CHAPTER

4 MODELING SPECIES DISTRIBUTION AND RANGE DYNAMICS, AND POPULATION DYNAMICS USING DYNAMIC OCCUPANCY MODELS

CHAPTER OUTLINE

Applied Hierarchical Modeling in Ecology. https://doi.org/10.1016/B978-0-12-809585-0.00004-1

4.1 INTRODUCTION TO THE MODELING OF PRESENCE/ABSENCE DYNAMICS

The subject of this chapter is modeling change in species presence/absence using dynamic, or multi-season, occupancy models (MacKenzie et al., 2003). These are discrete-space/discrete-time models for the Markovian dynamics of binary systems and are perhaps the most powerful class of models in this book. The dynamic occupancy, or dynocc, model has an extremely wide scope of application that stems from the very wide range of biological settings in which we can recognize the notion of a "site" that can be in one of two mutually exclusive states that we can broadly denote as "presence" and "absence." State transitions over discrete time steps ("seasons" or "years") are governed by two underlying processes: colonization and extinction, or alternatively, colonization and persistence.

The dynocc model has strong relationships with two other important classes of ecological models: matrix population models (Caswell, 2001) and metapopulation models (Hanski, 1998). The classical dynocc model has a component for observation error, either false-negatives only (in this chapter) or also including false-positives (Chapter 7). However, you can strip the model of its observation component and still use it for inference about "relative" presence/absence (Kéry, 2011; Monneret et al., 2018), i.e., as a model for patch occupancy dynamics where the state parameters are confounded with observation error (see Sections 4.11 and 4.12). This form of the model is essentially the basic model of metapopulation dynamics (Hanski, 1998), where detection probability is ignored and where patch occupancy is explained as a function of colonization and extinction probabilities. In classic metapopulation models, colonization and extinction probabilities are assumed to be dependent on patch features such as area or connectivity. Hence, the dynocc model can be called a generalized metapopulation model, where the generalization refers to the explicit modeling of the observation process. Finally, like presumably all discrete-state/discrete-time models in this book, the dynocc can be formulated as a hidden Markov model (Zucchini et al., 2016).

Inference for the original development of the model by MacKenzie et al. (2003) was based on marginal likelihood; and we later developed a hierarchical, or state-space, formulation (Royle and Kéry, 2007). Most published dynocc models assume sites are independent, but (spatial) autologistic models (see Chapter 9) were developed by Royle and Dorazio (2008, Chapter 9), Bled et al. (2011a,b), Broms et al. (2014, 2016b), and Molinari-Jobin et al. (2018). In metapopulation ecology, Bayesian implementations ignoring imperfect detection have been developed by several authors, including O'Hara et al. (2002) and Ter Braak and Etienne (2003). Bayesian implementations of dynocc models that include both connectivity-dependent colonization/extinction and observation error were developed by Risk et al. (2011), Sutherland et al. (2014), and Chandler et al. (2015); see also Chapter 9.

In spite of its extremely wide scope of application and the interest of science in processes, the dynocc model must be one of the most underutilized models in all of ecology. It is the most natural model for population dynamics when individuals cannot be individually identified (MacKenzie et al., 2012), e.g., for a population study where territories are defined as sites. There are a number of publications that use the model for population studies with territories (MacKenzie et al., 2003; Bruggeman et al., 2016; Monneret et al., 2018), but we think there should be many more of them. Even more glaring is the virtual lack of reception of the dynocc model in studies of species distribution or range change. Indeed, in a large proportion of studies involving species distribution models (SDMs), there is a primary interest in the dynamics of distributions, e.g. the spread of infectious diseases and invasive species, the range shifts of species under climate change, etc. In spite of this, the vast majority of SDMs are static: often, all that people do is gather two sets of covariate values (say, for climate or less frequently for habitat) at two different points in time, fit a static model to each, and predict and map some index of occurrence probability. The difference between two such maps is then interpreted in the context of range change. However, a much more powerful approach is to directly use a model that describes the change at each site explicitly, using parameters that describe the underlying processes. These parameters can then be made functions of possible drivers of that change. The dynocc is exactly such a model! In addition, all static SDMs make the equilibrium assumption, i.e., that a study organism occurs at all sites where it could occur, given the environmental conditions. This assumption will frequently be wrong, especially with species with dynamic ranges, e.g., owing to dispersal limitation. The dynocc model does not require the equilibrium assumption, and modeling covariate relationships in the rates of change underlying the distribution may be a much safer way of inferring factors that affect the distribution of a species (Yackulic et al., 2015).

How we define discrete units of time and space influences the inference of dynocc models for ecological processes, and the relationship between occurrence and abundance. For instance, the dynocc model can be adopted for the dynamics of a disease, where one occupied site, e.g., a pond, may have thousands of frogs, or where an individual frog may have billions of viruses (Lachish et al., 2012). Similarly, the model can be applied for species distribution change, i.e., as a dynamic species distribution model or dynamic SDM (e.g., Kéry et al., 2013; Clement et al., 2016; Rushing et al., 2019; Yackulic et al., 2019). In common with most presence/absence-based SDMs, there is no explicit notion of abundance (N) in the dynocc model other than that presence denotes $N > 0$ and absence, $N = 0$. At the other end of a spectrum of possible applications of the model, a site may be occupied by just one individual, breeding pair, or family group. In this case, there is a very tight relationship with a model for abundance and the number of occupied sites represents abundance, e.g., when we make sites sufficiently small or define them by some characteristic habitat feature that can only be occupied by a single such demographic unit, such as a nest-box for cavity-nesting passerines, a cliff for a cliff-nesting raptors, or a den for foxes and badgers. Table 4.1 gives an overview of the examples in this chapter in terms of how a site is defined and the range of possible values of abundance that underlie the "presence" state.

We prefer to call the model of MacKenzie et al. (2003) "dynamic" rather than "multi-season" occupancy model, because "dynamic" describes the nature of the model more succinctly (Royle and Kéry, 2007). In particular, a static occupancy model can be fit to multiple years of data by "stacking" the data and treating year as a factor (see Section 4.6), and this is also a "multi-season" model, but not a dynamic one.

Table 4.1 Overview of the occupancy examples in this chapter in terms of biology, definition of a "site," and the relationship between presence/absence (z) and the underlying abundance (*N*)

Chapter Section	Species	Definition of Site and Relationship Between z and N
4.3	Asp Viper	Site not explicitly defined, but presumably biologically defined (e.g., a suitable habitat patch), and may hold up to many dozen individuals
4.5	Eagle Owl	Classical "demography without marked individuals" (MacKenzie et al., 2012): a site is biologically defined and equals one territory and can hold either 0 or 1 pair (see also Sections 4.11, 4.12 and especially 6.4)
4.9	European Crossbill	Site is a 1 km^2 grid cell and thus not biologically defined; may hold up to several nesting pairs
4.10.2	Middle-spotted Woodpecker	Site is not precisely defined: citizen-science records are entered at some coordinate and then summarized somewhat arbitrarily in 1 km^2 grid cells; each may hold up to several nesting pairs
4.11 and 4.12	Peregrine	As for the Eagle Owl: a biologically defined site (a nesting cliff) can hold either 0 or 1 pair
4.13	Marbled White	Site is not biologically defined (a 250-m linear transect), along which there may be hundreds of individuals

4.2 DERIVATION OF THE DYNOCC MODEL FROM FIRST PRINCIPLES

As for the Binomial N-mixture model and the static site-occupancy model (Chapters 6 and 10, respectively, in AHM1), we here introduce the dynocc model from first principles by thinking about the processes involved. We want to model the dynamics of the discrete states (presence vs. absence) of discrete spatial units ("sites") in discrete time ("occasions"). One very powerful, and yet very simple, strategy to describe such binary dynamics is a Markov model: we describe the initial conditions of the system and then find a probabilistic rule that describes the transitions of the system from one time step to the next. You will see that when we do this, we will naturally arrive at the dynamic occupancy model, a wonderfully powerful and yet surprisingly simple model to describe the spatiotemporal dynamics of binary systems such as presence/absence. Such binary systems dynamics may serve to describe topics as different as the occupancy dynamics of Peregrine Falcons in a defined set of cliffs, the range dynamics of a species at a continental scale or the spatiotemporal patterns of outbreaks of some infectious disease.

Imagine that we want to describe spatiotemporal variation in the presence/absence state $z_{i,t}$ of some species at site i at time t. For now we drop the site index and consider just one particular site and look at how its state changes over time. Thus, if the site is occupied at time t we write $z_t = 1$ and otherwise $z_t = 0$. To describe the initial conditions of the system, we need a statistical distribution for the randomness in z_1. Clearly, the Bernoulli distribution is a natural candidate for this and so, for the start of our time series of presence/absence measurements, we write for the presence/absence status of the site:

$$z_1 \sim Bernoulli(\psi_1)$$

Hence, presence or absence at time 1 is a realization of a Bernoulli random variable governed by a parameter ψ_1, which is the initial occupancy probability. This description of the initial conditions of the system is exactly the same as that for the state process of the static occupancy model in Chapter 10 in AHM1.

Next, we need a probabilistic description of the random process by which sites change from occupied to unoccupied and vice versa. If we distinguish two states for a site (occupied and unoccupied) then there are exactly four possible transitions (where the vertical line below denotes conditioning and is read as "given"):

(1) from occupied at time t to occupied at time $t+1$: $z_{t+1} = 1 | z_t = 1$
(2) from occupied at t to unoccupied at $t+1$: $z_{t+1} = 0 | z_t = 1$
(3) from unoccupied at t to occupied at $t+1$: $z_{t+1} = 1 | z_t = 0$
(4) from unoccupied at t to unoccupied at $t+1$: $z_{t+1} = 0 | z_t = 0$

Let's start with a site that is occupied at time t, i.e., where $z_t = 1$, and let's ask: what is a suitable statistical model for the first transition? Of course, the natural answer is: another Bernoulli distribution, and so, for occupied sites we can write:

$$z_{t+1} = 1 | z_t = 1 \sim Bernoulli(\phi)$$

Hence, an occupied site remains occupied with a probability ϕ which we interpret as the "survival" or persistence probability for the site. We don't need to describe transition 2, since it is comprised in the description of the first transition above: every occupied site must either remain occupied or become unoccupied. Hence, the probability of transition 2 is simply $1 - \phi$, which is equal to the site extinction probability ε. Indeed, we can alternatively parameterize transitions 1 and 2 in terms of the persistence probability ϕ or the extinction probability ε, and we will see examples of both in this chapter. Traditionally, the ε parameterization is more widespread (e.g., in the original paper by MacKenzie et al., 2003) and is in `unmarked`, MARK, and PRESENCE, but our own preference is the ϕ parameterization, perhaps as an analogy with demographic models for individuals.

To complete the statistical description of the system dynamics, we also need a statistical model for transitions 3 and 4, i.e., for the possible transitions of a site that is unoccupied at time t. Here, too, a Bernoulli is a perfectly natural choice:

$$z_{t+1} = 1 | z_t = 0 \sim Bernoulli(\gamma)$$

Hence, an unoccupied site at t becomes occupied at $t+1$ with probability γ, which is the colonization probability (this is transition 3). And since, again, unoccupied sites must go into either of two states, the probability to remain unoccupied is simply $1 - \gamma$, which is the probability that governs transition 4.

This completes our description of the state dynamics of a binary system such as the presence/absence dynamics of a species. We describe it in terms of three Bernoulli random variables, one for the initial state and two for the state transitions. This model is exactly a metapopulation or stochastic patch-occupancy model, or SPOM (Hanski, 1998, 1999): sites transition between occupied and unoccupied in a random manner, but there is some regularity nonetheless, which is represented by the parameters for initial occupancy, colonization, and extinction. This model is also a Markov model of order 1 because the system state at $t+1$ depends only on the state at t and on the parameters that govern the transitions.

What if we want to robustify our inferences about such a system by acknowledging the possibility of presence/absence measurement errors? That is, we want to allow for four possible observations (y) as follows (also dropping the index of time t for now for clarity of exposition):

(1) An occupied site can be observed as occupied: $y = 1|z = 1$,
(2) An occupied site can be observed as unoccupied: $y = 0|z = 1$,
(3) An unoccupied site can be observed as unoccupied: $y = 0|z = 0$, and
(4) An unoccupied site can be observed as occupied: $y = 1|z = 0$.

Again, the probabilities of the two possible observations for an occupied or an unoccupied site must sum to 1. To describe the first two possibilities, we can write this:

$$y = 1|z = 1 \sim Bernoulli(p) \text{ and}$$

$$y = 0|z = 1 \sim Bernoulli(1 - p)$$

Thus, if a site is occupied, the observation process is governed by the (site-level) detection probability p, and $1 - p$ is the false-negative error rate. If $p = 1$, every occupied site is always detected.

Similarly, for unoccupied sites, two measurements are possible in principle and the following probability model is very natural for them.

$$y = 0|z = 0 \sim Bernoulli(1 - q) \text{ and}$$

$$y = 1|z = 0 \sim Bernoulli(q)$$

This part of the observation process is governed by the (site-level) false-positive probability q. Thus, if $q = 0$ an unoccupied site is never erroneously recorded as being an occupied site.

We can succinctly summarize the model described so far in only three lines (ignoring the index i for sites):

1. Initial state: $z_1 \sim Bernoulli(\psi_1)$
2. State dynamics: $z_{t+1}|z_t \sim Bernoulli(z_t\phi + (1 - z_t)\gamma)$
3. Observation process: $y_t|z_t \sim Bernoulli(z_t p + (1 - z_t)q)$

We write in a single line the two transition processes that comprise the state dynamics as an "if-else" statement, where z acts as a "switch": If a site is occupied at time t then we have $z_t\phi + (1 - z_t)\gamma = 1 * \phi + (1 - 1) * \gamma = \phi$ and thus the site will still be occupied at $t + 1$ with a probability that is the persistence probability ϕ. Alternatively, if the site is not occupied at t then we have $z_t\phi + (1 - z_t)\gamma = 0 * \phi + (1 - 0) * \gamma = \gamma$ and thus the site will become occupied at $t + 1$ with a probability that is the colonization probability γ. Similarly, we can write the two components of the observation process in a single line: if a site is occupied, we get a detection ($y = 1$) with a probability that is given by $1 * p + (1 - 1)q = p$ and if the site is unoccupied, then we get a detection with a probability of $0 * p + (1 - 0)q = q$.

This is perhaps the most general dynamic occupancy model for a binary system, i.e., where a site must be in one of two states, such as presence and absence and there are both types of errors: false-positives and false-negatives. In Chapter 7 we will deal with estimation under this model, but for now we assume the false-positive error rate q to be zero and hence, the observation process simplifies to:

4. Observation process: $y_t|z_t \sim Bernoulli(z_t p)$

As for the static occupancy model (Chapter 10 in AHM1), we need extra information to estimate all parameters of the dynocc model, e.g., time-to-detection measurements or, more commonly, replicated measurements at some or all sites, during some or all years or "seasons," within a short period so that we can make the usual closure assumption. In this robust-design protocol we have T primary occasions (years or seasons) and J nested, secondary occasions. That is, we need replicated data $y_{i,j,t}$ such that for site i in year t we have $j = 1...J$ repeated measurements, with $J \geq 2$ for at least some combinations of sites and years. As with any robust design, a primary occasion does not have to be a year, but is whatever time interval over which you are interested in population dynamics.

Balanced data are not required, and indeed, we have very seldom seen a balanced data set to which a dynamic occupancy model was applied. In many cases, unequal replication is simply a matter of missing values; hence, if the maximum replication at any site during any year is, say, 3, then any site with 0 or only 1 or 2 surveys in a year may be conceived of as having 3, 2, or 1 missing responses in that year. This does not pose a problem as long as this missingness is random: whether or not a survey is missing must not depend on the state of the site nor be related to the values of its dynamics parameters. An example where this general assumption about missing values would not hold is if better sites, which are more likely to be occupied and recolonized and less likely to become extinct, are more likely to get surveys or to be surveyed at all, perhaps because the observer likes to see the birds. In that case, detection probabilities will likely be overestimated and biased inferences may result. A solution is then to augment the model with a model for this adaptive or preferential sampling; see Sections 4.11 and 6.4.

The dynocc model can be described as a sort of fourfold logistic regression, one for the initial presence/absence state (governed by occupancy probability ψ_1), the transitions from an occupied site and from an unoccupied site (governed by the probabilities of persistence (ϕ) and colonization (γ), respectively), and for the observation process (governed by detection probability p). Hence, it is natural to introduce covariate information into this model in the usual linear model way and model effects of covariates and random effects on a link scale such as the logit. For example, to specify effects of a site covariate `elevation` on initial occupancy, a yearly site-level covariate `spring.precipitation` on persistence, or an observational covariate `wind.speed` on detection, we would write the following (where the parameters in each line are distinct and would need different names in BUGS code):

$$\text{logit}(\psi_{1,i}) = \alpha + \beta * elevation_i$$

$$\text{logit}(\phi_{i,t}) = \alpha + \beta * spring.precipitation_{i,t}$$

$$\text{logit}(p_{i,j,t}) = \alpha + \beta * wind.speed_{i,j,t}$$

Covariates in this model can have four different formats: they can vary by site (site covariates), by year (year covariates), by site and year (yearly site covariates), or by site, year, and occasion (observational or sampling covariates). We can only fit site covariates into initial occupancy, site, year and yearly site covariates into colonization and persistence, and all four types of covariates into detection.

The main assumptions of the classical dynocc model are the same as those of the static occupancy model, i.e., closure within each primary period, absence of false-positives, independence of detection across sites, and the specific parametric assumptions of the model. We can relax some or all of these assumptions in more complex models, but to estimate the additional parameters we need more data and often data of a different kind (e.g., "certain detection" data in false-positive models in Chapter 7). Some of these assumptions must be assessed based on your knowledge of the system and you may not be able to diagnose their violation from looking at the data. However, violations of other assumptions may show up in statistics that can be calculated from the data and the model, e.g., frequencies of detections or of capture histories, which can form the basis for goodness-of-fit (GoF) tests (see Section 4.8).

In this chapter, we focus on the multi-season model with full colonization/persistence dynamics, but we mention two simpler variants of multi-season models. Royle and Dorazio (2008, Section 9.4) describe a simpler model with a temporal autologistic dependence of $\psi_{i,t+1}$ on $z_{i,t}$ which has a parameter θ that governs the strength of the temporal autocorrelation:

$$\text{logit}(\psi_{i,t+1}) = \alpha + ... + \theta * z_{i,t}$$

This classical formulation of temporal autocorrelation in a binary time series may be preferable when you are not interested in the underlying processes of occupancy change, but simply in accommodating temporal autocorrelation (Zipkin et al., 2012; Grant et al., 2013; Tingley et al., 2016; Iknayan and Beissinger, 2018; Si et al., 2018). Finally, in the simplest multi-season occupancy model we merely stratify occupancy (and detection) probability by year (the "stacking" of Chapter 2) and assume the absence of temporal autocorrelation (see Section 4.6).

4.3 SIMULATION AND ANALYSIS OF THE SIMPLEST DYNAMIC OCCUPANCY MODEL

We start by simulating, "long-hand," a data set under the simplest possible dynocc model with constant parameters. Inspired by Kéry (2002), we assume we had surveyed 100 potential habitat patches for Asp Vipers (*Vipera aspis*; Fig. 4.1) over 12 years and conducted two surveys for some, but not all, site/year combinations (for once we illustrate the simulation of unbalanced data).

FIGURE 4.1

A wonderful male Asp Viper in the French Jura mountains, 2017 *(Photo courtesy of Thomas Ott)*.

We assume probabilities of initial occupancy $\psi_1 = 0.7$, persistence $\phi = 0.9$, colonization $\gamma = 0.05$, and per-visit detection $p = 0.25$. For constant dynamics parameters, the Markov chain has an equilibrium at $\psi^{eq} = \gamma/(\gamma + \varepsilon) = \gamma/(\gamma + (1 - \phi))$, where ψ^{eq} is the occupancy probability at equilibrium (see Chapter 7 in MacKenzie et al., 2006, and Royle and Kéry, 2007). Therefore, our Asp Viper population will tend toward an occupancy of $0.05/(0.05 + (1 - 0.9)) = 0.33$, which implies a decline from the current level of occupancy of 0.7. This emphasizes that a metapopulation can decline even with constant underlying vital rates!

```
# Choose sample sizes and prepare arrays for z and y
nsites <- 100                           # Number of sites
nyears <- 12                            # Number of years
nsurveys <- 2                           # Number of presence/absence measurements
z <- array(NA, dim = c(nsites, nyears)) # latent presence/absence
y <- array(NA, dim = c(nsites, nsurveys, nyears)) # observed data

# Set parameter values as per above
psi1 <- 0.7                             # Prob. of initial occupancy or presence
phi <- 0.9                              # Persistence probability
gamma <- 0.05                           # Colonization probability
p <- 0.25                               # Probability of detection
(psi.eq <- gamma / (gamma+(1-phi)))     # Equilibrium occupancy
```

We simulate presence/absence in the first year and then propagate the occurrence status forward through year 12 using system dynamics.

```
# Generate initial presence/absence (i.e., the truth in year 1)
set.seed(1)                             # So we all get same data set
z[,1] <- rbinom(n = nsites, size = 1, prob = psi1)
sum(z[,1]) / nsites                     # True occupancy proportion in year 1
[1] 0.68

# Generate presence/absence (i.e., the truth) in subsequent years
for(t in 2:nyears){
  exp.z <- z[,t-1] * phi + (1 - z[,t-1]) * gamma
  z[,t] <- rbinom(n = nsites, size = 1, prob = exp.z)
}
apply(z, 2, sum) / nsites               # True occupancy proportions
[1] 0.68 0.68 0.63 0.58 0.49 0.46 0.47 0.42 0.38 0.37 0.31 0.34
```

Next, we simulate the measurement of the system state, with false-negative errors only.

```
# Detection/nondetection data (i.e. presence/absence measurements)
for(t in 1:nyears){                     # Loop over years 1 to 12
  for(j in 1:nsurveys){                 # Loop over repeat visits 1 and 2
    y[,j,t] <- rbinom(n = nsites, size = 1, prob = z[,t] * p)
  }
}
y  ;  str(y)                           # Look at the data thus far simulated
```

We have simulated two presence/absence measurements for 100 sites over 12 years. Finally, we want to add realism and simulate an unbalanced data set, by "shooting holes" into our balanced data set and turning some of the data into missing values. The way in which we do this will be decisive about whether these missing values are relevant for the analysis or not. Here, the usual considerations enter about data "missing at random" (MAR) or "missing not at random" (MNAR) (Rubin, 1976): MAR means that whether or not a datum is missing does *not* depend on its value. There are many possible ways in which we may obtain data with MNAR, and in these cases the missing data-generating process must be modeled to avoid biased inferences (see Sections 4.11 and 6.4). We assume MAR here with a constant probability of 0.2 for a survey to be missing, regardless of whether the site is occupied or not. That is, we assume the simplest possible structure in our missing value data-generating model.

```
# Generate missing values: create simple version of unbalanced data set
prob.missing <- 0.2          # Constant NA probability
y2 <- y                      # Duplicate balanced data set
for(i in 1:nsites){          # Loop over every datum in 3D array
  for(j in 1:nsurveys){
    for(t in 1:nyears){
      turnNA <- rbinom(1, 1, prob.missing)
      y2[i,j,t] <- ifelse(turnNA == 1, NA, y2[i,j,t])
    }
  }
}
y2   ;   str(y2)             # Look at the data now
```

We tally up the number of visits per site and year and find that 46 site/year combinations had no visits at all, 422 had a single visit, and 732 had the nominal two visits.

```
table(nvisits <- apply(y2, c(1,3), function(x) sum(!is.na(x))))

# Compute true expected and realized occupancy (psi and psi.fs)
psi <- numeric(nyears)    ;   psi[1] <- psi1
for(t in 2:nyears){          # Compute true values of psi
  psi[t] <- psi[t-1] * phi + (1 - psi[t-1]) * gamma
}
psi.fs <- colSums(z) / 100 # True realized occupancy

# Compute observed occupancy proportion
zobs <- apply(y2, c(1,3), function(x) max(x, na.rm = TRUE))
zobs[zobs == '-Inf'] <- NA   # 13 site/years without visits
psi.obs <- apply(zobs, 2, sum, na.rm = TRUE) / apply(zobs, 2, function(x)
  sum(!is.na(x)))
```

Fig. 4.2 shows the trajectories of true expected occupancy (`psi`), true realized or finite-sample occupancy (`psi.fs`; Royle and Kéry, 2007), and observed occupancy. There is a clear decline in occupancy given our simulated parameter values, but examining the observed data alone (absent any modeling process) would only seem to suggest fluctuations around a constant level.

We first fit the dynocc model in `unmarked` and then in BUGS. For `unmarked`, we need to reformat the data into the "wide format"—a 2D array with dimension nsites × nyears × nsurveys, i.e., we need to bind the yearly data side by side.

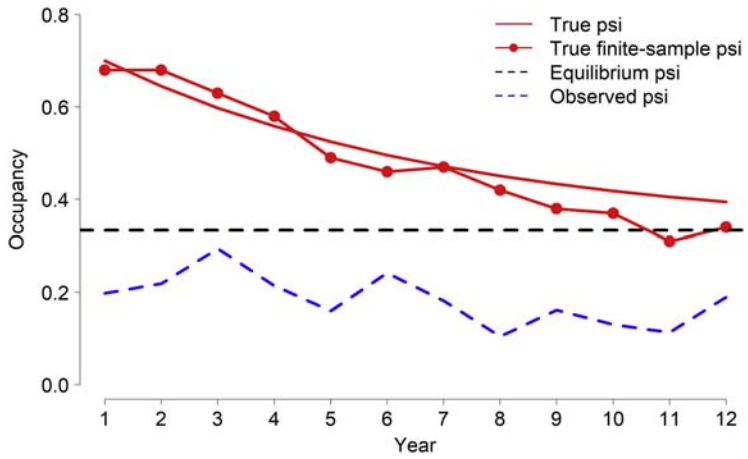

FIGURE 4.2

Trajectories of true expected occupancy (`psi`) and true realized occupancy (`psi.fs`) and of the observed proportion of occupied sites in the simulated Asp Viper population. The dashed black line shows the equilibrium occupancy that the population is heading to if the dynamics rates remain constant for long enough.

```
library(unmarked)
yy <- matrix(y2, nrow = nsites, ncol = nsurveys * nyears)
str(yy)
int [1:100, 1:24] 0 NA NA 0 0 NA NA 0 1 0 ...
```

We fit the model in `unmarked` using function `colext`, backtransform the estimates to the probability scale, and form confidence intervals.

```
# Package and summarize data set
summary(umf <- unmarkedMultFrame(y = yy, numPrimary = nyears))

unmarkedFrame Object

100 sites
Maximum number of observations per site: 24
Mean number of observations per site: 18.86
Number of primary survey periods: 12
Number of secondary survey periods: 2
Sites with at least one detection: 70

Tabulation of y observations:
   0    1 <NA>
1651  235  514

# Fit dynamic occupancy model and look at estimates
summary(fm <- colext(psiformula = ~1, # First-year occupancy
gammaformula = ~ 1,                   # Colonization
epsilonformula = ~ 1,                 # Extinction
pformula = ~ 1,                       # Detection
data = umf))
```

```
Call:
colext(psiformula = ~1, gammaformula = ~1, epsilonformula = ~1,
  pformula = ~1, data = umf)

Initial (logit-scale):
 Estimate    SE    z P(>|z|)
     1.07 0.435 2.47  0.0135

Colonization (logit-scale):
 Estimate   SE     z P(>|z|)
    -3.76 0.93 -4.04 5.3e-05

Extinction (logit-scale):
 Estimate    SE     z  P(>|z|)
     -2.4 0.304 -7.88 3.24e-15

Detection (logit-scale):
 Estimate    SE    z  P(>|z|)
    -1.16 0.127 -9.1 9.18e-20

AIC: 1167.49
Number of sites: 100
optim convergence code: 0
optim iterations: 47
Bootstrap iterations: 0

# Backtransform estimates to probability scale
backTransform(fm, type = "psi")    # First-year occupancy
backTransform(fm, type = "col")    # Colonization probability
backTransform(fm, type = "ext")    # Extinction probability
backTransform(fm, type = "det")    # Detection probability

 Estimate     SE LinComb (Intercept)
    0.745 0.0825    1.07           1

 Estimate     SE LinComb (Intercept)
   0.0228 0.0207   -3.76           1

 Estimate     SE LinComb (Intercept)
   0.0835 0.0233    -2.4           1

 Estimate     SE LinComb (Intercept)
    0.239 0.0231   -1.16           1

# For Null model point estimates can simply do this
(MLEs <- plogis(coef(fm)))

# Get 95% CIs on probability scale and print them along with MLEs
( MLEandCI <- cbind(MLEs, rbind(plogis(confint(fm, type = "psi")),
plogis(confint(fm, type = "col")),
plogis(confint(fm, type = "ext")),
plogis(confint(fm, type = "det"))) ))

               MLEs       0.025     0.975
psi(Int) 0.74517648 0.555131656 0.8726586
col(Int) 0.02277496 0.003751754 0.1260508
ext(Int) 0.08351979 0.047826505 0.1418819
p(Int)   0.23907422 0.196680790 0.2873369
```

Next, we fit the same model in BUGS. For group-organized data like this, we like to model the simulated 3D array directly because covariate information about site, replicate, and year is then contained in the array dimensions and because the BUGS code is nice and tidy. We parameterize the model in terms of persistence (`phi`) and compute the converse, extinction (`eps`), as a derived quantity.

```
# Bundle and summarize data set
str(bdata <- list(y = y2, nsites = dim(y2)[1], nsurveys = dim(y2)[2],
  nyears = dim(y2)[3]))

List of 4
 $ y        : int [1:100, 1:2, 1:12] 0 NA NA 0 0 NA NA 0 1 0 ...
 $ nsites   : int 100
 $ nsurveys : int 2
 $ nyears   : int 12

# Specify model in BUGS language
cat(file = "dynocc.txt","
model {

# Priors
psi1 ~ dunif(0, 1)
phi ~ dunif(0, 1)
gamma ~ dunif(0, 1)
p ~ dunif(0, 1)

# Likelihood
# Ecological submodel: Define state conditional on parameters
for (i in 1:nsites){   # Loop over nsites sites
  # Initial conditions of system
  z[i,1] ~ dbern(psi1) # Presence/absence at start of study
  # State transitions
  for (t in 2:nyears){ # Loop over nyears years
    z[i,t] ~ dbern(z[i,t-1] * phi + (1-z[i,t-1]) * gamma)
  }
}

# Observation model
for (i in 1:nsites){
  for (j in 1:nsurveys){
    for (t in 1:nyears){
      y[i,j,t] ~ dbern(z[i,t] * p)
    }
  }
}

# Derived parameters
eps <- 1 - phi          # Extinction probability
psi[1] <- psi1          # Population occupancy
for (t in 2:nyears){
  psi[t] <- psi[t-1] * phi + (1-psi[t-1]) * gamma
}
}
")

# Initial values
zst <- zobs             # Take observed presence/absence as inits
inits <- function(){ list(z = zst)}
```

```
# Parameters monitored
params <- c("psi1", "phi", "eps", "gamma", "p", "psi")  # Could add 'z'

# MCMC settings
na <- 1000  ;  ni <- 25000  ;  nt <- 10  ;  nb <- 5000  ;  nc <- 3

# Call JAGS (ART 2 min), check convergence and summarize posteriors
library(jagsUI)
out1 <- jags(bdata, inits, params, "dynocc.txt", n.adapt = na, n.chains = nc,
  n.thin = nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(3,3))  ;  traceplot(out1)
print(out1, 3)          # not shown
```

We compare the truth with the MLEs obtained from `unmarked` and the posterior means from JAGS and find them reasonably similar. We would get more similarity still with increased sample size (i.e., more sites, more replicates, and more years). One credible interval does not include the true value and another only just, but of course this will happen by chance about 5% of the time.

```
     truth  MLEs 0.025 0.975  mean  2.5% 97.5%
psi1  0.70 0.745 0.555 0.873 0.742 0.581 0.885
col   0.05 0.023 0.004 0.126 0.086 0.050 0.139
ext   0.10 0.084 0.048 0.142 0.032 0.002 0.079
det   0.25 0.239 0.197 0.287 0.234 0.198 0.277
```

Finally we compare true and estimated occupancy probability (Fig. 4.3). Without going into details, we simply note that these estimates, in this case of a model without site covariates, can be obtained from the `unmarked` model fit object in the slot `projected` (we see some more later; also you will see in Section 4.9 how to obtain CIs around the MLEs of psi for all years). Comparing the trajectories of true occupancy with the two sets of estimates and the observed raw data (Fig. 4.2) we see that without the dynamic occupancy model we might have concluded that the Asp Viper population is more or less stable, while in fact it is in steep decline in terms of the number of extant populations among the 100 studied.

To finish off this first encounter with the dynamic occupancy model, we emphasize again the striking similarity between the description of a statistical model when written in algebra, when data are simulated in R, and when the model is described in the BUGS model-fitting language (Table 4.2).

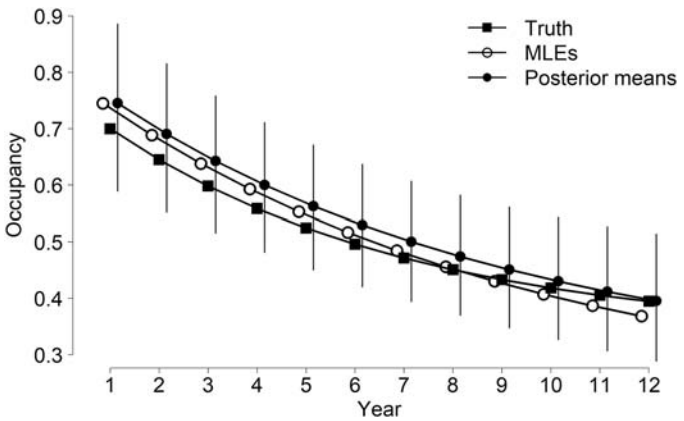

FIGURE 4.3

True and estimated population occupancy trajectory over 12 years in a simulated Asp Viper metapopulation studied at 100 sites. Uncertainty intervals are 95% CRIs.

Table 4.2 Descriptions of the dynamic occupancy model in terms of algebra, data simulation code in R, and the BUGS language.

Initial State	
Algebra	$z_1 \sim Bernoulli(\psi)$
R data simulation	`z[ ,1] <- rbinom(nsites,1,psi1)`
BUGS language	`z[ i,1] ~ dbern(psi1)`
State dynamics	
Algebra	$z_{i,t+1} \vert z_{i,t} \sim Bernoulli(z_{i,t}\phi + (1 - z_{i,t})\gamma)$
R data simulation	`z[ ,t] <- rbinom(nsites,1,z[ ,t-1]*phi+(1-z[ ,t-1])*gamma)`
BUGS language	`z[ i,t] ~ dbern(z[ i,t-1]*phi+(1-z[ i,t-1])*gamma)`
Observation model	
Algebra	$y_{i,j,t} \sim Bernoulli(z_{i,t}p)$
R data simulation	`y[ ,j,t] <- rbinom(nsites, 1, z[ ,t] * p)`
BUGS language	`y[ i,j,t] ~ dbern(z[ i,t] * p)`

Finally, we point out that when you have no patterns in detection across the J replicate surveys, the observation model can be simplified by fitting it to the aggregated site-specific data (see Section 10.3 in AHM1 and Section 4.10.2 in this volume). Also, other observation protocols such as removal sampling or time-to-detection (see Chapter 10 in AHM1) can easily be adopted for a dynamic occupancy model.

4.4 A GENERAL DATA SIMULATION FUNCTION FOR DYNOCC MODELS

Here we introduce a function to generate data sets under a very general dynamic occupancy model: `simDynocc`. As always, apart from producing a data set to be analyzed, this function may be used to get insights into the structure of the dynocc model, explore issues of parameter estimation, power, sample sizes requirements, and many more; see Chapter 4 in AHM1 about why data simulation is important for you. Function `simDynocc` allows to vary many things: the number of sites, years (primary occasions), and surveys (occasions, secondary occasions); and to choose constant or time-varying parameters for persistence (ϕ), colonization (γ), and detection (p). Every parameter type can be made a function of one continuous covariate, which is a site covariate for ψ_1, a yearly site covariate for ϕ and γ, and a sampling covariate for p. In addition, the function permits to add a quadratic seasonal effect on p to mimic the typical seasonal pattern in the detection probability of, say, an insect species, where first abundance increases at the beginning of a season and later decreases again, leading to a strong seasonal effect in detection probability as a function of the underlying abundance phenology curve (see Section 1.8). The function also allows to introduce random site and site-survey effects in detection probability and simulation of data under a BACI (before-after-control-impact; Russell et al., 2009; Popescu et al., 2012; Russell et al., 2015) design, where some event happens in a specified year and then reduces ϕ or γ by a specified percentage (only reductions are allowed); see Section 5.6.1 (Note that in Chapter 9 we will encounter a spatial variant of a function

to simulate data under a dynamic occupancy model, under a variant of the model of Bled et al. (2011a,b) with autologistic and other spatial effects in colonization and persistence parameters: `simDynoccSpatial`). The most complex model under which we can generate data with `simDynocc` is:

$$z_{i,1} \sim Bernoulli(\psi_i)$$

$$z_{i,t+1}|z_{i,t} \sim Bernoulli\left(z_{i,t}\phi_{i.t} + (1 - z_{i,t})\gamma_{i,t}\right)$$

$$y_{i,j,t}|z_{i,t} \sim Bernoulli\left(z_t p_{i,j,t}\right)$$

$$\text{logit}(\psi_i) = \text{logit}(\text{mean.psi1}) + \text{beta.Xpsi1} * \text{Xpsi1}_i$$

$$\begin{aligned}\text{logit}(\phi_{i,t}) = {} & \text{logit}(\text{mean.phi}_t - \\ & \text{mean.phi}_t * (impact.phi/100) * I(t \geq \text{Year.of.impact}) + \\ & \text{beta.Xphi} * \text{Xphi}_{i,t}\end{aligned}$$

$$\begin{aligned}\text{logit}(\gamma_{i,t}) = {} & \text{logit}(\text{mean.gamma}_t - \\ & \text{mean.gamma}_t * (impact.gamma/100) * I(t \geq \text{Year.of.impact}) + \\ & \text{beta.Xgamma} * \text{Xgamma}_{i,t}\end{aligned}$$

$$\begin{aligned}\text{logit}\left(p_{i,j,t}\right) = {} & \text{logit}(\text{mean.p}_t) + \text{beta.Xp} * \text{Xp}_{i,j,t} + eps1_i + eps2_j + eps3_{i,j,t} + \\ & \text{beta1}_t * (j - (nrep/2)) + \text{beta2}_t * (j - (nrep/2))^2\end{aligned}$$

This general function allows you to generate a wide variety of data sets under the dynamic occupancy generating model. The function defaults to annual variation in phi, gamma, and p. Here are some usages.

```
library(AHMbook)
str(data <- simDynocc(    # Explicit defaults
nsites = 250, nyears = 10, nsurveys = 3, year.of.impact = NA,
  mean.psi1 = 0.4, beta.Xpsi1 = 0,
  range.phi = c(0.5, 1), impact.phi = 0, beta.Xphi = 0,
  range.gamma = c(0, 0.5), impact.gamma = 0, beta.Xgamma = 0,
  range.p = c(0.1, 0.9), beta.Xp = 0,
  range.beta1.survey = c(0, 0), range.beta2.survey = c(0, 0),
  trend.sd.site = c(0, 0), trend.sd.survey = c(0, 0),
  trend.sd.site.survey = c(0, 0), show.plot = TRUE))

# All four parameters constant
str(data <- simDynocc(nsites = 250, nyears = 10, nsurveys = 3, mean.psi1 = 0.6,
  range.phi = c(0.7, 0.7), range.gamma = c(0.3, 0.3), range.p = c(0.5, 0.5)))

# Full time-dependence
str(data <- simDynocc(mean.psi1 = 0.6, range.phi = c(0.5, 0.8),
  range.gamma = c(0.1, 0.5), range.p = c(0.1, 0.9)))

# Constant intercepts, but covariates in all parameters
str(data <- simDynocc(mean.psi1 = 0.6, beta.Xpsi1 = 1,
  range.phi = c(0.6, 0.6), beta.Xphi = 2, range.gamma = c(0.3, 0.3),
  beta.Xgamma = 2, range.p = c(0.2, 0.2), beta.Xp = -2) )

# Full time-dependence and and effects of all covariates (incl. season)
str(data <- simDynocc(mean.psi1 = 0.6, beta.Xpsi1 = 1,
  range.phi = c(0.6, 1), beta.Xphi = 2, range.gamma = c(0, 0.2),
  beta.Xgamma = 2, range.p = c(0.1, 0.9), beta.Xp = -2,
  range.beta1.survey = c(2, 10), range.beta2.survey = c(-10, -20)) )
```

```
# No detection error (i.e., p = 1)
str(data <- simDynocc(range.p = c(1, 1)) )

# Can do a single site ...
str( data <- simDynocc(nsites = 1) )

# ... but must have at least two years
str(data <- simDynocc(nyears = 2) )
```

For seasonal variation in detection probability, as might be typical for an insect population in temperate latitudes, i.e., with seasonal effects in p which are different each year, do this (we assume we have one visit in each of 12 months).

```
str(data <- simDynocc(nsurveys = 12, mean.psi1 = 0.6,
  range.phi = c(0.6, 0.6), range.gamma = c(0.3, 0.3),
  range.p = c(0.5, 0.5), range.beta1.survey = c(-0.3, 0.4),
  range.beta2.survey = c(0, -0.7)) )

# Add detection heterogeneity at the site level
str(data <- simDynocc(trend.sd.site = c(3, 3)) )        # No time trend
str(data <- simDynocc(trend.sd.site = c(1, 3)) )        # With time trend

# Add detection heterogeneity at the level of the survey
str(data <- simDynocc(trend.sd.survey = c(3, 3)) )      # No time trend
str(data <- simDynocc(trend.sd.survey = c(1, 3)) )      # With time trend

# Add detection heterogeneity at the level of the individual visit
str(data <- simDynocc(trend.sd.site.survey = c(3, 3)) ) # No trend
str(data <- simDynocc(trend.sd.site.survey = c(1, 3)) ) # With trend
```

The following simulates data under a BACI design where an impact happens in year 10 (out of 20) and reduces `phi` by 80% and `gamma` by 50% (`year.of.impact` must be >1 and < `nyears`).

```
str(data <- simDynocc(nsites = 250, nyears = 20, nsurveys = 3,
  year.of.impact = 10, impact.phi = 80, impact.gamma = 50) )
```

We use this function to answer some design questions in Section 4.7 and in a meta-analysis in Section 5.6. For now, we use it to generate a data set with a special kind of a covariate, time (i.e., the factor `year`) and show how to fit a model with year effects in all of gamma, eps/phi, and p.

4.5 SIMULATION AND ANALYSIS OF A TIME-DEPENDENT DATA SET WITH `unmarked` AND BUGS

Let's imagine now that we were studying a population of 250 Eagle Owl territories (see Chapter 6) and that in each of 10 years, we made three surveys at each to assess whether a territory was occupied. We use `simDynocc` to obtain a data set (Fig. 4.4). We see substantial annual variation in the persistence (phi), colonization (gamma), and detection (p). The true number of pairs (= occupied sites) averages about 120 (range 67−152) and is declining. In contrast, the observed number of pairs appears to be stable, a spurious pattern that is driven by an increase in p over time.

```
set.seed(1)
str(data <- simDynocc(nsites = 250, nyears = 10, nsurveys = 3, mean.psi1 = 0.6,
  range.phi = c(0.5, 1), range.gamma = c(0, 0.5), range.p = c(0.1, 0.5)))
```

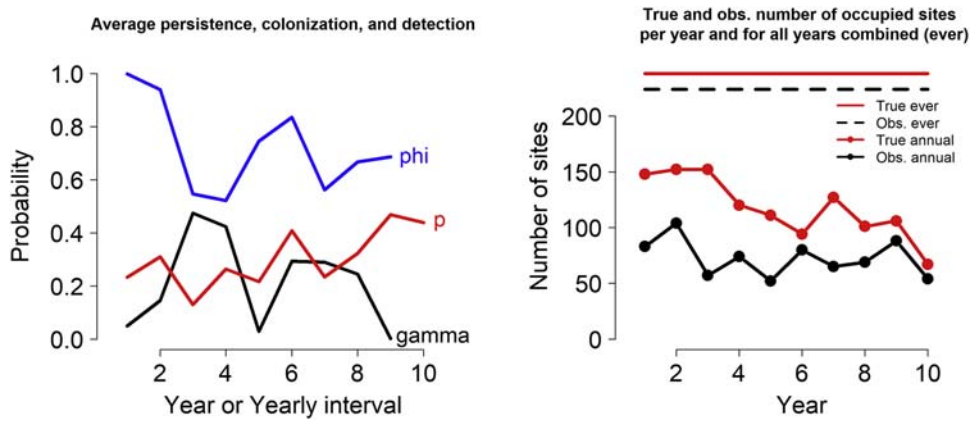

FIGURE 4.4

Simulated dynamic occupancy data for an imaginary study of Eagle Owls with 250 territories (sites), and with 3 replicate surveys in each of 10 years.

We first use the `unmarked` function `colext` to fit the basic, fully time-dependent dynamic occupancy model. There is a vignette on this function available on CRAN and this section draws much from it. We load the package and reformat the detection/nondetection data from a three-dimensional array (as generated) into a two-dimensional matrix with `nsites` rows.

```
library(unmarked)
str(yy <- matrix(data$y, data$nsites, data$nsurveys * data$nyears) )
int [1:250, 1:30] 0 0 0 1 0 0 0 1 1 0 ...

# Create a matrix indicating the year each site is surveyed
year <- matrix(c('01','02','03','04','05','06','07','08','09','10'), nrow = nrow(yy),
  ncol = data$nyears, byrow=TRUE)
head(year)        # not shown
```

To organize the data we use the function `unmarkedMultFrame`. The only required arguments now are `y`, the detection/nondetection data, and `numPrimary`, the number of seasons. Three types of covariates can be supplied using the arguments `siteCovs`, `yearlySiteCovs`, and `obsCovs`. We put all available covariates into the `unmarked` data frame for illustration though we are not going to use them for the moment, apart from `year`. In later examples, you will see all three types of covariates actually being used.

```
xxp <- matrix(data$Xp, data$nsites, data$nsurveys * data$nyears) # Xp in 2D
summary( umf <- unmarkedMultFrame(y = yy, siteCovs = data.frame(Xpsi1 = data$Xpsi1),
  yearlySiteCovs = list(year = year, Xphi = data$Xphi, Xgamma = data$Xgamma),
  obsCovs = list(Xp = xxp), numPrimary = data$nyears) )

unmarkedFrame Object

250 sites
Maximum number of observations per site: 30
Mean number of observations per site: 30
Number of primary survey periods: 10
Number of secondary survey periods: 3
Sites with at least one detection: 224
```

```
Tabulation of y observations:
   0    1
6492 1008

Site-level covariates:
     Xpsi1
 Min.   :-1.94769
 1st Qu.:-0.86551
 Median :-0.01341
 Mean   : 0.03930
 3rd Qu.: 0.92469
 Max.   : 1.97074

Observation-level covariates:
       Xp
 Min.   :-1.999574
 1st Qu.:-0.981910
 Median : 0.008498
 Mean   : 0.009213
 3rd Qu.: 1.003295
 Max.   : 1.999421

Yearly-site-level covariates:
      year           Xphi              Xgamma
 01     : 250   Min.   :-1.99758   Min.   :-1.999199
 02     : 250   1st Qu.:-1.07270   1st Qu.:-0.983174
 03     : 250   Median :-0.08815   Median :-0.008474
 04     : 250   Mean   :-0.03517   Mean   : 0.008517
 05     : 250   3rd Qu.: 1.01944   3rd Qu.: 1.092480
 06     : 250   Max.   : 1.99972   Max.   : 1.996779
 (Other):1000
```

Next, we fit a series of dynamic occupancy models that make different assumptions about the annual variation (or lack thereof) in the parameters. We will then use AIC to help us decide which model best describes the data set. We fit all possible combinations of time-dependence or constancy over time. Note that since all we are interested in at first is model comparison using AIC, we could speed up computation by requesting the SE not to be computed (i.e., setting `se = FALSE`) and by providing "informed" initial values. For the latter, we could use those from a neighboring model. Note also that we may have to increase the value of the `maxit` control argument.

```
fm1 <- colext(psiformula = ~1,    # First-year occupancy
gammaformula = ~ 1,               # Colonization
epsilonformula = ~ 1,             # Extinction
pformula = ~ 1,                   # Detection
data = umf)
system.time(summary(fm2 <- colext(psiformula = ~1, gammaformula = ~year-1,
  epsilonformula = ~1, pformula = ~1, data = umf,
  control = list(trace = TRUE, REPORT = 5), se = TRUE)))
system.time(summary(fm3 <- colext(psiformula = ~1, gammaformula = ~1,
  epsilonformula = ~year-1, pformula = ~1, data = umf,
  control = list(trace = TRUE, REPORT = 5), se = TRUE)))
```

```
system.time(summary(fm4 <- colext(psiformula = ~1, gammaformula = ~1,
  epsilonformula = ~1, pformula = ~ year-1, data = umf,
  control = list(trace = TRUE, REPORT = 5), se = TRUE)))
system.time(summary(fm5 <- colext(psiformula = ~1, gammaformula = ~year-1,
  epsilonformula = ~year-1, pformula = ~1, data = umf,
  control = list(trace = TRUE, REPORT = 5), se = TRUE)))
system.time(summary(fm6 <- colext(psiformula = ~1, gammaformula = ~year-1,
  epsilonformula = ~1, pformula = ~year-1, data = umf,
  control = list(trace = TRUE, REPORT = 5, maxit = 500), se = TRUE)))
system.time(summary(fm7 <- colext(psiformula = ~1, gammaformula = ~1,
  epsilonformula = ~year-1, pformula = ~year-1, data = umf,
  control = list(trace = TRUE, REPORT = 5), se = TRUE)))
system.time(summary(fm8 <- colext(psiformula = ~1, gammaformula = ~year-1,
  epsilonformula = ~year-1, pformula = ~year-1, data = umf,
  control = list(trace = TRUE, REPORT = 5), se = TRUE)))

# Generate a fit list and compare the models using AIC (and check out some German)
models <- fitList(
'psi(.)gam(.)eps(.)p(.)' = fm1,
'psi(.)gam(Y)eps(.)p(.)' = fm2,
'psi(.)gam(.)eps(Y)p(.)' = fm3,
'psi(.)gam(.)eps(.)p(Y)' = fm4,
'psi(.)gam(Y)eps(Y)p(.)' = fm5,
'psi(.)gam(Y)eps(.)p(Y)' = fm6,
'psi(.)gam(.)eps(Y)p(Y)' = fm7,
'psi(.)gam(Y)eps(Y)p(Y)' = fm8)

(ms <- modSel(models))
                      nPars     AIC  delta   AICwt cumltvWt
psi(.)gam(Y)eps(Y)p(Y)    29 5365.38   0.00 1.0e+00     1.00
psi(.)gam(Y)eps(.)p(Y)    21 5385.80  20.42 3.7e-05     1.00
psi(.)gam(.)eps(Y)p(Y)    21 5388.12  22.73 1.2e-05     1.00
psi(.)gam(Y)eps(Y)p(.)    20 5408.31  42.92 4.8e-10     1.00
psi(.)gam(.)eps(.)p(Y)    13 5413.76  48.38 3.1e-11     1.00
psi(.)gam(Y)eps(.)p(.)    12 5450.96  85.58 2.6e-19     1.00
psi(.)gam(.)eps(Y)p(.)    12 5463.51  98.12 4.9e-22     1.00
psi(.)gam(.)eps(.)p(.)     4 5496.68 131.29 3.1e-29     1.00
Warnmeldung:
In sqrt(diag(vcov(x, altNames = TRUE))) : NaNs wurden erzeugt
```

In this case, AIC did select the data-generating model with full time variation, but with other RNG seeds, it may not. Three times, a Hessian matrix was singular (this does not happen every time and obviously again depends on the data set generated), so some NaNs were generated when trying to invert the Hessian and not all standard errors are available.

Now before we look at predictions and parameter estimates, we want to fit the model in JAGS and then compare the estimates. We conduct the analysis using code from Royle and Kéry (2007), which includes estimation of the actual number of territories occupied (among the 250, which as a proportion is the finite-sample or finite-population occupancy, as opposed to the population parameter), the occupancy-based population growth rate, and the turnover rate; see that paper for a definition of these terms and more info. We only fit the fully time-dependent model for an illustration.

```
# Bundle and summarize data
str(bdata <- list(y = data$y, nsites = data$nsites, nsurveys = data$nsurveys,
  nyears = data$nyears))

# Specify model in BUGS language
cat(file = "dynocc.txt","    # overwrite previous file
model {

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

# Ecological submodel: Define state conditional on parameters
for (i in 1:nsites){
  z[i,1] ~ dbern(psi1)
  for (t in 2:nyears){
    z[i,t] ~ dbern(z[i,t-1]*phi[t-1] + (1-z[i,t-1])*gamma[t-1])
  }
}

# Observation model
for (i in 1:nsites){
  for (j in 1:nsurveys){
    for (t in 1:nyears){
      y[i,j,t] ~ dbern(z[i,t]*p[t])
    }
  }
}

# Derived parameters
# Sample and population occupancy, growth rate and turnover
# Also, logit-scale params for direct comparison with unmarked
lpsi1 <- logit(psi1)
lp[1] <- logit(p[1])
psi[1] <- psi1
n.occ[1] <- sum(z[1:nsites,1])
for (t in 2:nyears){
  psi[t] <- psi[t-1]*phi[t-1] + (1-psi[t-1])*gamma[t-1]
  n.occ[t] <- sum(z[1:nsites,t])
  growthr[t-1] <- psi[t]/psi[t-1]
  turnover[t-1] <- (1 - psi[t-1]) * gamma[t-1]/psi[t]
  lgamma[t-1] <- logit(gamma[t-1])
  leps[t-1] <- logit(1-phi[t-1])
  lp[t] <- logit(p[t])
}
}
")
```

```
# Initial values
zst <- apply(data$y, c(1, 3), max) # Obs. occurrence as inits for z
inits <- function(){ list(z = zst)}

# Parameters monitored
params <- c("psi", "phi", "gamma", "p", "n.occ", "growthr", "turnover",
  "lpsi1", "lgamma", "leps", "lp")    # could add 'z'

# MCMC settings
na <- 1000  ;  ni <- 20000  ;  nt <- 10  ;  nb <- 10000  ;  nc <- 3

# Call JAGS (ART 3 min), check convergence and summarize posteriors
library("jagsUI")
out2 <- jags(bdata, inits, params, "dynocc.txt", n.adapt = na, n.chains = nc,
  n.thin = nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(3, 3))  ;  traceplot(out2)
print(out2, dig = 2)                # not shown
```

Next, we compare the true occupancy and its estimates from `unmarked` and JAGS. Note that estimates of occupancy probability in years $t > 1$ must be derived from the estimates of first-year occupancy and the two parameters governing the dynamics, extinction/persistence, and colonization. As for all derived quantities, computation of these estimates along with their uncertainty is trivially easy in an MCMC-based analysis. Although a little more complicated with MLE, `unmarked` does this automatically in two ways. First, the population-level estimates of occupancy probability $\psi_{t+1} = \psi_t(1-\varepsilon_t) + (1-\psi_t)\gamma_t$ are calculated and stored in the slot named `projected`. Slots can be accessed using the `@` operator, e.g., `fm@projected`. Sometimes, you may want to infer the proportion of the *sampled* sites that are occupied, rather than of the entire population of sites. These estimates are contained in the `smoothed` slot of the fitted model. Thus, the projected values are estimates of population parameters, and the smoothed estimates are of the finite-sample quantities. Discussion of the differences can be found in Weir et al. (2009). Bootstrap methods can be used to compute standard errors of derived parameter estimates. Here we employ a nonparametric bootstrap to obtain standard errors of the smoothed or projected estimates of occupancy probability during each year (see also Section 4.9.5 where we program a nonparametric bootstrap "by hand").

```
system.time(fm8 <- nonparboot(fm8, B = 500) ) # Takes about 3 hours
cbind(psi = data$mean.psi, smoothed = smoothed(fm8)[2,],
  SE = fm8@smoothed.mean.bsse[2,])            # Finite sample occupancy
cbind(psi = data$mean.psi, projected = projected(fm8)[2,],
  SE = fm8@projected.mean.bsse[2,])           # Population occupancy
```

The next table compares Truth for population occupancy with the estimates from `unmarked` and JAGS (along with their asymptotic standard error and the posterior sd, respectively). In JAGS, the population estimates are called `psi` while for the finite-sample proportion of occupied sites, we could either divide `n.occ` (the estimated number of occupied sites) by the number of sites (250) outside of BUGS or else define this derived quantity in the BUGS model.

```
round(cbind(psi = data$mean.psi, ML.estimates = projected(fm8)[2,],
  ML.ASE = fm8@projected.mean.bsse[2,], Bayesian.estimates = out2$summary[1:10,
  c(1:2)]), 4)

          psi  ML.estimates  ML.ASE    mean      sd
Year1  0.6000         0.6272  0.0536  0.5790  0.0561
Year2  0.6192         0.6262  0.0406  0.6007  0.0420
Year3  0.6371         0.5905  0.0890  0.5272  0.0748
Year4  0.5209         0.4812  0.0827  0.5028  0.0712
Year5  0.4750         0.3359  0.0603  0.3471  0.0528
Year6  0.3698         0.3959  0.0383  0.4007  0.0386
Year7  0.4939         0.4541  0.0635  0.4553  0.0601
Year8  0.4244         0.4120  0.0576  0.4304  0.0580
Year9  0.4245         0.4000  0.0343  0.3989  0.0333
Year10 0.2922         0.3052  0.0459  0.3172  0.0479
```

Fig. 4.5 shows that MLEs agree well with Bayesian estimates and that the model was able to correct the observed data for the effects of imperfect detection. In particular, the dynocc model reveals that the apparent stability of the population was a mere artifact caused by an increase in detection probability and that the population was in reality in a decline.

Next, we also want to compare the classical and the Bayesian estimates of the time-dependent parameters (phi/eps, gamma, *p*). We need to compute the estimates on the probability scale in `unmarked`, where naturally, all estimates are on the (logit) link scale. Backtransforming estimates when covariates, such as year, are present involves an extra step. Specifically, we need to tell `unmarked` the values of our covariate at which we want an estimate. This can be done using `backTransform` in combination with `linearComb`, although it may be easier to use `predict` which allows the user to supply a data frame in which each row represents a combination of covariate values of interest. Below, we create data frames called `nd` with each row representing a year. Then we request yearly estimates of the probability of extinction, colonization, and detection, and compare them to "truth," i.e., the values with which we simulated the data set. *Importantly, note that there are*

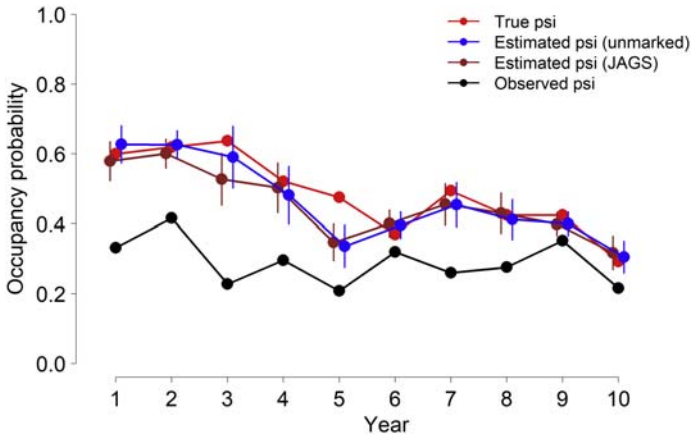

FIGURE 4.5

Comparison of the true and observed occupancy and the estimates (with 1 SE range) under the dynamic site-occupancy model fit in `unmarked` and `JAGS`.

nyear − 1 extinction and colonization parameters in this case, so we must not include year "10" in nd. For an example of how to produce predictions of initial occupancy (`type='psi'`), see Section 4.9.5.

```
nd1 <- data.frame(year = c('01','02','03','04','05','06','07','08','09'))
nd2 <- data.frame(year = c('01','02','03','04','05','06','07','08','09','10'))
E.ext <- predict(fm8, type = 'ext', newdata = nd1)
E.col <- predict(fm8, type = 'col', newdata = nd1)
E.det <- predict(fm8, type = 'det', newdata = nd2)
```

We get predictions along with standard errors and 95% CIs, which can be used to create plots. We plot the true values, estimates from `unmarked`, and those from JAGS in the same plot (Fig. 4.6). Maximum likelihood and Bayesian estimates are numerically very similar most of the times and their 95% uncertainty intervals (usually) include the true parameter values. However, in three cases no SE and CI could be computed by `unmarked` due to numerical reasons. In addition, we see several boundary estimates in colonization and extinction probability from `unmarked`: with no or few

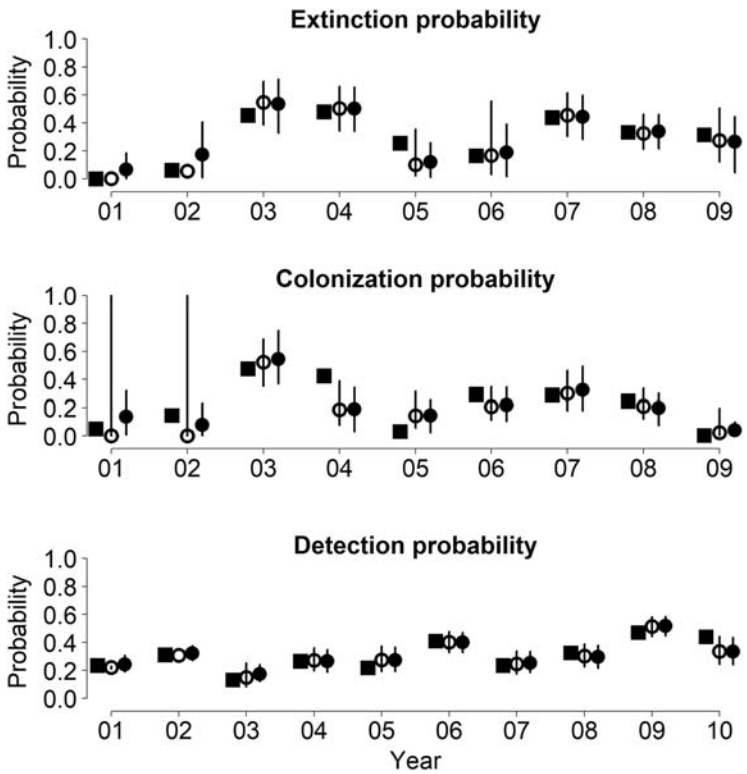

FIGURE 4.6

Comparison of the true values (black squares) with the estimated probabilities of extinction, colonization, and detection with the MLEs from `unmarked` (open circles) and the Bayesian estimates from JAGS (solid circles). Uncertainty intervals are 95% CIs and CRIs, respectively.

observed transitions the MLEs may become 0 or 1 and the CI infinite or very large (on the link scale). In contrast, the Bayesian posterior mean is never exactly on the boundary (only in the exceptional case when all the posterior mass is).

4.6 TREND ESTIMATION WITH OCCUPANCY DATA

Trend estimation is the *raison d'être* for most people in the biodiversity monitoring business. Now, the powerful dynocc model does not have a parameter representing a trend in occupancy. It doesn't even have parameters for occupancy probability for any year except the first. Hence, this model may not be ideal when you are simply interested in a trend of occupancy. We then need a simpler model where essentially we get rid of the parameters describing the dynamics and directly model occupancy probability for each primary period instead. These parameters can then be constrained by a linear or other model for the temporal patterns (e.g., we could also fit a spline of time; see Section 10.14 in AHM1). This model is a "static multi-season model" which treats seasons as strata.

In Chapter 2 we have used `unmarked` to fit models to multiyear data by "stacking," which assumes independence over the years. This may often be a sensible approximation, but we can use a nonparametric bootstrap to obtain SEs that are adjusted for dependency. In BUGS we can add zero-mean random year effects into a model with a trend in occupancy. We illustrate both with a simulated data set with a pronounced decline in occupancy (Fig. 4.7A).

```
# Pick colonization, extinction; compute equilibrium occupancy
gam <- 0.1
eps <- 0.2
( psi.eq <- gam / (gam + eps) )
[1] 0.3333333
```

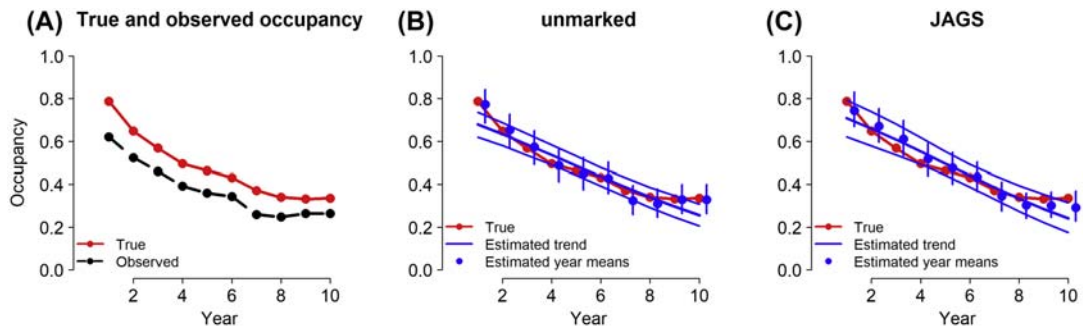

FIGURE 4.7

Trend analyses in a static multi-season occupancy model illustrated with a simulated data set. (A) True and observed occupancy. (B) Fitted values from two analyses of "stacked" yearly data in `unmarked` with year as a factor and with year as a continuous explanatory variable. (C) Fitted values from occupancy model with linear logistic trend and added random year effects. Uncertainty intervals in B are 95% CIs and 95% CRIs in C.

```
# Simulate a data set
set.seed(24)
data <- simDynocc(mean.psi1 = 0.8, range.phi = c(1-eps, 1-eps),
  range.gamma = c(gam, gam), range.p = c(0.4, 0.4)) # library(AHMbook)
```

We fit two occupancy models to the stacked data: first, treating Year as a factor and second as a continuous covariate, the coefficient of which is the trend.

```
# Stack detection histories
ystack <- array(NA, dim = c(data$nsites * data$nyears, data$nsurveys))
for(t in 1:data$nyears){
  ystack[((t-1)*data$nsites+1):(data$nsites*t),] <- data$y[,,t]
}

# Create year covariate and factor (both site covariates in unmarked)
year <- 1:data$nyears
yr <- rep(1:data$nyears, each = data$nsites)        # year as cont. covariate
yrfac <- as.factor(yr)                              # year as a factor

# Format and summarize data
library(unmarked)
summary(umf <- unmarkedFrameOccu(y = ystack, siteCovs = data.frame(yr = yr,
  yrfac = yrfac)) )                                 # require(unmarked)

# (1) Analysis of stacked data: treating year as a factor
summary(fm1 <- occu(~1 ~yrfac-1, data = umf))
nd <- data.frame(yrfac = as.factor(1:10))
pred.yrfac <- predict(fm1, type = "state", newdata = nd)

# (2) Analysis of stacked data: fitting a trend of year
summary(fm2 <- occu(~1 ~yr, data = umf))
nd <- data.frame(yr = 1:10)
pred.yr <- predict(fm2, type = "state", newdata = nd)
```

Not surprisingly, predictions from the first occupancy model follow the known truth more closely. The second model yields a useful and simple characterization of the temporal occupancy pattern in the form of a straight line on the logit scale (which will translate to a sigmoidal curve on the probability scale), i.e., a trend (Fig. 4.7B).

A nonparametric bootstrap to account for non-independence in the stacked-data analysis is coded as follows:

- We randomly draw a sample of *n* sites, *with replacement*, where *n* is the sample size: i.e., we create a new sample of identical size as that at hand, where some sites occur multiple times and others not at all.
- We do the stacking and fit the model of interest (we don't need the SEs) and save the estimates, predictions, or whatever we are interested in.
- We repeat this a large number of times (e.g., 1000), and use the sample SD of the parameters or other estimands (e.g., predictions) as the bootstrapped SEs and percentiles as bootstrapped CIs.

```
# Create array to hold predictions
simrep <- 1000
bs.pred <- array(NA, dim = c(nrow(pred.yr), simrep))

# Nonparametric bootstrap for prediction of trend line (ART 2 min)
for(b in 1:simrep){
  cat(paste("\n** Bootstrap rep", b, "**"))
  # Sample sites with replacement (get their index)
  bs.samp <- sample(1:250, 250, replace = TRUE)
  # Repeat stacking with bootstrap sample
  ystack <- array(NA, dim = c(250 * 10, 3))
  for(t in 1:data$nyears){
    ystack[((t-1)*data$nsites+1):(data$nsites*t),] <- data$y[bs.samp,,t]
  }
  # Create unmarked data frame, fit model and predict trend line
  umf <- unmarkedFrameOccu(y = ystack, siteCovs = data.frame(yr = yr))
  fm <- occu(~1 ~ yr, data = umf, se = FALSE)
  tmp <- predict(fm, type = "state", newdata = data.frame(yr = 1:10))
  # Save param estimates
  bs.pred[,b] <- tmp[,1]
}

# Get bootstrap SE and CI for all annual predictions
se.bs <- apply(bs.pred, 1, sd)
ci.bs <- t(apply(bs.pred, 1, function(x)quantile(x, c(0.025, 0.975))))

# Compare asymptotic and bootstrapped SEs and CIs
round(cbind('ASE' = pred.yr[,2], 'Asymp_LCL' = pred.yr[,3],
  'Asymp_UCL' = pred.yr[,4], 'Bootstrapped SE' = se.bs, 'Bootstrapped CI' = ci.bs), 3)

       ASE Asymp_LCL Asymp_UCL Bootstrapped SE  2.5% 97.5%
 [1,] 0.023     0.632     0.722           0.030 0.620 0.735
 [2,] 0.021     0.591     0.673           0.028 0.579 0.686
 [3,] 0.019     0.548     0.622           0.026 0.535 0.635
 [4,] 0.017     0.503     0.568           0.024 0.490 0.583
 [5,] 0.015     0.456     0.515           0.023 0.443 0.531
 [6,] 0.014     0.408     0.464           0.022 0.393 0.477
 [7,] 0.015     0.358     0.417           0.023 0.342 0.432
 [8,] 0.016     0.310     0.373           0.024 0.292 0.386
 [9,] 0.017     0.264     0.332           0.025 0.247 0.347
[10,] 0.018     0.222     0.295           0.026 0.206 0.308
```

We produce a sample of size 1000 in only about 2 min and see that in this example, the stacked-data analysis doesn't underestimate much the uncertainty (though this may be different in other examples). Next, we use JAGS to fit a trend model with random yearly variation around that trend, to accommodate possible non-independence of the data over time. This model is not directly comparable to the year-as-a-factor model in `unmarked`, which fits year as a fixed-effects factor. Here now, we treat year as a random-effects factor in a model that already contains a linear trend, so year effects will be shrunk toward the trend line. Comparison of Fig. 4.7B and C shows similar, though not identical, estimates of year-specific occupancy and of the trend in occupancy.

```
# Bundle and summarize data
str(bdata <- list(y = data$y, nsites = dim(data$y)[1],
  nsurveys = dim(data$y)[2], nyears = dim(data$y)[3]))

# Specify model in BUGS language
cat(file = "occ.txt","
model {

# Specify priors and specify model for occupancy
for (t in 1:nyears){
  logit(psi[t]) <- alpha + beta.trend * (t-5.5) + eps.year[t]
  p[t] ~ dunif(0, 1)
  eps.year[t] ~ dnorm(0, tau.lpsi)
}
alpha <- logit(mean.psi)
mean.psi ~ dunif(0,1)
beta.trend ~ dnorm(0, 0.01)
tau.lpsi <- pow(sd.lpsi, -2)
sd.lpsi ~ dunif(0, 10)

# Ecological submodel: Define state conditional on parameters
for (i in 1:nsites){
  for (t in 1:nyears){
    z[i,t] ~ dbern(psi[t])
    # Observation model
    for (j in 1:nsurveys){
      y[i,j,t] ~ dbern(z[i,t]*p[t])
    }
  }
}

# Derived parameters
for (t in 1:nyears){
  n.occ[t] <- sum(z[1:nsites,t])   # Finite sample occupancy
  logit(psi.trend[t]) <- alpha + beta.trend * (t-5.5)
}
}
")

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

# Parameters monitored
params <- c("psi", "psi.trend", "mean.psi", "alpha", "beta.trend",
  "sd.lpsi", "p", "n.occ")

# MCMC settings
na <- 1000  ;  ni <- 6000   ;   nt <- 1   ;   nb <- 2000   ;   nc <- 3

# Call JAGS (ART 2 min), check convergence and summarize posteriors
out <- jags(bdata, inits, params, "occ.txt", n.adapt = na, n.chains = nc,
  n.thin = nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(3,3))  ;  traceplot(out)
print(out, dig = 2)               # not shown
```

Occupancy models with trends are often useful in biodiversity monitoring applications and you can fit a wide range of trend models as part of an occupancy model. For instance, you could fit polynomial terms or splines to accommodate curvilinear relationships, as in a GAM (Fewster et al., 2000), or specify a random walk as another method of smoothing (see Section 4.11). We close by noting that as an alternative, Weir et al. (2009) and van Strien et al. (2010, 2013) fit a dynamic model first, and then compute the least-squares linear regression solutions from the estimates of annual occupancy. Doing this for each MCMC sample achieves a proper error propagation and correct uncertainty assessment for the slope parameter that represents the occupancy-based trend.

4.7 STUDY DESIGN, BIAS, AND PRECISION OF ESTIMATORS

Here we illustrate, in the context of the dynocc model, some of the many useful things you can do with simulated data sets. In the first example, we explore the quality of estimates when the dynocc model is applied to unreplicated data, i.e., "single-visit data." Whether or not, and under which conditions, hierarchical models for abundance and occurrence are estimable for such data has been subject for some debate, which has focused mostly on static models (Lele et al., 2012, Sólymos et al., 2012, Dorazio, 2012, 2014, Knape and Korner-Nievergelt, 2015, Sólymos and Lele, 2016). In dynamic models, there is additional information from the among-season replication, which is exploited in a Markovian model such as the Dail-Madsen (see Chapters 1 and 2; also see Bellier et al., 2016). We are not aware of studies that have formally examined estimability of the dynocc model with single replicates (but see Dail and Madsen, 2013, and Peach et al., 2017). Here, we use simulation to explore this topic. In the second example, we emphasize that estimates under the model are of variable quality depending on sample size and the values of detection probability (see also McKann et al., 2013). In the third example, we conduct a power analysis to identify the optimal design for a given target in the context of trend estimation using an occupancy model. In the fourth example (and also in Section 4.11.1), we explore the effects of unmodeled site-level detection heterogeneity on the dynocc estimators. You find all the code for this section on the website; here we only summarize the results.

4.7.1 CAN WE FIT DYNOCC MODELS TO SINGLE-VISIT DATA?

To assess parameter estimability, we repeatedly generated a data set and fitted the model to each replicated data set, and then compared the estimates of each replicate with the known truth. If the model parameters are estimable, then the estimates should cluster around the true values. We did this with three models: the fully time-dependent model, the model with all parameters constant, and a model where we have a single continuous covariate affecting detection probability.

We started as a first case with the fully time-dependent model. To make the estimation task a little harder, but also more alike to real-life, the analysis model did not quite match the data simulation model: we added random noise into detection probability both at the site and the site-by-survey levels, using the last two arguments in `simDynocc`. We ran 100 simulation replicates where; for each data set, we randomly drew a value for all four parameters from Uniform distributions on the range 0.01−0.99. As a second case, we examined the least parameter-rich dynocc model, which has an intercept only for each of the four parameters. Since we only got one estimate for each parameter and simulation replicate instead of 10 previously (because we have 10 years), we simulated 1000 data sets instead of only 100. In our third case, we explored a scenario where a single covariate affects detection

probability. As for the first case, all parameters were fully year-dependent, but now there was additional random noise in p stemming from two components, and in addition there was a site covariate affecting detection probability, which we also used in model fitting. We simulated 100 data sets.

In Fig. 4.8, we compare truth and estimates for occupancy, colonization, and detection probabilities for all three cases. Estimates under the fully time-dependent model are not strongly biased, since the

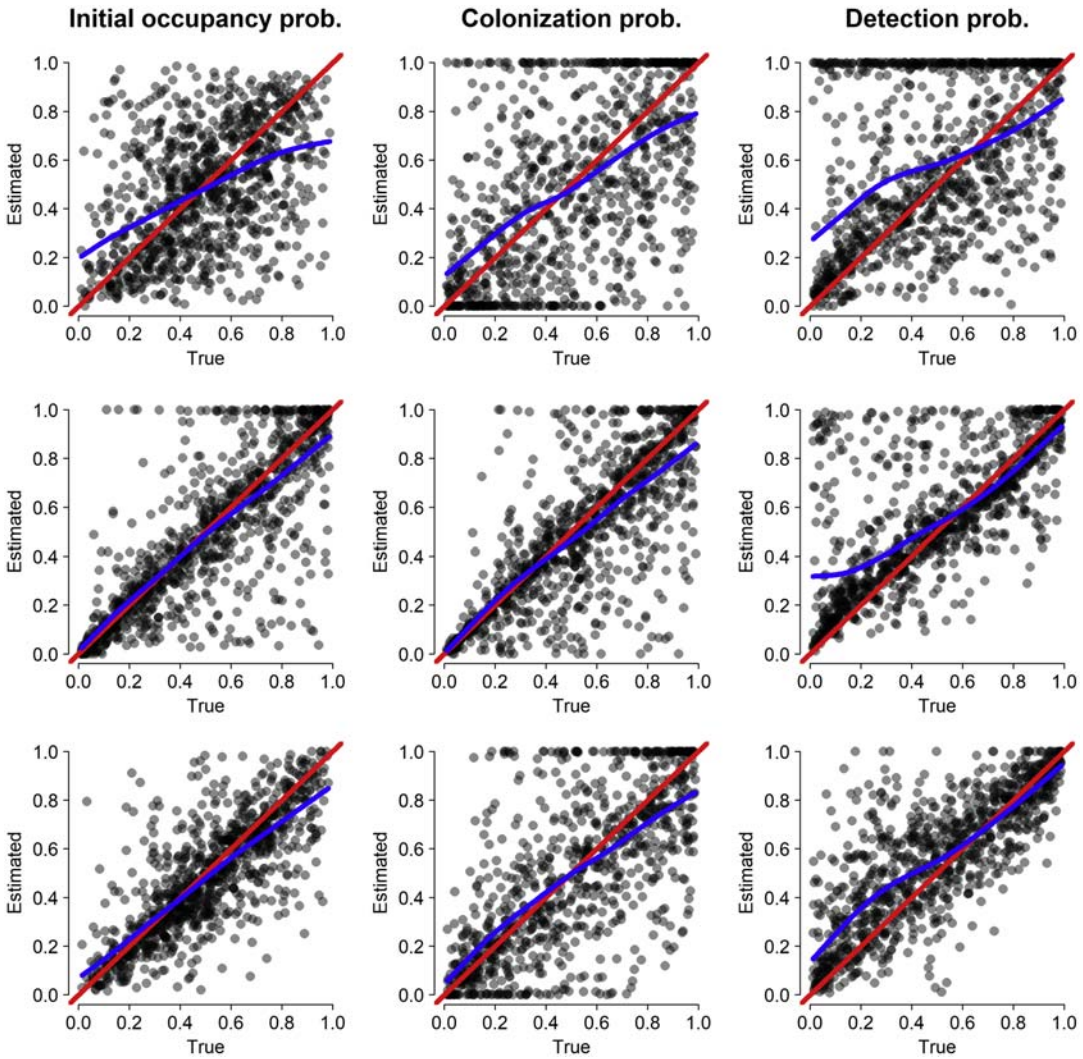

FIGURE 4.8

Comparison of estimates and true values of probabilities of occupancy (psi), colonization (gamma), and detection (p) under three different dynocc models fit to single-visit data sets. The first model (top row) has full time-dependence in gamma, extinction (eps), and p; the second model (middle row) is the intercepts-only model; while the third model (bottom row) is the same as the first, but in addition has a single site covariate in detection. The 1:1 line is shown in red and the blue line is a spline fit; coincidence of the two suggests unbiasedness.

blue smoother lies mostly in the vicinity of the 1:1 line. However, estimates are so imprecise as to be practically useless. In addition, we see a high incidence of boundary estimates. In contrast, estimates under the simpler intercepts-only model have substantially reduced bias, are much more precise, and have much fewer boundary estimates. Finally, if we can leverage additional information stemming from a single site-covariate in the detection model, we again get improved estimates in terms of bias and precision and hardly see any boundary estimates. Thus, the absence of temporal variation in the parameters and the availability of a covariate that is informative about detection both take us from a situation of almost useless estimates to much more useful estimates.

We do *not* want to encourage uncritical application of the dynocc model to single-visit data. However, we want to illustrate that with a little additional information the model may yield fairly decent estimates in terms of bias and precision even in this case. Presumably, estimates will become better when more information is introduced, e.g., with additional covariates (Peach et al., 2017). However, we imagine that these covariates must be "private," i.e., not shared between detection and a parameter in the ecological submodel (Lele et al., 2012; Sólymos et al., 2012; Dorazio, 2014; see also Section 10.6). Note also the model by Peach et al. (2017) which ensures estimability of a single-visit model by imposing a parametric function on the aggregate detection probability (from multiple visits) and the known number of visits to a site. Hepler et al. (2018) develop a spatio-temporal occupancy model and show that it can be fit to single-visit data as long as the number of seasons is sufficiently large (e.g., >30). Estimability is due to the sharing of information both within the local neighbourhood and over time.

4.7.2 BIAS AND PRECISION AS A FUNCTION OF `nsites`, `nsurveys`, and `p`

Sample size and detection probability are major factors that affect the quality of the estimates in any kind of capture-recapture model, including occupancy models. Frequently we hear the question: "What is the minimal sample size … or detection probability … that still guarantees decent estimates?"—Our reply to this is: "*Simulation is the answer!*" Here, we show how trivially easy it is to run a simulation with varying values of the number of sites and surveys and of detection probability and then keep track of estimator bias and precision. For a range of combinations of these factors, we simulated a large number of data sets with known data-generating parameters and then compared the resulting estimates with this known truth. Similar to Section 10.7 (in AHM1) for the static occupancy model, we studied estimator quality in a design that varies the number of sites (`nsites` = 20, 120, or 250) and of surveys (`nsurveys` = 2, 5, or 10) and the magnitude of detection probability (covering almost the entire range between 0 and 1). We assumed initial occupancy to be equal to 0.6 and annual values of phi and gamma to vary between 0.7 and 0.9 and 0.1 and 0.5, respectively. For each combination of the two sample size dimensions (sites, surveys) we repeated the following 100 times: (1) randomly pick (and save) a value for detection probability from a Uniform(0.01, 0.99), (2) use `simDynocc` to generate a data set, and (3) estimate parameters using MLE with `unmarked` and save the estimates. We don't need the SEs so we didn't compute them, to avoid the simulation from breaking whenever a Hessian becomes singular, but we did save the 10 projected values of population occupancy.

Fig. 4.9 visualizes the simulation results for the nine scenarios that combine three levels each of the site (`nsites` = 20, 120, or 250) and the survey factor (`nsurveys` = 2, 5, or 10) and for the whole range of values of p between 0.01 and 0.99. We see substantial variation in the quality of the estimates under the dynocc model. We see hardly any bias (the blue smoother coincides with the red line most of the time), but with small sample size (in terms of the number of sites M or visits J), the precision of the

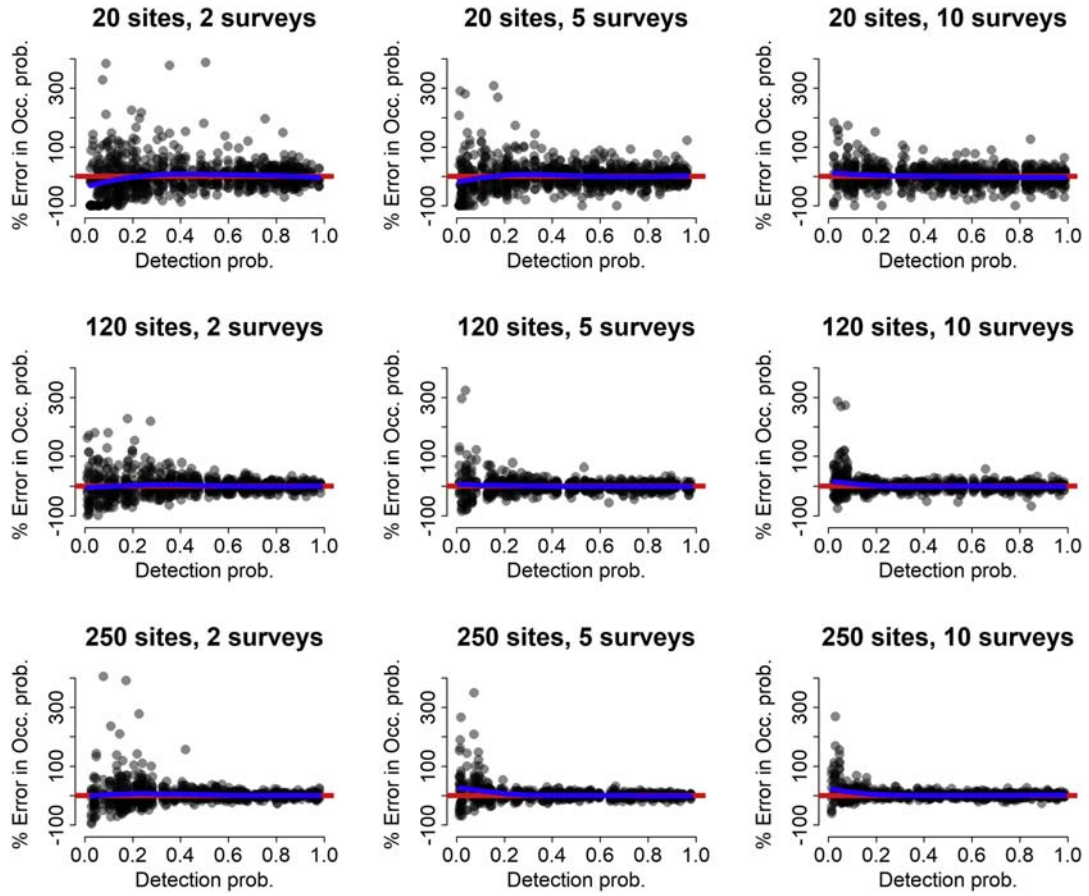

FIGURE 4.9

Percent error in population occupancy estimates as a function of number of sites, number of surveys per season, and detection probability (choice of y-axis leaves out up to six even more extreme cases per plot). Red line shows absence of error and the blue line is a spline smoother to show the average behavior of the estimator for a given value of p. Sample size is 100 simulation replicates for each plot. Coincidence of red and blue lines suggests absence of bias.

estimates can be fairly low. Moreover, we clearly see the "First Law of Capture-Recapture": the quality of the estimators becomes bad when p is low for any combination of M and J. However, interestingly, this only applies to the precision, but we don't see the positive bias in the occupancy estimator of the dynamic occupancy model when p is less than about 0.1 or 0.2, unlike what is known for the static occupancy model (see MacKenzie et al. (2002) and Guillera-Arroita et al. (2010) and compare Fig. 10−7 in AHM1 and Fig. 4.9). Hence, it seems that the Markovian structure of the dynamic model allows to reduce the uncertainty about occupancy in this case. We note that we could conduct analogous assessments also for the other parameters in the model, especially colonization and persistence.

4.7.3 A POWER ANALYSIS FOR OCCUPANCY TREND ESTIMATION

Power analysis is a classic use of data simulation: we build a data set with a certain effect included (e.g., a time trend in occupancy), fit the model and record the p-value of a test for that effect, and repeat this a large number of times. Power is given by the proportion of times that the test yields a significant result. Here, we provide an illustration with an occupancy-based trend, where we simulated metapopulations with a decline of 20%, in terms of the proportion of occupied sites, over 10 years. We varied the number of sites and surveys to see which gave the highest power to detect the decline. In addition, we also varied detection probability p for each data set. We wanted to identify the optimal allocation of visits to sites and replicates, for a constant duration of 10 years, such that we are most likely to detect the trend of the magnitude that we built into the data. Our design is factorial, and also lets us compare four different types of allocation for a constant number of 500 visits (i.e., $250 * 2 = 100 * 5 = 50 * 10 = 250 * 2 = 500$).

Here we summarize the results only. We see that the power to detect a 20% decline in 10 years, using this occupancy design, varies between about 17% and 80% and increases both with more sites and more visits, but faster by increasing the number of sites. In particular, when comparing different allocations to sites and visits for a total of 500 visits, the power more than doubles from 20 sites and 25 surveys to 250 sites and 2 surveys.

```
power
             nsurveys = 2   nsurveys = 5   nsurveys = 10   nsurveys = 25
nsites = 20         0.173          0.219           0.233           0.247
nsites = 50         0.218          0.298           0.361           0.378
nsites = 100        0.317          0.427           0.469           0.523
nsites = 250        0.563          0.687           0.752           0.795
```

4.7.4 EFFECTS OF UNMODELED DETECTION HETEROGENEITY

The "Second Law of Capture-Recapture" states that unmodeled individual detection heterogeneity causes negative bias in abundance estimators. In occupancy analyses, the site corresponds to the individual, and occupancy (strictly, the number of occupied sites) is analogous to abundance. Hence, we expect similar bias when detection probability varies among sites in a fashion that is unaccounted for by the model, e.g., via covariates (Royle, 2006). Indeed, Miller et al. (2015) have found in a field experiment that uncontrolled detection heterogeneity was the major biasing factor in occupancy modeling. To study the effects of unmodeled detection heterogeneity, we simulated data sets where detection was year-specific and, in addition, contained site-level random (logit-)Normal noise. Then, we fitted the standard model with year-dependent detection that did not account for this noise. We repeated this for three levels of heterogeneity (SD of the random effect: 0.2, 1, and 2; see Fig. 4.10) and 100 data sets each. We studied the effects of these three levels of site-specific detection heterogeneity on all four main parameters of the model by conducting a simulation with 250 sites, 8 years, and 3 surveys. We note that SD = 1 and 2 represent very strong to extreme levels of heterogeneity and hence, that the results shown here are best seen as worst-case scenarios.

Perhaps somewhat surprisingly given the results of Miller et al. (2015), all four estimators were fairly robust to small to moderate levels of site-level detection heterogeneity (left column in Fig. 4.11). However, large (middle) or extreme levels (right) of site-level detection heterogeneity produced a negative bias in occupancy and colonization probability and a positive bias in extinction and detection probability, where the magnitude of the bias depended also on the magnitude of p itself. It would be trivial to investigate the effects of the number of sites on these patterns, e.g., by simulating only 30 sites as did McKann et al. (2013). Going from left (small detection heterogeneity) to right (larger detection heterogeneity), the negative bias in the occupancy estimator increased. Hence, we can hypothesize that

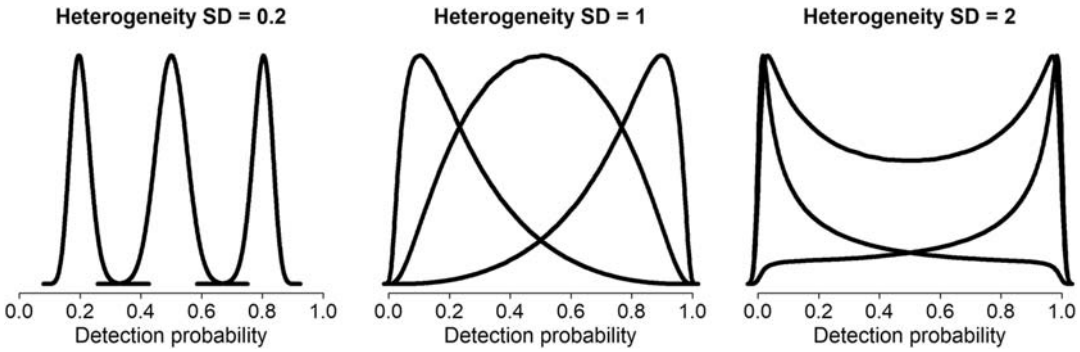

FIGURE 4.10

Illustration of the effects of the three magnitudes of random logit-Normal variability in site-level detection probability (p) in the simulation. In each year, a value of p is chosen randomly on the interval (0.01, 0.99) and then additional noise is added. As a visualization of the magnitude of heterogeneity represented by the three levels of SD = 0.2, 1, or 2, we here show just three values of the basic p of 0.2, 0.5, and 0.8, to which logit-Normal noise of the given magnitude is then added on the logit scale, resulting in the distributions shown.

an unmodeled change over time in the magnitude of detection heterogeneity may lead to an increasingly strong negative bias in occupancy. This could lead to a spurious decline in occupancy, the masking or the understatement of the strength of an increase. We test this conjecture in Section 4.10.1.

4.8 GOODNESS-OF-FIT

To assess the goodness-of-fit (GoF) of a model we compare the observed data with the data that we would expect to see under our model using some fit statistic, or discrepancy measure, such as residuals, Chi-squared, Freeman-Tukey, or deviance. However, for binary-data models such as occupancy models, such standard fit statistics are a simple deterministic function of sample size and therefore uninformative about fit (see Section 4.4.5 in McCullagh and Nelder (1989) and Section 8.4.1.1 in Royle et al. (2014)). To test GoF in this case we must aggregate the binary responses in some way and it is unclear which aggregation scheme is best to indicate a particular assumption violation or how sensitive a test based on any such aggregation is. The MacKenzie and Bailey (2004) test aggregates the data by unique detection history (see Section 10.8 in AHM1) and compares observed and expected numbers of sites with a certain detection history by Chi-squared. It is implemented in `AICcmodavg` as function `mb.gof.test` for both static and dynamic occupancy models (Mazerolle, 2016). Other possible aggregations include row or column sums of the detection history matrix.

We illustrate two techniques for GoF assessment of the dynamic occupancy model. The first uses parametric bootstrapping and aggregation of the binary responses by capture history, i.e., it is an extension of the MB test to a dynamic occupancy model. The second is based on posterior predictive distributions in a Bayesian analysis, where a discrepancy measure is simultaneously computed for the actual data set and for a replicate (simulated) data set at every iteration of the MCMC algorithm. Inspired by Rankin et al. (2016), we decompose fit into a component that describes the open part of the model and another that describes the closed part of the model. We illustrate both with a simulated data set that contains a lot of structure which the analyzing model does not, by simulating a data set with effects of covariates and heterogeneities. We then fit a much simpler model and see whether we can pick up the resulting lack of fit.

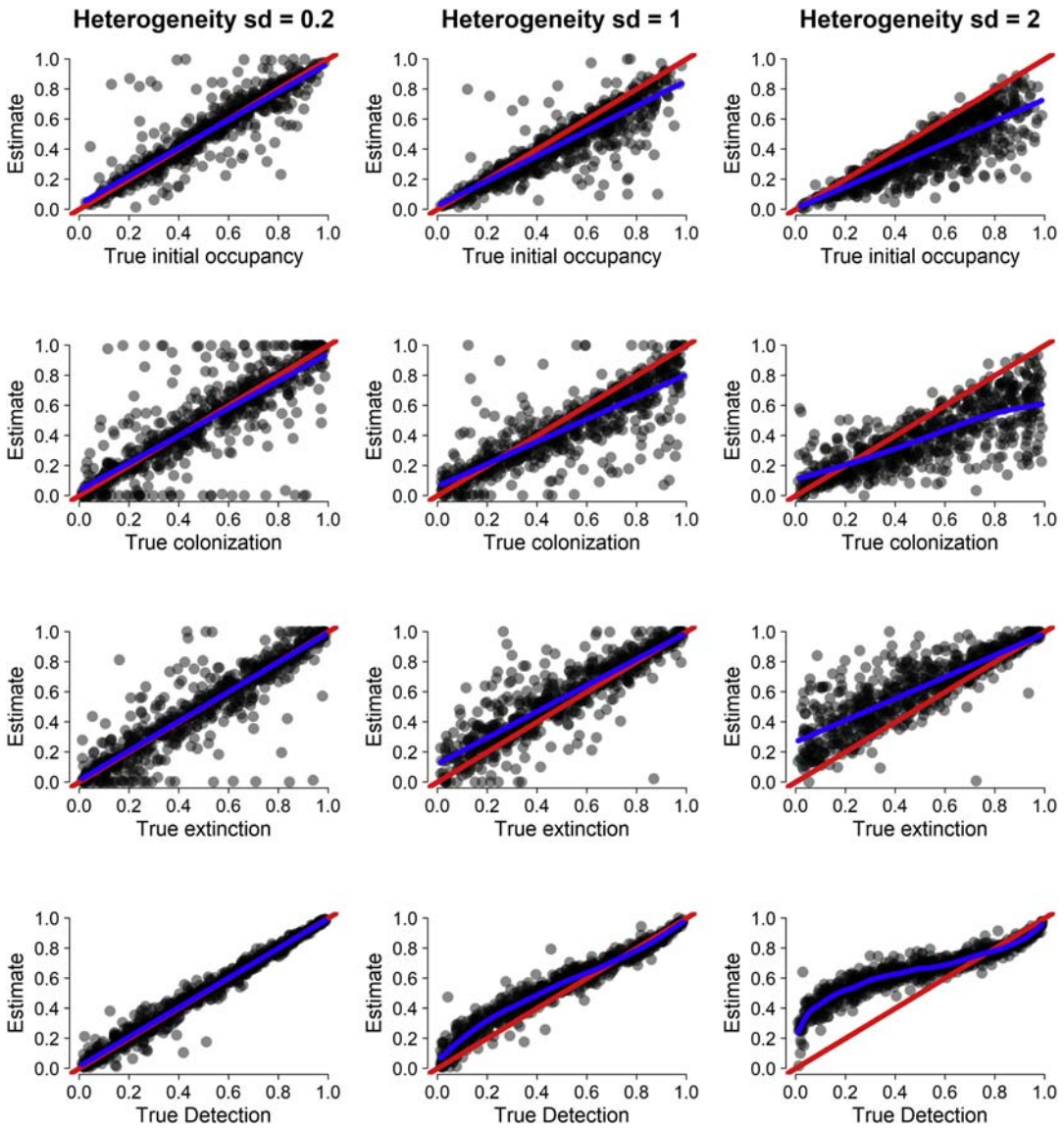

FIGURE 4.11

Scatterplots of MLEs versus the true parameter values in 100 data sets with site-level detection heterogeneity of magnitude 0.2 (small/moderate; left column), 1 (large; middle column), or 2 (extreme; right column). Red is the 1:1 line and blue is a smoothing spline. Coincidence of the two suggests absence of bias.

```
set.seed(68)
str(data <- simDynocc(nsites = 250, nyears = 10, nsurveys = 3,
  mean.psi1 = 0.4, beta.Xpsi1 = 1,
  range.phi = c(0.2, 1), beta.Xphi = 1,
  range.gamma = c(0, 0.8), beta.Xgamma = -1,
  range.p = c(0.1, 0.9), beta.Xp = 2,
  range.beta1.survey = c(0, 3), range.beta2.survey = c(-3, 0),
  trend.sd.site = c(0, 2), trend.sd.survey = c(0, 2)) )
```

Diagnosing lack of fit in an occupancy model remains difficult and we should perhaps not rely too much on just a single number, such as a GoF p-value. Residual plots against modeled or unmodeled covariates or spatial coordinates may be used to detect systematic patterns of departures of the data from the model and these could then be taken account of in an improved version of the model; see also the following important sources: Broms et al. (2016a), Warton et al. (2017), MacKenzie et al. (2018) and Wright et al. (2019).

4.8.1 MACKENZIE AND BAILEY GOODNESS-OF-FIT TEST FOR DYNOCC MODELS

We illustrate the extension of the test of MacKenzie and Bailey (2004) to the dynocc model, which is implemented in the `AICcmodavg` R package (Mazerolle, 2016), by fitting an ill-fitting model to the complex data set just simulated. In this example, we fit the intercepts-only model.

```
# Fit constant dynocc model
library(unmarked)                  # Load the package
yy <- matrix(data$y, nrow = data$nsites, ncol = data$nsurveys *
  data$nyears)
summary(umf <- unmarkedMultFrame(y = yy, numPrimary = data$nyears))
summary(fm <- colext(~1, ~ 1, ~ 1, ~ 1, data = umf))

# Compute Chi-square test statistic for actual data by season and
# generate reference distribution of test statistic under H0 (~1.1 h)
library(AICcmodavg)
system.time(gof <- mb.gof.test(fm, print.table = F, nsim = 1000,
  plot.hist = TRUE, plot.seasons = TRUE, report = 1) )
gof

Goodness-of-fit for dynamic occupancy model
Number of seasons:  10

Chi-square statistic:
 Season 1  Season 2  Season 3  Season 4  Season 5
  18.5134   73.8895   20.3665   54.8196   13.8125
 Season 6  Season 7  Season 8  Season 9 Season 10
  84.9100   67.5141   67.5075   96.9058   56.9778

Total chi-square = 555.2166
Number of bootstrap samples = 1000
P-value = 0

Quantiles of bootstrapped statistics:
  0%  25%  50%  75% 100%
  39   58   66   74  117

Estimate of c-hat = 8.37
```

We find out what we know already: that the constant model is not an adequate representation of the structure in the data and therefore, the model does not fit. The ratio of the value of the fit statistic for the

observed data and the mean of the values of the fit statistic for the 1000 simulated, "perfect" data sets, called c-hat, and which expresses the magnitude of the lack of fit, is large at 8.37.

If this were a real analysis, we would be faced with the question of what to do with a model that doesn't fit. The discussions in Sections 6.9 and 8.4.3 in AHM1 remain relevant here. Other than the extreme solution (to declare a data set as unanalyzable), there are several ways forward. The first would be to try and improve the model to explain a larger amount of the variation in the data, which would then bring down the value of c-hat. You could experiment here by fitting a progression of more complex models, which incorporate more and more of the stuff that we put into the simulated data set, to see how you can bring c-hat down, and therefore, how much you can approach a fitting model. (Hint: we also added random variability associated with sites and site-visits, and we can't address this with `colext` in `unmarked`, but we can with BUGS; see Section 4.10.) A second solution might be to filter your data to make them more homogeneous. Third might be to accept a certain lack of fit and assume that it is not due to a structural breakdown of the model but merely represents some unstructured extra noise (i.e., "overdispersion"). We could then account for the resulting additional uncertainty in the inferences by increasing the SEs and CIs and in AIC-based model selection by using QAIC rather than AIC (Burnham and Anderson, 2002). However, the usual rule of thumb is that if the value of c-hat is greater than, say, 3−4, we should not make this assumption, but rather not trust the model. In Section 4.9 you will see GoF testing and accommodation of a c-hat > 1 for a real data set.

4.8.2 GOF TESTS BASED ON BAYESIAN POSTERIOR PREDICTIVE DISTRIBUTIONS

In principle, we could try to program the aggregation by capture history as in the MB test, but this might be complicated inside of the BUGS program (it might be doable in R based on MCMC output from BUGS, perhaps using code snippets from `AICcmodavg`). Simpler forms of aggregations are row or column sums of the detection histories (see Section 10.8 in AHM1). Presumably, we would do this for each year separately and then obtain a fit statistic for each year, which could be summed for an overall GoF measure.

Rankin et al. (2016) developed a Bayesian GoF test for a robust-design capture-recapture model where they decompose fit into one component for the dynamics and another for the within-season fit. The observed data for the former are the m-array, while for the latter they are the detection frequencies and the number of unique individuals captured. Here, we apply their ideas to the dynocc model and construct tables for the observed and expected numbers of state transitions to gauge fit of the open part of the model and use the detection frequencies at individual sites to test the fit of the closed part of the model. For the open part, counting transitions requires a bit of BUGS coding. For each MCMC iteration, we create a replicate data set under the model and do the following for both the observed and the replicate data:

(1) Compute the observed presence/absence matrix `zobs` for each site and year.
(2) Identify the four possible transitions for each interval and count them for observed and replicate data sets.
(3) Compute the expected number of transitions under the model for each interval.
(4) Compute a Chi-squared (or other) discrepancy between the expected number of transitions for both observed and replicate data sets. To avoid division by zero, we add a small number (e) to the denominator in the Chi-squared.
(5) Add up the Chi-squared discrepancies over all transitions and years.
(6) For the closed part of the GoF test, we compute the detection frequencies, i.e., the number of times that the species was detected per site and season (i.e., i,t). We employ a Chi-squared discrepancy for both and, for comparison, also a Freeman-Tukey for the closed part of the GoF test.

We illustrate with a model with all covariate effects in, but without the year effects, and carry out computations for both open and closed tests in one model. The code becomes fairly unwieldy and is shown on the website only; here we simply show the results.

We found that the lack of fit was greater in the open than in the closed parts of the model and, moreover, in the latter, a Chi-squared discrepancy appeared to be more sensitive than a Freeman-Tukey in our case (Fig. 4.12). The magnitude of the ratios of these fit statistics for the actual and the expected (under H0) data agreed qualitatively with those of the Bayesian p-value and were 2.38, 1.12, and 1.04, in the order of the figure. To get more intuition about where the lack of fit may come from in the open model part, we compared the observed and the expected number of transitions for each interval (which we computed in the model; note that here the table of expected number of transitions is constant over years because of the constant parameters in our model). Here is just one example for the first interval.

```
*** Interval 1  ***
* Observed transitions*
             To Non-occ To Occ
From Non-occ         72     83
From Occ             56     39

* Replicate data transitions*
             To Non-occ   To Occ
From Non-occ   83.29733 74.59733
From Occ       57.13000 34.97533

* Expected transitions*
             To Non-occ   To Occ
From Non-occ   68.87431 66.69349
From Occ       61.81234 52.61985
[ .... truncated output ...]
```

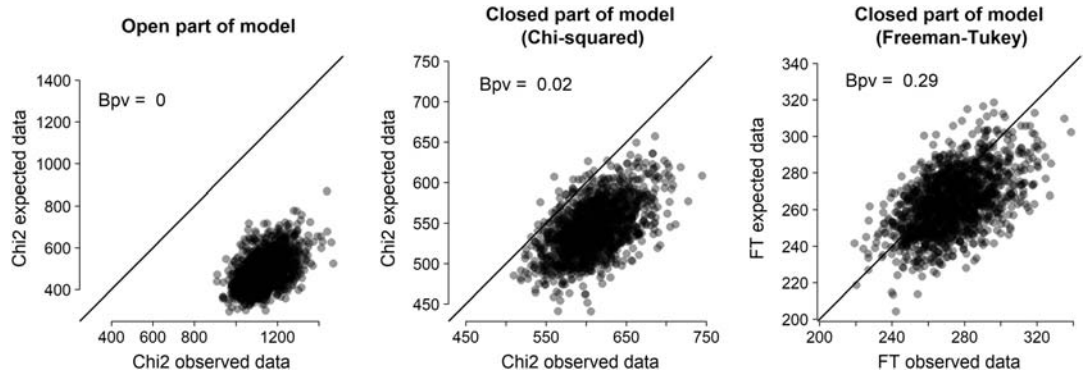

FIGURE 4.12

Posterior predictive checks of goodness-of-fit of the open and the closed parts of the dynocc model using a Chi-squared and a Freeman-Tukey discrepancy measure. The Bayesian p-value (Bpv) is the proportion of points above the 1:1 line and represents the probability to obtain, under the null hypothesis of a fitting model, a more extreme (i.e., larger) value of a test statistic.

In a Bayesian or a frequentist analysis, inspecting observed and expected counts may help give ideas about how the model could be improved to enhance its fit. As a second option, we may account for additional uncertainty by adding site and/or time-specific random effects for some/all parameters. This will naturally inflate the uncertainty around estimates and produce a qualitatively similar result as the frequentist method of inflating SEs using an estimate of an "overdispersion factor."

It would now be easy for you to conduct a simulation study to gauge the power of such GoF tests using these or other test statistics to selected forms of assumption violations and for variable values of the sample size settings `nsites`, `nsurveys`, and `nyears`. Indeed, more studies of GoF methods in hierarchical models such as the dynocc would be welcome (Conn et al., 2018).

4.9 ANALYSIS AND MAPPING OF CROSSBILL DISTRIBUTION AND RANGE DYNAMICS IN SWITZERLAND

Throughout this book we emphasize that any model of distribution and abundance with spatially indexed covariates can be used as a species distribution model (SDM). That is, if you know the values of the spatial covariates in the model over some larger area, then you can form predictions for this entire area and draw maps. In AHM1, we have demonstrated this for abundance and density in Chapters 6–9, for occupancy in Chapter 10 and for species richness in Chapter 11. Here now, we use data from the European Crossbill (*Loxia curvirostra*) to produce Swiss distribution maps for this species during the years 2001–2012. In addition, we emphasize how we can just as well produce predictive maps of the dynamic rate parameters that underlie distribution change, i.e., colonization and extinction, or even of the observation error in the surveys (i.e., 1-p). Hence, as we have seen for Willow Warbler survival in Chapter 3, we can easily produce predictive maps of modeled population quantities other than distribution or abundance.

We keep the usual workflow in our analyses of hierarchical models using `unmarked`. Our analysis is similar to that in Kéry et al. (2013) but differs slightly in the range of years used and the strategy of model selection. We do this:

- Fit a large, "global" model with all potentially interesting or important effects. We approach this model in incremental, small steps, as a precaution to the identification of local optima in the likelihood.
- Run a parametric bootstrap GoF of the global model and compute c-hat.
- Use stepwise backwards elimination, based on QAIC, to identify the best model for prediction.
- Predict the probabilities of occupancy, colonization, extinction, and detection, as a function of one or more of the following explanatory variables: year, elevation, forest cover, and date. In particular, we produce predictive maps of all these quantities for Switzerland.

The actual code for the analyses is rather lengthy, and we only provide a fairly brief summary. On the book website we offer the full code underlying this section. This will enable you to replicate all analyses in this section.

4.9.1 DATA MANIPULATIONS AND CREATION OF `unmarked` DATA FRAME

We load the crossbill data, which contains detection/nondetection data for 267 1-km^2 quadrats in the Swiss breeding bird survey MHB (see Section 6.9 in AHM1), along with the survey dates for each breeding season 2001–2012. The file also contains coordinates, elevation, forest cover, and the number of surveys per breeding season for each quadrat.

```
# Read in data set from AHMbook
library(AHMbook)
data(crossbillAHM)
str(cb <- crossbillAHM)

# Extract response (detection/nondetection data) and survey dates
y <- as.matrix(cb[,6:41])           # Detection/nondetection data
dates <- as.matrix(cb[,42:77])      # Survey dates

# Standardize covariates for elevation, forest and survey date
mean.ele <- mean(cb$ele, na.rm=TRUE)
sd.ele <- sd(cb$ele, na.rm=TRUE)
elev <- (cb$ele - mean.ele) / sd.ele
mean.forest <- mean(cb$forest, na.rm=TRUE)
sd.forest <- sd(cb$forest, na.rm=TRUE)
forest <- (cb$forest - mean.forest) / sd.forest
mean.date <- mean(dates, na.rm=TRUE)
sd.date <- sd(c(dates), na.rm=TRUE)
DATE <- (dates - mean.date) / sd.date
DATE[is.na(DATE)] <- 0              # Mean-impute missing dates

# Generate unmarked data frame
library(unmarked)
year <- matrix(as.character(2001:2012), 267, 12, byrow = T) # Year covar.
summary(umf <- unmarkedMultFrame(y = y, siteCovs = data.frame(elev, forest),
  yearlySiteCovs = list(year = year), obsCovs=list(date = DATE), numPrimary = 12) )
```

We note the three types of covariates possible in this kind of data set. The yearly-site-level covariate in this analysis is simply year, but might be any other covariate that varies by year and site, but not by replicate survey. Only its first 11 columns will be used to model interannual variation in colonization or extinction, but all 12 columns will be used for modeling annual variation in detection.

4.9.2 FITTING A LARGE, "GLOBAL" DYNAMIC OCCUPANCY MODEL IN `unmarked`

We first investigated whether there was annual variation in the parameters by fitting five models with time-dependence in no, one, or all three parameter types that can vary by year. AIC favored the fully time-dependent model by a large margin of 32 AIC units. Based on this and our biological intuition, we wanted to fit the following "global model" with all effects of potential interest. (Again, see the website for the full code.)

```
Model for initial occupancy (psi1)
(elev + I(elev^2)) * (forest + I(forest^2))

Model for colonization (gamma)
(year-1) + (elev + I(elev^2)) * (forest + I(forest^2))

Model for extinction (eps)
(year-1) + (elev + I(elev^2)) * (forest + I(forest^2))

Model for detection (p)
(year-1) + (elev + I(elev^2)) * (forest + I(forest^2)) +
   date + I(date^2) + date:elev + date:I(elev^2) +
   I(date^2):elev + I(date^2):I(elev^2)
```

We could have tried to fit this model directly, but experience shows that when fitting somewhat complex dynocc models with several covariates, likelihood maximization algorithms are prone to getting stuck at local rather than the global optima. Hence, we homed in on the global model in a

stepwise manner, by adding terms one at a time. We monitored the negative log-likelihood (NLL) as we went, knowing that the NLL of a more complex model must always be at least very slightly lower than that of all earlier models in such a progression. In the end we fit the global model from multiple sets of starting values as another guard against local optima. Fig. 4.13 shows the trajectories of the NLL for the progression of models as we added more terms into the model, first approaching the desired covariate structures in psi1, then in gamma and in eps, and finally in p. With the exception of psi1, there were always some models where `unmarked` got stuck at a local optimum.

In general, the following points are important when fitting complex models with `colext` or other likelihood routines: always scale your covariates; fit models in a stepwise progression of increasing complexity and monitor NLL; start optimization at dispersed starting values as a partial insurance against local minima; start optimization at solutions of related model to speed up convergence; and don't compute SEs in the model selection phase.

We also see that the run for the most complex model (38) did not identify the global maximum likelihood solution because the NLL (2996.729) was slightly higher than for the less complex neighboring model 37 (2995.707). We then tried to find the global optimum for model 38 by initializing a search at the solutions of the neighboring model 37 and also by starting the search repeatedly from various sets of random starting values in the vicinity of a previous set of solutions of model 38.

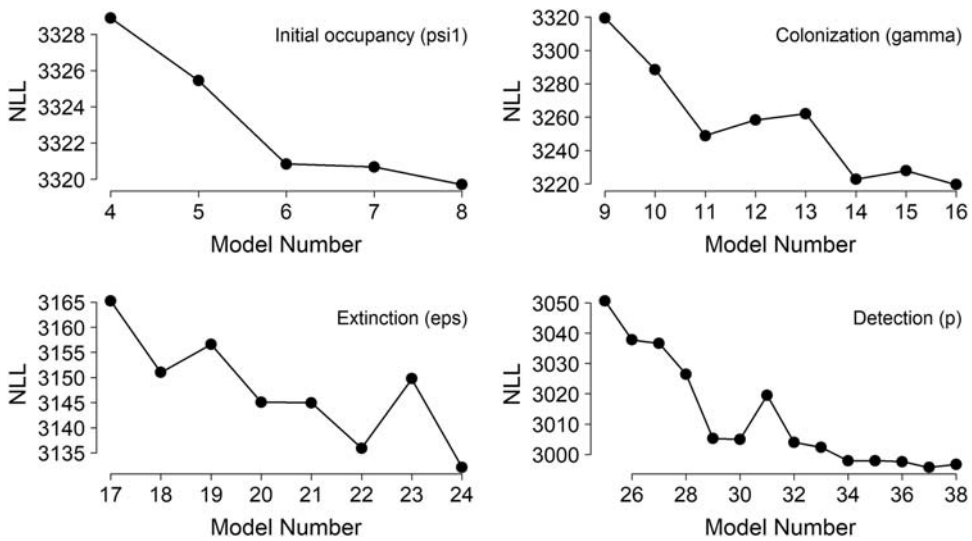

FIGURE 4.13

Negative log-likelihood (NLL) for a series of models fit to the Swiss Crossbill data where each step corresponds to one parameter more or less, in the stated part of the model. In every case, the function `optim` (called by `colext`) indicated convergence of the function minimization. Any increase in the NLL, going from left to right, likely indicates that a local rather than a global maximum was found by `optim`. Model 4 is the fully time-dependent model without any covariate.

```
                                                NLL
free search for model 38:                  2996.728978
using solutions from model 37 as inits:    2995.688954
random around previous solution 1          2995.646178
random around previous solution 2          2995.646266
random around previous solution 3          2995.646051
random around previous solution 4          2995.646011
random around previous solution 5          2995.646451
random around previous solution 6          2995.646279
random around previous solution 7          2995.645961
random around previous solution 8          2995.646141
model 37:                                  2995.707745
```

The free search did not permit `optim` to find the global optimum here either. Using random starting values around a previous solution suggested NLL = 2995.646 to be the minimum NLL and showed that `fm38.list[[9]]` (i.e., "random … solution" 7) had the smallest NLL. We therefore take its solution to be the MLEs.

```
# Identify model run with smallest NLL in tmptab
best <- which.min(tmptab)
```

Fig. 4.14 shows that these 10 model fits produce virtually identical estimates for most parameters, and that discrepancies occurred only for a few parameters with extreme values.

We refitted the model with solutions from `fm38.list[[best]]` used as initial values to obtain the standard errors and called the resulting model fit `fm38X`.

```
inits <- coef(fm38.list[[best]])
summary(fm38X <- colext(~ (elev + I(elev^2))* (forest + I(forest^2)),
  ~ (year-1) + (elev + I(elev^2))* (forest + I(forest^2)),
  ~ (year-1) + (elev + I(elev^2))* (forest + I(forest^2)),
  ~ (year-1) + (elev + I(elev^2))* (forest + I(forest^2)) +
  date + I(date^2) + date:elev + date:I(elev^2) +
  I(date^2):elev + I(date^2):I(elev^2), umf, starts = inits,
  control = list(maxit = 500), se = T))
```

We considered this as our "global" model. Following common usage in capture-recapture and occupancy modeling, we then tested for GoF of this maximal model, so that we could, for instance, adjust AIC for any moderate lack of fit. We used the adaptation of the MacKenzie-Bailey GoF test (MacKenzie and Bailey, 2004) to the dynamic occupancy model (see Section 4.8.1) and produced 1000 bootstrap replicates to assess the magnitude of any lack of fit.

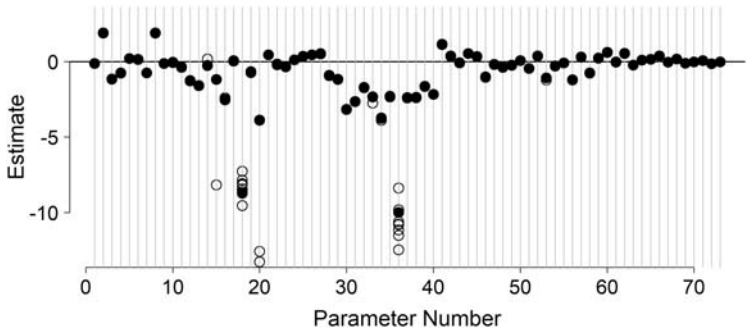

FIGURE 4.14

Estimates of the 73 parameters in model 38 fit to the Swiss Crossbill data, with presumable MLEs shown with solid and other (up to 9), non-MLE solutions as open circles.

```
# Compute Chi-square test statistic for actual data by season
library(AICcmodavg)
mb.chisq(fm38X, print.table = TRUE)

# Generate reference distribution of test statistic under H0 (takes 7h)
system.time( gof <- mb.gof.test(fm38X, print.table = F, nsim = 1000,
  plot.hist = TRUE, plot.seasons = TRUE, report = 1) )
(c.hat <- gof$c.hat.est)
[1] 2.102584
```

This suggested that the global model does not quite fit. Rather, there is a moderate degree of overdispersion measured by a value of c.hat of 2.10. We assumed this does not indicate a major structural deficiency of our model, but rather unstructured noise, and used QAICc for model selection in our search for the best model for prediction.

```
AICc(fm38X, c.hat = c.hat)      # QAICc of the global model
[1] 3055.302
```

4.9.3 MODEL SELECTION BY BACKWARDS ELIMINATION

We next simplified this most complex model by backwards elimination. We dropped one term at a time, starting with the one with the highest *p*-value and continuing until the QAICc no longer went down. We respected the rules of marginality for linear models (McCullagh and Nelder, 1989): that a significant higher order term, e.g., a quadratic term or an interaction, precludes the drop of any of the lower order terms involved such as a linear term or a main effect. After about a dozen such steps (see next table) we identified model 50 as best, in a QAICc sense, and used it for inference.

```
Model selection based on QAICc:
(c-hat estimate = 2.102584)

       K   QAICc Delta_QAICc QAICcWt Cum.Wt Quasi.LL
fm50  62 3017.34        0.00    0.53   0.53 -1427.52
fm51  61 3018.17        0.83    0.35   0.87 -1429.64
fm49  63 3020.73        3.39    0.10   0.97 -1427.50
fm48  64 3024.19        6.85    0.02   0.99 -1427.50
fm47  65 3025.69        8.35    0.01   1.00 -1426.50
   [ ... list truncated ...]
fm39  73 3051.49       34.15    0.00   1.00 -1424.75
fm38X 74 3055.30       37.96    0.00   1.00 -1424.74
```

4.9.4 INFERENCE UNDER THE AIC-BEST MODEL

We inspect the summary of `fm50` and produce a table of MLEs, SEs, and CIs. We note two estimates for the yearly intercepts that are very close to 0 and have very large SEs, colonization 2009 and extinction 2008; these may be boundary estimates and hence, their 95% CIs span 0−1 on the probability scale (do `plogis(c(-130, 130))`). Their SE and CI should best not be interpreted.

```
                                  MLE      SE       0.025     0.975
psi(Int)                      -0.0793  0.3833   -0.83043   0.67191
psi(elev)                      1.9507  0.4551    1.05876   2.84256
psi(I(elev^2))                -1.2445  0.5098   -2.24373  -0.24534
psi(forest)                   -0.7112  0.5104   -1.71165   0.28924
psi(I(forest^2))               0.1655  0.2613   -0.34668   0.67766
psi(elev:forest)               0.1665  0.4511   -0.71761   1.05057
psi(elev:I(forest^2))         -0.7809  0.3540   -1.47469  -0.08721
psi(I(elev^2):forest)          1.8743  0.5393    0.81743   2.93127
col(year2001)                 -0.0926  0.3264   -0.73241   0.54716
col(year2002)                 -0.3715  0.3821   -1.12040   0.37739
....
col(year2009)                 -8.7195 55.8571 -118.19737 100.75829
col(year2010)                 -0.7878  0.4690   -1.70704   0.13149
col(year2011)                 -3.4583  2.3500   -8.06411   1.14756
col(elev)                      0.5002  0.2060    0.09642   0.90405
col(I(elev^2))                -0.2159  0.2914   -0.78699   0.35515
col(forest)                   -0.3927  0.2801   -0.94163   0.15632
col(I(forest^2))               0.2008  0.2139   -0.21852   0.62003
col(elev:forest)               0.3629  0.1822    0.00591   0.71994
col(elev:I(forest^2))          0.4318  0.2128    0.01465   0.84887
col(I(elev^2):forest)          0.5789  0.2884    0.01368   1.14412
col(I(elev^2):I(forest^2)) -0.9509  0.2876   -1.51459  -0.38726
ext(year2001)                 -1.3218  0.5058   -2.31307  -0.33057
ext(year2002)                 -3.2547  0.7072   -4.64071  -1.86872
...
ext(year2008)                 -9.9830 61.7477 -131.00622 111.04026
ext(year2009)                 -2.4937  0.4330   -3.34226  -1.64512
ext(year2010)                 -2.4372  0.5266   -3.46928  -1.40514
ext(year2011)                 -1.7589  0.4253   -2.59239  -0.92533
ext(elev)                     -1.9752  0.2615   -2.48781  -1.46257
ext(I(elev^2))                 1.1320  0.2998    0.54445   1.71959
ext(forest)                    0.4192  0.2536   -0.07780   0.91622
ext(elev:forest)               0.8572  0.2235    0.41918   1.29519
ext(I(elev^2):forest)         -1.3221  0.2844   -1.87960  -0.76469
p(year2001)                   -0.3512  0.1977   -0.73858   0.03622
p(year2002)                   -0.2137  0.1623   -0.53188   0.10452
...
p(year2011)                   -0.7098  0.1449   -0.99376  -0.42582
p(year2012)                    0.2540  0.1630   -0.06543   0.57334
p(elev)                        0.4714  0.0774    0.31958   0.62316
p(forest)                      0.6493  0.0790    0.49455   0.80403
p(I(forest^2))                -0.2784  0.0488   -0.37412  -0.18274
p(date)                        0.0816  0.0440   -0.00465   0.16781
p(I(date^2))                   0.1135  0.0383    0.03839   0.18862
p(elev:forest)                 0.4188  0.0674    0.28672   0.55079
```

These SEs and CIs fail to account for the extra uncertainty associated with the overdispersion detected in the GoF test. We used function `modavg` in `AICcmodavg` to inflate them properly and illustrate this here for the parameters in the psi1 part of the model, comparing them for the standard `unmarked` output.

```
# Compare estimates side by side
print(cbind(tmp[1:8,], inflated.uncertainty), 2)
                        MLE   SE 0.025  0.975 SE(infl) 95%LCL(infl) 95%UCL(infl)
psi(Int)             -0.079 0.38 -0.83  0.672     0.56        -1.17         1.01
psi(elev)             1.951 0.46  1.06  2.843     0.66         0.66         3.24
psi(I(elev^2))       -1.245 0.51 -2.24 -0.245     0.74        -2.69         0.20
psi(forest)          -0.711 0.51 -1.71  0.289     0.74        -2.16         0.74
psi(I(forest^2))      0.165 0.26 -0.35  0.678     0.38        -0.58         0.91
psi(elev:forest)      0.166 0.45 -0.72  1.051     0.65        -1.12         1.45
psi(elev:I(forest^2)) -0.781 0.35 -1.47 -0.087     0.51        -1.79         0.22
psi(I(elev^2):forest)  1.874 0.54  0.82  2.931     0.78         0.34         3.41
```

We see that the SEs are multiplied by the square root of c.hat and the CIs are inflated accordingly.

4.9.5 FORMING PREDICTIONS IN ONE OR TWO DIMENSIONS

Forming predictions is essential for any regression analysis to understand the relationships between covariates and modeled parameters, and to present the results of the analysis in a paper. Prediction means using the model to compute the expected values of these parameters when the values of one or more covariates are varied. We have earlier described the dynamic occupancy model as a combination of four logistic regressions, with one probability parameter in each. Whenever we add covariates into our analysis, that probability parameter (on the logit scale) is replaced by a linear function of those covariates. Prediction then means that we take values of interest for these covariates and put them into the linear regression equation to compute the expected values of the probability parameters. We now illustrate this with our best model (`fm50`) to generate the following types of predictions:

- Predictions for a single categorical covariate
- Predictions for a single continuous covariate
- Predictions for a categorical and a continuous covariate simultaneously
- Predictions for two continuous covariates simultaneously
- Predictions for covariates in geographic space for predictive mapping (see next section)

We may form predictions "by hand," but in most cases it is more practical to use for this one of the `predict` functions in R. We illustrate the use of two such functions: the `predict` function in `unmarked` and then similar functionality in `AICcmodavg`, which in addition allows to account for lack of fit in a model (as quantified by an estimate of `c.hat`) by inflating the uncertainty around predictions. We could use these functions to estimate the predicted values for our actual data set, i.e., for the observed values of the covariates. However, more typically we will want to show the covariate relationships "in general," i.e., for a hypothetical new data set that spans a whole range of these covariates, and this is what we do here. Hence, we will first generate values of such "prediction covariates." Remember that we need to transform them exactly as we transformed the covariates in our analysis of the actual data set, but then, we typically want to plot predictions against the untransformed values of the prediction covariates.

We start by creating prediction covariates for elevation, forest cover, and survey date and generate covariates with 100 values equally spaced within a reasonable range, with the latter informed by the range of the observed covariate values in our data set.

```
# Let elevation only go till 2250 (no forest higher up)
n.pred.points <- 100            # not too high
ep.original <- seq(min(cb$elev), 2250, length.out = n.pred.points)
ep <- (ep.original - mean.ele)/sd.ele
fp.original <- seq(min(cb$forest), max(cb$forest), length.out = n.pred.points)
fp <- (fp.original - mean.forest)/sd.forest
first.survey <- min(as.numeric(dates), na.rm = TRUE)
dp.original <- seq(from = first.survey, to = 105, length.out = n.pred.points)
dp <- (dp.original - mean.date) / sd.date
```

4.9.5.1 Predictions of Year Effects on Occupancy, Colonization, Extinction, and Detection

The `predict` function can only be used for the basic parameters in the model, which for occupancy is only the first year, while for later years, occupancy is a derived quantity that is a function of psi1, gamma, and eps. Hence, if we want to predict occupancy probability for later years, we must compute

this ourselves and for their uncertainty, when using maximum likelihood use a nonparametric bootstrap or the delta method, or else Bayesian posterior inference. We illustrate the former here. We could use the `nonparboot` function in `unmarked`, but here we code it up ourselves for illustration of what it does. For a nonparametric bootstrap estimate of the variance of annual occupancy, we repeat the following many times:

- Draw a data set of 267 sites from the original data set *with replacement*. Hence, some sites will appear multiple times in this new data set, while others will appear not at all.
- Fit the desired model (`fm50`) and save the estimates of annual occupancy.
- We could also compute further functions of the model parameters in the same loop, such as range size (see Section 4.9.6).

```
# Get bootstrapped estimates of SE and CI for annual occupancy
nboot <- 1000           # number of bootstrap samples
boot.psi.hat <- array(NA, dim = c(12, nboot))
for(i in 1:nboot){      # Start loop
  cat(paste("\n ** Nonparametric bootstrap rep", i, "**\n") )
  # Draw bootstrap sample
  bootsamp <- sample(1:267, replace = TRUE)
  # Create unmarked data frame
  umfboot <- unmarkedMultFrame(y = as.matrix(cb[bootsamp,6:41]),
    siteCovs = data.frame(elev = elev[bootsamp], forest = forest[bootsamp]),
    yearlySiteCovs = list(year = year), obsCovs = list(date = DATE[bootsamp,]),
    numPrimary = 12)
  # Fit model 50, use estimates from fm50 as inits, do not compute SEs
  inits <- coef(fm50)
  fmtmp <- colext(~ (elev + I(elev^2)) * (forest + I(forest^2)) -
     I(elev^2):I(forest^2),
  ~ (year-1) + (elev + I(elev^2)) * (forest + I(forest^2)),
  ~ (year-1) + (elev + I(elev^2)) * (forest + I(forest^2)) -
    I(elev^2):I(forest^2) - elev:I(forest^2) - I(forest^2),
  ~ (year-1) + elev + forest + I(forest^2) + date + I(date^2) +
    elev:forest, umfboot, starts = inits,
    control = list(trace = T, REPORT = 25, maxit = 500), se = F)
  # Compute and save estimates of annual occupancy
  # -> Here could bootstrap additional functions of model parameters
  boot.psi.hat[,i] <- projected(fmtmp)[2,]
}
```

Once we have accumulated a sample of sufficient size (best >1000), we can summarize it by using the SD of the replicate values (the "bootstrapped SE") or percentiles defining central 95% (the "bootstrapped 95% CI").

```
SE.occ <- apply(boot.psi.hat, 1, sd)
CI.occ <- apply(boot.psi.hat, 1, function(x) quantile(x, prob = c(0.025, 0.975)))
```

Now that we have an estimate of the uncertainty of occupancy for all years, we continue with prediction of the year-specific expected values of colonization, extinction, and detection. Here is an overview of two available methods, producing predictions of the annual colonization. Remember that the year factor in the prediction data frame (nd, for "new data") for gamma and eps must contain only as many years as there are intervals, which is one fewer than in the new data frame for p.

```
library(unmarked)
library(AICcmodavg)
# Choose covariate values for the prediction
nd <- data.frame(year=factor(c('2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008',
  '2009', '2010', '2011')), elev = rep(0,11), forest = rep(0,11))
# Prediction of annual colonization with unmarked predict function
E.col.1 <- predict(fm50, type = 'col', newdata = nd)
# With modavgPred (can inflate variances)
E.col.2 <- modavgPred(cand.set = list(fm50), newdata = nd, conf.level = 0.95,
   parm.type = 'gamma', c.hat = c.hat) # Here for c.hat of 2.10

# Compare estimates for two predict methods
print(cbind(year = 2001:2011, as.matrix(E.col.1), E.col.2$matrix.output),2)
```

```
      year Predicted     SE   lower upper mod.avg.pred uncond.se lower.CL upper.CL
 [1,] 2001   0.47686 0.0814 3.2e-01  0.63      0.47686     0.118  2.6e-01     0.70
 [2,] 2002   0.40818 0.0923 2.5e-01  0.59      0.40818     0.134  1.9e-01     0.67
 [3,] 2003   0.21682 0.0864 9.3e-02  0.43      0.21682     0.125  6.1e-02     0.54
 [4,] 2004   0.18211 0.0758 7.6e-02  0.38      0.18211     0.110  5.0e-02     0.49
 [5,] 2005   0.43028 0.1041 2.5e-01  0.63      0.43028     0.151  1.8e-01     0.72
 [6,] 2006   0.24293 0.1347 7.1e-02  0.57      0.24293     0.195  3.8e-02     0.72
 [7,] 2007   0.04486 0.1012 4.6e-04  0.83      0.04486     0.147  5.7e-05     0.97
 [8,] 2008   0.51520 0.0999 3.3e-01  0.70      0.51520     0.145  2.5e-01     0.77
 [9,] 2009   0.00016 0.0091 4.7e-52  1.00      0.00016     0.013  1.9e-73     1.00
[10,] 2010   0.31265 0.1008 1.5e-01  0.53      0.31265     0.146  1.1e-01     0.63
[11,] 2011   0.03052 0.0695 3.1e-04  0.76      0.03052     0.101  4.0e-05     0.96
```

We see well the effect of the accommodation of overdispersion on the SEs and CIs in the right side of the table (coming from `AICcmodavg`). We produce the remaining predictions (for eps and p) over time (see website for code) and then plot annual predictions for all four parameters of the model (Fig. 4.15). We see considerable variation among the years.

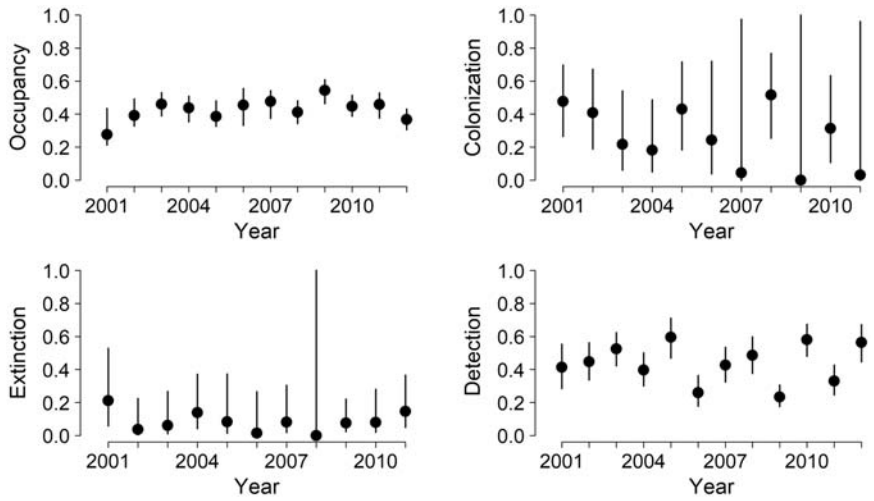

FIGURE 4.15

Annual estimates of the probabilities of occupancy, colonization, extinction, and detection in Swiss crossbills between 2001 and 2012, with 95% CIs, based on the AIC-best model (`fm50`).

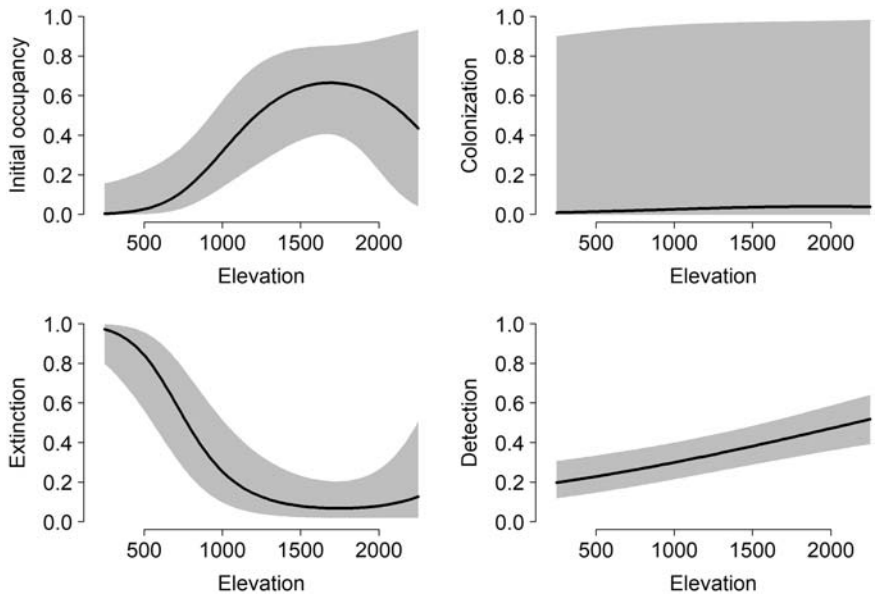

FIGURE 4.16

Predicted elevation profiles of probabilities of occupancy, colonization, extinction, and detection in Swiss crossbills, with 95% CIs, based on the AIC-best model (`fm50`), and inflated for overdispersion of magnitude c.hat = 2.10.

4.9.5.2 Predictions of a Single Continuous Covariate

We illustrate with predicted elevation profiles for the four basic model parameters. We have to keep constant the values of all other continuous covariates and choose one level for any factors in the model. For the former, the observed mean value is a typical choice, while for the latter, we have year as a factor and arbitrarily choose 2011 for illustration (Fig. 4.16) See website for code.

4.9.5.3 Predictions for Two Continuous Covariates Simultaneously

Finally, we give examples of how you can investigate how the model parameters vary as two continuous covariates are varied simultaneously (Fig. 4.17). Except for the last example, we use the `predict` function because we don't compute overdispersion-inflated SEs for now. Besides being visually pleasing, such plots may help forming hypotheses about drivers of the spatiotemporal patterns of all four parameters.

4.9.6 PREDICTION IN GEOGRAPHICAL SPACE TO PRODUCE SPECIES DISTRIBUTION MAPS

In the last section, we have seen how to produce "maps" in covariate space, i.e., in a space spanned by two environmental covariates: forest cover, and elevation or date. Next, we plot predictions in *geographical space*. If we map them, then these predictions represent predictive maps of species distribution, colonization, extinction, and detection for our Swiss crossbills. We also want to produce maps of the prediction uncertainty, a frequently forgotten, but important, topic in species distribution modeling. Especially, it is insufficient to conclude that a range is changing without some assessment of the precision of our estimate of that change. Now, we will not predict for values of the explanatory

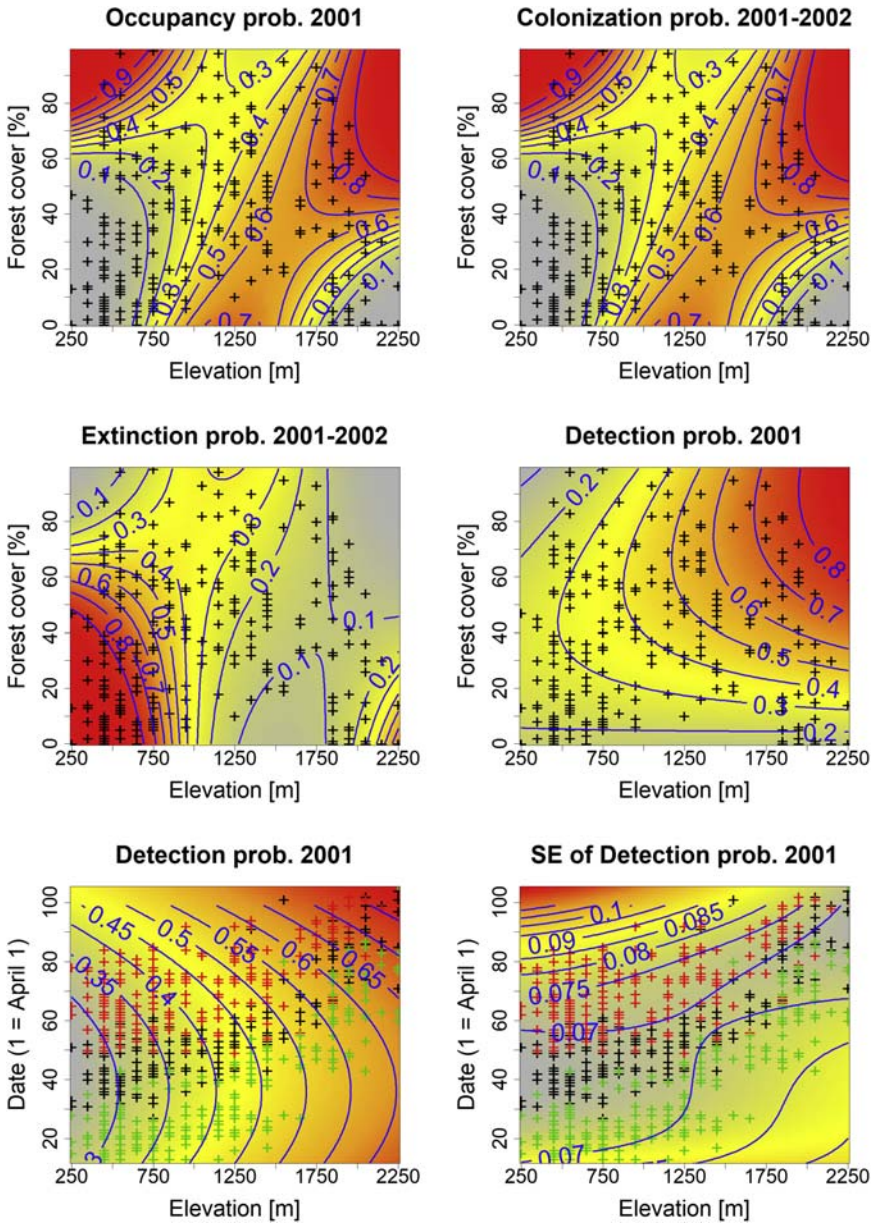

FIGURE 4.17

Two-dimensional predictions of a parameter along gradients of forest cover, and elevation (first two rows) or date (bottom row). Crosses show the actual values of the covariates in the observed data. In the two plots at the bottom, first to third surveys are shown in green, black, and red respectively.

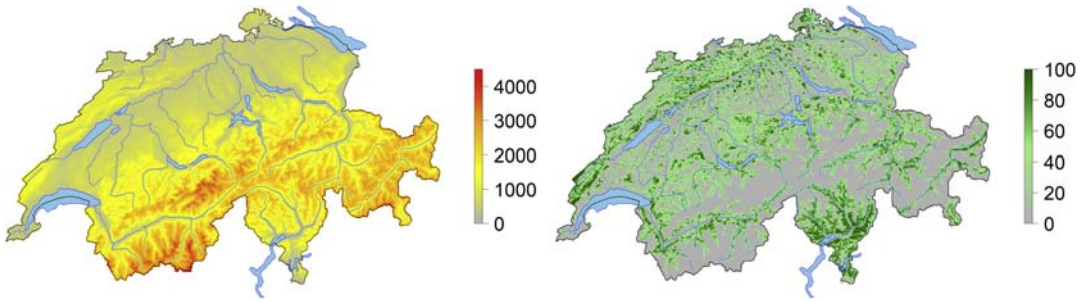

FIGURE 4.18

Maps of two of the most powerfully predicting landscape features in Switzerland: elevation (in meters, left) and forest cover (in percent, right).

variables in some hypothetical landscape, but for the real Swiss landscape, using the actual values of elevation and forest cover for each of the 42,275 1-km^2 quadrats in the Swiss landscape. We load the Swiss landscape data from `unmarked` first.

```
require(unmarked)
data(Switzerland)          # Load Swiss landscape data in unmarked
str(ch <- Switzerland)
```

First, we remind ourselves of how Switzerland looks like in terms of elevation and forest cover by mapping these covariates from the Swiss landscape data set (Fig. 4.18).

We use the values of these covariates to predict what values of occupancy, colonization, extinction, and detection we would expect in each 1-km^2 quadrat under our AIC-best model `fm50`. We could again use `predict` methods from above, but they also compute the SEs and hence are much slower. Since we only want the point prediction, we do them "by hand." You will recognize most of the following steps for producing predictions from the earlier sections.

```
nyears <- 12

# Standardize using same values as for original data set in model fitting
EP <- (ch$elevation - mean.ele) / sd.ele
FP <- (ch$forest - mean.forest) / sd.forest

# Extract parameter estimates for each parameter type
tmp <- summary(fm50)
psipar <- tmp$psi[,1]              # Params in model for initial occupancy
names(psipar) <- rownames(tmp$psi)
colpar <- tmp$col[,1]              # Params in model for colonization
names(colpar) <- rownames(tmp$col)
extpar <- tmp$ext[,1]              # Params in model for extinction
names(extpar) <- rownames(tmp$ext)
detpar <- tmp$det[,1]              # Params in model for detection
names(detpar) <- rownames(tmp$det)

# Create arrays to contain predictions for each km2 in Switzerland
# First-year occupancy
pred.occ1 <- numeric(nrow(ch))
# Colonization and extinction prob. (note one fewer year)
pred.col <- pred.ext <- matrix(NA, nrow = nrow(ch), ncol = (nyears-1))
# Occupancy (all years) and detection prob.
pred.occ <- pred.det <- matrix(NA, nrow = nrow(ch), ncol = nyears)
```

We predict initial occupancy in each quadrat in the first year (2001). For this step, we could also have used a `predict` function directly.

```
# Compute predicted first-year occupancy probability
pred.occ1 <- plogis(psipar[1] + psipar["elev"]*EP +
  psipar["I(elev^2)"]*EP^2 + psipar["forest"]*FP +
  psipar["I(forest^2)"]*FP^2 + psipar["elev:forest"]*EP*FP +
  psipar["I(elev^2):forest"]*EP^2*FP)
pred.occ[,1] <- pred.occ1
```

Next, we compute the expected colonization and extinction for each inter-year interval between 2001 and 2012 (again avoiding use of a `predict` function for greater speed). At the bottom of this section of code, we compute occupancy for years after 2001 as a function of the basic model parameters. For this step, there isn't any ready predict function.

```
# Compute predicted colonization and extinction probability and occupancy
# for later years (j>1)
for(j in 1:(nyears-1)){
  # Colonization
  pred.col[,j] <- plogis(colpar[j] + colpar["elev"]*EP +
    colpar["I(elev^2)"]*EP^2 + colpar["forest"]*FP +
    colpar["I(forest^2)"]*FP^2 + colpar["elev:forest"]*EP*FP +
    colpar["elev:I(forest^2)"]*EP*FP^2 +
    colpar["I(elev^2):forest"]*EP^2*FP +
    colpar["I(elev^2):I(forest^2)"]*EP^2*FP^2)

  # Extinction
  pred.ext[,j] <- plogis(extpar[j] + extpar["elev"]*EP +
    extpar["I(elev^2)"]*EP^2 + extpar["forest"]*FP +
    extpar["elev:forest"]*EP*FP + extpar["I(elev^2):forest"]*EP^2*FP)

  # Compute occupancy for later years recursively
  pred.occ[,(j+1)] <- pred.occ[,j] * (1-pred.ext[,j]) +
    (1-pred.occ[,j]) * pred.col[,j]
}
```

Finally, we predict (per-survey) detection probability. Note that the third covariate in the model for detection, survey date, is not one that can be readily associated with a spatial unit; hence, we predict at the average survey date. These are the terms involving the multiplication with zero below, which we could have simply dropped (since date is centered), but show to make explicit the prediction at average date.

```
# Compute predicted detection probability (for average date)
for(j in 1:nyears){
  pred.det[,j] <- plogis(detpar[j] + detpar["elev"]*EP +
    detpar["forest"]*FP + detpar["I(forest^2)"]*FP^2 + detpar["date"] * 0 +
    detpar["I(date^2)"] * 0 + detpar["elev:forest"]*EP*FP)
}
```

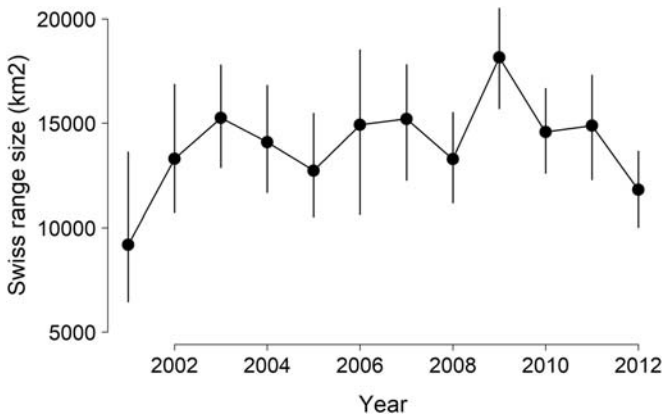

FIGURE 4.19

Estimated range size (number of occupied 1-km^2 quadrats) of the European Crossbill (*Loxia curvirostra*) in Switzerland between 2001 and 2012 with bootstrapped 95% CIs.

We computed the annual crossbill range size 2001−2012 by simply adding the predicted occupancy probability over Switzerland. To get the CIs, we used a nonparametric bootstrap exactly as in Section 4.9.5. We see that the Swiss range of the crossbill varies twofold over only a dozen years (Fig. 4.19).

We have computed predictions of the probabilities of occupancy and detection for 12 years and of colonization and extinction for 11 interannual intervals for every Swiss 1-km^2 quadrat. We can now plot them, producing maps not only of a static quantity (occupancy probability, as one representation of a species distribution), but also of the two components underlying the range dynamics (colonization and extinction). In addition, we can produce predictive maps of the presence/absence measurement error (or rather of its converse, detection probability). Fig. 4.20 shows a sample of such maps which highlights the variety of quantities underlying our species distribution data that we can estimate and then map.

An underrated topic in SDMs is prediction uncertainty. Using one of the two prediction functions illustrated, we can readily obtain such maps for the primary parameters in the model, and we want to illustrate this for just one of them, colonization probability from 2001−2002 (Fig. 4.21). To compute the prediction uncertainty of occupancy for years 2−12 one could use the delta method or a bootstrap; see Clement et al. (2016) for an SDM application of the former.

```
# Obtain prediction uncertainty for Swiss colonization 2001-2002
library(AICcmodavg)
se.pred.col <- matrix(NA, nrow = nrow(ch), ncol = 1)
newData <- data.frame(year = factor('2001',
  levels = c('2001','2002','2003','2004','2005','2006','2007','2008','2009','2010','2011')),
  elev = EP, forest = FP)                    # NOTE: Only 11 yearly intervals
system.time(
pred <- modavgPred(cand.set = list(fm50), newdata = newData,
  parm.type = 'gamma', c.hat = c.hat))       # Takes a while
se.pred <- pred$matrix.output[,2]            # Grab inflated SE
```

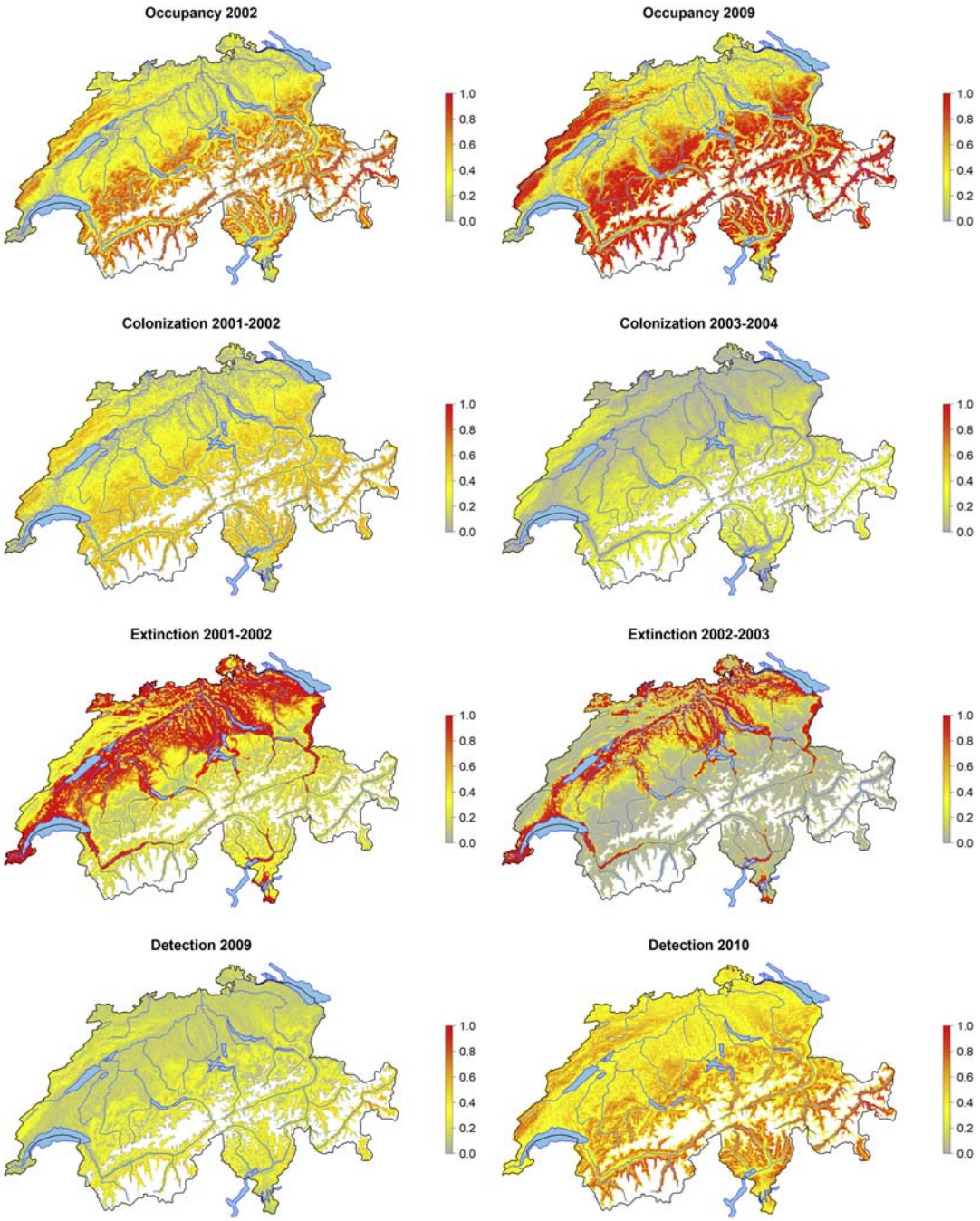

FIGURE 4.20

A sample of predictive maps of the probability of occupancy, colonization, extinction, and detection of Swiss crossbills during 2001–2012 (areas above 2250 m a.s.l. where there is no forest are masked).

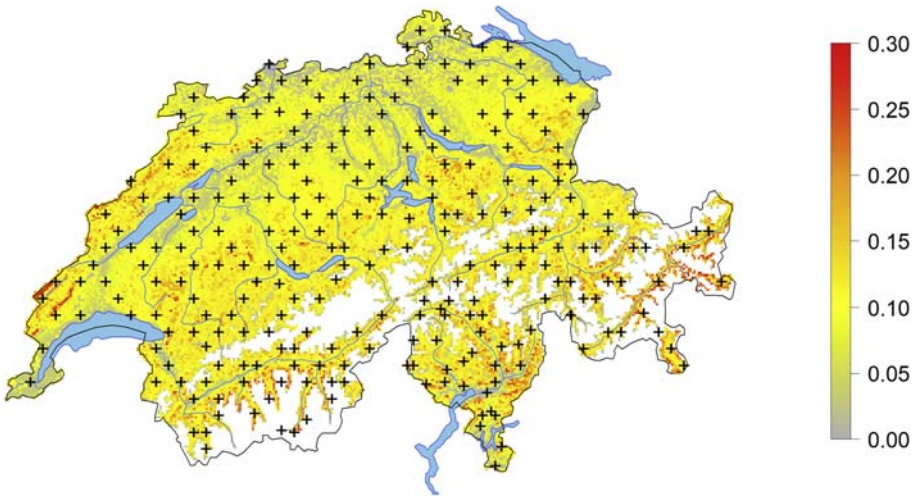

FIGURE 4.21

Swiss map of the prediction uncertainty (overdispersion-inflated standard errors) of colonization probability in 2001–2002. Crosses show locations of 267 survey quadrats. Note that we could also map the lower and upper prediction bounds, perhaps for the supplementary material of a paper.

4.9.7 BRIEF COMMENTS ON THE ANALYSIS OF DISTRIBUTION DYNAMICS OF SWISS CROSSBILLS

We have illustrated a typical workflow for the analysis of distribution or range dynamics using the dynocc model in `unmarked`: (1) we fit a series of sensible models and use AIC to pick the best one; (2) we test GoF; if lack of fit is not too bad, we may assume it represents simply extra noise, for which we adjust by increasing our measures of uncertainty (SE, CI) and of model selection (i.e., use QAIC); (3) we make inferences about functional relationships with covariates and form predictions; and (4) we may map predictions in geographic space for any of the model parameters. Many variations on this workflow are possible, including the use of just a single complex model that may be of interest for inference and GoF testing (ver Hoef and Boveng, 2015), model averaging to produce ensemble predictions (Dormann et al., 2018b), or use of different criteria for model selection, such as cross-validation (Hooten and Hobbs, 2015).

Importantly, *any* regression model for a parameter as a function of spatially indexed covariates can be extrapolated to a wider area, provided that the values of these covariates are known, and, hence, predictive maps for that parameter can be produced. Such mapped parameters may be expected abundance or density (as we have shown in every single Chapter from 6–9 in AHM1) or the probability of occupancy (as in Chapter 10 in AHM1 and also in this chapter), all of them representing instances of an SDM. However, predictive maps may also be produced of other parameters, e.g., survival probabilities (Chapter 3), or colonization and extinction probabilities, as we have just seen, where we have even produced predictive maps of the measurement error in our SDM (or rather, its complement, detection probability). Yet another useful map might be for a derived quantity such as an occupancy growth rate or turnover. Thus, we can visualize the spatial or spatiotemporal patterns in all of these parameters. This may be a powerful method for generating hypotheses about drivers of the processes that cause these patterns.

The dynocc is a very powerful model, and likelihood inference, for example, using `colext`, is very convenient, but care has to be taken to avoid local minima, which are frequent when fitting this model with more than just a handful of covariates. We have shown some strategies to avoid pitfalls. In addition, with other dynocc modeling exercises with likelihood inference (MK, unpublished own work) we have repeatedly noticed almost binary predictions of occupancy probability for some of the earliest years in a series, where almost all predictions were very close to either zero or one. One possible reason for this is "complete separation" (p. 104 in Gelman and Hill, 2007), where the observed zeros and ones separate the covariate values into two nonoverlapping sets of values. Bayesian inference is better able to deal with this, especially when weakly informative priors are used. Therefore, you may want to choose a Bayesian analysis to mitigate such problems.

4.10 ANALYSIS OF CITIZEN SCIENCE DATA USING OCCUPANCY MODELS

All over the world there has been a huge increase in recent decades in the amount of detection/nondetection data collected in citizen-science schemes, where little or no design is imposed on the data collection protocol. These data may be valuable because of their sheer volume and because their collection does not cost much. However, they may be very costly at the analysis stage because design deficiencies must then be compensated for with a more complex model. Clearly, in such programs detection probability is likely to be very heterogeneous and to change systematically in space and time. Hence, occupancy models appear to be an ideal analytical framework for detection/nondetection data from citizen-science schemes (Kéry et al., 2006; Altwegg et al., 2008; Kéry et al., 2010a,b; 2013; van Strien et al., 2010, 2011; 2013; Bled et al., 2013; Broms et al., 2014, 2016a,b; Isaac et al., 2014; Clement et al., 2016; Crum et al., 2017; Outhwaite et al., 2018). See also the excellent recent synthesis by Altwegg & Nichols (2019).

But even when detection error can be accounted for, pitfalls remain for the analysis of citizen-science data. For instance, preferential sampling is arguably the rule in most citizen-science schemes and unless accounted for in an analysis will lead to biased inferences (see Section 4.11). Further, detection probability is likely to be much more variable in space and in time in a citizen-science program than in a designed program (such as many national breeding bird surveys including the Swiss MHB). We have seen in Section 4.7.4 that unmodeled site-level heterogeneity in detection biases low the occupancy estimator. In many citizen-science surveys with mass participation an increase in the number of people and in the number of sites surveyed may increase detection heterogeneity. This is worrying because it may then cause spurious negative trends in occupancy estimates and even in observed occupancy proportions.

Next, we first demonstrate via simulation the ability of increasing site-level detection heterogeneity to cause spurious downward trends in occupancy estimates. At the same time we show that this pattern of heterogeneity can be corrected for via random site effects in the detection part of the occupancy model. In the second section we fit occupancy models with and without heterogeneous detection to a real-world data set from a citizen-science scheme, analyzing opportunistic detection/nondetection data from a Swiss woodpecker.

4.10.1 EFFECTS OF TRENDS IN THE MAGNITUDE OF UNMODELED DETECTION HETEROGENEITY

To illustrate the effects of changes over time in the magnitude of unmodeled, site-level detection heterogeneity, we simulate a single data set over 20 years from a perfectly stable population. With colonization at 0.3 and extinction at 0.2, equilibrium occupancy is 0.6 and this is the value we choose

for psi1. Detection probability is 0.5 for each of three surveys, with zero-mean, logit-Normal random site-effects with SD *that increases from 0 to 2 over the course of the study.* Hence, heterogeneity among sites in detection probability increases over time from zero to very large levels (see the SD = 2 case in Fig. 4.10). Then, we fit two models with fully time-dependent parameters: one assumes no additional detection heterogeneity, while the other accounts for site-level detection heterogeneity that varies by year. The former is the model from Section 4.5, so here we show only the second model with heterogeneity (as usual, you find full code on the website).

```
# Generate data set with increase in site-level heterogeneity
set.seed(1)
str(data <- simDynocc(nsites = 250, nyears = 20, nsurveys = 3, mean.psi1 = 0.6,
  range.p = c(0.5, 0.5), range.phi = c(0.8, 0.8), range.gamma = c(0.3, 0.3),
  trend.sd.site = c(0, 2)))  # library(AHMbook)

# Heterogeneity occupancy model
# Bundle data
str(bdata <- list(y = data$y, nsites = dim(data$y)[1],
  nsurveys = dim(data$y)[2], nyears = dim(data$y)[3]))

List of 4
 $ y       : int [1:250, 1:3, 1:20] 0 0 1 1 0 0 1 0 0 0 ...
 $ nsites  : int 250
 $ nsurveys: int 3
 $ nyears  : int 20

# Specify model in BUGS language
cat(file = "dynoccH.txt", "
model {

# Specify priors
psi1 ~ dunif(0, 1)
for (t in 1:(nyears-1)){
  phi[t] ~ dunif(0, 1)
  gamma[t] ~ dunif(0, 1)
}
for (t in 1:nyears){
  lp[t] <- logit(mean.p[t])
  mean.p[t] ~ dunif(0, 1)
}

# Random effects priors
for (t in 1:nyears){
  for (i in 1:nsites){
    eps[i,t] ~ dnorm(0, tau.eps[t])
  }
  tau.eps[t] <- pow(sd.eps[t], -2)
  sd.eps[t] ~ dunif(0, 10)   # Note different variance in every year
}

# Ecological submodel
for (i in 1:nsites){
  z[i,1] ~ dbern(psi1)
  for (t in 2:nyears){
    z[i,t] ~ dbern(z[i,t-1]*phi[t-1] + (1-z[i,t-1])*gamma[t-1])
  }
}
```

```
# Observation model
for (i in 1:nsites){
  for (j in 1:nsurveys){
    for (t in 1:nyears){
      logit(p[i,j,t]) <- lp[t] + eps[i,t] # time + site.survey effects
      y[i,j,t] ~ dbern(z[i,t] * p[i,j,t])
    }
  }
}

# Compute population and sample occupancy
psi[1] <- psi1                              # Population occupancy
psi.fs[1] <- sum(z[1:nsites,1]) / 250       # Sample occupancy
for (t in 2:nyears){
  psi[t] <- psi[t-1]*phi[t-1] + (1-psi[t-1])*gamma[t-1]
  psi.fs[t] <- sum(z[1:nsites,t]) /  250
}
}
")

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

# Parameters monitored
params <- c("psi", "psi.fs", "phi", "gamma", "mean.p", "sd.eps")

# MCMC settings
na <- 5000  ;  ni <- 100000  ;  nb <- 20000  ;  nt <- 80  ;  nc <- 3

# Call JAGS (ART 163 min), check convergence and summarize posteriors
out <- jags(bdata, inits, params, "dynoccH.txt", n.adapt = na,
  n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, parallel = T)
par(mfrow = c(4,4))  ;  traceplot(out)
print(out, 2)                               # not shown
```

Interestingly, time-varying detection heterogeneity at the site level biased both the observed occupancy proportion and also the occupancy probability under a standard model that did not account for this extra variability (Fig. 4.22, left), such that both showed a spurious decline over time with the simulated increase in that heterogeneity. Unpublished simulations revealed that the opposite holds with a decline in detection heterogeneity over time. In contrast, the heterogeneity model correctly identified a more or less stable population and was well able to estimate the increase in the magnitude of site-level detection heterogeneity over time (Fig. 4.22, middle and right).

Hence, in a clear occupancy analogy of the Second Law of Capture-Recapture (Chapter 6 in AHM1), detection heterogeneity at the site level leads to a negative bias in the occupancy estimator (Royle, 2006). If the magnitude of that heterogeneity changes over time, then so will this bias. A directed change of detection heterogeneity will therefore cause a directed change both in the observed proportion of occupied sites as well as the estimated occupancy under a model without such heterogeneity. This is worrisome for analyses of citizen science data, since we might well expect trends in heterogeneity, both in space and especially over time. Ideally, such patterns in the detection heterogeneity ought to be investigated and, where necessary, corrected for by appropriate modeling. This may be done via random effects or via covariates that may explain part of such variability. Also note that a more parsimonious representation of a heterogeneity trend would be a linear model for the heterogeneity at the log scale.

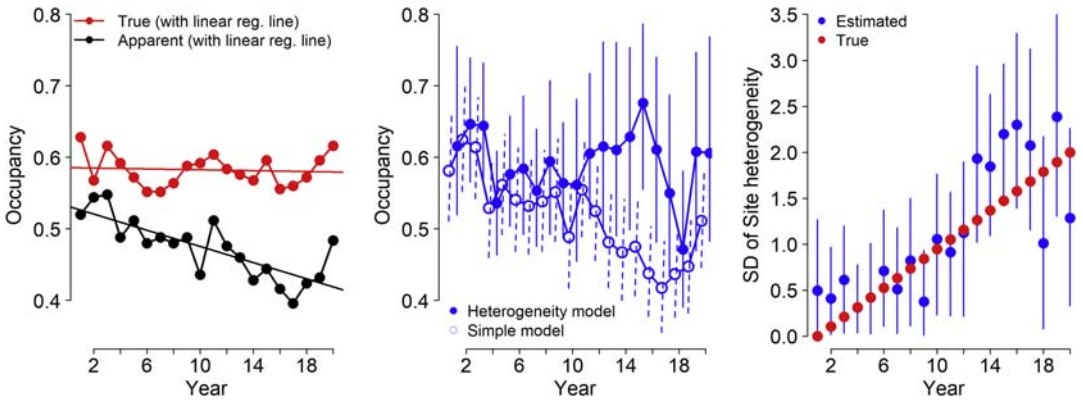

FIGURE 4.22

Effects of time-varying, unmodeled site-level detection heterogeneity on occupancy analyses. The simulated system was stable, but both the observed occupancy (left) and that estimated under a model without heterogeneity (middle, open blue circles) had a spurious downward trend. In contrast, the model which accommodated that heterogeneity (middle, solid blue circles) did not estimate such a negative trend. Instead, it correctly identified an increase in the detection heterogeneity (right, red solid red circles shows true values in each year). Uncertainty intervals are 95% CRIs.

4.10.2 ANALYSIS OF CITIZEN SCIENCE DATA ON SWISS MIDDLE-SPOTTED WOODPECKERS

Biological records centers have always been challenged by temporal changes in sampling effort. Often, sampling effort is not recorded or quantified for citizen-science records. Hence, an increase in the number of recorded presences of a species could mean that the species is increasing in abundance or extending its range, or simply that more people are looking harder for the species, or a combination of both. With the advent of internet-based data entry for biological records, the number of records submitted often increases greatly. Then, by an increase in observer activity alone, many species may appear to increase, even though their populations are known to be stable, or they appear stable even though their populations are declining. It would be entirely possible for a population to *appear* to increase, whereas in fact it is declining, simply because of a strong increase in the observer effort (Tingley and Beissinger, 2013).

Occupancy models are perfectly suited to deal with varying observation effort in a process-based manner and to correct for both the number and the quality of surveys. If the observer effort is increasing over time, then this will be reflected in more surveys and possibly also increased detection probability per survey. Both components of the overall chance for an occupied site to be detected are accommodated in an occupancy model. Therefore, they appear ideal for inference about species distribution and its change based on unstructured citizen-science records (Kéry et al., 2006; Altwegg et al., 2008; Kéry et al., 2010a,b, 2013; van Strien et al., 2010, 2011, 2013; Bled et al., 2013; Broms et al., 2014, 2016b; Clement et al., 2016; Johnston et al., 2018; Altwegg and Nichols, 2019).

One frequent practical challenge is the extremely large proportion of empty cells if the data are formatted in a multidimensional array, as we usually do in BUGS, or as a matrix in `unmarked`. This may lead to crashes in `unmarked`, or endless run times in BUGS because any missing response in the

data set is automatically estimated. To reduce run times in BUGS, we therefore format the detection/nondetection data as a vector and thereby avoid having to update missing values. We illustrate this here and show how we can flexibly rewrite the models in BUGS for such data in this "long" format. We analyze Swiss citizen science data of the Middle-spotted Woodpecker (*Dendrocopos medius*; Fig. 4.23) during the 26 breeding seasons of 1990−2015.

This data set is based on checklists which volunteers submit from their bird-watching trips and which are then summarized by site (1-km^2 quadrat) and day, such that we have the number of surveys (= checklists) per quadrat and day and the number among these on which a Middle-spotted Woodpecker was recorded, during each breeding season of 162 days (Julian days 51−212, corresponding to February 20−July 31). Data are available from a total of 144,517 recorded surveys on 116,204 quadrat-day combinations from 1,545 1-km^2 quadrats in which the species was ever recorded since 1985. Both the number of surveys and the number of surveys with the target species recorded have greatly increased over the years (Fig. 4.24).

In our analysis, we reduce run times in four ways: first, we get rid of the NAs that are necessary to fill up a regular array (see above); second, we fit a Binomial instead of a Bernoulli observation model (which allows for the aggregation of Bernoulli trials which have a constant parameter p); and third, we randomly subsample 30% of the full data set. For a "real" analysis in a paper, we would only do the subsampling in the exploratory phases of the analysis during model development and in the

FIGURE 4.23

Middle-spotted Woodpecker eating a cherry, Hagenthal-le-Haut, 2012 *(Photo courtesy of Alex Labhardt).*

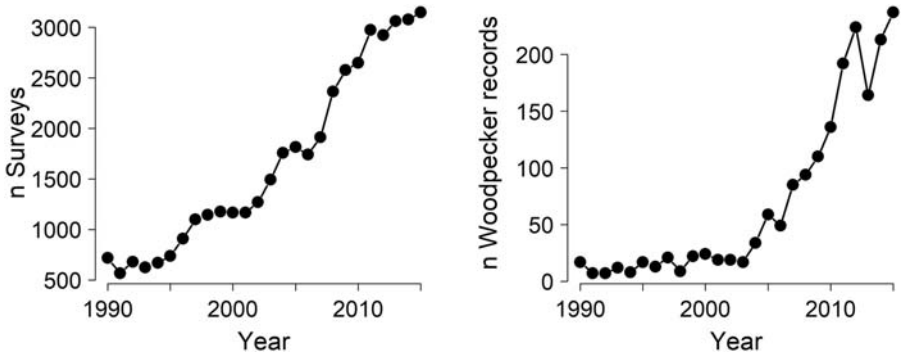

FIGURE 4.24

Annual number surveys in the potential distribution area of the Middle-spotted Woodpecker in Switzerland (left) and number of Middle-spotted Woodpecker records (right).

end run the chosen model for the full data set, but here we don't bother. Developing models for a subsample of a large data set is an important strategy for speeding up analyses (Gelman and Hill, 2007), but one that is only too rarely adopted. Finally, and fourth, since we're interested in temporal occupancy patterns, we restrict the analysis to what could be called the (current) "potential distribution area" of our study species in Switzerland. This means that an occupancy of, say, 20% no longer represents 20% of Switzerland, but rather this percentage refers to some subset of Switzerland that is not clearly defined. We could have carried along all quadrats with volunteer observations in the entire country, but this would have increased the data set tremendously and at no benefit for the purpose of estimation of occupancy trajectories over time.

After subsampling, we end up with data from 1,433 sites only which, as is typical for such citizen-science data sets, are extremely unbalanced. Of the $1{,}433 \times 26$ site/year combinations 73% are empty, i.e., do not have any visits. And of the $1{,}433 \times 26 \times 162$ site/year/day combinations, over 99% are empty; thus, had we tried to fit the model in the usual multi-dimensional array format, we would have had a 100 times larger data set. The number of days (among 162) with recorded visits varied tremendously by site and year, ranging from 0 to 152. The observed occupancy in the set of sites where the species was ever detected after 1985 fluctuated around 5% in the first 15 years, but since 2005 has increased about fourfold (Fig. 4.24).

```
# Read in the data set from AHMbook package
data(spottedWoodpecker)
str(dat <- spottedWoodpecker)            # Look at overview of data set

# Add to data scaled date (such that 0 is 1 May and 1 unit is 1 month)
dat$date <- (dat$jdate - 121) / 30

# Check sample sizes in original data set
nsites <- length(unique(dat$site))       # 1545 sites
nyears <- length(unique(dat$year))       # 26 years
ndays <- length(unique(dat$jdate))       # 162 days in breeding season
```

```
# Randomly thin out the data set by subsampling 30%
dat.full <- dat                          # Make a copy of full data set
prop.data <- 0.3                         # Proportion of data to be used
ncase <- nrow(dat)                       # 116204
set.seed(1, sample.kind = "Rounding")    # Ensures you get the same subset
sel.cases <- sort(sample(1:ncase, ncase * prop.data))
dat <- dat[sel.cases,]                   # Smaller data set

# Look at subsampled data set
str(dat)
'data.frame':   34861 obs. of  8 variables:
 $ site    : int  6 1155 1261 1262 608 741 821 907 1076 1262 ...
 $ coordx  : num  907942 1068942 1095942 1095942 1025942 ...
 $ coordy  : num  55276 169276 186276 187276 171276 ...
 $ year    : int  1990 1990 1990 1990 1992 1992 1992 1992 1992 1992 ...
 $ jdate   : int  51 51 51 51 51 51 51 51 51 51 ...
 $ y       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ nsurveys: int  1 1 1 1 1 1 1 1 1 3 ...
 $ date    : num  -2.33 -2.33 -2.33 -2.33 -2.33 ...

# Have to renumber the sites (since lost some in subsampling)
dat$site <- as.numeric(as.factor(dat[,"site"]))

# Sample sizes in new (subsampled) data set
nsites <- length(unique(dat$site))       # 1433 sites
nyears <- length(unique(dat$year))       # 26 years
ndays <- length(unique(dat$jdate))       # 162 days

# Plot annual total N of records and of middle spotted records (Fig. 4.24)
par(mfrow = c(1,2), mar = c(5,5,4,3), cex.lab = 1.5, cex.axis = 1.5)
plot(1990:2015, tapply(dat$nsurvey, list(dat$year), sum, na.rm = TRUE),
  cex = 2, type = 'b', pch = 16, ylab = 'Number of surveys', xlab = 'Year',
  frame = F)
plot(1990:2015, tapply(dat$y, list(dat$year), sum, na.rm = TRUE),
  cex = 2, type = 'b', pch = 16, ylab = 'Number of middle spotted records',
  xlab = 'Year', frame = F)

# Compute number of middle spotted records per site/year (det. frequency)
table(df <- tapply(dat$y, list(dat$site, dat$year), sum, na.rm = TRUE))

   0    1    2    3    4    5    6    7    8   10
8582 1063  178   40   21    6    9    8    2    3

# Proportion of missing values per site x year combo ?
(prop.NA <- sum(is.na(df)) / prod(dim(df)))
[1] 0.7339632

# Proportion of missing value per site x year X day combo ?
(prop.NA <- 1 - (nrow(dat) / (nsites * nyears * ndays)))
[1] 0.9942243    # This is HUGE !!!

# Compute observed occupancy
zobs <- tapply(dat$y, list(dat$site, dat$year), max, na.rm = TRUE)
zobs[zobs>1] <- 1
psiobs <- apply(zobs, 2, mean, na.rm = TRUE)
```

We then explored seven multi-season models: static and dynamic models, models with linear or quadratic time trends in occupancy, models with fixed or random year effects, and a model with annually varying magnitude of the heterogeneity in site-level detection probability (as per the last section):

- Model 1: Static model with years as blocks (treated as fixed effects)
- Model 2: Static model with years as blocks (treated as random effects)
- Model 3: Static model with a linear trend in occupancy and random yearly deviations
- Model 4: Static model with a quadratic trend in occupancy and random yearly deviations
- Model 5: Dynamic model with fixed effects of year in phi, gamma, and *p*
- Model 6: Dynamic model with random effects of year in phi, gamma, and *p*
- Model 7: As 6, but in addition with annual heterogeneity in site-level extra-dispersion in *p*

Some BUGS analysis ingredients, such as data bundle and inits function, are identical for all models, so we won't repeat them for every model. Here, we only show data bundle and inits and then the analysis for model 7; please check the website for the complete code.

```
# Bundle data (same for all models)
str(bdata <- list(y = dat[,'y'], nsurveys = dat[,'nsurvey'],
  site = dat[,'site'], year = dat[,'year']-1989, date = dat$date,
  nsites = nsites, nyears = nyears, nobs = nrow(dat)) )

List of 8
 $ y       : int [1:34861] 0 0 0 0 0 0 0 0 0 0 ...
 $ nsurveys: int [1:34861] 1 1 1 1 1 1 1 1 1 3 ...
 $ site    : num [1:34861] 6 1092 1192 1193 573 ...
 $ year    : num [1:34861] 1 1 1 1 3 3 3 3 3 3 ...
 $ date    : num [1:34861] -2.33 -2.33 -2.33 -2.33 -2.33 ...
 $ nsites  : int 1433
 $ nyears  : int 26
 $ nobs    : int 34861

# Initial values
zst <- zobs   ;   zst[is.na(zst)] <- 1
inits <- function(){list(z = zst)}
```

When fitting the models to the "vertical" data, the state model remains the same, but the observation model is different: we now have a single loop over all `nobs` observations and need indices to link each observation with the right site and year.

As our final model, we fit an extension of model 6 where we now add site- and year-specific random effects in the detection model: `eps.site`. These are given independent priors for every year, making the associated dispersion parameters (the `sd.p.site[ t]`) fixed effects (we could also fit a model for these dispersion parameters on the log scale, e.g., impose a linear trend or treat them as random effects).

```
# Model 7: Same as 6, with annual heterogeneity in
# site-level extra dispersion in p
# ------------------------------------------------

# Specify model in BUGS language for vertical data format
cat(file = "occmodel7.txt", "
model {
```

```
# Specify priors
psi1 ~ dunif(0, 1)                    # Initial occupancy
for (t in 1:(nyears-1)){              # For survival and persistence
  logit(phi[t]) <- lphi[t]
  lphi[t] ~ dnorm(mu.lphi, tau.lphi)
  logit(gamma[t]) <- lgamma[t]
  lgamma[t] ~ dnorm(mu.lgamma, tau.lgamma)
}
for (t in 1:nyears){                  # For detection parameters
  alpha.lp[t] ~ dnorm(mu.alpha.lp, tau.alpha.lp)
  beta.lp.1[t] ~ dnorm(mu.beta.lp1, tau.beta.lp1)
  beta.lp.2[t] ~ dnorm(mu.beta.lp2, tau.beta.lp2)
}

# Hyperpriors for hyperparameters
mu.lphi <- logit(mean.phi)
mean.phi ~ dunif(0, 1)
mu.lgamma <- logit(mean.gamma)
mean.gamma ~ dunif(0, 1)
mu.alpha.lp <- logit(mean.p)
mean.p ~ dunif(0, 1)
mu.beta.lp1 ~ dnorm(0, 0.1)
mu.beta.lp2 ~ dnorm(0, 0.1)
tau.lphi <- pow(sd.lphi, -2)
tau.lgamma <- pow(sd.lgamma, -2)
tau.alpha.lp <- pow(sd.lp, -2)
tau.beta.lp1 <- pow(sd.beta.lp1, -2)
tau.beta.lp2 <- pow(sd.beta.lp2, -2)
sd.lphi ~ dunif(0, 3)
sd.lgamma  ~ dunif(0, 10)
sd.lp  ~ dunif(0, 1)
sd.beta.lp1  ~ dunif(0, 1)
sd.beta.lp2  ~ dunif(0, 1)

# Annually varying site random effects in detection
for(t in 1:nyears){
  for(i in 1:nsites){
    eps.site[i,t] ~ dnorm(0, tau.p.site[year[i]])
  }
  tau.p.site[t] <- pow(sd.p.site[t],-2)
  sd.p.site[t] ~ dunif(0.001, 10)    # SD's estimated as fixed effects
}

# Ecological submodel: Define state conditional on parameters
for (i in 1:nsites){
  z[i,1] ~ dbern(psi1)
  for (t in 2:nyears){
    z[i,t] ~ dbern(z[i,t-1]*phi[t-1] + (1-z[i,t-1])*gamma[t-1])
  }
}

# Observation model
for (i in 1:nobs){
  logit(p[i]) <- alpha.lp[year[i]] + beta.lp.1[year[i]] * date[i] +
    beta.lp.2[year[i]] * pow(date[i],2) + eps.site[site[i], year[i]]
  y[i] ~ dbin(z[site[i],year[i]]*p[i], nsurveys[i])
}

# Derived parameters
psi[1] <- psi1                        # Population occupancy
n.occ[1] <- sum(z[1:nsites,1])        # Number of occupied sites in sample
for (t in 2:nyears){
  psi[t] <- psi[t-1]*phi[t-1] + (1-psi[t-1])*gamma[t-1]
  n.occ[t] <- sum(z[1:nsites,t])
}
}
")
```

```
# Parameters monitored
params <- c("psi", "phi", "gamma", "n.occ", "mean.phi", "mu.lphi",
  "sd.lphi", "mean.gamma", "mu.lgamma", "sd.lgamma", "mean.p",
  "mu.alpha.lp", "mu.beta.lp1", "mu.beta.lp2", "sd.lp", "sd.beta.lp1",
  "sd.beta.lp2", "alpha.lp", "beta.lp.1", "beta.lp.2", "sd.p.site")

# MCMC settings
na <- 5000  ;  ni <- 50000  ;  nb <- 25000  ;  nt <- 25  ;  nc <- 3

# Call JAGS (ART 832 min), check convergence and summarize posteriors
out7 <- jags(bdata, inits, params, "occmodel7.txt", n.adapt = na,
  n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, parallel = T)
par(mfrow = c(3,3))  ;  traceplot(out7)
print(out7, dig = 2)
```

We compare the estimated occupancy trajectories under the seven models (Fig. 4.25) and plot the annual estimates of phi, gamma, *p*, and the annual magnitude of site-level detection heterogeneity (Fig. 4.26), as well as the season- and year-specific patterns in detection probability (Fig. 4.27, left). It appears that in the broad area of Switzerland where the Middle-spotted Woodpecker occurs, the species is about twice as widespread as what the observed data suggest. This is perhaps not surprising given that detection probability is low during most of the breeding season (though it may be as high as about 0.4 at its start; Fig. 4.27, left). Most models agree with the observed data that there was little if any trend before about 2005, but then came a rapid range expansion. Static models produced more jagged trajectories than did dynamic models, where the Markovian treatment of temporal autocorrelation in occupancy has a smoothing effect. In addition, dynamic models with random year effects (models 6 and 7) have particularly smooth trajectories because the shrinkage in the annual estimates adds smoothing. Estimates of colonization (gamma) did not differ much depending on whether year effects were specified as fixed, as in model 5, or as random, as in model 7 (Fig. 4.26), but they did for

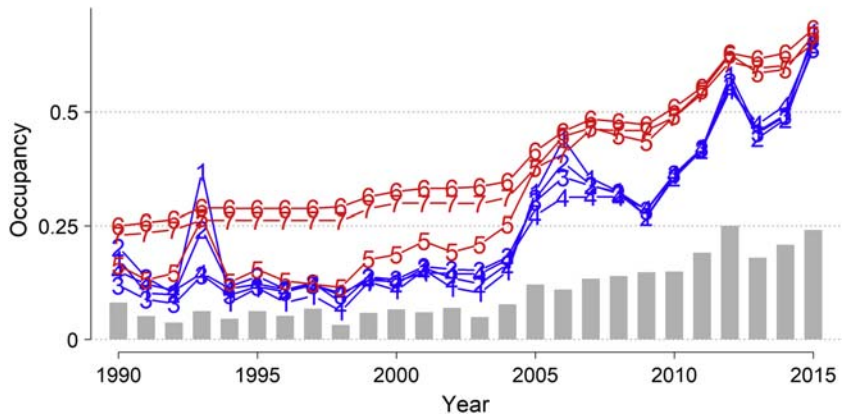

FIGURE 4.25

Observed occupancy and estimated occupancy trajectories of Swiss Middle-spotted Woodpeckers under the seven multi-season models (denoted by model number). Static models are in blue and dynamic models in red; gray bars denote the observed proportion of sites occupied.

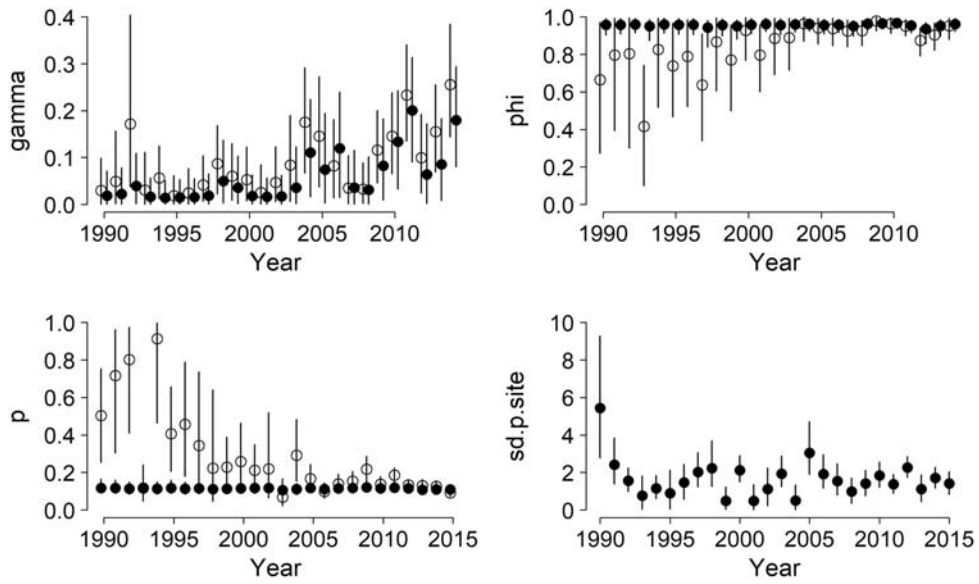

FIGURE 4.26

Annual estimates (with 95% CRIs) of colonization (gamma), persistence (phi), and detection probability (*p*) under fixed-effects model 5 (open circles) and under random-effects model 7 (solid circles), and of the annual site-level detection heterogeneity (*sd.p.site*) under model 7 for Swiss Middle-spotted Woodpeckers.

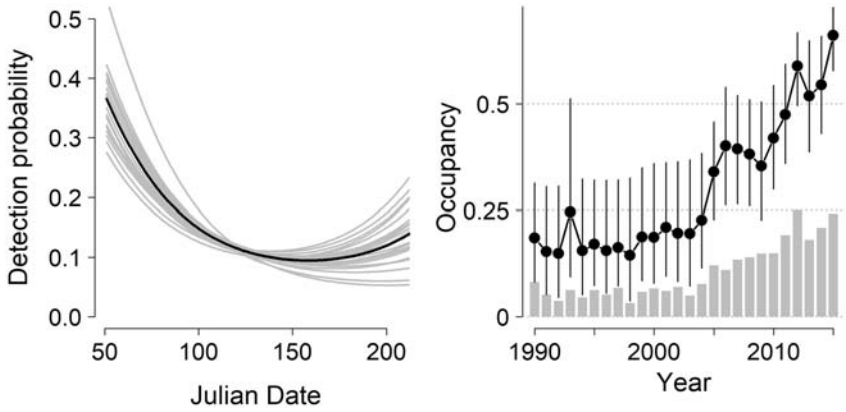

FIGURE 4.27

(Left) Seasonal trajectories of detection probability for every year under model 7 (average is black). (Right) Observed occupancy (gray bars) and model-averaged occupancy trajectory from all seven fitted models (solid circles, with 95% CRIs).

persistence (phi) and detection (p). Up to about 2005, estimates of persistence in the fixed-effects model 5 were very imprecise and much lower than those under the random-effects model 7, where strong shrinkage pulled in all annual estimates so they became almost constant over time. A similar but inverse pattern prevailed for detection, where the fixed-effects model 5 estimated much higher (and much more imprecise) values than did the random-effects model 7, where yearly estimates were again essentially shrunken to values that appeared to be determined mostly by the years after 2005.

Thus, something seems to have happened around 2005 in the distribution dynamics of Swiss Middle-spotted Woodpeckers, but we don't know what this might be. Therefore, perhaps it would be better to have different priors for the annual random effects parameters to allow the mean for the random year effects to change over time. For instance, we could arbitrarily distinguish 5- to 10-year periods, fit a linear or quadratic effect of time, or smooth the annual parameters via a random walk in time. This would then allow us to see in more detail how colonization and extinction, the demographic drivers of distribution change, have evolved over time. We will not do this here though, but you will see an example of such smoothing in the next two sections.

Finally, we found considerable site-level noise in detection probability, but no trend over time in the magnitude of that extra dispersion in p (perhaps ignoring one very large and very imprecise estimate in 1990). Hence, accounting for heterogeneous heterogeneity in site-level detection (in model 7) hardly made any difference in the estimated occupancy trajectory (compared to model 6) of Swiss Middle-spotted Woodpeckers.

Since we did not have any *a priori* reason to prefer one of the models for its estimated occupancy trajectory, we decided to model average the occupancy trajectories with equal weights for all models. Equal model weighting has been found to be fairly good in many cases (Dormann et al., 2018b). We have the same number of posterior samples for every model (3k), so we simply merged them for all models and obtained a total of 21k MCMC samples for psi in every year. Thus, we would call the trajectory in Fig. 4.27 (right) our best guess of how the extent of the Swiss distribution of the Middle-spotted Woodpecker has changed over time.

4.10.3 BRIEF COMMENTS ON THE USE OF OCCUPANCY MODELS FOR CITIZEN SCIENCE DATA

Both static and dynamic occupancy models offer an incredibly useful analytical framework for many citizen science data sets on occurrence because they fully account for the two major kinds of components of observation effort: the *number* of surveys and their "*quality*," where the latter is quantified by detection probability. The models shown in this section are only a start and can be extended in many ways in BUGS, some of which we illustrate later, e.g., to models with temporal smoothing of the annual parameters (in Sections 4.11 and 4.12), modeling of false-positives (Chapter 7), and accounting for spatial autocorrelation (Chapter 9), to name but a few possible extensions of this wonderfully powerful, yet conceptually so simple model.

4.11 ACCOUNTING FOR PREFERENTIAL SAMPLING IN A BIRD POPULATION STUDY

When "sites" in an occupancy analysis are defined in such a way that at most a single pair or individual can occupy it, then occupancy modeling essentially becomes abundance modeling. The dynocc model then enables you to make inferences about population dynamics, in terms of the territory (i.e., site-level) colonization and persistence processes that underlie abundance changes. This may be very helpful to understand the mechanisms that underlie population dynamics, even without being able to

individually identify the animals. That is, the dynocc model can be valuable to enhance your understanding of population dynamics without the need for any marked animals, which are logistically costly and often infeasible to get (MacKenzie et al., 2012; Tempel and Gutierrez, 2013).

There are many ways in which sites may be defined such that they can hold at most a single pair, the most straightforward being to define a site to be the territory of a pair. Many species, especially larger birds such as birds of prey (Newton, 1979, 1986; Ratcliffe, 1993), have well-defined territories that are reasonably stable over time so that we can use occupancy modeling for inferences about abundance (= number of occupied territories) and the colonization and persistence/extinction processes that govern abundance change. There are plenty of population studies especially of birds where the dynocc model may be powerful to understand population dynamics at the level of an occupied territory (Bruggemann et al., 2016; Monneret et al., 2018) or even the individual nest (Bled et al., 2011a; Cruz et al., 2018;). Indeed, the dynocc model appears perfect to apply to data from the countless nest-box studies on small passerines such as the famous great tits (*Parus major*; see Fig. 4.1 in AHM1).

Two frequent challenges, however, are missing values and preferential sampling. First, it is very rare that each site, or territory, is surveyed in every year. Rather, there will be missing values that need to be accounted for to obtain unbiased estimates of population size. Thus, we need to *estimate* the status of a site when it is not surveyed. Fortunately, this does not pose any problems in a Bayesian analysis, since an MCMC algorithm can naturally impute, i.e., estimate, these missing values. This is a considerable benefit for volunteer surveys, which often produce very imbalanced data, as we have just seen in the last section. Second, such missing values due to nonvisitation of a site frequently occur not at random: "better" sites are typically more likely to be surveyed and sites that are less likely to hold a pair or to have breeding success are less likely to be visited. Indeed, in some raptor studies it is even recommended to first target surveys on the "good" sites in every year and only visit less promising sites when there is enough time (Monneret et al., 2018).

When the measurement that would be obtained at a site affects the likelihood with which a site is surveyed and, therefore, the likelihood with which we obtain a missing value for that site, we speak of adaptive, informative, or *preferential sampling* (Conroy et al., 2008; Diggle et al., 2010, Pati et al., 2011, Pacifici et al., 2012, 2016, Conn et al., 2017, Monneret et al., 2018; Dinsdale and Salibian-Barrera, 2019, Gelfand and Shirota, 2019; see also Section 6.4). In the typical situation where "good" sites, or more generally, sites with high values of some response of interest, are more likely to be sampled, we speak of positive preferential sampling (PS). Naive extrapolation to the wider area will then lead to overestimation of means or totals. Hence, in a raptor study with PS, simple extrapolation of the proportion of occupied sites in the sample to the entire sampling frame of all territories will lead to an overestimation of population size. To avoid such bias, as always when we have missing values that are not occurring at random, the missing value-generating process is not ignorable and must be modeled. Here, we have to specify a *joint model* for the response data and the visitation data and make the visitation probability dependent upon some measure of the latent quality of the site that is estimated from the occupancy data (Fig. 4.28). There are different ways in which we can express site quality (i.e., eps.site in the scheme), and they may yield different estimates.

To formalize the notion of PS, we suppose there exists some covariate x_i which is related to the state variable of interest, which in the present case is occupancy at site i, but could also be abundance (Conroy et al., 2008; Wong et al., 2020). The covariate x_i may be observed or latent and might be a habitat metric, or it could be a count index, detection/nondetection, or even estimated occurrence probabilities (Pacifici et al., 2012). The key idea is that we decide to sample a given site or not based on the outcome of this covariate x_i, either in a deterministic or in a stochastic manner. In the former case, this means that we obtain occupancy data only at sites which meet some target outcome of x_i, such as exceeding a threshold,

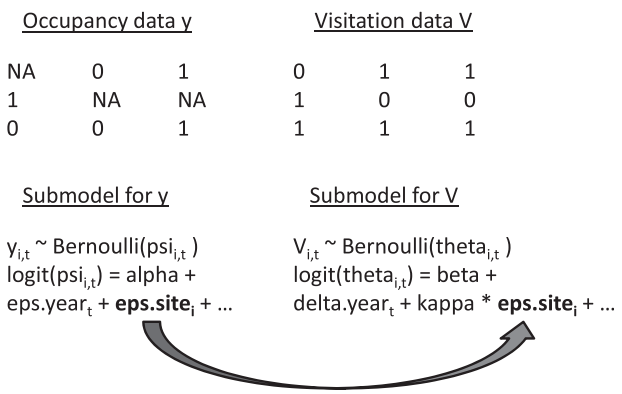

FIGURE 4.28

Schematic of a simple species distribution model that accounts for preferential sampling (PS): a joint logistic regression is specified for species occupancy data and site visitation data, and the probability that a site is visited, theta, depends in part on a measure of latent site quality that is estimated from the occupancy data (here, eps.site_i, though below we also use functions of y). That measure of site quality may be constant or variable over time, i.e., be indexed *i* or (*i, t*). We have positive PS when kappa is positive and (very rarely) negative PS when kappa is negative. For simplicity, this schematic ignores dynamics and imperfect detection.

or detection in a preliminary survey, or based perhaps on the gestalt of a site or some ill-defined prior knowledge, as is the case in our example where we don't have an explicit "trigger." For inference in this biased sampling situation, there are two data sets that arise: (1) the occupancy data for the sites that are formally surveyed which met the sampling criterion and (2) the observed values of the sample covariate itself. The ordinary model we would use for unbiased data also applies to the biased data, as long as we account for the "first stage data" (Pacifici et al., 2016)—i.e., the covariate x_i, which in the context of our current raptor sampling problem is the binary index $V_{i,t}$, where if it takes on the value $V_{i,t} = 1$ we decide to visit the site, and if it takes on the value $V_{i,t} = 0$ we decide to not visit the site. Interestingly, essentially identical models have been developed in two separate lines of research: in the context of formal adaptive sampling on the one hand (Thompson and Seber, 1996; Conroy et al., 2008; Pacifici et al., 2012, 2016; Wong et al., 2020) and for the analysis of opportunistically sampled data on the other hand (Diggle et al., 2010; Pati et al., 2011; Conn et al., 2017; Monneret et al., 2018; Dinsdale and Salibian-Barrera, 2019; Gelfand and Shirota, 2019). In both cases the inference is based on a joint model for the occupancy or abundance data and the visitation or index data, with a link between the two, such that sites with greater occupancy or abundance have a higher visitation probability, or value of the index covariate.

We use the long-term Peregrine Falcon survey in the French Jura mountains conducted by René-Jean Monneret, René Ruffinoni, and their colleagues (Monneret et al., 2018) to illustrate the following topics in the context of a dynocc model:

- We can have single-visit data and still use the powerful machinery of the dynocc model for inferences about the colonization/persistence processes, even when we don't estimate detection probability. We just have to keep in mind the confounding with p and discuss this honestly in our paper or technical report.
- We show how an MCMC analysis can automatically estimate missing values and thus provide estimates of population totals that are corrected for incomplete sampling of a set of sites in a study, i.e., for coverage bias: when not all sites are visited in every year.

- We can smooth a time series of parameters (here, colonization and persistence). Temporal patterns may then become more readily recognizable. A model with such smoothing accommodates the fact that factors affecting population dynamics are likely to act over multiple years and hence these rate parameters will tend to be more similar in two years that are closer in time than in two years further apart. Here, we specify a random-effects model for the first differences in the time series, i.e., a random walk (Link and Barker, 2010). This is an alternative for smoothing a time series to the autoregressive time series smoothing (Johnson and Hoeting, 2003) that you saw in Section 1.5.4.
- And, importantly, we show how you can correct inferences for PS by adopting a joint model of site occupancy and site visitation (see Section 6.4 for another example of modeling of PS). We experiment with three formulations of the link between occupancy and visitation data and will discover that inferences are sensitive to our choice of the formulation of PS.

We analyze a data set on territory occupancy of the magnificent Peregrine Falcon (*Falco peregrinus*; Fig. 4.29) in the French part of the glorious Jura mountains in an area of 12,714 km^2. The data contain the observed occupancy status (1—occupied by a pair, 0—unoccupied, NA—site not visited) over 53 years (1965–2016) for a total of 284 cliffs that held an adult pair at least once. Most sites were checked multiple times during a breeding season (early March to late June), but the visit-specific results are not available, only the aggregate over a breeding season. This precludes estimation of detection probability using standard methods (but see Section 4.8.1). Multiple visits by experienced observers will lead to a high combined detection probability; hence, here we model the unreplicated, aggregate data under the assumption that detection is nearly perfect, knowing that any violation of this assumption will bias all our estimates.

```
# Load the data set from AHMbook
data(FrenchPeregrines)
str(dat <- FrenchPeregrines)

# Extract data for modeling
ain <- which(dat$department == 'Ain')                    # Sites in Dep. Ain
jura <- which(dat$department == 'Jura')                  # Sites in Dep. Jura
doubs <- which(dat$department == 'Doubs')                # Sites in Dep. Doubs
ht <- as.numeric(dat$height)                             # Cliff height
y <- as.matrix(dat[,4:56])                               # Detection/Nondetection data
nsites <- nrow(y)
nyears <- ncol(y)
year <- 1964:2016

# Produce some summaries, including ratio estimator
n.occ.obs <- apply(y, 2, sum, na.rm = TRUE)              # Observed N pairs
n.visited <- apply(y, 2, function(x) sum(!is.na(x)))     # Number of visited sites
n.ratio <- n.occ.obs / (n.visited / 284)
```

As a result of the pesticide crash (Ratcliffe, 1993) the number of peregrine pairs declined to a low of only 17 observed pairs in 1972 and then recovered to a maximum of 196 pairs in 2008, after which it declined again. However, the number of surveyed sites doubled from less than 40% to almost 90% over this time period (Fig. 4.30). Hence, owing to this coverage bias the observed number of pairs can clearly not represent the total population size in the French Jura. In addition, PS is evident in the raw data (Fig. 4.31): there is a significantly positive correlation between the proportion of years when a site was observed to be occupied (after it became "recruited" into the survey) and the proportion of years it was surveyed.

FIGURE 4.29

A gorgeous tiercel of the Peregrine Falcon (*Falco peregrinus*), Vosges, France *(Photo courtesy of Vincent Michel)*.

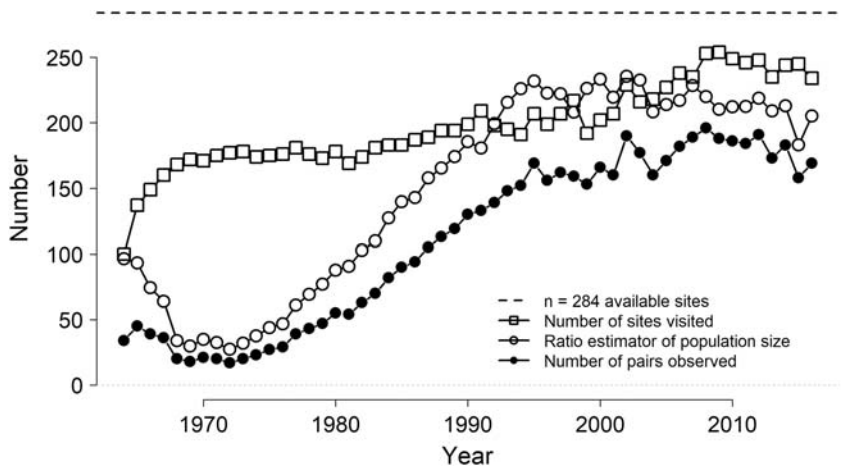

FIGURE 4.30

Number of sites visited, observed, and estimated number of adult peregrine pairs in the French Jura 1964–2016 among the known total of 284 sites. The ratio estimator assumes that sites are surveyed at random.

FIGURE 4.31

Preferential sampling in the French peregrine population survey. There is a significant positive correlation between the proportion of years when a site was visited and the proportion of years when it was found occupied ($r = 0.24$, $p < 0.001$).

```
# Compute visitation data V
V <- y  ;  V[V == 0] <- 1  ;  V[is.na(V)] <- 0
y[1:5, 1:5]  ;  V[1:5, 1:5]    # Look at parts of y and V

     yr1964 yr1965 yr1966 yr1967 yr1968
[1,]      1      1      1      1      1
[2,]     NA     NA     NA     NA     NA
[3,]     NA     NA     NA     NA     NA
[4,]     NA     NA      0      0      0
[5,]     NA     NA     NA     NA     NA

     yr1964 yr1965 yr1966 yr1967 yr1968
[1,]      1      1      1      1      1
[2,]      0      0      0      0      0
[3,]      0      0      0      0      0
[4,]      0      0      1      1      1
[5,]      0      0      0      0      0

# Compute prop. years with pairs (given surveyed) and of years with
# surveys and plot, both starting from first ever visit of a site
first.visit <- apply(!is.na(y), 1, function(x) which(x)[1])
prop.visit <- prop.pair <- numeric(length = nsites)
for(i in 1:nsites){
  prop.visit[i] <- mean(V[i, first.visit[i]:nyears],na.rm = TRUE)
  prop.pair[i] <- mean(y[i, first.visit[i]:nyears],na.rm = TRUE)
}


# Fig. 4.31
par(mar = c(5,5,4,2), cex.lab = 1.5, cex.axis = 1.5)
plot(prop.pair, prop.visit, xlab = 'Proportion of years occupied',
  ylab = 'Proportion of years visited', pch = 16, cex = 2, ylim = c(0,1),
  frame = FALSE, col = rgb(0,0,0,0.4))
```

One main aim of our analysis is to correct for this coverage bias. The simplest approach would be a ratio estimator given by the observed number of occupied divided by the proportion of visited sites. This approach assumes that sites are visited at random in every year. This simple correction for coverage bias

suggests a much more drastic decline at the beginning of the study period and a more rapid increase after the crash, and around 50 missed pairs after recovery over those detected (Fig. 4.30).

The ratio estimator does not provide any information about demographic mechanisms underlying the dynamics of the population; hence, we next fit four dynocc models, three of which accommodate PS. The three models with PS have two linked submodels: one for the detection/nondetection data (y) and the other for the visitation data (V), and they differ only by what part of the occupancy model they use as a predictor for visitation probability. Model 2 has two sets of random site effects, one for persistence and the other for colonization, and each of them is allowed to affect the visitation probability in an additive manner. Models 3 and 4 use a function of the detection/nondetection data y as a predictor for visitation probability: Model 3 uses $y_{i,t-1}$, while model 4 uses the average y over the past 10 years.

The (sub)model for the detection/nondetection data is identical in all four models. In addition to effects of cliff-height (a factor with three levels), there are random year effects in persistence and colonization, with a random-walk smoother to better evaluate temporal patterns in these key parameters. We specify this random walk in a hierarchically centered manner, i.e., as `lphi.year[ t] ~ dnorm(lphi.year[ t-1], tau.eps.lphi)`. We found this yields much better mixing than a noncentered parameterization such as `lphi.year[ t] <- lphi.year[ t-1] + eps.lphi [ t-1]`. In general, we have noticed terrible mixing for some models with PS and sometimes no mixing at all. Initializing the chains at some of the solutions from model 1 has helped though. However, we caution that we need a better understanding of these models, for example, to find out whether there are local extrema or ridges in the likelihood (see also comments on this in Section 6.4). We here show code only for models 1 and 2; as always, you can find the full code on the website.

```
# Data bundle
str(bdata <- list(y = y, height = ht, nsites = nsites, nyears = nyears,
  ain = ain, jura = jura, doubs = doubs, year = ((1964:2016)-1990) / 26))

List of 8
 $ y     : int [1:284, 1:53] 1 NA NA NA NA NA NA NA 1 1 ...
 $ height: num [1:284] 2 2 1 3 1 3 2 3 3 3 ...
 $ nsites: int 284
 $ nyears: int 53
 $ ain   : int [1:93] 1 2 3 4 5 6 7 8 9 10 ...
 $ jura  : int [1:89] 94 95 96 97 98 99 100 101 102 103 ...
 $ doubs : int [1:102] 183 184 185 186 187 188 189 190 191 192 ...
 $ year  : num [1:53] -1 -0.962 -0.923 -0.885 -0.846 ...

# Model 1: no preferential sampling (PS)
# Specify model in BUGS language
cat(file = "dynocc1.txt", "
model {

# Priors and models for parameters
psi1 ~ dbeta(1, 1)                  # Initial occupancy

# Model for phi and gamma: cliff height + site + smooth random year effects
for (i in 1:nsites){
  for(t in 1:(nyears-1)){
    logit(phi[i,t]) <- lphi[i,t]
    lphi[i,t] <- lphi.site[i] + lphi.year[t]
    logit(gamma[i,t]) <- lgamma[i,t]
    lgamma[i, t] <- lgamma.site[i] + lgamma.year[t]
  }
  lphi.site[i] ~ dnorm(alpha.lphi[height[i]], tau.lphi.site)
  lgamma.site[i] ~ dnorm(alpha.lgamma[height[i]], tau.lgamma.site)
}
```

```
# Priors for phi and gamma intercepts
for(k in 1:3){
  alpha.lphi[k] <- logit(initial.phi[k])
  initial.phi[k] ~ dbeta(1, 1)
  alpha.lgamma[k] <- logit(initial.gamma[k])
  initial.gamma[k] ~ dbeta(1, 1)
}
tau.lphi.site <- pow(sd.lphi.site, -2)
sd.lphi.site ~ dunif(0, 2)
tau.lgamma.site <- pow(sd.lgamma.site, -2)
sd.lgamma.site ~ dunif(0, 2)

# Priors for year effects on phi and gamma with rw smoothers
lphi.year[1] <- 0                  # Set to zero to avoid overparameterization
lgamma.year[1] <- 0
for (t in 2:(nyears-1)){
  lphi.year[t] ~ dnorm(lphi.year[t-1], tau.eps.lphi)
  lgamma.year[t] ~ dnorm(lgamma.year[t-1], tau.eps.lgamma)
}
tau.eps.lphi <- pow(sd.eps.lphi,-2) # Hyperpriors for variances
sd.eps.lphi ~ dunif(0, 1)
tau.eps.lgamma <- pow(sd.eps.lgamma,-2)
sd.eps.lgamma ~ dunif(0, 1)

# Ecological and observation submodels confounded (no p)
for (i in 1:nsites){
  y[i,1] ~ dbern(psi1)
  for (t in 2:nyears){
    y[i,t] ~ dbern(y[i,t-1]*phi[i,t-1] + (1-y[i,t-1])*gamma[i,t-1])
  }
}

# Derived parameters
# Population occupancy and population size
psi[1] <- psi1
n.occ[1] <- sum(y[1:nsites,1])
for (t in 2:nyears){
  n.occ[t] <- sum(y[1:nsites,t])     # Number of occupied sites
}
# Year-specific average values of phi and gamma
for(t in 1:(nyears-1)){
  mean.phi.year[t] <- mean(phi[,t])
  mean.gamma.year[t] <- mean(gamma[,t])
}
# Average gamma and phi per cliff height category in central year (1989)
for(k in 1:3){
  logit(gamma.cliff[k]) <- alpha.lgamma[k] + lgamma.year[26]
  logit(phi.cliff[k]) <- alpha.lphi[k] + lphi.year[26]
}

# Population size in each French Jura Departement
for (t in 1:nyears){
  n.ain[t] <- sum(y[ain, t])
  n.jura[t] <- sum(y[jura, t])
  n.doubs[t] <- sum(y[doubs, t])
}
}
")

# Initial values
inits <- function(){ list(psi1 = runif(1))}
```

```
# Parameters monitored
params <- c("psi1", "alpha.lphi", "initial.phi", "alpha.lgamma",
  "initial.gamma", "lphi.year", "lgamma.year", "mean.phi.year",
  "mean.gamma.year", "sd.eps.lphi", "sd.eps.lgamma", "gamma.cliff",
  "phi.cliff", "n.occ", "n.ain", "n.jura", "n.doubs", "y")

# MCMC settings
na <- 1000  ;  ni <- 20000  ;  nt <- 10  ;  nb <- 10000  ;  nc <- 3

# Call JAGS (ART 43 min), check convergence and summarize posteriors
out1 <- jags(bdata, inits, params, "dynocc1.txt", n.adapt = na, n.chains = nc,
  n.thin = nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(3,3))   ;   traceplot(out1)
summary(out1)   ;   jags.View(out1)  # not shown
```

Next is a model that accommodates PS. The first submodel is identical to model 1, but then there is a second submodel, which is linked to the first via two site random effects in persistence and colonization probability that are estimated in the former.

```
# Model 2: modeling preferential sampling (PS) by joint modeling of
# occupancy and visitation
# Specify model in BUGS language
cat(file = "dynocc2.txt", "
model {

# Submodel 1: model for detection/nondetection data (y)
# ------------------------------------------------

# all this first part is identical to dynocc1 above and omitted here

# Submodel 2: model for whether a site is visited or not (V)
# ------------------------------------------------------------
# Priors and linear models
# Including an effect of past occupation history in visitation from t=2
for (i in 1:nsites){
  for (t in 1:nyears){
    theta[i,t] <- ilogit(alpha.visit + beta.visit[1] * year[t] +
      beta.visit[2] * pow(year[t],2) + beta.visit[3] * pow(year[t],3) +
      kappa.lphi * lphi.site[i] + kappa.lgamma * lgamma.site[i])
  }
}
alpha.visit <- logit(theta.int)
theta.int ~ dbeta(1, 1)
for(v in 1:3){                          # Coefficients of time
  beta.visit[v] ~ dnorm(0, 0.1)
}
kappa.lphi ~ dnorm(0, 0.5)              # first coefficient for PS
kappa.lgamma ~ dnorm(0,0.5)             # second coefficient for PS
# curve(dnorm(x, 0, sqrt(1/0.5)), 0, 20) # howsit look like ?

# Logistic regression for visits (V)
for (i in 1:nsites){
  for (t in 1:nyears){
    V[i,t] ~ dbern(theta[i,t])
  }
}
}
")
```

```
# Initial values
# initialize at solutions of model 1 without PS and with a guess of kappa
# (For this, you have to run model 1 beforehand)
tmp <- out1$mean
inits <- function(){list(psi1 = tmp$psi1, initial.phi = tmp$initial.phi,
  initial.gamma = tmp$initial.gamma, sd.eps.lphi = tmp$sd.eps.lphi,
  sd.eps.lgamma = tmp$sd.eps.lgamma, kappa.lphi = 3, kappa.lgamma = 3)}

# Parameters monitored
params <- c("psi1", "alpha.lphi", "initial.phi", "sd.lphi.site",
  "alpha.lgamma", "initial.gamma", "sd.lgamma.site", "lphi.year",
  "lgamma.year", "mean.phi.year", "mean.gamma.year", "sd.eps.lphi",
  "sd.eps.lgamma", "gamma.cliff", "phi.cliff", "alpha.visit", "beta.visit",
  "kappa.lphi", "kappa.lgamma", "n.occ", "n.ain", "n.jura", "n.doubs",
  "lphi.site", "lgamma.site", "y")

# MCMC settings
na <- 1000  ;  ni <- 300000  ;  nt <- 150  ;  nb <- 150000  ;  nc <- 3

# Call JAGS (ART 18 h), check convergence and summarize posteriors
out2 <- jags(bdata, inits, params, "dynocc2.txt", n.adapt = na, n.chains = nc,
  n.thin = nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(2,2))   ;   traceplot(out2)
summary(out2)   ;   jags.View(out2)      # not shown
```

There was clear evidence for positive PS in the French Jura peregrine survey under all three models that did allow for PS (posterior means and 95% CRIs):

- Model 2: kappa.lphi: 3.87 (3.18, 4.69); kappa.lgamma: -0.92 (-1.39, -0.52)
- Model 3: kappa: 3.20 (3.05, 3.36)
- Model 4: kappa: 5.54 (5.10, 5.63)

The negative sign of the effect of estimated site effects in colonization on visitation is somewhat puzzling but seems to be more than compensated for by the strong positive effect of the site effects on persistence. We are particularly interested in the following inferences: population size corrected for coverage bias and the trajectories of persistence and colonization.

We find that the dynocc model 1 without PS yields population size estimates that resemble the ratio estimator and that model 2 with PS expressed via two constant measures of site quality yields very similar estimates as well. In contrast, the two models that express PS via the results of previous surveys yield much lower population size estimates (Fig. 4.32). In the absence of a model selection criterion, we decide to simply average the estimates under the three PS models. We estimate that in any given year between 2 and 35 pairs were missed due to coverage bias and that the proportion of pairs detected ranged from 51 to 96%. Persistence probability recovered very quickly after the pesticide crash and likely reflect individual survival probability (Fig. 4.33), while colonization probability increased much more slowly over time and likely reflect the size and fecundity of the population. Correction for PS reduced persistence estimates especially at the very start of the study, while colonization estimates were fairly unaffected. Both persistence and colonization probability were higher for the tallest cliffs. Fig. 4.34 shows a key result from the analysis: the model-averaged (over the three PS models)

estimates of the full detection/nondetection matrix y, which we interpret as the presence/absence matrix z under the assumption that detection error is negligible. We could sum over regions within the study area to obtain regional population size estimates, such as for the three French Jura departments individually (as shown in the model code).

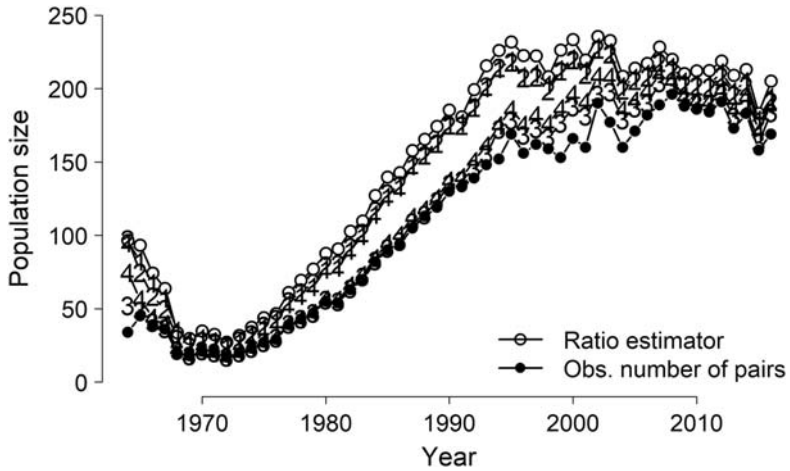

FIGURE 4.32

Peregrine population size estimation in the French Jura accounting for coverage bias and preferential sampling (PS). Numbers sandwiched between the ratio estimator and observed population size give the trajectories of estimates under model 1 (without PS) and models 2–4 (with three formulations of PS). This analysis does not correct for imperfect detection.

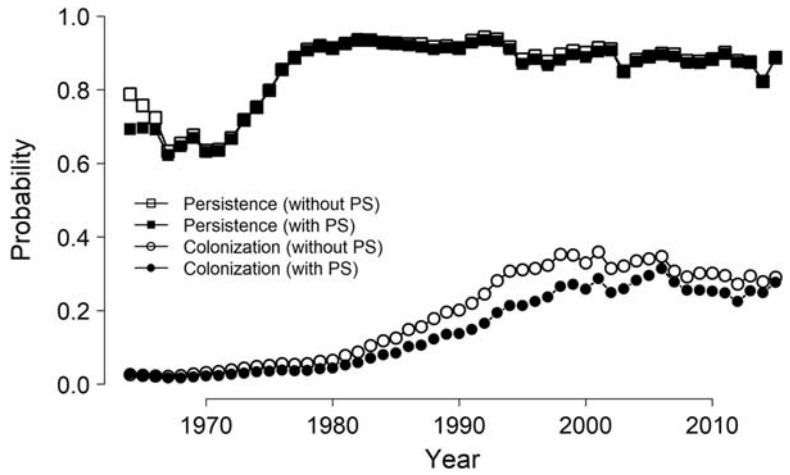

FIGURE 4.33

Estimated trajectories of persistence and colonization probabilities under the simple dynocc model 1 and averaged for models 2 and 3. Analysis does not correct for imperfect detection.

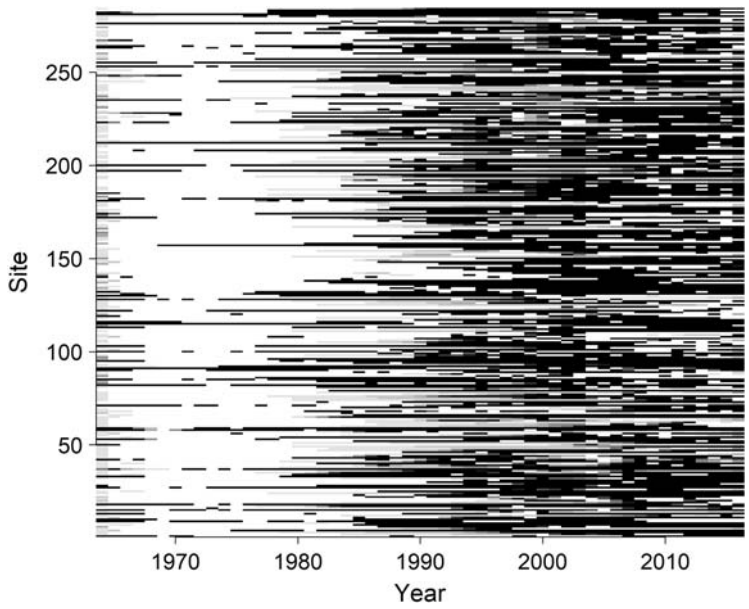

FIGURE 4.34

Estimated detection/nondetection matrix model-averaged for models 2—4 in an analysis that does not correct for imperfect detection. White denotes absence, gray increasing degrees of certainty of presence, and black is certain presence. Under the assumption of negligible detection error, this matrix can be interpreted as the true presence/absence matrix.

```
# Population size comparisons in a table (not all printed out)
# This code requires fits from all 4 models
n.occ.ma <- colMeans(rbind(out2$sims.list$n.occ, out3$sims.list$n.occ,
  out3$sims.list$n.occ))  # Model average population size estimates from 3 PS models
print(cbind(n.visited, n.occ.obs, n.ratio, 'model1' = out1$mean$n.occ,
  'model2' = out2$mean$n.occ, 'model3' = out3$mean$n.occ, 'model4' = out4$mean$n.occ,
  'PS average' = n.occ.ma, 'missed' = n.occ.ma - n.occ.obs,
  'Prop.detected' = n.occ.obs / n.occ.ma), 2)
```

```
       n.visited n.occ.obs n.ratio model1 model2 model3 model4 PS average missed Prop.detected
yr1964       100        34      97     95     98     52     75         67   33.3          0.51
yr1965       137        45      93     83     75     49     56         57   12.3          0.79
yr1966       149        39      74     68     59     41     44         47    7.7          0.83
yr1967       160        36      64     57     50     37     39         41    5.0          0.88
yr1968       168        20      34     35     30     22     22         25    4.5          0.82
yr1969       172        18      30     28     25     18     18         20    2.3          0.89
yr1970       171        21      35     30     28     22     21         24    2.8          0.88
.......
yr2010       249       186     212    203    202    190    192        194    7.8          0.96
yr2011       246       184     212    203    201    190    191        193    9.4          0.95
yr2012       248       191     219    211    208    196    199        200    9.4          0.95
yr2013       235       173     209    198    196    181    186        186   12.8          0.93
yr2014       244       183     213    205    202    186    194        192    8.5          0.96
yr2015       245       158     183    179    175    163    171        167    9.0          0.95
yr2016       234       169     205    194    191    184    189        186   17.2          0.91
```

```
# Model-averaged phi and gamma per cliff-height category
       post.mean phi  2.5% 97.5% post.mean gamma   2.5% 97.5%
low            0.897 0.842 0.937           0.162 0.0878 0.312
medium         0.886 0.797 0.938           0.143 0.0597 0.342
tall           0.943 0.914 0.966           0.305 0.1788 0.549
```

Thus, the dynocc model holds considerable promise for studying "population dynamics without marked animals" (MacKenzie et al., 2012). Accounting for PS will often be important in such studies, but arguably even more so for species distribution modeling efforts based on opportunistic records. Understanding of PS is still in its infancy, however, but will likely increase over the coming years, given the ubiquity of data sets that are affected by such nonrandom sampling processes and the need to correct for it to avoid biased inferences.

4.12 A DEMOGRAPHIC DYNAMIC OCCUPANCY MODEL

In the modeling of occurrence you can never "escape" abundance: that is, both occurrence and detection will be related to the underlying abundance. For instance, a locally more common species will often tend to also be more locally widespread, and there is almost always a spatial or temporal signal of abundance in detection probability, which tends to be higher when abundance is high. This latter relationship can even be formalized to estimate abundance from detection/nondetection data as in the Royle-Nichols model (Section 6.13.1), which Rossman et al. (2016) have extended to the dynocc model to estimate the parameters of the Dail-Madsen model from detection/nondetection data considered in this chapter.

In the dynocc example in the previous section we already had a tight connection between abundance and occurrence because each cliff site could only be occupied by a single peregrine pair. Hence, the number of occupied sites was identical to abundance at the set of studied sites. However, this equality does not hold for the dynamics rates (site colonization and persistence) because a colonization and an extinction event may represent multiple survival or recruitment events at the level of the individual bird occupying a site. Excitingly, though, we can write the former as a function of the latter and then estimate individual vital rates from site-occupancy data!

Roth and Amrhein (2010) have developed such a variant of a dynamic occupancy model which makes the abundance-occupancy relationship particularly clear. They express the probability that a territory is occupied as a function of the individual "local" survival of the territory holder and a recruitment parameter. "Local survival" is the product of the probabilities to remain on a given territory and to survive, given that a bird remains in the same territory. With simulated data and in comparison to capture-recapture data from their Nightingale population study, they show that their model provides excellent estimates of apparent survival.

The model makes the assumption that a territory can only be occupied by a single bird, or rather, that we only record territory-holders of one sex, typically males. Between occasions, a bird may die or permanently leave *the territory* with probability $1 - \phi$, where ϕ is "local survival," which can be thought of as an extreme form of the "apparent survival" in the CJS model (see Chapter 3). An empty territory or a territory whose owner has just died or emigrated may be recolonized with recruitment probability r. It makes sense to allow for such a "rescue effect" and to assume that during the same interval a territory owner may die/leave permanently and a new bird may recruit immediately, such that the territory continues to be occupied, albeit by two different birds. The information to separate the two cases of continued occupancy by the same bird and of a "rescue event" will come mostly from sites that change their occupancy status from 0 to 1, i.e., that become colonized. Hence, parameters in this model will not be estimable when all territories are always occupied. In addition, since Roth and Amrhein (2010) specify their model as a conditional model, i.e., they condition on the year of first detection, there is hardly any information on recruitment during the first interval. It is possible to specify an unconditional model too, but we illustrate the conditional model here.

During the first year in which territory i is detected (f_i), we know its state for certain.

$$z_{i,f_i} \sim Bernoulli(1)$$

In later years, whether a territory is occupied or not will depend on three things: whether it was occupied or not during the year before and on the probabilities of "local survival" of the territory owner and of recruitment.

$$z_{i,t}|z_{i,t-1} \sim Bernoulli\left(z_{i,t-1} * (\phi + (1-\phi) * r) + \left(1 - z_{i,t-1}\right) * r\right)$$

Hence, if a territory is occupied during year $t-1$ we have $z_{i,t-1} = 1$ and the territory will still be occupied with probability ϕ or with probability $(1-\phi) * r$. The former denotes the case when the same bird survives, while the latter denotes a "rescue effect" where the owner dies, but the territory becomes reoccupied by a new bird during the same interval. Finally, if the territory is not occupied during year $t-1$, it will become occupied in year t with probability r. Thus, the model assumes that during one interval, never more than a single exit and entry event may occur. We presume that estimates of survival will be biased high and of recruitment low when multiple exit and entry events occur during one interval. Roth and Amrhein (2010) formulate their model with two latent variables, where one is the true latent state of territory occupancy and the other is a "fidelity state," indicating whether the same or a different owner is in an occupied territory. Our model here is a simplified version of their original model.

To be able to estimate imperfect detection of the individual birds and therefore of territory occupancy, as usual we need replicate surveys to at least some territories in at least some of the years during a relatively short period such that the closure assumption is tenable (or time-to-detection measurements). We can then adopt the usual Bernoulli measurement error model with p being the detection probability which in this case refers both to an individual and to the territory (because each territory is represented by a single individual only) and $y_{i,j,t}$ is the detection/nondetection observation at site i during replicate j in year t:

$$y_{i,j,t}|z_{i,t} \sim Bernoulli\left(z_{i,t} * p\right)$$

Thus, in the conditional model, we have three probability parameters: local survival ϕ, recruitment r, and detection p, while in the unconditional model we would have the initial proportion of occupied territories as an additional parameter. All can be modeled, i.e., we can allow survival and recruitment to differ by site and annual interval, and detection by site, interval, and individual visit, and then model them with covariates or random effects. We will illustrate the model with full time-dependence using simulated data produced by function `simDemoDynocc` from the `AHMbook` package. Here is the function call with explicit default arguments and where phi, r, and p are made year-dependent, by specification of a range within which annual values will be drawn from Uniform distributions.

```
simDemoDynocc(nsites = 100, nyears = 10, nvisit = 5, psi1 = 0.6,
  range.phi = c(0.2, 0.9), range.r = c(0, 0.4), range.p = c(0.1, 0.9),
  show.plot = TRUE)

# Some functionality of function simDemoDynocc
str(data <- simDemoDynocc(psi1 = 1))                # All sites initially occupied
str(data <- simDemoDynocc(nsites = 1000))           # Plenty more sites
str(data <- simDemoDynocc(nyears = 100))            # Plenty more years
str(data <- simDemoDynocc(nvisit = 20))             # Plenty more visits
str(data <- simDemoDynocc(range.phi = c(0.8, 0.8))) # Constant survival
str(data <- simDemoDynocc(range.phi = c(0.2,0.3), range.r = c(0,0.2))) # Decline
str(data <- simDemoDynocc(range.phi = c(0.8,1), range.r = c(0.5,0.7))) # Increase
str(data <- simDemoDynocc(nvisit = 1))              # Single visit
str(data <- simDemoDynocc(range.p = c(1,1)))        # Perfect detection
```

We generate a single data set with 100 sites and 20 years and fit the time-dependent model.

```
# Generate a data set with year-specific parameters
set.seed(24)
str(data <- simDemoDynocc(psi1 = 0.6, nsites = 100, nyears = 20, nvisit = 5,
  range.phi = c(0.1, 0.9), range.r = c(0, 0.5), range.p = c(0.1, 0.9)))

# Bundle and summarize data
str(bdata <- list(y = data$y, nsites = data$nsites, nyears = data$nyears,
  nvisit = data$nvisit, first = data$f) )

List of 5
 $ y     : int [1:100, 1:5, 1:20] 1 1 1 0 1 1 1 0 1 1 ...
 $ nsites: num 100
 $ nyears: num 20
 $ nvisit: num 5
 $ first : num [1:100] 1 1 1 5 1 1 1 8 1 1 ...

# Specify model in BUGS language
cat(file = "DemoDynocc1.txt", "
model {

# Priors
for(t in 1:(nyears-1)){
  phi[t] ~ dunif(0,1)
  r[t] ~ dunif(0,1)
  p[t] ~ dunif(0,1)         # only nyears-1 p params in conditional model !
}

# Likelihood
# Alive/dead process specified conditional on first detection
for(i in 1:nsites){
  z[i, first[i]] ~ dbern(1) # Condition on year of first detection
  for(t in (first[i]+1):nyears) {
    z[i, t] ~ dbern(z[i,t-1] * (phi[t-1] + (1-phi[t-1]) * r[t-1]) +
      (1-z[i,t-1]) * r[t-1])
  }
}

# Observations conditional on Alive/dead process
for(i in 1:nsites){
  for(t in (first[i]+1):nyears) {
    for(j in 1:nvisit){
      y[i,j,t] ~ dbern(z[i,t] * p[t-1])
    }
  }
}
}
")

# Initial values
zst <- zinit(apply(bdata$y, c(1,3), max))
inits <- function() list(z = zst)

# Parameters monitored
params <- c("phi", "r", "p")

# MCMC settings
na <- 1000  ;  ni <- 6000   ;   nt <- 4   ;   nb <- 2000   ;   nc <- 3
```

```
# Call JAGS (ART 2 min), check convergence and summarize posteriors
out1 <- jags(bdata, inits, params, "DemoDynocc1.txt", n.adapt = na,
  n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, parallel = T)
par(mfrow = c(2,2))   ;   traceplot(out1)
print(out1, dig = 2)
```

We see a fair coincidence of the dashed and the solid lines in Fig. 4.35. Estimates of annual survival are fairly imprecise, while those for recruitment are a little better and those for detection better still in this respect. It might therefore pay to reduce model complexity by introduction of yearly covariates or temporal random effects. A small simulation study with 100 data sets (not shown) suggested essentially no bias for detection, but a slight positive bias for small values of survival and recruitment and a slight negative bias for large values of these parameters.

We think that this model is an important one, both for conceptual and for practical reasons. First, it may help focusing our thinking on the individual demography that always underlies the dynamic rate parameters in the dynocc model and may provide motivation to develop more models that take detection/nondetection data and estimate parameters of abundance and/or the demography of individuals. Second, it is potentially a very useful model because it enables us to obtain estimates of two key demographic parameters, survival and recruitment, from fairly "cheap" occupancy data without the need to mark and/or individually recognize individuals. We are surprised that the model does not seem to have been applied much since its first development and would hope for this to change somewhat in the future. As always, it might be a good idea if possible to combine territory occupancy data with, say, capture-recapture data, which directly inform about apparent survival in an integrated population model (Besbeas et al., 2002; Brooks et al., 2004; see Chapter 10). One would then have to reconcile "local" and apparent survival, perhaps via some "territory emigration rate", although sometimes we would expect them to be approximately identical, as they seemed to be in the case study of Roth and Amrhein (2010). Another useful combination of the Roth-Amrhein model for detection/nondetection data might be with a Dail-Madsen model for replicated counts (Zipkin et al., 2017).

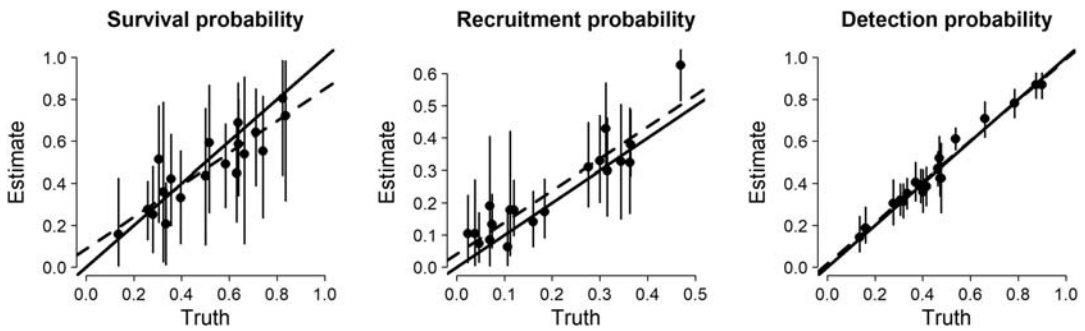

FIGURE 4.35

Estimated versus true values of the time-dependent parameters of local survival, recruitment, and detection in the demographic occupancy model of Roth and Amrhein (2010) for a single simulated data set. The 1:1 lines are solid and the linear regressions of estimates on truth are shown as a dashed line. Errors are 95% CRIs.

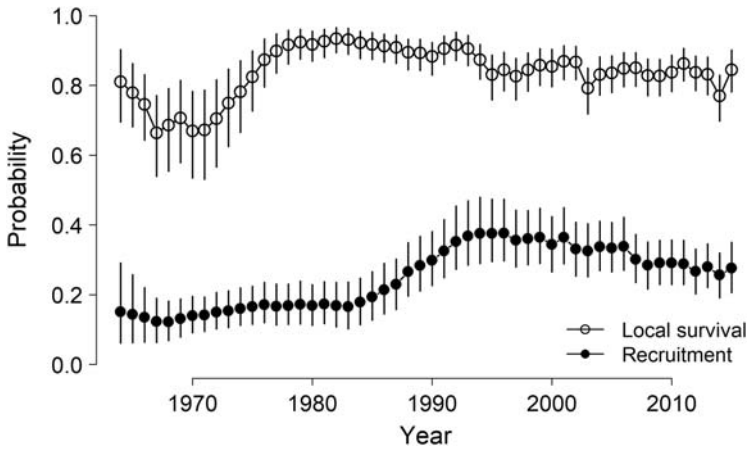

FIGURE 4.36

Estimates of local survival and recruitment of peregrines in the French Jura mountains between 1964 and 2016 under a variant of the Roth and Amrhein (2010) demographic occupancy model.

For illustration, we next apply the model to the French peregrines where, strictly, we have adult pairs rather than individual (male) territory holders in the original application by Roth and Amrhein (2010). However, it would appear that the observation of an adult pair is equivalent to observation of, say, just the female, so unless local survival and recruitment are different when there is no bird present or one bird already present, it should be possible to use also pair data to fit the model and estimate rates that represent averages for the sexes. In this application we drop the observation process, and we again smooth the vital rates by a random walk. We see that we obtain fairly similar patterns of the (individual) vital rates as for the site-level probability of persistence and colonization (Fig. 4.36).

```
# Data loading from AHMbook and preparation
data(FrenchPeregrines)
fp <- FrenchPeregrines
y <- as.matrix(fp[,4:56])
f <- apply(y, 1, function(x) min(which(x!=0)))
f[f == 'Inf'] <- ncol(y)

# Bundle and summarize data
str(bdata <- list(y = y, nsites = nrow(y), nyears = ncol(y), first = f) )
List of 4
 $ y     : int [1:284, 1:53] 1 NA NA NA NA NA NA NA 1 1 ...
 $ nsites: int 284
 $ nyears: int 53
 $ first : int [1:284] 1 39 35 12 32 24 22 23 1 1 ...
```

```
# Specify model in BUGS language
cat(file = "DemoDynocc2.txt", "
model {

# Priors
phi[1] ~ dunif(0, 1)         # Survival in first interval
r[1] ~ dunif(0, 1)           # Recruitment in first interval
lphi[1] <- logit(phi[1])
lr[1] <- logit(r[1])
for(t in 2:(nyears-1)){
  logit(phi[t]) <- lphi[t]
  logit(r[t]) <- lr[t]
}

# Random-walk smoother
for (t in 2:(nyears-1)){    # Survival and recruitment in later intervals
  lphi[t] ~ dnorm(lphi[t-1], tau.lphi)
  lr[t] ~ dnorm(lr[t-1], tau.lr)
}
tau.lphi <- pow(sd.lphi,-2) # Hyperpriors for variances
sd.lphi ~ dunif(0, 1)
tau.lr <- pow(sd.lr,-2)
sd.lr ~ dunif(0, 1)

# Likelihood
# Alive/dead process specified conditional on first detection
for(i in 1:nsites){
  y[i, first[i]] ~ dbern(1) # Condition on year of first detection
  for(t in (first[i]+1):nyears) {
    y[i, t] ~ dbern(y[i,t-1] * (phi[t-1] + (1-phi[t-1]) * r[t-1]) +
      (1-y[i,t-1]) * r[t-1])
  }
}
}
")

# Initial values
inits <- function(){ list(sd.lphi = runif(1), sd.lr = runif(1))}

# Parameters monitored
params <- c("phi", "r", "sd.lphi", "sd.lr")

# MCMC settings
na <- 1000  ;  ni <- 12000   ;   nt <- 6   ;   nb <- 6000   ;   nc <- 3

# Call JAGS (ART 3 min), check convergence and summarize posteriors
out2 <- jags(bdata, inits, params, "DemoDynocc2.txt", n.adapt = na,
  n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, parallel = T)
par(mfrow = c(2,2))   ;   traceplot(out2)
print(out2, dig = 2)   # not shown
```

4.13 ACCOUNTING FOR TEMPORARY EMIGRATION AND MODELING PHENOLOGIES USING OCCUPANCY DATA: ESTIMATION OF ARRIVAL AND DEPARTURE IN INSECTS OR MIGRATORY ANIMALS

The classical dynocc model assumes closure over the entire duration of the secondary periods within a given primary period. Violation of the closure assumption due to what is often called *temporary emigration* (TE) leads to an increase in occupancy if TE is random. In that case, the occupancy parameter must be interpreted as "use sometimes during the secondary occasions" but otherwise

remains unbiased with this meaning. In contrast, under Markovian TE the probability that a site is occupied or not during a secondary occasion depends on whether or not it was occupied during the previous occasion. This form of TE will bias the occupancy estimator in a way that no longer enables us to simply redefine its interpretation as some sort of use (Kendall et al., 1997; Kendall, 1999). Hence, there is an interest in accounting for any nonrandom patterns of presence and absence among the secondary occasions to estimate occupancy free of bias.

In addition, the processes underlying TE are often of biological interest and therefore we may want to estimate and model parameters describing them. One good example is the modeling of phenologies—when a species has a distinct period of the year in which it is detectable in principle, perhaps because it is completely absent from a study area outside of that period or because it is present, but in a life-state that is invisible to the survey method employed. Examples for the former are migratory birds, bats, or insects which arrive on their breeding areas at some date and leave again later. Examples for the latter are insects, which often are recorded in their adult forms only, but not as eggs, larvae, or pupae; or vascular plants which may be detectable only when they are flowering. In these cases, we may be interested in the timing of the arrival and departure processes, for instance, to see whether they are affected by climate change.

Several occupancy models have been developed that relax the strict closure assumption of standard occupancy models by describing some kind of phenology via Markovian TE (Kendall et al., 2013; Roth et al., 2014; Chambert et al., 2015). The basic, within-season model structure is identical whether applied to single- or to multi-season data. Here, we describe a variant of the model of Roth et al. (2014) for a single season, but the extension to multiple seasons, either via a dynamical formulation or via a treatment of years as strata, is straightforward and our case study will feature multiple seasons. Roth et al. (2014) assume that site i is occupied ($z = 1$) at some point during a season with occupancy probability ψ:

$$z_i \sim Bernoulli(\psi)$$

The closure assumption is then relaxed by assuming that the species arrives at site i at some time (arr_i) and departs again at a later time (dep_i). Neither is usually observed, either because surveys are not conducted on every day or because of detection error, so they must be estimated. But we can't expect to be able to estimate each date of arrival and departure of a species at every single site, so instead we estimate a distribution for them, i.e., treat these dates as random effects. Roth et al. (2014) specify an overdispersed Poisson, but here we adopt a Normal distribution instead: $arr_i \sim Normal(\mu_a, \sigma_a)$ for arrival and $dep_i \sim Normal(\mu_d, \sigma_d)$ for departure.

For J surveys at a site, we obtain the site- and occasion-specific detection data $y_{i,j}$ for which we assume the usual observation process of an occupancy model, but where the argument of the Bernoulli distribution contains an additional term, $I_{i,j}$:

$$y_{i,j} \sim Bernoulli\left(z_i * I_{i,j} * p_{i,j}\right).$$

Hence, detection of the species ($y_{i,j} = 1$) at site i during survey j depends on three things: whether a site is occupied "in principle" ($z_i = 1$), whether the species is present at site i during the particular occasion j given that the site is occupied ($I_{i,j} = 1$), and on the detection probability given the species is present ($p_{i,j}$). The binary presence indicator $I_{i,j}$ is a function of the latent arrival and the departure dates (arr_i, dep_i) and of the known survey dates ($date_{i,j}$).

$$I_{i,j} = \begin{cases} 1 \text{ when } arr_i \leq date_{i,j} \text{ and } dep_i > date_{i,j} \\ 0 \text{ when } arr_i > date_{i,j} \text{ or } dep_i \leq date_{i,j} \end{cases}.$$

Two special cases of the model allow arrivals only or departures only and differ only in their definition of I. For a model with arrivals only, we have $I_{i,j} = 1$ when $arr_i \leq date_{i,j}$ and 0 otherwise. For a departure-only model, we have $I_{i,j} = 1$ when $\text{dep}_i > date_{i,j}$ and 0 otherwise.

We can cast this model as a multi-scale occupancy model with three levels (see Section 10.10 in AHM1) where the middle level, availability, is a deterministic function of the dates of arrival, of departure, and of the survey. Hence, for a single season we can write:

$$z_i \sim Bernoulli(\psi_i)$$

$$h_{i,j}|z_i \sim Bernoulli\big(z_i * I_{i,j}\big)$$

$$y_{i,j}|h_{i,j} \sim Bernoulli\big(h_{i,j} * p_{i,j}\big)$$

Notation is the same as before and $h_{i,j}$ (you can think of its meaning as "here") is the presence or absence during the particular occasion j at site i which is occupied in principle, i.e., for which $z_i = 1$. Estimability stems from the fact that one parameter, I, is a deterministic function of known or estimable data, i.e., $date_{i,j}$, arr_i, and dep_i, and further helped by treating the latter two as random effects. For multiple years we can simply add an index for year to every quantity. We can view this model in two ways: first, it corrects the estimates of occupancy (ψ) for a specific type of violation of the closure assumption (as specified by the Markovian availability process for h) and second, it lets us study the biologically relevant availability process by estimating its parameters and exploring covariate relationships.

We illustrate the model with a simplified version of the butterfly case study from the Roth et al. (2014) paper; thanks to Tobi Roth for providing code and data (see also their paper and its Supplementary material). Their data come from multiple years; hence, we need to account for this additional "openness" in the modeled process across years. In principle, we could choose any of a number of modeling solutions, e.g., treating years as strata (with year-specific parameters either as fixed or as random effects), or modeling some temporal autocorrelation over the years, which of course includes the typical colonization-extinction processes of the traditional dynocc model. We follow Roth et al. and fit a linear logistic regression of occupancy probability on year, i.e., a trend model with random annual deviations. We analyze data from the Marbled White (*Melanargia galathea*; see Chapter 1 and Fig. 1.19) collected from 1998 to 2010 at 519 study sites in the Swiss canton Aargau. Each site was a 250-m-long transect and surveyed by Pollard walks (Pollard et al., 1995) once every 5 years with 11 seasonal visits, resulting in 1,337 observed site-level detection histories. See Roth et al. (2014) for further description of the survey and the data.

We fit linear models (possibly on a link scale) on year (1−13) for the following four parameters: the probability of occupancy (psi) and detection (p, for each of the 11 survey periods separately) and the dates of arrival and departure. We do not scale date (if we did, we would have had to express arrival and departure, along with their priors, on the identical scale).

```
# Read in Marbled White data and do some data management help page
data(SwissMarbledWhite)             # from AHMbook
str(dat <- SwissMarbledWhite)       # rename and look at data overview

# Data preparation
y <- as.matrix(dat[,14:24])         # Grab detection/nondetection data
DATE <- as.matrix(dat[,3:13])       # Grab survey dates
for(t in 1:11) {                    # Mean-impute date (but don't transform)
  DATE[is.na(DATE[,t]),t] <- mean(DATE[,t], na.rm = T)
}
year <- dat$year
nsites <- length(unique(dat$site))
nyears <- length(unique(dat$year))
nsurveys <- ncol(y)
nobs <- nrow(y)
```

We had originally tried to write the model in the full, three-level hierarchical formulation of Roth et al. (2014), but never obtained convergence. Hence, we fit a simpler version of the model, which defines the z variables only for the observed site-year combinations and does not define the intermediate availability indicators h. Even so the model is hard to get to convergence, and we run long chains, use weakly informative priors on the variance parameters, and numerically "stabilize" some of the estimates (see "*min-max*" in the BUGS code).

```
# Bundle and summarize data
# Use year as factor and yr as regressor
str(bdata <- list(y = y, DATE = DATE, year = year-1997, yr = year-2004,
  site = dat$site, nobs = nobs, nsites = nsites, nyears = nyears,
  nsurveys = nsurveys))

List of 9
 $ y       : int [1:1337, 1:11] 0 0 0 0 0 0 0 0 0 0 ...
 $ DATE    : num [1:1337, 1:11] 23 23 23 23 23 23 23 23 23 25 ...
 $ year    : num [1:1337] 1 1 1 1 1 1 1 1 1 ...
 $ yr      : num [1:1337] -6 -6 -6 -6 -6 -6 -6 -6 -6 -6 ...
 $ site    : int [1:1337] 1 2 3 4 5 6 7 8 9 10 ...
 $ nobs    : int 1337
 $ nsites  : int 519
 $ nyears  : int 13
 $ nsurveys: int 11

# Specify phenological occupancy model in BUGS language
cat(file = "PhenoOcc.txt", "
model {

# Linear model for annual site occupancy (psi) with its priors
for(i in 1:nobs) {
  z[i] ~ dbern(psi[i])
  logit(psi[i]) <- min(10, max(-10, lpsi[i]))
  lpsi[i] <- beta.lpsi[1] + beta.lpsi[2] * yr[i] + eps.lpsi[year[i]]
}
for(t in 1:nyears){
  eps.lpsi[t] ~ dnorm(0, tau.lpsi)
}

# Priors for occupancy
beta.lpsi[1] <- logit(mean.psi)
mean.psi ~ dunif(0, 1)
beta.lpsi[2] ~ dnorm(0, 0.1)
tau.lpsi <- pow(sigma.lpsi, -2)
sigma.lpsi ~ dnorm(0, 0.1)I(0.01,)

# Logit-linear model of detection on year
for(t in 1:nyears) {
  for(j in 1:nsurveys) {
    logit(p[t,j]) <- min(10, max(-10, lp[t,j]))
    lp[t,j] <- beta.lp[1,j] + beta.lp[2,j]* (t-7) # year centered
  }
}
# Priors for detection
for (j in 1:nsurveys) {                # visit-specific regression coefs on year
  beta.lp[1,j] ~ dnorm(mu.lp1, tau.lp1)
  beta.lp[2,j] ~ dnorm(mu.lp2, tau.lp2)
}
mu.lp1 ~ dnorm(0, 0.1)
mu.lp2 ~ dnorm(0, 1)
tau.lp1 <- pow(sigma.lp1, -2)
tau.lp2 <- pow(sigma.lp2, -2)
sigma.lp1 ~ dnorm(0, 1)I(0.01,)
sigma.lp2 ~ dnorm(0, 1)I(0.01,)
# curve(dnorm(x, 0, sqrt(1)), 0, 10)   # how does this look like ?
```

```
# Linear regression of arrival date (arr) on year
for(i in 1:nobs) {
  arr[i] ~ dnorm(mu.arr1[i], tau.arr)
  mu.arr1[i] <- min(200, max(30, mu.arr[i]))
  mu.arr[i] <- beta.arr[1] + beta.arr[2] * yr[i]
}

# Priors for arrival model
beta.arr[1] ~ dnorm(90, 0.1)
beta.arr[2] ~ dnorm(0, 1)
tau.arr <- pow(sigma.arr, -2)
sigma.arr ~ dnorm(0, 0.01)I(0.01,)
# curve(dnorm(x, 0, sqrt(100)), 0, 20) # how does this look like ?

# Linear regression of departure date (dep) on year
for(i in 1:nobs) {
  dep[i] ~ dnorm(mu.dep1[i], tau.dep)
  mu.dep1[i] <- min(500, max(0, mu.dep[i]))
  mu.dep[i] <- beta.dep[1] + beta.dep[2] * yr[i]
}

# Priors for departure model
beta.dep[1] ~ dnorm(120, 0.1)
beta.dep[2] ~ dnorm(0, 1)
tau.dep <- pow(sigma.dep, -2)
sigma.dep ~ dnorm(0, 0.01)I(0.01,)

# Model for the observed data
for(i in 1:nobs) {
  for(j in 1:nsurveys) {
    y[i,j] ~ dbern(mu1[i,j])
    mu1[i,j] <- min(0.99, max(0.01, mu[i,j]))
    mu[i,j] <- z[i] * step(DATE[i,j] - arr[i]) * step(dep[i] -
    DATE[i,j]) * p[year[i],j]
  }
}

# Derived quantities
# Average occupancy per year
for(t in 1:nyears){
  for(s in 1:nsites){
    logit(tmp[s, t]) <- beta.lpsi[1] + beta.lpsi[2] * (t-7) + eps.lpsi[t]
  }
  psi.pred[t] <- mean(tmp[,t])
}

# Average detection per year and visit
for(t in 1:nyears){
  for(j in 1:nsurveys){
  logit(p.pred[t,j]) <- beta.lp[1,j] + beta.lp[2,j] * (t-7)
  }
}

# Average arrival and departure time per year and length of flight period
for(t in 1:nyears){
  arr.pred[t] <- beta.arr[1] + beta.arr[2] * (t-7)
  dep.pred[t] <- beta.dep[1] + beta.dep[2] * (t-7)
  fp.pred[t] <- dep.pred[t] - arr.pred[t]
}
}
")

# Initial values
zst <- apply(y, 1, max, na.rm = T)
inits <- function() {list(z = zst)}
```

```
# Parameters monitored
params <- c('mean.psi', 'beta.lpsi', 'sigma.lpsi', 'beta.lp', 'mu.lp1',
  'mu.lp2', 'sigma.lp1', 'sigma.lp2', 'beta.arr', 'sigma.arr', 'beta.dep',
  'sigma.dep', 'psi.pred', 'p.pred', 'arr.pred', 'dep.pred', 'fp.pred')

# MCMC settings
na <- 5000  ;  ni <- 100000  ;  nt <- 80  ;  nb <- 20000  ;  nc <- 3

# Call JAGS (ART 400 min), check convergence and summarize posteriors
out <- jags(bdata, inits, params, "PhenoOcc.txt", n.chains = nc,
  n.thin = nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(3,3))   ;  traceplot(out)
print(out, dig = 2)
```

In Fig. 4.37 we see that the Marbled White is becoming more widespread over time, but that its detection probability declines over the years, which could be a consequence of declining abundance. Both the arrival and the departure dates have a slight tendency to become earlier, while the length of the flight period is essentially constant. None of these trends are significant though, since the uncertainty around the predictions is substantial.

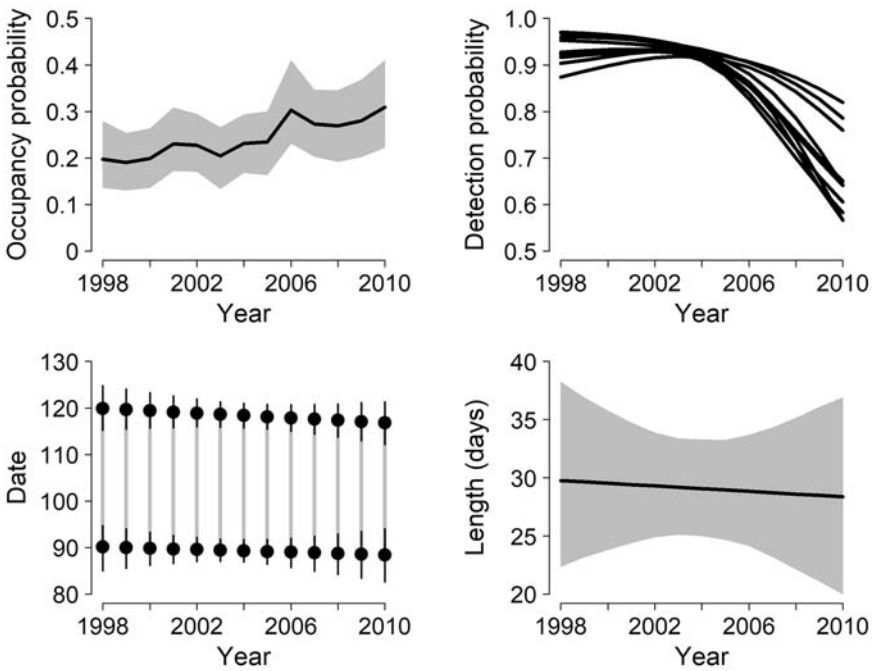

FIGURE 4.37

Predictions of annual trajectories of occupancy (top left), detection probability for each of the 11 survey periods (top right), arrival and departure dates with intervening flight period in gray (bottom left), and the length of the flight period (bottom right) in the Marbled White butterfly (*Melanargia galathea*) in the Swiss canton of Aargau. Uncertainty polygons and intervals are 95% CRIs.

This is an interesting model because it allows us to relax the closure assumption for species which have a predictable pattern of presence/absence over the course of a season, as do many migratory birds or insects with distinct phenology of appearance of the adults and probably also many species of plants. Moreover, the arrival/departure processes are often of biological interest. Since the model has parameters for them, we can directly estimate these parameters and model them, i.e., see whether covariates may influence them. It is likely that for a "full" 3D data set with no or only few missing data, we would be able to fit the fully hierarchical model, with z indexed by site and year, and h indexed by site, year, and visit. This might then let us formulate the traditional colonization/extinction dynamics at the top level of the model, add up the occupied sites per year, and so forth. Hence, for a balanced data set, the inferences under this model might be even richer. One possible alternative might be to model the mean and the spread of the presence period using a Gaussian kernel of date, similar to what we did for the count models for the UK Marbled Whites in Chapter 1.

4.14 SUMMARY AND OUTLOOK

We presented the dynamic occupancy (or "dynocc") model of MacKenzie et al. (2003), which is a Markov model for a latent binary process that is partially observed in discrete time. The model describes the initial system state in terms of occupancy probability ψ_1 at time $t = 1$, the latent presence/absence transition dynamics in terms of the probabilities of colonization (γ) and persistence (ϕ; or alternatively colonization γ and extinction ε), and the observation process in terms of detection probability (p). This model has a close relationship to the classical metapopulation models (Hanski, 1998) and it can be represented as a matrix (Caswell, 2001) or a hidden Markov model (Zucchini et al., 2016). The dynocc model, and its simplified static version in Section 4.6, is beautifully simple, and yet it can be applied to a tremendous range of different scientific settings, ranging from population dynamics (MacKenzie et al., 2003, 2012), species distribution change (Kéry et al., 2013), nest-site occupancy dynamics (Bled et al., 2011a), disease ecology (Lachish et al., 2012), and for the analysis of data from monitoring programs and unstructured citizen science data collections (Kéry et al., 2006; Altwegg et al., 2008; van Strien et al., 2010, 2011, 2013; Altwegg and Nichols, 2019). In a fascinating new development, Joseph (2020a) combines the dynocc model with machine-learning-based covariate regressions and fits it to a whole community of species over a large number of years in what he calls *neural hierarchical models*.

There is always a mathematical relationship between occupancy (ψ) and the underlying expected abundance (λ): $\psi = \text{Prob}(N > 0|\lambda)$, such that, in a Poisson abundance model, occupancy probability is given by $1 - e^{-\lambda}$ (Royle and Dorazio, 2008). In this chapter, we have seen cases where the abundance-occupancy relationship is particularly tight (the Peregrine and Eagle Owl analyses in Table 4.1) to those where it is less direct, as for the Asp Vipers and the butterfly; and it would be more loose still for occupancy modeling of a disease. But in each case, the characterization of a population or a spatial sample of a larger ("meta"-) population by the distinction between $N = 0$ and $N > 0$, i.e., by presence/absence, is almost always useful (Although if you *do* have abundance information, it is usually best to directly model it, rather than collapsing counts to binary detection/nondetection data and using an occupancy model on this degraded abundance information).

We think of the dynocc model as one of the most powerful models in all of ecological statistics. But at the same time, it must also be one of the most underutilized ones. The power of the dynocc model is due to the following:

- Very wide variety of settings in which it may be applied

- Much wider availability of detection/nondetection data compared to count or other data that are more directly informative about abundance
- Greater computational ease with which models for binary data can be fitted compared to models for abundance (e.g., compare the dynocc model with its abundance counterpart, the Dail-Madsen model in Chapter 2)

In this chapter we also touched upon sampling issues such as imbalanced data sets, which may lead to extremely large numbers of missing responses when the data are organized in the usual multidimensional arrays that we like for BUGS or even for `unmarked` and, much more importantly, nonignorable missingness in distribution and abundance data, i.e., preferential sampling (PS). This is an important and, even in the statistical sciences, fairly new topic. Opportunistic data collection protocols underlie virtually all citizen-science schemes and in all of them, we must frequently expect some degree of positive PS—"better" sites have a higher chance of being visited by the observers. This will lead to positive biases in the perceived means or totals unless corrected for in a joint model for the observations and the visitation. In addition, it seems possible that if the magnitude of PS changes over time, spurious trends may be perceived in a PS-naive analysis. In Chapter 6, we revisit PS and see an example where ignoring PS leads to very erroneous assessments of population trends of Swiss Eagle Owls. Clearly, accounting for PS is important for inferences about distribution and abundance from opportunistic data sets, although the understanding of it appears to still be in its infancy (Dinsdale and Salibian-Barrera, 2019). More research is needed here, both in statistics, but also in ecology and related fields.

In this chapter we have seen how the limits between abundance and occupancy modeling become blurred with decreasing average abundance per site. Indeed, the two coincide when a site can hold only either 0 or 1 individual or pair: population size then equals the number of presence sites. However, this equality does not hold for the dynamic rate parameters, since a colonization and persistence event in the occupancy model may correspond to multiple possible pathways in terms of the survival and recruitment of the individual(s) that occupy the site. But we can describe the former in terms of the latter and this opens up the possibility to truly estimate and model individual vital rates from site-level presence/absence observations; this is what the model of Roth and Amrhein (2010) does.

Surprisingly, the Roth-Amrhein model has hardly ever been used in practice, although it seems to hold considerable potential for all bird population studies where a site can only be occupied by a single pair. Examples include nest-box studies or studies including species that have fairly well-defined territories that do not change over the years, such as most birds of prey. We think therefore that there may be a large amount of untapped demographic information in these studies that waits to be extracted using the Roth-Amrhein model. Nevertheless, a better understanding of the model is required, for instance, of the exact assumptions that this model makes (can we really apply it directly for pair data as we did?), how these assumptions can be violated, and how such assumption violations affect model performance. There is a place for both simulation and empirical studies in answering these questions. We also think that there is a great potential to combine "cheap and risky" data with more "expensive and reliable" data sets in an integrated model (Chapter 10). For instance, if both territory-visitation data and ringing data are available from a bird population study, then it should be possible to specify a joint model for both data sets by adopting a Roth-Amrhein model for the former and a CJS or ring-recovery model for the latter. (By "expensive" we mean harder to obtain and by "reliable" we mean that well-understood and well-performing models are available for such data.)

A recent trend in the modeling of changing distributions has been to move back from the simplistic presence/absence characterization of population abundance to a more explicit characterization in terms of abundance classes or true abundance. This also includes attempts to estimate parameters of

population dynamics, sometimes including dispersal, from mere binary presence/absence observations, sometimes in a joint modeling framework with other data (e.g., the dynamic range models of Pagel and Schurr, 2012) and sometimes alone (Sutherland et al., 2014). Some of the models discussed by Hefley and Hooten (2016) also fall in this class. More generally, we think that it will be interesting to see how we can infer from simple occupancy patterns the demography of the individuals that inhabit the sites.

Over the past two decades, reaction-diffusion models have increasingly appeared in ecological statistics as a powerful framework for changing species distributions (e.g., Wikle et al., 1998; Wikle, 2003; Hooten et al., 2007; Wikle and Hooten, 2010; Hefley and Hooten, 2016; Hefley et al., 2017b,c; Williams et al., 2017, 2018; Louvrier et al., 2020). These models are more mechanistic still than is the dynocc because they describe distributional change in terms of parameters for population growth and dispersal that are defined at the individual level. We don't doubt that their use will increase in ecology, but we think that there will always be a place for the dynocc model, because it is much easier to fit and to understand, especially for nonstatisticians. In terms of its mechanistic feel, we see the dynocc model as somewhat intermediate between purely descriptive SDMs and diffusion models. In addition, there are several ways in which we can "bring more space" into the dynocc model; see Chapter 9. Thereby, we can make the dynocc model even more mechanistic as a model that describes the consequences of dispersal in terms of colonization and persistence probabilities, while still retaining its great conceptual and practical simplicity to a large degree.

The dynocc model has so many other uses and applications that despite the length of this chapter, we have only scratched the surface. Some of the important topics that we haven't covered include perturbation analysis to study the effects of colonization, extinction, covariates, or other drivers on the growth rate trajectory in an occupancy study (Martin et al., 2009b; Miller et al., 2012), other observation protocols such as time-to-detection or removal designs (Section 10.12 in AHM1, Reich 2020; Strebel et al., in prep.), Royle-Nichols variants of the model (Royle and Nichols, 2003; Rossman et al., 2016), and the extension of the multiscale occupancy model (Nichols et al., 2008; Mordecai et al., 2011, Section 10.10 in AHM1) to the dynamic case (Tingley et al., 2018). We could have used simulation to study the effects of unmodeled heterogeneity in parameters other than detection. For instance, we could ask whether unmodeled heterogeneity in colonization or persistence has an effect on the estimators. Many such extensions and developments have already been achieved and others would be fairly trivial, while yet others await more research. Hence, we envision much more research on and use of the dynocc model, both in terms of the method, but especially also with applications of this powerful model.

In later chapters, we will address many other extensions of the dynocc model. Indeed, a majority of the remaining chapters in this volume deal with what could be described as mere variants of this model. In the next chapter (5) we cover the extension to multiple species, or (meta)communities, in what can be called dynamic community models (Dorazio et al., 2010). In Chapter 6, we extend occupancy models to more than a single state of "presence" to become multi-state occupancy models (Royle and Link, 2005; Nichols et al., 2007; MacKenzie et al., 2009). In Chapter 7, we extend the dynocc model to allow for false-positives also (Royle and Link, 2006; Miller et al., 2011, 2013a; Sutherland et al., 2013). In Chapter 8, we jointly model multiple species in a dynocc model and let them interact, while in Chapter 9 we extend spatially ignorant dynocc models to become spatially more explicit (Royle and Dorazio, 2008; Bled et al., 2011a,b); see also the exciting recent work by Hepler et al. (2018) and Hepler and Erhardt (2020).

This page intentionally left blank