piles of firewood stocked in the forest only temporarily. Larvae develop over 3–4 years; hence, eggs laid in firewood are normally doomed. Nevertheless, checking firewood piles in forests is an efficient search strategy for this rare and elusive beetle. In 2009, one of us (MK) surveyed one of the few Swiss areas where the species is known to occur, the hills around Movelier in the Swiss Jura mountains.

The complete data set ("bluebug.txt") contains replicated counts at a total of 27 sites (woodpiles) in the Movelier region in 2009. There were up to six replicate counts at each woodpile; the count result of which is called `detX`. Woodpiles were either at the forest edge or more in the interior of a forest (covariate `forest_edge`), and individual visits took place at varying dates (covariate `dateX`) and times of day (hours in the afternoon, covariates `hX`).

A summary of these data is shown in Table 13.1. We see that *Rosalia* was detected at 10 of 27 woodpiles and from 1 to 5 times. Clearly, detection probability at an occupied woodpile is not perfect; for instance, the woodpile in row 10 was surveyed six times and *Rosalia* was seen only once. It is natural to wonder whether other woodpiles might have been occupied but *Rosalia* was simply missed. Another question might be to ask how many times a woodpile might have to be checked in order to detect *Rosalia* at least once when it occurs. And finally, we may wonder

TABLE 13.1 A Summary of the Blue Bug Data Set (`bluebug.txt`) That Keeps Track Only of Detections and Nondetections.

0	1	1	1	1	1	5
1	1	1	1	1	–	5
1	0	1	0	0	1	3
1	0	0	0	1	1	3
1	1	–	–	–	–	2
1	–	–	–	–	–	1
0	0	0	0	1	0	1
1	–	–	–	–	–	1
1	–	–	–	–	–	1
1	0	0	0	0	0	1
0	0	0	0	0	–	0
0	0	0	0	0	–	0
0	0	0	0	0	–	0
0	–	–	–	–	–	0
0	–	–	–	–	–	0
0	–	–	–	–	–	0
0	–	–	–	–	–	0
0	0	–	–	–	–	0
0	0	–	–	–	–	0
0	–	–	–	–	–	0
0	0	–	–	–	–	0
0	0	–	–	–	–	0
0	0	–	–	–	–	0
0	0	0	–	–	–	0
0	–	–	–	–	–	0
0	–	–	–	–	–	0
0	–	–	–	–	–	0

Note: Rows denote woodpiles and columns, except for the right-most column, denote survey occasions. The total number of surveys with detections is shown in the right-most column. Surveys with *Rosalia* detections are shown in buff color, those without *Rosalia* detections in yellow, and missing values shown as dashes. For pure convenience, sites have been ordered by decreasing number of surveys with detections.

whether the location of a woodpile, at the forest edge or in the interior, may affect the probability of it being occupied, and similarly, whether there were relationships between detection probability and the date and time of day, respectively, at which a survey took place, or whether *Rosalia* was detected before (behavioral effect, see Section 6.2.3.). We will answer these questions with a site-occupancy species distribution model now.

```
# Read in the data
data <- read.table("bluebug.txt", header = TRUE)

# Collect the data into suitable structures
y <- as.matrix(data[,4:9])       # as.matrix essential for WinBUGS
y[y>1] <- 1                      # Reduce counts to 0/1
edge <- data$forest_edge
dates <- as.matrix(data[,10:15])
hours <- as.matrix(data[,16:21])

# Standardize covariates
mean.date <- mean(dates, na.rm = TRUE)
sd.date <- sd(dates[!is.na(dates)])
DATES <- (dates-mean.date)/sd.date       # Standardise date
DATES[is.na(DATES)] <- 0                 # Impute zeroes (means)

mean.hour <- mean(hours, na.rm = TRUE)
sd.hour <- sd(hours[!is.na(hours)])
HOURS <- (hours-mean.hour)/sd.hour       # Standardise hour
HOURS[is.na(HOURS)] <- 0                 # Impute zeroes (means)
```

In the BUGS code below, we "stabilize" the logit to avoid numerical under- or overflow by truncating values more extreme than (−999, 999) on the logit scale. This should hardly affect the inference because this restricts the value of the linear predictor to the range (plogis(−999), plogis(999)).

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

# Priors
alpha.psi ~ dnorm(0, 0.01)
beta.psi ~ dnorm(0, 0.01)
alpha.p ~ dnorm(0, 0.01)
beta1.p ~ dnorm(0, 0.01)
beta2.p ~ dnorm(0, 0.01)
beta3.p ~ dnorm(0, 0.01)
beta4.p ~ dnorm(0, 0.01)

# Likelihood
# Ecological model for the partially observed true state
for (i in 1:R) {
   z[i] ~ dbern(psi[i]) # True occurrence z at site i
   psi[i] <- 1 / (1 + exp(-lpsi.lim[i]))
   lpsi.lim[i] <- min(999, max(-999, lpsi[i]))
   lpsi[i] <- alpha.psi + beta.psi * edge[i]
```

```
    # Observation model for the observations
    for (j in 1:T) {
        y[i,j] ~ dbern(mu.p[i,j]) # Detection-nondetection at i and j
        mu.p[i,j] <- z[i] * p[i,j]
        p[i,j] <- 1 / (1 + exp(-lp.lim[i,j]))
        lp.lim[i,j] <- min(999, max(-999, lp[i,j]))
        lp[i,j] <- alpha.p + beta1.p * DATES[i,j] + beta2.p *
          pow(DATES[i,j], 2) + beta3.p * HOURS[i,j] + beta4.p *
          pow(HOURS[i,j], 2)
        } #j
    } #i

# Derived quantities
occ.fs <- sum(z[])                              # Number of occupied sites
mean.p <- exp(alpha.p) / (1 + exp(alpha.p)) # Average detection
}
",fill = TRUE)
sink()

# Bundle data
win.data <- list(y = y, R = nrow(y), T = ncol(y), edge = edge, DATES =
   DATES, HOURS = HOURS)

# Initial values
zst <- apply(y, 1, max, na.rm = TRUE)     # Good starting values crucial
inits <- function(){list(z = zst, alpha.psi=runif(1, -3, 3), alpha.p =
   runif(1, -3, 3))}

# Parameters monitored
params <- c("alpha.psi", "beta.psi", "mean.p", "occ.fs", "alpha.p",
   "beta1.p", "beta2.p", "beta3.p", "beta4.p")

# MCMC settings
ni <- 30000
nt <- 10
nb <- 20000
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())
```

We inspect the estimates and then illustrate.

```
# Summarize posteriors
print(out, dig = 2)
           mean    sd   2.5%   25%   50%   75%  97.5%  Rhat n.eff
alpha.psi  5.83  5.26  -0.10  1.73  4.26  8.67  17.98  1.10    46
beta.psi  -6.61  5.26 -18.83 -9.38 -5.13 -2.60  -0.44  1.10    48
mean.p     0.56  0.15   0.27  0.46  0.56  0.67   0.85  1.01   200
occ.fs    17.02  2.38  11.00 16.00 17.00 18.00  21.00  1.01   220
alpha.p    0.29  0.66  -0.97 -0.15  0.26  0.72   1.71  1.01   160
beta1.p    0.34  0.40  -0.42  0.06  0.33  0.60   1.13  1.00  2400
beta2.p    0.21  0.47  -0.71 -0.10  0.19  0.51   1.17  1.01   230
beta3.p   -0.48  0.42  -1.37 -0.75 -0.46 -0.20   0.31  1.01   330
beta4.p   -0.59  0.32  -1.28 -0.79 -0.57 -0.37   0.00  1.00  1600
```

We note that convergence for the occupancy parameters could be better (Rhat = 1.10). We also note (not shown) that parameter estimates are quite sensitive to the priors chosen in the model. This is not quite unexpected, given the small size of the data set. Thus, we should state our inferences with caution.

Earlier on, we asked a series of questions that we wanted to answer with the site-occupancy model. The first was "How many woodpiles were likely occupied by *Rosalia alpina*, given the detection probability estimated?". We find the answer in the tabular summary of the estimates above, it is 17.02 (95% CRI 11–21). Since this is a key quantity in our analysis, we want to visualize its entire posterior distribution (Fig. 13.5).

```
# Posterior distribution of the number of occupied woodpiles in actual
  sample
hist(out$sims.list$occ.fs, nclass = 30, col = "gray", main = "", xlab =
   "Number of occupied woodpiles (occ.fs)", xlim = c(9, 27))
abline(v = 10, lwd = 2) # The observed number
```

The second question of interest was "Given that we may overlook the species at occupied woodpiles, how many times must we survey a

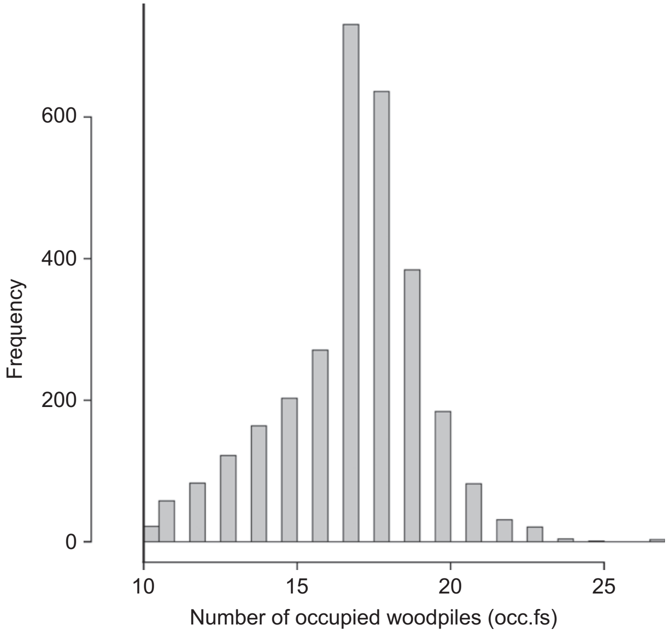

FIGURE 13.5 Posterior distribution of the number of woodpiles occupied by the cerambycid beetle *Rosalia alpina* in the Movelier region in 2009 among the 27 surveyed woodpiles. Vertical line indicates the observed number of 10.

woodpile before we can be "almost certain" to detect it at least once, when it occurs?". We can answer this question by using a simple binomial argument put forwards in Kéry (2002) and many times elsewhere: the probability $P*$ to detect the species during n identical and independent surveys is $P* = 1 - (1 - p)^n$, where p is the detection probability from a site-occupancy model. Since detection varies in all sorts of ways (see below), we have to decide on one "useful" value of p. We take the `mean.p` monitored in the analysis. Using the MCMC samples for that quantity, we can incorporate our uncertainty about detection probability into the answer to our question. We will compute $P*$ for values of n between 1 and 10 and see where it is at least 95%, which will be our definition of "almost certain".

```
Pstar <- array(NA, dim = c(out$n.sims, 10))
x <- cbind(rep(1, 3000), rep(2, 3000), rep(3, 3000), rep(4, 3000), rep
   (5, 3000), rep(6, 3000), rep(7, 3000), rep(8, 3000), rep(9, 3000),
   rep(10, 3000))
for (i in 1:out$n.sims) {
    for (j in 1:10){
        Pstar[i,j] <- 1 - (1 - out$sims.list$mean.p[i])^j
        } #j
    } #i

boxplot(Pstar ~ x, col = "gray", las = 1, ylab = "Pstar", xlab = "Number of
   surveys", outline = FALSE)
abline(h = 0.95, lty = 2, lwd = 2)
```

Hence, 3 – 4 "average" surveys were required to be almost certain to detect *Rosalia alpina* at a woodpile where it occurred (Fig. 13.6).

What about the occupancy at woodpiles at the forest edge as compared to the forest interior? Our parameter `beta.psi` represents the difference in occupancy probability, on the logit scale between woodpiles at the forest edge and those in the interior. The 95% CRI of its estimate does not cover 0; hence, we can be rather confident in that *Rosalia* was more widespread at woodpiles in the forest interior. We convert the occupancy parameters into an estimate of occupancy in both locations and plot that.

```
par(mfrow = c(2, 1))
hist(plogis(out$sims.list$alpha.psi), nclass = 40, col = "gray", main =
   "Forest interior", xlab = "Occupancy probability", xlim = c(0, 1))
hist(plogis(out$sims.list$alpha.psi+ out$sims.list$beta.psi),
   nclass = 40, col = "gray", main = "Forest edge", xlab = "Occupancy
   probability", xlim = c(0, 1))
```

So, indeed, there appears to be a big effect of the location of a woodpile on the probability that it is occupied by *Rosalia alpina*: the forest interior is much preferred (Fig. 13.7).

Finally, we want to answer the questions about a relationship between detection probability and date and time of day, respectively. We can see from the 95% CRI in the summary results table above that the regression

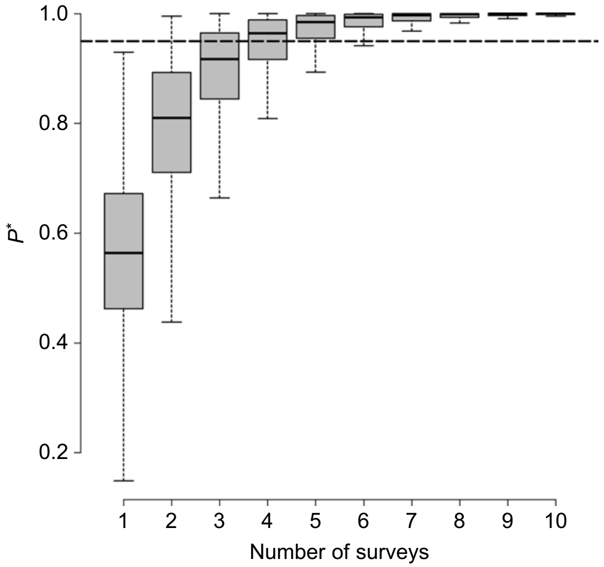

FIGURE 13.6 The relationship between *P**, the probability to detect *Rosalia alpina* at a woodpile at least once during *n* surveys, and *n* for the blue bug data set. The dashed line indicates 95% certainty to detect the species when present.

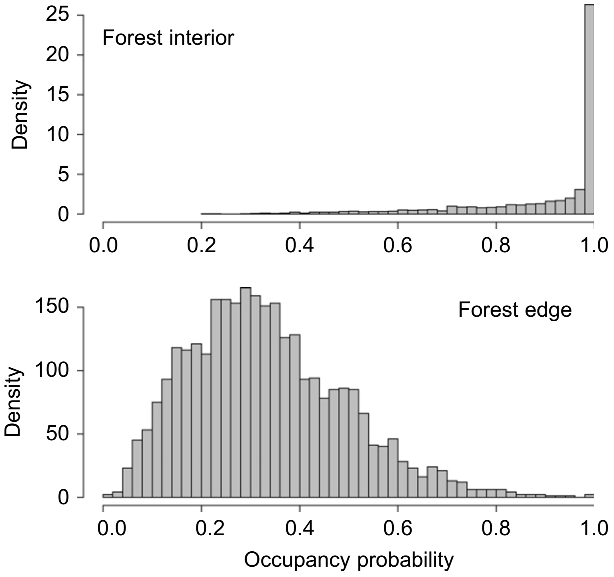

FIGURE 13.7 Posterior distributions of the probability of occupancy by *Rosalia alpina* for a woodpile in the forest interior (top) and at the forest edge (bottom) in Movelier, 2009.

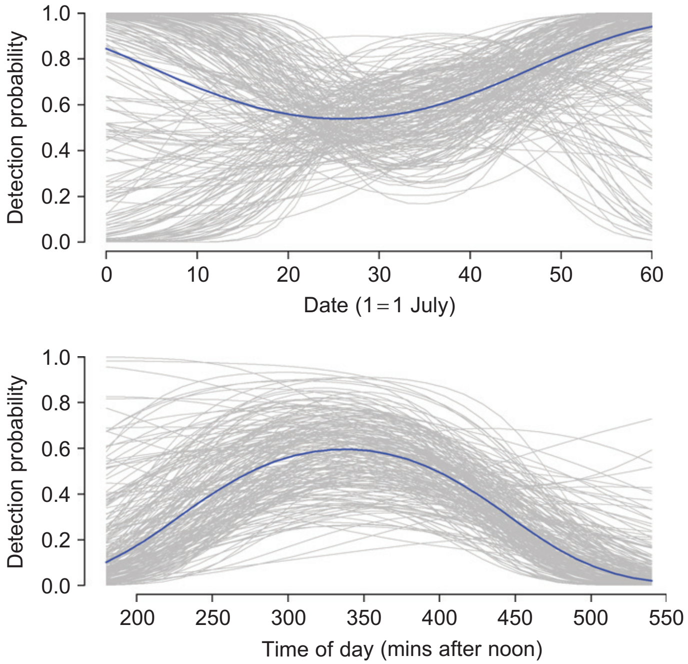

FIGURE 13.8 Predictions of the covariate relationships that account for estimation uncertainty. Top, effect of date; bottom, effect of time of day. Blue lines show the posterior mean, and gray lines show the relationships based on a random posterior sample of size 200 to visualize estimation uncertainty.

parameters for date, `beta1.p` and `beta2.p`, largely overlap zero but that those for time of day, `beta3.p` and `beta4.p`, do not do this so clearly (at least not `beta4.p`, which just about straddles 0). We will plot the predicted relationship in a figure that also shows the uncertainty in the estimates by plotting the relationships for a random MCMC sample of the regression coefficients involved in their computation (Fig. 13.8). This again suggests the absence of a date effect on detection probability (top panel); however, detection probability seems to be highest around 5 – 6 pm (bottom panel). These results can be interesting for designing a monitoring program for this endangered species.

```
# Predict effect of time of day with uncertainty
mcmc.sample <- out$n.sims

original.date.pred <- seq(0, 60, length.out = 30)
original.hour.pred <- seq(180, 540, length.out = 30)
date.pred <- (original.date.pred - mean.date)/sd.date
hour.pred <- (original.hour.pred - mean.hour)/sd.hour
p.pred.date <- plogis(out$mean$alpha.p + out$mean$beta1.p *
   date.pred + out$mean$beta2.p * date.pred^2 )
```

```
p.pred.hour <- plogis(out$mean$alpha.p + out$mean$beta3.p *
   hour.pred + out$mean$beta4.p * hour.pred^2 )

array.p.pred.hour <- array.p.pred.date <- array(NA, dim = c(length
   (hour.pred), mcmc.sample))
for (i in 1:mcmc.sample){
    array.p.pred.date[,i] <- plogis(out$sims.list$alpha.p[i] +
      out$sims.list$beta1.p[i] * date.pred + out$sims.list$beta2.p[i] *
      date.pred^2)
    array.p.pred.hour[,i] <- plogis(out$sims.list$alpha.p[i] +
      out$sims.list$beta3.p[i] * hour.pred + out$sims.list$beta4.p[i] *
      hour.pred^2)
    }

# Plot for a subsample of MCMC draws
sub.set <- sort(sample(1:mcmc.sample, size = 200))

par(mfrow = c(2, 1))
plot(original.date.pred, p.pred.date, main = "", ylab = "Detection
   probability", xlab = "Date (1 = 1 July)", ylim = c(0, 1), type = "l",
   lwd = 3, frame.plot = FALSE)
for (i in sub.set){
    lines(original.date.pred, array.p.pred.date[,i], type = "l",
      lwd = 1, col = "gray")
    }
lines(original.date.pred, p.pred.date, type = "l", lwd = 3,
   col = "blue")

plot(original.hour.pred, p.pred.hour, main = "", ylab = "Detection
   probability", xlab = "Time of day (mins after noon)", ylim = c(0, 1),
   type = "l", lwd = 3, frame.plot = FALSE)
for (i in sub.set){
    lines(original.hour.pred, array.p.pred.hour[,i], type = "l",
      lwd = 1, col = "gray")
    }
lines(original.hour.pred, p.pred.hour, type = "l", lwd = 3,
   col = "blue")
```

13.5 DYNAMIC (MULTISEASON) SITE-OCCUPANCY MODELS

So far we have been modeling detection/nondetection observations from R sites and J replicate surveys, yielding data $y_{i,j}$ for site i and survey j. We required a so-called closed population, which in the occupancy context means that the occurrence state of site i must not change over the J replicates. The closure assumption is often a reasonable approximation for studies that are short relative to the dynamics of the system investigated. However, in other cases, closure may not hold for all replicate surveys, for instance, when animals randomly move onto and off study sites. This specific form of nonclosure is called random temporary emigration, and the models of the preceding sections may still be applied. The probability of random temporary emigration, that is, of being temporarily unavailable for detection, is confounded with the probability of detection given availability (Kendall, 1999). In other words, the detection parameter refers to the product of the probability of being available for detection and that of being detected, given being present. According to conventional wisdom, the interpretation of the occupancy parameter simply changes from the *probability of permanent presence* to the *probability of use sometime* during the study period (MacKenzie, 2005).

However, there may be cases when temporary emigration (dispersal) is so strong as to make the resulting estimates of probability of use meaningless, for example, effectively 1. In other cases, temporary emigration may be Markovian: whether a site is occupied at time $t = 2$ depends on whether it was so at $t = 1$. Probability of (un-)availability is then no longer confounded with the probability of detection given availability, and naive application of single-season occupancy models results in biased estimates of occupancy (Kendall, 1999; Rota et al., 2009).

As a remedy, the J survey occasions may be assigned to subgroups and closure assumed only within each such subgroup. Owing to the seasonality of nature in most parts of the world, seasons over a series of years represent an extremely common, natural grouping factor. As an example, for birds or amphibians, replicate surveys are often conducted during the breeding season and this may be repeated over multiple years. Such a sampling at two temporal scales is called the robust design (Williams et al., 2002); each year, or breeding season, is called a primary sampling occasion, and the surveys within each season are called secondary sampling occasions. It is natural then to assume closure among secondary seasons only, that is, within each primary season, and allow change in the occurrence state among primary seasons. In the context of site-occupancy models, we then have observations from R sites, J replicate surveys (secondary sampling occasions), and K primary seasons (such as years), yielding detection/nondetection data $y_{i,j,k}$ for site i, within-season survey j, and season k. Note that up to now in this chapter, index j was for all occasions, while in this section j will index secondary occasions only.

Given our expectation that occupancy changes among seasons k, how should we model occupancy dynamics? It would be simplest to treat season as a group and fit separate parameters for each, as we did in Section 12.3 in the context of abundance estimation in an open population. This is a reasonable approach, but there may be two issues with it. First, it treats observations from a site surveyed in different seasons as independent. However, whether a site is occupied at one time may depend on whether it was occupied previously, violating the independence assumption and representing a form of pseudoreplication (Hurlbert, 1984). This may result in too short standard error estimates, so it may be desirable to account for the repeated-measures nature of multiseason data. Second, the interest of a study may focus on the parameters that govern occupancy dynamics, that is, colonization and extinction/survival. Occupancy is the quantity that metapopulation ecologists also call incidence (Hanski, 1994). Rather than simply describing changes of incidence over time, a metapopulation ecologist is interested in estimating probabilities of patch survival (or extinction) and patch colonization. This provides us with the motivation to explicitly model occupancy dynamics in terms of parameters describing the demographic components of that dynamics. This is achieved by the multiseason, or dynamic, site-occupancy model of MacKenzie et al. (2003). Moving from a single-season to a dynamic site-occupancy model is analogous to moving from a closed capture–recapture model (Chapter 6) to a Jolly-Seber model (Chapter 10) or from a classic binomial mixture model (Chapter 12) to the generalized binomial mixture model of Dail and Madsen (2011).

To describe detection/nondetection data $y_{i,j,k}$ for site i and (within-season) replicate survey j in season k, we follow the hierarchical, or state-space, formulation of the model by Royle and Kéry (2007). We describe the observed data in a two-level random-effects model, that is, as a set of two linked stochastic processes or equations. The first equation describes the ecological process, that is, the evolution of the latent occurrence state $z_{i,k}$ of site i over season k. Occurrence is latent because it is only partly observable and hence must be estimated from the observations $y_{i,j,k}$. The second equation describes the observation process, that is, the mapping of the latent state $z_{i,k}$ on observation $y_{i,j,k}$. The basic model is thus the following:

$z_{i,k} \sim \text{Bernoulli}(\psi_{i,k})$ 1. Ecological process yields true state

$y_{i,j,k} \mid z_{i,k} \sim \text{Bernoulli}(z_{i,k} p_{i,j,k})$ 2. Observation process yields observations

The sole change to the single-season occupancy model is the addition of an index for season, k. The model now describes the latent occurrence state $z_{i,k}$ at site i in season k as a Bernoulli trial with occupancy parameter $\psi_{i,k}$. Observation $y_{i,j,k}$ is equal to 1 if a species is detected during temporal replicate j at site i in season k, and zero otherwise, and is another Bernoulli trial governed by the product of the occurrence state at i and k and detection probability $p_{i,j,k}$.

As said above, we could model $y_{i,j,k}$ by simply treating season k as a group, which would be equivalent to fitting separate occupancy models to the data from each season. This is how we modeled changes in abundance over multiple seasons in Section 12.3. But now, we will describe the state dynamics in an explicit, Markovian way instead: we will specify an initial state and two sets of parameters that govern subsequent changes in a first-order autoregressive manner. This is a simple extension of the ecological process model above. For clarity, we will drop the site index (i).

$z_1 \sim \text{Bernoulli}(\psi_1)$	1a. Initial ecological state in first season
$z_{k+1} \mid z_k \sim \text{Bernoulli}(z_k \phi_k + (1 - z_k)\gamma_k)$	1b. Markovian transitions in later seasons

Hence, in season 1, occurrence is a simple Bernoulli trial as before. In all later seasons, the occurrence state z_{k+1} of a site in season $k+1$ is a Bernoulli trial with a success parameter that depends on two things: whether the site was occupied at time k and on the value of either a survival or a colonization parameter. Hence, if a site was occupied during season k (i.e., $z_k = 1$ and therefore, $1 - z_k = 0$), it will be re-occupied in the following season with probability ϕ_k; this is the (site) survival probability. Of course, we could equivalently describe this in terms of the complement of survival, extinction probability $1 - \phi_k$. On the other hand, if a site was unoccupied during season k (i.e., $z_k = 0$ and therefore, $1 - z_k = 1$), it will be occupied at $k+1$ with probability γ_k; this is the (site) colonization probability.

The state process of the dynamic site-occupancy model is exactly equivalent to a classical metapopulation model (Hanski, 1998), which expresses changes between time t and $t+1$ in the occurrence state of a collection of patches as a function of the probabilities of colonization of patches unoccupied at time k, and of survival (or alternatively, of extinction) of patches that were occupied at time k. This model makes the important assumption that the occurrence state of each patch can be determined perfectly, that is, that detection probability is equal to 1. Dynamic site-occupancy models represent an extended metapopulation model: the extension lies in an explicit accounting for imperfect detection (MacKenzie et al., 2003; Royle and Kéry, 2007), which becomes possible whenever replicated detection/nondetection observations are available within single periods of closure for at least some sites and/or such periods. Not accounting for imperfect detection in conventional metapopulation models will lead to biased estimates of *all* estimated quantities: incidence will be estimated too low and the probabilities of extinction, colonization, and turnover will all be estimated too high (Moilanen (2002); Royle and Dorazio (2008); see also Risk et al. (2011), for a robust-design incidence function model).

We will next simulate a data set under the dynamic site-occupancy model and analyze that. Afterwards, we will analyze a real data set. You will find another example of a dynamic occupancy model in the OpenBUGS manual (Examples > Ecology examples > Sparrowhawks).

13.5.1 Generation and Analysis of Simulated Data

We assume that we have data from a typical population study of a (nocturnal) bird of prey, the Long-eared owl (Fig. 13.9). Each of a total of R territories was surveyed on J occasions during each of K breeding seasons (years), and it was recorded whether any sign of territory occupation was

FIGURE 13.9 Long-eared owl (*Asio otus*), Finland, 2008 (Photograph by T. Muukkonen).

detected. Our data $y_{i,j,k}$ represent detection ($y = 1$) or nondetection ($y = 0$) of an owl in territory i, during replicate survey j in breeding season (year) k. Note that here, occupancy is equivalent to abundance because the number of occupied sites is exactly the local population size of owls.

We define a function to generate a data set. As always, apart from generating a data set to be analyzed later, this function may be used to get insights into the structure of the model used to analyze the data, issues of parameter estimation, or required samples sizes (see Section 1.5).

```
data.fn <- function(R = 250, J = 3, K = 10, psi1 = 0.4, range.p =
   c(0.2, 0.4), range.phi = c(0.6, 0.8), range.gamma = c(0, 0.1)) {
# Function to simulate detection/nondetection data for dynamic
  site-occ model
# Annual variation in probabilities of patch survival, colonization and
# detection is specified by the bounds of a uniform distribution.
# Function arguments:
# R - Number of sites
# J - Number of replicate surveys
# K - Number of years
# psi1 - occupancy probability in first year
# range.p - bounds of uniform distribution from which annual p drawn
# range.psi and range.gamma - same for survival and colonization
  probability
```

```
   # Set up some required arrays
   site <- 1:R                        # Sites
   year <- 1:K                        # Years
   psi <- rep(NA, K)                  # Occupancy probability
   muZ <- z <- array(dim = c(R, K))   # Expected and realized occurrence
   y <- array(NA, dim = c(R, J, K))   # Detection histories

   # Determine initial occupancy and demographic parameters
   psi[1] <- psi1                     # Initial occupancy probability
   p <- runif(n = K, min = range.p[1], max = range.p[2])
   phi <- runif(n = K-1, min = range.phi[1], max = range.phi[2])
   gamma <- runif(n = K-1, min = range.gamma[1], max = range.gamma[2])

   # Generate latent states of occurrence
   # First year
   z[,1] <- rbinom(R, 1, psi[1])         # Initial occupancy state

   # Later years
   for(i in 1:R){                        # Loop over sites
      for(k in 2:K){                     # Loop over years
         muZ[k] <- z[i, k-1]*phi[k-1] + (1-z[i, k-1])*gamma[k-1]
            # Prob for occ.
         z[i,k] <- rbinom(1, 1, muZ[k])
         } #k
      } #i

   # Plot realised occupancy
   plot(year, apply(z, 2, mean), type = "l", xlab = "Year", ylab =
     "Occupancy or Detection prob.", col = "red", xlim = c(0,K+1),
     ylim = c(0,1), lwd = 2, lty = 1, frame.plot = FALSE, las = 1)
   lines(year, p, type = "l", col = "red", lwd = 2, lty = 2)

   # Generate detection/nondetection data
   for(i in 1:R){
      for(k in 1:K){
         prob <- z[i,k] * p[k]
         for(j in 1:J){
            y[i,j,k] <- rbinom(1, 1, prob)
            } #j
         } #k
      } #i

   # Compute annual population occupancy
   for (k in 2:K){
      psi[k] <- psi[k-1]*phi[k-1] + (1-psi[k-1])*gamma[k-1]
      }

   # Plot apparent occupancy
   psi.app <- apply(apply(y, c(1,3), max), 2, mean)
   lines(year, psi.app, type = "l", col = "black", lwd = 2)
   text(0.85*K, 0.06, labels = "red solid - true occupancy\n red
     dashed - detection\n black - observed occupancy")

   # Return data
   return(list(R = R, J = J, K = K, psi = psi, psi.app = psi.app, z = z,
     phi = phi, gamma = gamma, p = p, y = y))
}
```

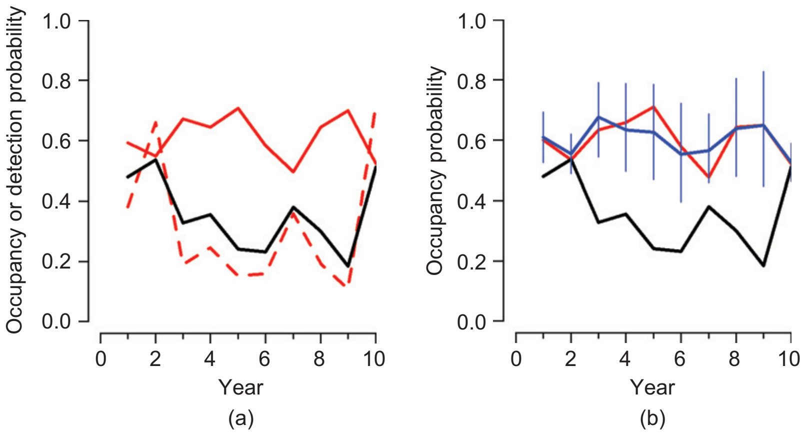

FIGURE 13.10 (a) Simulated territory occupancy data for long-eared owls. Truth is shown in red (solid, occupancy probability; dashed, detection probability) and the observed occupancy probability in black. The difference between the red and the black lines is due to detection error. (b) Comparison between true, observed, and estimated occupancy probability. Truth is shown in red, estimates under the site-occupancy model (with 95% CRI) are in blue, and naïve estimates (observed values) are in black. (Note: Using the R code in the book, you will generate each plot separately.)

We execute the function once to obtain a data set for 250 owl territories with three surveys in each of 10 years (Fig. 13.10a).

```
data <- data.fn(R = 250, J = 3, K = 10, psi1 = 0.6, range.p = c(0.1, 0.9),
   range.phi = c(0.7, 0.9), range.gamma = c(0.1, 0.5))
```

We attach the data set and produce a simple summary.

```
attach(data)
str(data)
> str(data)
List of 10
 $ R        : num  250
 $ J        : num  3
 $ K        : num  10
 $ psi      : num  [1:10] 0.6 0.535 0.635 0.658 0.71 ...
 $ psi.app  : num  [1:10] 0.48 0.536 0.328 0.356 0.24 0.232 0.38 0.3 0.184 0.512
 $ z        : num  [1:250, 1:10] 0 1 0 1 1 1 1 1 0 1 ...
 $ phi      : num  [1:9] 0.791 0.761 0.805 0.879 0.772 ...
 $ gamma    : num  [1:9] 0.151 0.489 0.403 0.384 0.105 ...
 $ p        : num  [1:10] 0.382 0.659 0.19 0.246 0.151 ...
 $ y        : num  [1:250, 1:3, 1:10] 0 0 0 1 1 0 1 0 0 0 ...
```

We conduct the analysis using code from Royle and Kéry (2007), which includes the estimation of the actual number of occupied territories (among the 250), the occupancy-based population growth rate, and the turnover rate.

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

# Specify priors
psi1 ~ dunif(0, 1)
for (k in 1:(nyear-1)){
    phi[k] ~ dunif(0, 1)
    gamma[k] ~ dunif(0, 1)
    p[k] ~ dunif(0, 1)
    }
p[nyear] ~ dunif(0, 1)

# Ecological submodel: Define state conditional on parameters
for (i in 1:nsite){
    z[i,1] ~ dbern(psi1)
    for (k in 2:nyear){
        muZ[i,k]<- z[i,k-1]*phi[k-1] + (1-z[i,k-1])*gamma[k-1]
        z[i,k] ~ dbern(muZ[i,k])
        } #k
    } #i

# Observation model
for (i in 1:nsite){
    for (j in 1:nrep){
        for (k in 1:nyear){
            muy[i,j,k] <- z[i,k]*p[k]
            y[i,j,k] ~ dbern(muy[i,j,k])
            } #k
        } #j
    } #i

# Derived parameters: Sample and population occupancy, growth rate
  and turnover
psi[1] <- psi1
n.occ[1]<-sum(z[1:nsite,1])
for (k in 2:nyear){
    psi[k] <- psi[k-1]*phi[k-1] + (1-psi[k-1])*gamma[k-1]
    n.occ[k] <- sum(z[1:nsite,k])
    growthr[k] <- psi[k]/psi[k-1]
    turnover[k-1] <- (1 - psi[k-1]) * gamma[k-1]/psi[k]
    }
}
",fill = TRUE)
sink()

# Bundle data
win.data <- list(y = y, nsite = dim(y)[1], nrep = dim(y)[2], nyear = dim
   (y)[3])

# Initial values
Zst <- apply(y, c(1, 3), max) # Observed occurrence as inits for z
inits <- function(){ list(z = zst)}

# Parameters monitored
params <- c("psi", "phi", "gamma", "p", "n.occ", "growthr",
   "turnover")
```

```
# MCMC settings
ni <- 2500
nt <- 4
nb <- 500
nc <- 3

# Call WinBUGS from R (BRT 3 min)
out <- bugs(win.data, inits, params, "Dynocc.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 = 2)
[...]
               mean   sd   2.5%    25%    50%    75%  97.5% Rhat n.eff
psi[1]         0.61 0.04   0.53   0.58   0.61   0.64   0.69 1.00   720
[...]
psi[10]        0.53 0.03   0.46   0.50   0.53   0.55   0.59 1.00   810
phi[1]         0.83 0.04   0.75   0.81   0.83   0.86   0.90 1.00  1500
[...]
phi[9]         0.67 0.06   0.55   0.63   0.68   0.71   0.80 1.00  1500
gamma[1]       0.12 0.06   0.01   0.08   0.12   0.17   0.25 1.00  1500
[...]
gamma[9]       0.24 0.12   0.02   0.15   0.24   0.33   0.44 1.00  1500
p[1]           0.40 0.03   0.35   0.38   0.40   0.42   0.47 1.00  1100
[...]
p[10]          0.71 0.03   0.66   0.69   0.71   0.72   0.75 1.00  1500
n.occ[1]     152.32 7.48 138.00 147.00 152.00 158.00 166.00 1.00  1500
[...]
n.occ[10]    131.51 2.18 128.00 130.00 131.00 133.00 137.00 1.00  1100
growthr[2]     0.91 0.06   0.80   0.87   0.91   0.95   1.04 1.00  1500
[...]
growthr[10]    0.83 0.14   0.62   0.73   0.81   0.90   1.16 1.00  1400
turnover[1]    0.09 0.05   0.01   0.05   0.08   0.12   0.20 1.00  1500
[...]
turnover[9]    0.17 0.11   0.01   0.09   0.16   0.24   0.41 1.00  1500
[...]
```

We compare truth and estimates of truth (posterior mean, sd, and 95% CRI) in tables …

```
print(cbind(data$psi, out$summary[1:K, c(1, 2, 3, 7)]), dig = 3)
print(cbind(data$phi, out$summary[(K+1):(K+(K-1)), c(1, 2, 3, 7)]),
   dig = 3)
print(cbind(data$gamma, out$summary[(2*K):(2*K+(K-2)), c(1, 2, 3,
   7)]), dig = 3)
print(cbind(data$p, out$summary[(3*K-1):(4*K-2), c(1, 2, 3, 7)]),
   dig = 3)
```

… and in a picture (Fig. 13.10b).

```
plot(1:K, data$psi, type = "l", xlab = "Year", ylab = "Occupancy
   probability", col = "red", xlim = c(0,K+1), ylim = c(0,1), lwd = 2,
   lty = 1, frame.plot = FALSE, las = 1)
```

```
lines(1:K, data$psi.app, type = "l", col = "black", lwd = 2)
points(1:K, out$mean$psi, type = "l", col = "blue", lwd = 2)
segments(1:K, out$summary[1:K,3], 1:K,out$summary[1:K,7],
   col = "blue", lwd = 1)
```

We are rather satisfied with the performance of the metapopulation estimators of the model.

13.5.2 Dynamic Occupancy Modeling in a Real Data Set

As another illustration of the dynamic occupancy model of MacKenzie et al. (2003), we will use data from the Six-spot burnet (Fig. 13.11) collected in the Swiss butterfly monitoring program. Remember that we have 95 sites with two replications in each of 7 seasons, and that a "season"

FIGURE 13.11 The Six-spot burnet *Zygaena filipendulae*, a day-flying moth, Switzerland, 2004 (Photograph by T. Marent).

represents one day, within which a transect is surveyed back and forth (for further description, see Section 12.3; Kéry et al. (2009b); Dorazio et al. (2010)). This is a more typical example of an occupancy model, where a "site" represents a 2.5 km transect in a 1 km^2 square and is so large relative to the space requirements of the study species that is can be inhabited by many (hundred) individuals. Thus, there is no longer a 1:1 relationship between occupancy and abundance as in the owl example.

After reading the count data into R, we will first reformat the data into a 3-dimensional array, as we did for the multiseason binomial mixture model in Section 12.3. We start with a format where butterfly counts from different "seasons" (days) are stacked. For this code to work, the data must be balanced, that is, we must have the same number of surveyed sites in each "season" (day). This is not a requirement of the model, simply of our code. If you have variation in the number of sites surveyed, then you have to "fill in" the data using NAs to make them balanced or else vectorize the BUGS model description (see chapter 21 in Kéry, 2010).

```
# Read in the data and put it into 3D array
bdat <- read.table(file = "burnet.txt", header = T)
str(bdat)

y <- array(NA, dim = c(95, 2, 7)) # 95 sites, 2 reps, 7 days

for (i in 1:7){
    sel.rows <- bdat$day == i
    y[,,i] <- as.matrix(bdat)[sel.rows, 3:4]
    }
str(y)

# Convert counts to detection/nondetection data
y[y>0] <- 1

# Look at the number of sites with detections for each day
tmp <- apply(y, c(1,3), max, na.rm = TRUE)
tmp[tmp == "-Inf"] <- NA
apply(tmp, 2, sum, na.rm = TRUE)
 [1] 0 0 3 10 17 17 6
```

There are no detections of burnets at all during the first two days. We are now ready to fit the dynamic occupancy model in WinBUGS. The code is the same as before (Section 13.5.1) so we simply recycle the BUGS model description from there.

```
# Bundle data
win.data <- list(y = y, nsite = dim(y)[1], nrep = dim(y)[2], nyear = dim
   (y)[3])

# Initial values
inits <- function(){ list(z = apply(y, c(1, 3), max))}

# Parameters monitored
params <- c("psi", "phi", "gamma", "p", "n.occ", "growthr",
   "turnover")
```

```
# MCMC settings
ni <- 5000
nt <- 4
nb <- 1000
nc <- 3

# Call WinBUGS from R (BRT 1 min)
out1 <- bugs(win.data, inits, params, "Dynocc.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(out1, dig = 3)
                mean      sd  2.5%   25%   50%    75%   97.5%  Rhat n.eff
psi[1]         0.106   0.208 0.001 0.009 0.025  0.078   0.895 1.048    60
[...]
psi[7]         0.117   0.052 0.044 0.080 0.106  0.142   0.246 1.005   490
phi[1]         0.442   0.294 0.014 0.177 0.419  0.687   0.969 1.007   870
[...]
phi[6]         0.447   0.183 0.170 0.313 0.415  0.551   0.900 1.002  1000
gamma[1]       0.151   0.230 0.001 0.014 0.050  0.171   0.905 1.025   100
[...]
gamma[6]       0.026   0.026 0.001 0.008 0.018  0.036   0.095 1.004   650
p[1]           0.294   0.296 0.002 0.038 0.179  0.497   0.948 1.037    79
[...]
p[7]           0.536   0.183 0.195 0.398 0.540  0.679   0.864 1.005   430
n.occ[1]       9.268  19.972 0.000 0.000 1.000  6.000  85.000 1.171    32
[...]
n.occ[7]       9.515   3.971 6.000 7.000 8.000 11.000  20.000 1.009   300
growthr[2]    21.130 257.481 0.078 0.841 2.037  6.563 117.412 1.026    88
[...]
growthr[7]     0.548   0.230 0.222 0.384 0.504  0.671   1.101 1.003   710
turnover[1]    0.714   0.287 0.054 0.539 0.826  0.953   0.998 1.011   260
[...]
turnover[6]    0.172   0.139 0.005 0.061 0.137  0.251   0.505 1.003   940
```

We see that some of the parameters associated with the first two days, when no burnets were observed, are not estimable. An indication of this is that their posterior distributions cover (almost) the entire range of their prior distributions, that is, the 95% CRI essentially covers the range from 0 to 1 for the probability parameters. This means that the data contain no information about these parameters. The parameters describing the dynamics of occupancy, survival (phi), colonization (gamma), and the growth rate, may all offer interesting insights into the factors that drive the population dynamics of a species in the context of occurrence.

Apart from the third day, when very few burnets were observed (and during the first two, see above), detection probability appears to be similar. Hence, we pool the detection parameters and fit a model with constant detection probability. In addition, as an exercise we aggregate the binary response over the two replicates per day and specify a binomial(2, p) data distribution instead of a Bernoulli(p). When there is no modeled structure among replicate surveys, this model parameterization is computationally more efficient than the one with a Bernoulli response.

```
# Specify model in BUGS language
sink("Dynocc2.txt")
cat("
model {

# Specify priors
psi1 ~ dunif(0, 1)
for (k in 1:(nyear-1)){
    phi[k] ~ dunif(0, 1)
    gamma[k] ~ dunif(0, 1)
    }
p ~ dunif(0, 1)

# Both models at once
for (i in 1:nsite){
    z[i,1] ~ dbern(psi1)        # State model 1: Initial state
    for (k in 2:nyear){         # State model 2: State dynamics
        muZ[i,k] <- z[i,k-1]*phi[k-1] + (1-z[i,k-1])*gamma[k-1]
        z[i,k] ~ dbern(muZ[i,k])

        # Observation model
        muy[i,k] <- z[i,k]*p
        y[i,k] ~ dbin(muy[i,k], 2)
        } #k
    } #i

# Derived parameters: Sample and population occupancy, growth
  rate and turnover
psi[1] <- psi1
n.occ[1] <- sum(z[1:nsite,1])
for (k in 2:nyear){
    psi[k] <- psi[k-1]*phi[k-1] + (1-psi[k-1])*gamma[k-1]
    n.occ[k] <- sum(z[1:nsite,k])
    growthr[k] <- psi[k]/psi[k-1]
    turnover[k-1] <- (1 - psi[k-1]) * gamma[k-1]/psi[k]
    }
}
",fill = TRUE)
sink()

# Aggregate detections over reps within a day and bundle data
yy <- apply(y, c(1, 3), sum, na.rm = TRUE)
win.data <- list(y = yy, nsite = dim(yy)[1], nyear = dim(yy)[2])

# Initial values
inits <- function(){list(z = apply(y, c(1, 3), max))}

# Parameters monitored
params <- c("psi", "phi", "gamma", "p", "n.occ", "growthr",
   "turnover")
```

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

# Call WinBUGS from R (BRT 1 min)
out2 <- bugs(win.data, inits, params, "Dynocc2.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(out2, dig = 3)
               mean     sd  2.5%   25%    50%    75%  97.5%  Rhat n.eff
psi[1]        0.461  0.394 0.003 0.058  0.387  0.902  0.997 1.122    21
[...]
psi[7]        0.093  0.032 0.040 0.070  0.090  0.114  0.165 1.002  1700
phi[1]        0.121  0.207 0.001 0.008  0.026  0.121  0.778 1.087    28
[...]
phi[6]        0.368  0.120 0.154 0.280  0.362  0.447  0.625 1.001  3000
gamma[1]      0.108  0.202 0.001 0.007  0.020  0.083  0.790 1.070    33
[...]
gamma[6]      0.017  0.017 0.000 0.005  0.012  0.024  0.063 1.001  3000
p             0.646  0.059 0.525 0.608  0.648  0.687  0.756 1.001  3000
n.occ[1]     43.739 38.086 0.000 5.000 36.000 87.000 95.000 1.176    16
[...]
n.occ[7]      7.241  1.301 6.000 6.000  7.000  8.000 10.000 1.001  3000
growthr[2]    0.742  4.958 0.004 0.021  0.060  0.327  5.190 1.096    26
[...]
growthr[7]    0.433  0.139 0.196 0.334  0.422  0.517  0.742 1.002  1900
turnover[1]   0.507  0.300 0.016 0.244  0.507  0.777  0.981 1.002  1400
[...]
turnover[6]   0.143  0.126 0.004 0.048  0.108  0.206  0.464 1.001  3000
```

We plot what we have learnt about the occupancy, or incidence, of Swiss burnets over the season (Fig. 13.12).

```
DAY <- cbind(rep(1, out2$n.sims), rep(2, out2$n.sims), rep(3,
   out2$n.sims), rep(4, out2$n.sims), rep(5, out2$n.sims), rep(6,
   out2$n.sims), rep(7, out2$n.sims))
boxplot(out2$sims.list$psi ~ DAY, col = "gray", ylab = "Occupancy
   probability", xlab = "Day of survey", las = 1, frame.plot = FALSE)
```

We see the typical unimodal phenology of an insect in temperate latitudes (Kéry et al., 2009). Six-spot burnets are most widespread in Switzerland during the period in which survey number 5 is made. Interestingly, although no burnets were seen during either the first or the second day, the posterior distribution for occupancy was quite different for the two days. There are two reasons for this. First, the Markovian model propagates information backwards in time and so occurrence at $k = 2$ ($z_{i,2}$) is informed directly by $z_{i,3}$ because there are data at $k = 3$. Conversely, $z_{i,1}$ gets no direct information

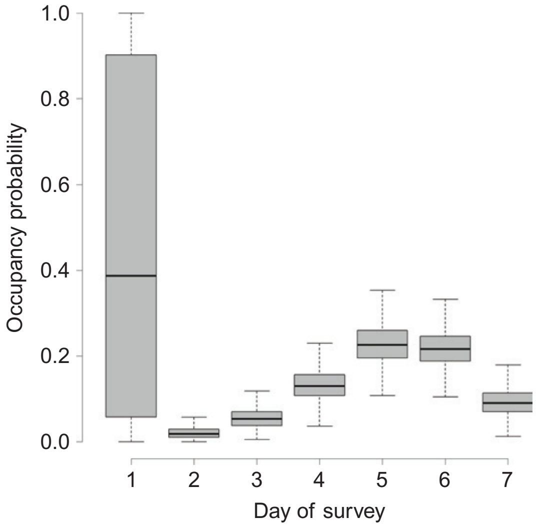

FIGURE 13.12 Occupancy probability of the burnet over a season: summary of posterior distributions for survey day 1 through survey day 7. No burnets at all were seen during the first two days.

at all from $k = 2$ because there were no observations (J.A. Royle, pers. comm.). Second, different amounts of information are available for occupancy on the two days. There were only 75 sites with surveys on the first, but 87 sites with surveys on the second day. As a consequence, the occupancy parameter was not estimable on the first day; the posterior samples simply reflected the prior. In contrast, on the second day, occupancy was estimated at effectively zero. You can compare the sample sizes for each day like the following:

```
apply(apply(y, c(1, 3), max), 2, function(x){sum(!is.na(x))})
[1] 75 87 95 95 95 95 87
```

In summary, the dynamic site-occupancy model is a powerful extension to the classical metapopulation model. Depending on the definition of a site and the state of occurrence, dynamic occupancy models can be used to describe the dynamics of a vast array of systems. Covariates can be introduced for all parameters via the usual GLM link functions. The main challenge when applying the model may be a data management and parameter bookkeeping one: to put the data in the required multidimensional arrays and not to get confused with multidimensional model code.

13.6 MULTISTATE OCCUPANCY MODELS

So far, we have been treating occurrence as a binary variable. However, frequently we can distinguish different states of occurrence. Examples include “single bird”, “nonreproductive pair”, and “reproductive pair” when studying territory occupancy, “breeding possible”, “breeding probable”, or “breeding confirmed” in bird atlas studies (Schmid et al., 1998), different population size classes in the monitoring of vocal amphibians (Royle and Link, 2005), or “occupied by species A only”, “occupied by species B only”, or “occupied by both species” in studies of species interactions. Apart from detection uncertainty, there is an additional potential component of uncertainty in these examples: state uncertainty, that is, whether a site observed in one state truly is in that state. For example, when species A is observed at a site, the true state of that site could either be “occupied by species A only” or “occupied by both species”.

The multistate site-occupancy model is used for inference about multiple states of occupancy in the presence of both state and detection uncertainty. This model seems to have been independently developed by Royle and Link (2005) and Nichols et al. (2007), providing another example for the independent and (more or less) simultaneous development of a model, such as the Cormack-Jolly-Seber model (Cormack, 1964; Jolly, 1965; Seber, 1965), single-state site-occupancy models (MacKenzie et al., 2002; Tyre et al., 2003), and spatial capture–recapture models (Borchers and Efford, 2008; Royle and Young, 2008). The explicit merging of site-occupancy models with multistate models (Chapter 9) holds promise because the combination of two already very general model classes likely results in even more flexible models. It is likely that many ideas that are well understood and applied in the multistate arena may be taken over to site-occupancy models as well.

In the following, we focus on the simplest possible multistate occupancy model, where two occurrence states along with the third state “unoccupied” are distinguished in a closed population. The generalization to more than two states is straightforward. We illustrate with data from a survey of long-eared owls (Fig. 13.9), where either hooting adult males or begging young are detected, or nothing at all. If a hooting male is heard, we are unsure about whether reproduction is taking place at a site. If we fail to hear anything, we are unsure about whether a site is occupied at all as well as whether there is reproduction. In contrast, when hearing begging young, there is no uncertainty about state and occurrence.

The development of the model is nearly identical to that of the multistate model (Chapter 9). First, we need to define lists of true and of observed states. The true states in our example are “not occupied”, “occupied without reproduction”, and “occupied with reproduction”. The list of observed states comprises “not seen”, “seen without reproduction”, and “seen with reproduction”. A hierarchical model for data from this system distinguishes a description of the state and another of the observation process. We therefore introduce the latent variable z, which defines the true state of each site and can take values 1 (site not occupied), 2 (site occupied without reproduction), or 3 (site occupied with reproduction). The probability of the state of site i is modeled using a categorical distribution as

$$z_i \sim \text{categorical}(\Omega_i),$$

where Ω_i is the state vector. The state vector has as many elements as there are states and each element is the probability that site i is in a given state. In our example, we have

$$\Omega_i = \begin{bmatrix} 1 - \psi_{1,i} - \psi_{2,i} \\ \psi_{1,i} \\ \psi_{2,i} \end{bmatrix},$$

where $\psi_{1,i}$ is the probability that site i is occupied without reproduction, $\psi_{2,i}$ is the probability that site i is occupied and reproduction takes place, and $1 - \psi_{1,i} - \psi_{2,i}$ is the probability that site i is unoccupied. Obviously, the three probabilities need to sum to 1, and we often assume that the probabilities are the same at all sites (no index i).

Given the true state z_i of site i, the observation process links the true state with the observations ($y_{i,j}$). We write

$$y_{i,j} \,|\, z_i \sim \text{categorical}(\Theta_{z_i,1\ldots O,i,j}),$$

where Θ is the observation array and O is the number of observed states. The array has four dimensions; the last two refer to site (i) and survey (j). If detection is assumed to be the same at all sites and constant among surveys, the observation array becomes a two-dimensional matrix. The first dimension refers to the true state and the second to the observed states. The elements of the matrix are the probabilities of an observation given a state. Assuming constancy over sites and surveys, the most general observation matrix Θ is

$$\begin{array}{c|ccc|} & \begin{array}{c}\text{not}\\ \text{seen}\end{array} & \begin{array}{c}\text{seen without}\\ \text{rep.}\end{array} & \begin{array}{c}\text{seen with}\\ \text{rep.}\end{array} \\ \text{not occupied} & \pi_{1,1} & \pi_{1,2} & \pi_{1,3} \\ \text{occupied without reproduction} & \pi_{2,1} & \pi_{2,2} & \pi_{2,3} \\ \text{occupied with reproduction} & \pi_{3,1} & \pi_{3,2} & \pi_{3,3} \end{array}.$$

The true states are in the rows and the observed states in the columns. Thus, $\pi_{m,k}$ denotes the probability of classifying a site in state m as being in state k. These probabilities are either detection or genuine classification probabilities or both. Clearly, the probabilities of correct classification are in the diagonal, while the off-diagonals contain the probabilities of incorrect classification. The matrix is row-stochastic, so the three probabilities in the same row are not independent; rather, they sum to one.

This matrix defines the most general multistate model that could be fitted in a site-occupancy context. Given sufficient data, all parameters should be estimable. However, frequently, there is a natural order in the modeled states and some errors are unlikely or impossible. Typically, it can be assumed that classification errors only occur in one direction, so that a "higher" state can be erroneously taken to be a lower state, but not the other way round. For instance, a site with reproduction could be classified as having no reproduction if only an adult is heard hooting and no begging young are heard, but not the other way round. The result of this is that we model a restricted version of the fully general observation matrix (now we also make explicit the relationships among cell probabilities within a row):

$$\begin{array}{l c}
 & \begin{array}{ccc} \text{not seen} & \text{seen without rep.} & \text{seen with rep.} \end{array} \\
\begin{array}{c} \text{not occupied} \\ \text{occupied without reproduction} \\ \text{occupied with reproduction} \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 1-\pi_{2,2} & \pi_{2,2} & 0 \\ 1-\pi_{3,2}-\pi_{3,3} & \pi_{3,2} & \pi_{3,3} \end{bmatrix}
\end{array}$$

Both Royle and Link (2005) and Nichols et al. (2007) describe restricted models of this kind, where a site in state 1 (unoccupied) can only be observed in state 1 (we assume there are no false positives), but sites in state 2 can be observed in state 1 or 2 and sites in state 3 in all three states (1, 2, or 3).

This model can be re-expressed in various parameterizations. What this means is that the elements of the state vector and the elements $\pi_{m,k}$ in the observation matrix can be rewritten as functions of other parameters that may be more interesting biologically or that may allow a more natural formulation of covariate effects; see Royle and Link (2005) and Nichols et al. (2007). Our parameterization in this chapter is as follows:

$$\begin{array}{cc}
\text{State vector} & \text{Observation matrix} \\
\begin{bmatrix} 1-\psi \\ \psi(1-r) \\ \psi r \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 1-p_2 & p_2 & 0 \\ p_{3,1} & p_{3,2} & p_{3,3} \end{bmatrix}
\end{array}$$

Here, ψ is the probability of occupancy, regardless of reproduction, and r is the probability that reproduction takes place at an occupied site. In the observation matrix, p_2 is the detection probability of a site without reproduction, $p_{3,3}$ is the probability that at a site with reproduction, the species is detected and reproduction is observed (i.e., the state is correctly classified), $p_{3,2}$ is the probability that at a site with reproduction, the species is detected but reproduction is not observed (i.e., the state is misclassified), and $p_{3,1}$ is the probability that the species is not detected at a site with reproduction. The three probabilities, $p_{3,k}$, must sum to one, and this is accounted for by our choice of a Dirichlet prior in the BUGS model description; see below.

To illustrate this model, we use data on territory occupancy of the long-eared owl (Fig. 13.9) from a long-term population study of our colleague Simon Birrer at the Swiss Ornithological Institute. Birrer has been surveying 40 owl territories repeatedly in every breeding season since 1989. Not all sites were checked in every year and we chose the data from 2009, when 31 sites were checked up to 5 times. We read in the data and briefly look at them.

```
owls <- read.table("owls.txt", header = TRUE)
str(owls)
```

The variables entitled `obs1-obs5` denote the result of each survey: detection of no owl at all (0), of a hooting owl (1) or of begging young (2). The variables entitled `date1-date5` give the Julian date of each survey. To fit the model, we must relabel the states because WinBUGS does not allow indices of 0. Hence, we denote the states in the same way as defined above. This relabeling is done in the data bundle statement below.

We specify the model with default vague priors for all parameters. The beta terms are used to specify a vague Dirichlet prior for the multinomial distribution represented by row three in the observation matrix above (see also Section 9.6). Our model could accommodate time variation in the observation matrix, but at first we will assume constancy of parameters over time.

```
# Specify model in BUGS language
sink("model1.txt")
cat("
model {

# Priors
p2 ~ dunif(0, 1)
psi ~ dunif(0, 1)
r ~ dunif(0, 1)
for (i in 1:3) {
   beta[i] ~ dgamma(1, 1) # Induce Dirichlet prior
   p3[i] <- beta[i]/sum(beta[])
   }

# Define state vector
for (s in 1:R){
   phi[s,1] <- 1 - psi              # Prob. of nonoccupation
   phi[s,2] <- psi * (1 - r)        # Prob. of occupancy without repro
   phi[s,3] <- psi * r              # Prob. of occupancy and repro
   }
# Define observation matrix
# Order of indices: true state, time, observed state
for (t in 1:T){
    p[1,t,1] <- 1
    p[1,t,2] <- 0
    p[1,t,3] <- 0
    p[2,t,1] <- 1-p2
    p[2,t,2] <- p2
    p[2,t,3] <- 0
    p[3,t,1] <- p3[1]
    p[3,t,2] <- p3[2]
    p[3,t,3] <- p3[3]
    }

# State-space likelihood
# State equation: model of true states (z)
for (s in 1:R){
    z[s] ~ dcat(phi[s,])
    }

# Observation equation
for (s in 1:R){
    for (t in 1:T){
        y[s,t] ~ dcat(p[z[s],t,])
        } #t
    } #s

# Derived quantities
for (s in 1:R){
    occ1[s] <- equals(z[s], 1)
    occ2[s] <- equals(z[s], 2)
    occ3[s] <- equals(z[s], 3)
    }
n.occ[1] <- sum(occ1[]) # Sites in state 1
n.occ[2] <- sum(occ2[]) # Sites in state 2
n.occ[3] <- sum(occ3[]) # Sites in state 3
}
",fill=TRUE)
sink()
```

We analyze rows 2–6 in the owls data frame and convert them to a matrix called **y**.

```
# Bundle data
y <- as.matrix(owls[, 2:6])
y <- y + 1
win.data <- list(y = y, R = dim(Y)[1], T = dim(Y)[2])

# Initial values
zst <- apply(y, 1, max, na.rm = TRUE)
zst[zst == "-Inf"] <- 1
inits <- function(){list(z = zst)}

# Parameters monitored
params <- c("p2", "p3", "r", "psi", "n.occ") # Might want to add "z"
```

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

# Call WinBUGS from R (BRT <1 min)
out1 <- bugs(win.data, inits, params, "model1.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(out1, dig = 2)
              mean    sd  2.5%   25%   50%   75% 97.5% Rhat n.eff
p2            0.35  0.19  0.04  0.21  0.33  0.46  0.82 1.00  2400
p3[1]         0.55  0.12  0.30  0.47  0.56  0.64  0.77 1.00  3000
p3[2]         0.21  0.09  0.05  0.14  0.20  0.26  0.40 1.00   530
p3[3]         0.24  0.12  0.07  0.16  0.22  0.31  0.51 1.00   550
r             0.64  0.21  0.24  0.48  0.64  0.81  0.98 1.01   350
psi           0.52  0.15  0.28  0.42  0.50  0.61  0.86 1.00  1900
n.occ[1]     19.11  5.39  5.00 16.00 20.00 23.00 27.00 1.00  1100
n.occ[2]      7.51  5.46  0.00  3.00  7.00 10.00 21.03 1.01   260
n.occ[3]     13.38  5.10  6.00 10.00 13.00 16.00 25.00 1.00   600
```

We estimate that 52% of sites are occupied, of which 64% by reproductive owls. For our specific sample of 40 sites, this translates into an estimated 13.4 occupied sites with and 7.5 sites without reproduction and 19.1 unoccupied sites. Detection probability of a site without reproduction is estimated at 0.35 and for a site with reproduction at 0.24. There is a probability of 0.21 to detect only hooting adults at a site with reproduction and one of 0.55 to miss it altogether. The parameters describing state uncertainty and detection error all refer to a single survey.

This model assumes that all parameters are constant, but the surveys take place over an extended time period (early March–early September), so this assumption may be unlikely. For instance, begging young will not be available over the entire period. Therefore, a more realistic model may be one that allows for these parameters to vary by occasion (i.e., survey 1–5).

```
# Specify model in BUGS language
sink("model2.txt")
cat("
model {

# Priors
psi ~ dunif(0, 1)
r ~ dunif(0,1 )

for (t in 1:T){
   p2[t] ~ dunif(0, 1)
   for (i in 1:3) {
      beta[i,t] ~ dgamma(1, 1)     # Induce Dirichlet prior
      p3[i,t] <- beta[i,t]/sum(beta[,t])
      } #i
   } #t
# Define state vector
for (s in 1:R){
   phi[s,1] <- 1 - psi          # Prob. of nonoccupation
   phi[s,2] <- psi * (1 - r)    # Prob. of occupancy without repro.
   phi[s,3] <- psi * r          # Prob. of occupancy and repro.
   }

# Define observation matrix
# Order of indices: true state, time, observed state
for (t in 1:T){
   p[1,t,1] <- 1
   p[1,t,2] <- 0
   p[1,t,3] <- 0
   p[2,t,1] <- 1-p2[t]
   p[2,t,2] <- p2[t]
   p[2,t,3] <- 0
   p[3,t,1] <- p3[1,t]
   p[3,t,2] <- p3[2,t]
   p[3,t,3] <- p3[3,t]
   }

# State-space likelihood
# State equation: model of true states (z)
for (s in 1:R){
   z[s] ~ dcat(phi[s,])
   }

# Observation equation
for (s in 1:R){
   for (t in 1:T){
      y[s,t] ~ dcat(p[z[s],t,])
      } #t
   } #s

# Derived quantities
for (s in 1:R){
   occ1[s] <- equals(z[s], 1)
   occ2[s] <- equals(z[s], 2)
   occ3[s] <- equals(z[s], 3)
   }
n.occ[1] <- sum(occ1[]) # Sites in state 1
n.occ[2] <- sum(occ2[]) # Sites in state 2
n.occ[3] <- sum(occ3[]) # Sites in state 3
}
",fill=TRUE)
sink()
```

We recycle the remaining "ingredients" for the call to `bugs()` below.

```
# Call WinBUGS from R (BRT 1 min)
out2 <- bugs(win.data, inits, params, "model2.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(out2, dig = 2)
              mean    sd  2.5%   25%   50%   75% 97.5%  Rhat n.eff
p2[1]         0.76  0.19  0.32  0.65  0.80  0.92  0.99  1.01   440
p2[2]         0.57  0.21  0.17  0.41  0.57  0.72  0.94  1.00  3000
p2[3]         0.16  0.16  0.00  0.05  0.12  0.23  0.58  1.00  1600
p2[4]         0.34  0.20  0.04  0.19  0.32  0.47  0.78  1.01   670
p2[5]         0.27  0.21  0.01  0.09  0.22  0.39  0.78  1.00  3000
p3[1,1]       0.53  0.17  0.20  0.40  0.53  0.65  0.84  1.00  1300
p3[1,2]       0.33  0.17  0.06  0.20  0.32  0.44  0.68  1.00  3000
p3[1,3]       0.41  0.19  0.08  0.26  0.40  0.55  0.80  1.00  3000
p3[1,4]       0.53  0.22  0.11  0.36  0.54  0.70  0.91  1.00  3000
p3[1,5]       0.37  0.25  0.02  0.16  0.34  0.55  0.87  1.00  3000
p3[2,1]       0.37  0.16  0.09  0.25  0.36  0.48  0.70  1.00  3000
p3[2,2]       0.14  0.12  0.00  0.04  0.10  0.19  0.44  1.00  1300
p3[2,3]       0.15  0.13  0.00  0.04  0.11  0.21  0.48  1.00  1000
p3[2,4]       0.24  0.19  0.01  0.09  0.20  0.36  0.69  1.00  3000
p3[2,5]       0.31  0.23  0.01  0.11  0.27  0.47  0.82  1.00  2000
p3[3,1]       0.10  0.09  0.00  0.03  0.08  0.14  0.35  1.00  3000
p3[3,2]       0.54  0.18  0.20  0.41  0.54  0.67  0.86  1.00  2000
p3[3,3]       0.45  0.19  0.11  0.30  0.44  0.58  0.82  1.01   600
p3[3,4]       0.23  0.19  0.01  0.08  0.19  0.33  0.69  1.00  3000
p3[3,5]       0.32  0.23  0.01  0.13  0.28  0.48  0.83  1.00  3000
r             0.58  0.17  0.27  0.47  0.59  0.70  0.91  1.00  3000
psi           0.40  0.10  0.22  0.33  0.40  0.47  0.62  1.00  3000
n.occ[1]     24.13  2.94 17.00 22.00 25.00 26.00 29.00  1.00  1100
n.occ[2]      6.33  2.48  1.00  5.00  6.00  8.00 11.03  1.00  3000
n.occ[3]      9.54  2.90  5.00  7.00  9.00 11.00 16.00  1.00  1900
```

Many parameters are estimated with little precision, but we see that occupancy (psi) and the conditional (on occupancy) probability of successful reproduction (r) are estimated at higher values under model 2 than under model 1. We could also specify a model with covariate effects (Julian date in our data set) on these time-dependent parameters, but leave this for the exercises.

The multistate occupancy model can be extended in two important ways. First, the generalization to more than two occupancy states is straightforward. Second, a dynamic multistate occupancy model has been developed recently (MacKenzie et al., 2009). Similar to the multistate models of Chapter 9, these models estimate state transition probabilities. Technically, the state transition is an element of the state equation and can be included in WinBUGS by using a categorical distribution. The parameters of the state transition matrix may then be, for example, the probability that a site with reproduction in year t is abandoned in year $t+1$, or the probability that a site without reproduction in year t produces young in year $t+1$. Dynamic multistate occupancy models are conceptually analogous to multievent models (Pradel, 2005).

13.7 SUMMARY AND OUTLOOK

We have introduced site-occupancy models, a class of hierarchical logistic regression model for occurrence data that jointly estimate detection probability to account for imperfect detection. Occurrence may be a proxy for the local metapopulation abundance, which is the focus of interest in the binomial mixture model of the previous chapter. Alternatively, occupancy may be the focus of interest such as in species distribution models, disease ecology, or metapopulation ecology. When detection of occupied sites (patches) is not perfect, the extent of occurrence of species will be underestimated and covariate relationships will be estimated with bias, regardless of whether there are patterns in detection probability or whether it is constant. Given suitable data (occurrence observations that are replicated in both space and time within a short period), occupancy probability can be estimated separately from detection probability, and covariate relationships with either parameter can be estimated, even when the same covariate is affecting both occurrence and detection. Knowing typical values of detection probability and how the latter varies with measurable covariates can be invaluable for the planning of surveys.

We have furthermore illustrated a dynamic, multiseason version of a site-occupancy model (MacKenzie et al., 2003; Royle and Kéry, 2007), which is precisely a generalization of a classical metapopulation model for incidence, colonization, and extinction probability that accounts for imperfect detection; imperfect detection biases virtually all parameter estimates in classical metapopulation models unless corrected for. Static and especially dynamic site-occupancy models have increasingly been used to correct for variation in effort over long time scales when studying changes in species distributions from historic data (Altwegg et al., 2008; Moritz et al., 2008; Tingley and Beissinger, 2009; Tingley et al., 2009; Kéry et al., 2010b; van Strien et al., 2011). We have also illustrated another important generalization, the multistate site-occupancy model (Royle and Link, 2005; Nichols et al., 2007). These models allow one to simultaneously deal with detection error and state uncertainty and thus considerably extend the range of possible applications of this model class. For instance, Miller et al. (2011) use multistate occupancy models to deal with false-negative (detection) *and* false-positive (misclassification) errors in occupancy data.

Further extensions of the basic model include Royle and Nichols (2003), who describe a heterogeneity site-occupancy model that allows one, under certain conditions, to estimate the mean abundance at a collection of sites from detection/nondetection data alone (see also Dorazio (2007); Conroy et al. (2008) for a Bayesian implementation). In an exciting new development, Bled et al. (2011b) describe complex, spatially explicit, dynamic occupancy model for the spread of invasive species. Roth and Amrhein (2009) have developed a site-occupancy model to estimate local survival and recruitment from territory occupancy data with unmarked animals. Dorazio and Royle (2005) have described a multispecies site-occupancy model that enables one, among other things, to estimate species richness for each site (i.e., community size) as well as for the collection of sites (i.e., metacommunity size). The Bayesian implementation of this model using data augmentation (Dorazio et al., 2006) has been very seminal for community studies; see series of papers by Kéry and Royle (2008, 2009), Russell et al. (2009), Zipkin et al. (2009, 2010), and Ruiz-Gutiérrez and Zipkin (2011). This model has been extended to open population by Kéry et al. (2009; not including dynamics) and Dorazio et al. (2010; including occurrence dynamics); Yamaura et al. (2011) developed a version of the open multispecies site-occupancy model with the Royle-Nichols (2003) formulation of detection heterogeneity. In addition, MacKenzie et al. (2009) developed a multistate, dynamic occupancy model, which appears to be a very general and unifying model—most other occupancy models can be described as special cases of this overarching model. In summary, site-occupancy models represent an extremely powerful and flexible class of models for inference about populations of animals and plants.

13.8 EXERCISES

1. In the blue bug example, fit a "behavioral response" effect, that is, fit a separate detection probability dependent on whether the species has been detected ever before at a site or not. Hint, you can use the following R code to generate the "seen-before" covariate matrix. How do you interpret the results? Would you use the behavioral response model for inference about the system behind the blue bug data set? Discuss.

```
# Generate a 'seen-before' covariate
sb <- array(NA, dim = dim(y))
for (i in 1:27){
    for (j in 1:6){
        sb[i,j] <- max(y[i, 1:(j-1)])
        }
    }
sb[is.na(y)] <- 0                       # Impute 'irrelevant' zeroes
```

2. In the dynamic occupancy model of Section 13.5.1, ignore the detection process and aggregate the temporal within-day replicates. Adapt the WinBUGS code to fit a conventional metapopulation model and see how the estimated quantities are biased; see also Ruiz-Gutiérrez and Zipkin (2011).

3. Fit a multiseason, nondynamic version of the site-occupancy model to the burnet data. That is, treat days as a group and model occupancy independent between successive days (similar to how we modeled abundance in Section 12.3). In this way, you commit some pseudoreplication, but treating days as a group allows you to model occupancy as a function of temporally varying covariates.

4. Site-occupancy models represent the only currently available species distribution modeling framework that can estimate true, rather than apparent distributions (Kéry et al., 2010a; Kéry, 2011b). However, modeling occurrence and observation jointly can be difficult in marginal data situations. Devise a simulation study, where you vary the number of sites, occupancy, and detection probability as well as the number of replicate visits per site to see that in small-data situations, occupancy estimates will be biased high, and sometimes severely so. Do so in a model with constant detection and occurrence probability. Hint: this is a somewhat larger project.

5. In the multistate occupancy model, add an effect of Julian date on detection probability of hooting adults and begging young, that is, p_2, $p_{3,2}$ and $p_{3,3}$. Do not forget to standardize the covariate.