bba253f30603a496429325e0286b311e
ebrary

CHAPTER

# 10

# Estimation of Survival, Recruitment, and Population Size from Capture–Recapture Data Using the Jolly–Seber Model

OUTLINE



bba253f30603a496429325e0286b311e
ebrary

*Bayesian Population Analysis using WinBUGS*
DOI: 10.1016/B978-0-12-387020-9.00010-9



Copyright © 2012 Elsevier Inc. All rights reserved.

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


## 10.1 INTRODUCTION

Capture–recapture data contain information not only about the disappearance of marked individuals from a study population (i.e., mortality and permanent emigration) but also about their arrival into the population (i.e., recruitment by locally born individuals and immigration). Estimation of these recruitment-related parameters is possible when the *complete* capture-histories of the marked individuals are analyzed, not just the part following first capture, as for the CJS model (Chapter 7). The leading capture-history zeros (those before initial capture) along with the first capture contain the information about the arrival process. A leading zero is observed either because an individual has not yet arrived (recruited or immigrated) in the population or because it was already in the population but has not been captured. The part of the capture-histories after the initial capture contains information about survival, which is modeled by the CJS model (Chapter 7). All open-population capture–recapture types of models described in the Chapters 7–9 condition on first capture. This means that only the part of a capture-history *after* first capture is modeled; the information about entry of individuals is discarded. In this chapter, we introduce the Jolly–Seber (JS) model (Jolly, 1965; Seber, 1965), which allows estimation of gains to and losses from the population by *not* conditioning on first capture. Additionally, population size and the number of recruits per occasion can be estimated. The JS model is an extension of the closed-population models in Chapter 6, which do not condition on first capture either but assume demographic closure (i.e., the absence of mortality, recruitment, or dispersal), and of the CJS model.

Since the JS model uses the complete capture-history, additional assumptions are necessary. All assumptions listed for the CJS model (Section 7.1) also apply to the JS model. In addition, the JS model requires that *all* individuals (marked *and* unmarked) in the population have the same capture probability. In other words, newly captured individuals must be a random sample of all unmarked individuals in a population (Williams et al., 2002). This assumption is likely to be met when the same sampling protocol is applied for capture and recapture, but not when protocols differ. In many studies where color marks are applied, initial capture is a physical capture, whereas recaptures are just sightings from a distance. In this case, the equal-capturability assumption is likely to be violated. Moreover, the assumption of homogeneous capture is violated in the presence of a permanent trap effect. Violation of the equal-catchability assumption biases estimates of population size and recruitment, but not of survival (Nichols et al., 1984). Therefore, permanent trap effects do not violate the CJS assumptions. Williams et al. (2002) provide an in-depth overview of JS model assumptions and consequences of their failure for the parameter estimates.

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



There are a number of different parameterizations of the JS model, which differ basically in the way how recruitment is modeled. They include the original parameterizations of Jolly (1965) and Seber (1965), the superpopulation approach (Crosbie and Manly, 1985; Schwarz and Arnason, 1996), the reverse-time formulation (Pradel, 1996), the parameterization of Link and Barker (2005), and finally the formulation as a restricted dynamic occupancy model (Royle and Dorazio, 2008). All JS model parameterizations are related and should give the same estimates of population size and recruitment parameters. Here, we illustrate three JS model variants: the restricted occupancy formulation, the description as a multistate model, and the superpopulation formulation (Royle and Dorazio, 2008). We make extensive use of parameter-expanded data augmentation (introduced in Chapter 6) because it makes the modeling much easier (Royle et al., 2007a; Royle and Dorazio, 2011).

Numerous JS model variants have been described in the frequentist framework (e.g., Pollock et al., 1990; Pradel, 1996; Schwarz and Arnason, 1996; Williams et al., 2002) and implemented in software such as MARK (White and Burnham, 1999) or POPAN (Arnason and Schwarz, 1999). In contrast, Bayesian applications are still relatively rare (but see Link and Barker, 2005; Dupuis and Schwarz, 2007; Schofield and Barker, 2008; Royle and Dorazio, 2008; Gardner et al., 2010; Link and Barker, 2010; Royle and Dorazio, 2011).

## 10.2 THE JS MODEL AS A STATE-SPACE MODEL

We formulate the JS model in the hierarchical state-space framework. As mentioned repeatedly, the observed capture–recapture data are described as the result of a state process (the ecological process) and the observation process, which depends on the result of the state process. We denote the number of individuals ever alive during the study $N_s$ and call it the superpopulation size (Schwarz and Arnason, 1996). We further assume that a fraction of $N_s$ is already alive and in the study area at the first capture occasion, and that all remaining individuals have entered the population by the end of the study. The probability that a member of $N_s$ enters the population at occasion $t$ is $b_t$ ($t = 1, \ldots, T$) and is called the entry probability (Schwarz and Arnason, 1996). It is the probability that an individual is new in the population, that is, it has entered the population since the preceding occasion. Entry could result either from in situ recruitment (locally born individuals) or from immigration. Sometimes the entry probability is called recruitment probability, but this is inaccurate. The number of individuals entering the population at $t$ is $B_t = N_s b_t$. The fraction of individuals already present at the first occasion is $b_1$; this "entry" probability has no clear ecological meaning because it is a complex function of all entries before the first occasion. All entry probabilities must sum to 1 to

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


ensure that all $N_s$ individuals enter the population sometime during the study. The number of individuals entering at each occasion can be modeled with a multinomial distribution as $\mathbf{B} \sim \text{multinomial}(N_s, \mathbf{b})$ (note that **B** and **b** are vectors).

As in the CJS model, we denote the latent state of individual $i$ at occasion $t$ as $z_{i,t} = 1$, if it is alive and present in the population, and as $z_{i,t} = 0$, if it is either dead or has not yet entered the population. Thus, if individual $i$ enters the population at $t$, its latent state changes from $z_{i,t-1} = 0$ to $z_{i,t} = 1$ (Fig. 10.1). On entry, the survival process starts, which is a simple coin flip and identical to that in the CJS model (Chapter 7). Technically, the latent state $z_{i,t+1}$ at $t+1$ is determined by a Bernoulli trial with success probability $\phi_{i,t}$ $(t = 1, \ldots, T-1)$. The two processes defined so far, the entry and the survival process, represent the latent state processes. The observation process is defined for individuals that are alive ($z = 1$). As usual, we assume that the detection of individual $i$ at occasion $t$ is determined by another coin flip with success probability $p_{i,t}$ $(t = 1, \ldots, T)$, that is, by another Bernoulli trial.

The resulting capture–recapture data consist of the capture-histories of $n$ individuals. When capture probability is $< 1$, typically not all individuals in a population are captured; hence, $n < N_s$. If $N_s$ were known, the capture–recapture data would contain in addition $N_s$-$n$ all-zero capture-histories, and the model specification would be relatively easy. We could just use a multinomial distribution to estimate entry probabilities. However, $N_s$ is unknown and so the multinomial index is also unknown and must be estimated. Moreover, parameters such as entry and capture probabilities refer to the complete population ($N_s$), not just

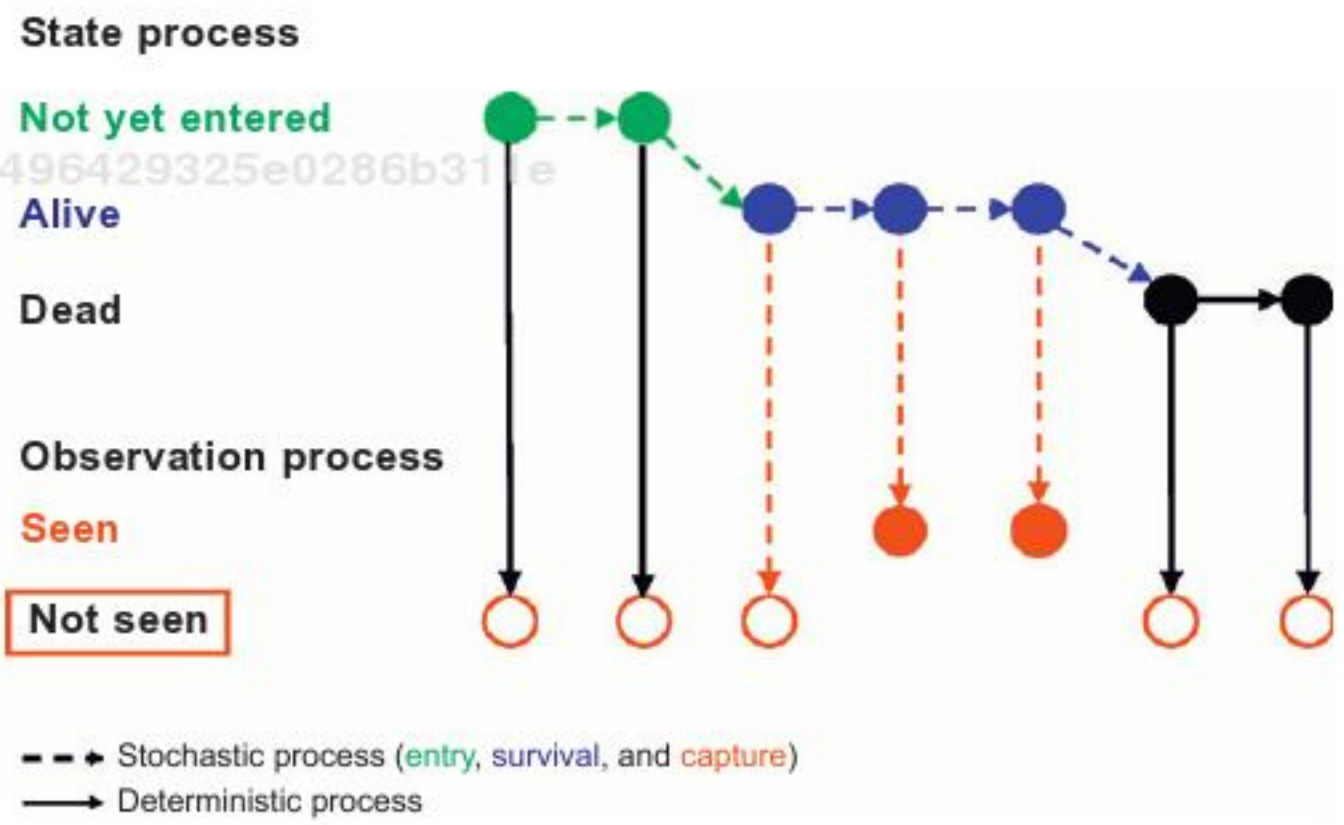


FIGURE 10.1 Example of the state and observation process of a marked individual over time in the JS model. The sequence of true states for this individual under the restricted occupancy and superpopulation formulation is $z = [0, 0, 1, 1, 1, 0, 0]$, and the observed capture-history is $y = [0, 0, 0, 1, 1, 0, 0]$. In the multistate formulation, $z = [1, 1, 2, 2, 2, 3, 3]$ and $y = [2, 2, 2, 1, 1, 2, 2]$.

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



to the $n$ individuals ever captured. To deal with these challenges, we use parameter-expanded data augmentation (PX-DA; Tanner and Wong, 1987; Liu and Wu, 1999; Royle et al., 2007a; Royle and Dorazio, 2011), as we did in Chapter 6. The key idea is to fix the dimension of the parameter space in the analysis by augmenting the observed data with a large number of all-zero capture-histories, resulting in a larger data set of fixed dimension $M$, and to analyze the augmented data set using a reparameterized (zero-inflated) version of the model that would be applied if $N_s$ were known. For a useful recent review of PX-DA, see Royle and Dorazio (2011).

## 10.3 FITTING THE JS MODEL WITH DATA AUGMENTATION

After augmentation, the capture–recapture data set contains $M$ individuals, of which $N_s$ are genuine and $M$-$N_s$ are pseudo-individuals (Fig. 10.2). We do not know the proportions of genuine and pseudo-individuals, but we can estimate them. There are different ways to parameterize a JS model, and we present three here. First, the entry to the population is described as a removal process from $M$ so that at the end of the study, $N_s$ ($N_s \leq M$) individuals have entered. This model can be developed either as a restricted version of a dynamic occupancy model

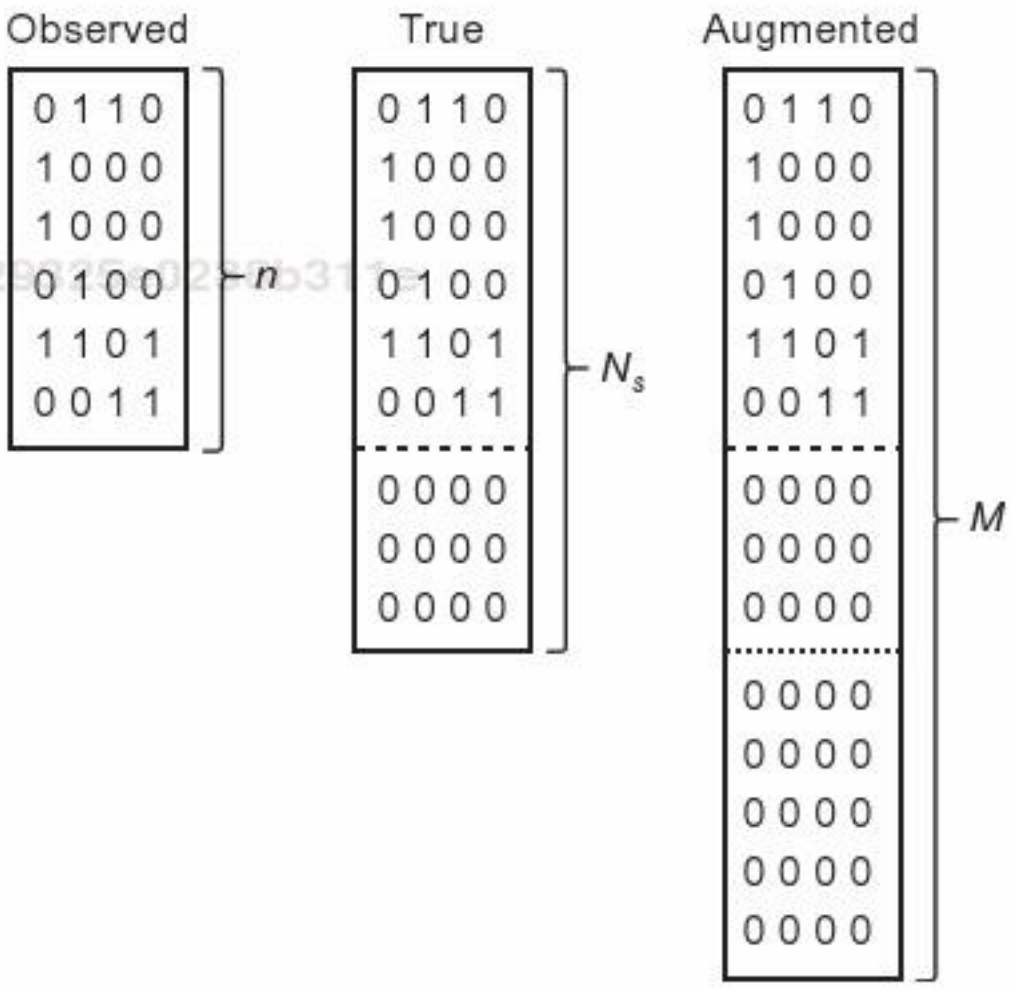


FIGURE 10.2 Observed, true, and augmented capture-history matrix. The observed number of capture-histories is $n$, true population size $N_s$, and the size of the augmented data set $M$; $n$ and $M$ are known, and $N_s$ can be estimated.

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


(Section 13.5) or as a multistate model (Chapter 9). As a third approach, we use a zero-inflated version of the superpopulation formulation.

### 10.3.1 The JS Model as a Restricted Dynamic Occupancy Model

Royle and Dorazio (2008) showed that the JS model can be formulated as a restricted dynamic occupancy model. The entry process is defined slightly differently from what we described earlier. We imagine that individuals can be in one of three possible states: "not yet entered", "alive", and "dead" (Fig. 10.1). The state transitions are governed by two ecological processes, entry and survival, which we estimate. We denote as $\gamma_t$ ($t = 1, \ldots, T$), the probability that an available individual in $M$ enters the population at occasion $t$. This corresponds to the transition probability from state "not yet entered" to the state "alive". Importantly, $\gamma$ refers to available individuals, that is, to those in $M$ that have not yet entered. The entry process is thus a removal process; over time, fewer and fewer individuals will be in the state "not yet entered" and thus available to entering the population. As a result, $\gamma$ will increase over time on average, even with constant per-capita recruitment. It is a pure "nuisance parameter", which is needed to describe the system, but without an ecological meaning. We refer to $\gamma$ as a removal entry probability. The expected number of individuals present at the first occasion is $E(B_1) = M\gamma_1$. The expected number of individuals entering at the second occasion is the product of the number of individuals still available to enter and $\gamma_2$, thus $E(B_2) = M(1-\gamma_1)\gamma_2$. More generally, the expected number of individuals entering the population at $t$ is $E(B_t) = M\prod_{i=1}^{t-1}(1-\gamma_i)\gamma_t$, and the total number of individuals that ever enter is $N_s = \sum \mathbf{B}$.

The state of individual $i$ at the first occasion is

$$z_{i,1} \sim \text{Bernoulli}(\gamma_1).$$

Subsequent states are determined either by survival, for an individual already entered ($z_{i,t} = 1$), or by entry for one that has not ($z_{i,t} = 0$). Thus,

$$z_{i,t+1} \,|\, z_{i,t}, \ldots, z_{i,1} \sim \text{Bernoulli}\left(z_{i,t}\phi_{i,t} + \gamma_{t+1}\prod_{k=1}^{t}(1-z_{i,k})\right).$$

Survival probability between occasion $t$ and $t + 1$ for individual $i$ is denoted as $\phi_{i,t}$. In contrast, $\gamma_t$ is only indexed by time because an index for individual would not be meaningful. The two equations above describe the state process of the JS model. The observation process is the same as in the CJS model (Section 7.2). It conditions on the state process and is

$$y_{i,t} \,|\, z_{i,t} \sim \text{Bernoulli}(z_{i,t}p_{i,t}).$$

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

This model is very similar to the dynamic occupancy model (Section 13.5). In the JS model, we just need to add a constraint that recruitment is not possible after death, whereas in occupancy models, recolonization after local extinction is possible.

Several quantities of interest can be derived from the latent state variable $z$. Population size at $t$ is $N_t = \sum_{i=1}^{M} z_{i,t}$, the number of "fresh recruits" (newly entered individuals) at $t$ is $B_t = \sum_{i=1}^{M} (1 - z_{i,t-1}) z_{i,t}$, and superpopulation size is $N_S = \sum \mathbf{B}$.

In a Bayesian analysis of the model, the following parameters require priors: survival ($\phi$), capture ($p$) and removal entry probabilities ($\gamma$). To express ignorance, we might specify a uniform prior U(0, 1) on all of them. The superpopulation size is a derived parameter; hence, a prior for $N_s$ is defined implicitly, but it is not clear how it would have to look like. In any case, it is not a discrete uniform prior such as U(0, M), which is what we might like to use (Royle and Dorazio, 2008), and what we use in the closed-population case (Chapter 6).

The BUGS code for this model is not difficult conceptually, but relatively long, because several quantities of interest are calculated (all after "Calculate derived population parameters"). This could be removed from the code and calculated outside BUGS; it just requires monitoring of the latent variable z. We present the code of a model with constant survival and recapture probabilities and time-dependent recruitment (entry).

```
# Specify model in BUGS language
sink("js-rest.occ.bug")
cat("
model {

# Priors and constraints
for (i in 1:M){
    for (t in 1:(n.occasions-1)){
        phi[i,t] <- mean.phi
        } #t
    for (t in 1:n.occasions){
        p[i,t] <- mean.p
        } #t
    } #i
mean.phi ~ dunif(0, 1)
mean.p ~ dunif(0, 1)

for (t in 1:n.occasions){
    gamma[t] ~ dunif(0, 1)
    } #t

# Likelihood
for (i in 1:M){
    # First occasion
    # State process
```

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


```
    z[i,1] ~ dbern(gamma[1])
    mu1[i] <- z[i,1] * p[i,1]
    # Observation process
    y[i,1] ~ dbern(mu1[i])
    # Subsequent occasions
    for (t in 2:n.occasions){
        # State process
        q[i,t-1] <- 1-z[i,t-1]          # Availability for recruitment
        mu2[i,t] <- phi[i,t-1] * z[i,t-1] + gamma[t] * prod(q[i,1:(t-1)])
        z[i,t] ~ dbern(mu2[i,t])
        # Observation process
        mu3[i,t] <- z[i,t] * p[i,t]
        y[i,t] ~ dbern(mu3[i,t])
        } #t
    } #i

# Calculate derived population parameters
for (t in 1:n.occasions){
    qgamma[t] <- 1-gamma[t]
    }
cprob[1] <- gamma[1]
for (t in 2:n.occasions){
    cprob[t] <- gamma[t] * prod(qgamma[1:(t-1)])
    } #t
psi <- sum(cprob[])                     # Inclusion probability
for (t in 1:n.occasions){
    b[t] <- cprob[t] / psi              # Entry probability
    } #t
for (i in 1:M){
    recruit[i,1] <- z[i,1]
    for (t in 2:n.occasions){
        recruit[i,t] <- (1-z[i,t-1]) * z[i,t]
        } #t
    } #i
for (t in 1:n.occasions){
    N[t] <- sum(z[1:M,t])               # Actual population size
    B[t] <- sum(recruit[1:M,t])         # Number of entries
    } #t
for (i in 1:M){
    Nind[i] <- sum(z[i,1:n.occasions])
    Nalive[i] <- 1-equals(Nind[i], 0)
    } #i
Nsuper <- sum(Nalive[])                 # Superpopulation size
}
",fill=TRUE)
sink()
```

## 10.3.2 The JS Model as a Multistate Model

It is obvious from Fig. 10.1 that the JS model can also be viewed as a multistate model (Royle and Dorazio, 2011). This formulation has the

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



advantage that it can be extended flexibly to include age classes, multiple sites, dead recoveries, or others, as we showed in Chapter 9. To fit the JS as a multistate model in BUGS, we again first have to define lists of true and observed states and to describe the transitions between them with appropriate parameters. For the JS model, the true states are "not yet entered", "alive", and "dead" (Fig. 10.1), and the state transition matrix therefore is

$$\begin{array}{c} \\ \text{not yet entered} \\ \text{alive} \\ \text{dead} \end{array} \begin{array}{c} \begin{array}{ccc} \text{not yet entered} & \text{alive} & \text{dead} \end{array} \\ \begin{bmatrix} 1-\gamma & \gamma & 0 \\ 0 & \phi & 1-\phi \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

The parameters are the same as in the restricted occupancy formulation; thus, $\gamma$ is the removal entry probability and $\phi$ is survival. The observation process maps the three true states on the two observed states "seen" and "not seen":

$$\begin{array}{c} \\ \text{not yet entered} \\ \text{alive} \\ \text{dead} \end{array} \begin{array}{c} \begin{array}{cc} \text{seen} & \text{not seen} \end{array} \\ \begin{bmatrix} 0 & 1 \\ p & 1-p \\ 0 & 1 \end{bmatrix} \end{array}$$

The implementation as a state-space model works similarly as for other multistate models (Chapter 9). However, since the traditional multistate models condition on initial capture, there is no way to estimate $\gamma_1$ at the first occasion. This can be overcome easily by adding a dummy occasion that contains only "0" before the first real occasion in the data. In the model specification, we then need to ensure that all individuals in the augmented data set are in state "not yet entered" at this first dummy occasion with probability 1. In this way, we solve two problems. First, the proportion of individuals present already at the first real occasion is estimated by the first transition, and second, the analyzed capture-histories condition on the first dummy occasion, which means that the model becomes unconditional for all real occasions.

The BUGS code to implement this model is given below. As discussed earlier, we assume constant survival and capture and time-dependent entry probability. The same quantities as before can be derived, we just have to remember that the latent state variable $z$ now takes values 1 ("not yet entered"), 2 ("alive"), and 3 ("dead"), and thus these quantities must be calculated slightly differently. Priors are specified in the same way as for the restricted occupancy formulation.

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

```
# Specify model in BUGS language
sink("js-ms.bug")
cat("
model {

# -------------------------------------
# Parameters:
# phi: survival probability
# gamma: removal entry probability
# p: capture probability
# -------------------------------------
# States (S):
# 1 not yet entered
# 2 alive
# 3 dead
# Observations (O):
# 1 seen
# 2 not seen
# -------------------------------------

# Priors and constraints
for (t in 1:(n.occasions-1)){
   phi[t] <- mean.phi
   gamma[t] ~ dunif(0, 1) # Prior for entry probabilities
   p[t] <- mean.p
   }

mean.phi ~ dunif(0, 1)      # Prior for mean survival
mean.p ~ dunif(0, 1)        # Prior for mean capture

# Define state-transition and observation matrices
for (i in 1:M){
   # Define probabilities of state S(t+1) given S(t)
   for (t in 1:(n.occasions-1)){
      ps[1,i,t,1] <- 1-gamma[t]
      ps[1,i,t,2] <- gamma[t]
      ps[1,i,t,3] <- 0
      ps[2,i,t,1] <- 0
      ps[2,i,t,2] <- phi[t]
      ps[2,i,t,3] <- 1-phi[t]
      ps[3,i,t,1] <- 0
      ps[3,i,t,2] <- 0
      ps[3,i,t,3] <- 1

      # Define probabilities of O(t) given S(t)
      po[1,i,t,1] <- 0
      po[1,i,t,2] <- 1
      po[2,i,t,1] <- p[t]
      po[2,i,t,2] <- 1-p[t]
      po[3,i,t,1] <- 0
      po[3,i,t,2] <- 1
      } #t
   } #i

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

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



```
    # Define latent state at first occasion
    z[i,1] <- 1       # Make sure that all M individuals are in state 1 at t=1
    for (t in 2:n.occasions){
        # State process: draw S(t) given S(t-1)
        z[i,t] ~ dcat(ps[z[i,t-1], i, t-1,])
        # Observation process: draw O(t) given S(t)
        y[i,t] ~ dcat(po[z[i,t], i, t-1,])
        } #t
    } #i

# Calculate derived population parameters
for (t in 1:(n.occasions-1)){
    qgamma[t] <- 1-gamma[t]
    }
cprob[1] <- gamma[1]
for (t in 2:(n.occasions-1)){
    cprob[t] <- gamma[t] * prod(qgamma[1:(t-1)])
    } #t
psi <- sum(cprob[])                    # Inclusion probability
for (t in 1:(n.occasions-1)){
    b[t] <- cprob[t] / psi             # Entry probability
    } #t

for (i in 1:M){
    for (t in 2:n.occasions){
        al[i,t-1] <- equals(z[i,t], 2)
        } #t
    for (t in 1:(n.occasions-1)){
        d[i,t] <- equals(z[i,t]-al[i,t],0)
        } #t
    alive[i] <- sum(al[i,])
    } #i

for (t in 1:(n.occasions-1)){
    N[t] <- sum(al[,t])                # Actual population size
    B[t] <- sum(d[,t])                 # Number of entries
    } #t
for (i in 1:M){
    w[i] <- 1-equals(alive[i],0)
    } #i
Nsuper <- sum(w[])                     # Superpopulation size
}
",fill = TRUE)
sink()
```

## 10.3.3 The Superpopulation Parameterization

An important parameterization of the JS model was developed by Crosbie and Manly (1985), extended by Schwarz and Arnason (1996), and implemented as a hierarchical model by Royle and Dorazio (2008) and Link and Barker (2010). This parameterization uses entry probabilities $b$ and an inclusion parameter $\psi$. To keep the sequential specification of the

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

state process model, we reexpress the entry probabilities ($b$) as conditional entry probabilities ($\eta$), and thus

$$\eta_1 = b_1, \quad \eta_2 = \frac{b_2}{1-b_1}, \quad \ldots, \quad \eta_t = \frac{b_t}{1-\sum_{i=1}^{t-1} b_i}$$

The conditional entry probabilities ($\eta$) are not the same as the removal entry probabilities ($\gamma$) in Sections 10.3.1 and 10.3.2. Nevertheless, the state process is identical to that in the restricted occupancy parameterization, except that it contains $\eta$ instead of $\gamma$. For the observation process, we suppose that each individual of $M$ has an associated latent variable $w_i \sim$ Bernoulli($\psi$). Individuals with $w_i = 1$ are exposed to sampling if alive, whereas individuals with $w_i = 0$ are not exposed to sampling. Thus, the observation model is

$$y_{i,t} \,|\, z_{i,t} \sim \text{Bernoulli}(w_i z_{i,t} p_{i,t}).$$

This observation model formally admits the zero-inflation of the augmented data set.

Here, we can derive some population estimates of interest as well. The vector of latent state variables $z$ is inflated under the superpopulation formulation because it has length $M$, rather than $N_s$. We calculate $u_{i,t} = z_{i,t} w_i$ to account for this. By using $u$ instead of $z$, we can use the same formulas as for the restricted occupancy formulation to calculate the derived population estimates.

We need to specify priors for the survival and capture probabilities as before, but not for the conditional entry probabilities ($\eta$). Instead, we specify priors for the entry probabilities ($b$), and a convenient one is the Dirichlet prior: $b_t \sim$ Dirichlet($\boldsymbol{\alpha}$). To express ignorance and allocate the entries of all individuals uniformly over the $T$ occasions, we set $\alpha_t = 1$ for all $t$. In addition, we give a U(0, 1) prior for the inclusion probability $\psi$. This induces a discrete U(0, $M$) prior for the superpopulation size $N_s$ (Royle and Dorazio, 2008), as we did for the closed-population models (Chapter 6).

The BUGS code for the superpopulation parameterization again has constant survival and capture probabilities and time-dependent entry probabilities.

```
# Specify model in BUGS language
sink("js-super.bug")
cat("
model {

# Priors and constraints
for (i in 1:M){
    for (t in 1:(n.occasions-1)){
        phi[i,t] <- mean.phi
        } #t
```

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

```
    for (t in 1:n.occasions){
        p[i,t] <- mean.p
        } #t
    } #i

mean.phi ~ dunif(0, 1)          # Prior for mean survival
mean.p ~ dunif(0, 1)            # Prior for mean capture
psi ~ dunif(0, 1)               # Prior for inclusion probability

# Dirichlet prior for entry probabilities
for (t in 1:n.occasions){
    beta[t] ~ dgamma(1, 1)
    b[t] <- beta[t] / sum(beta[1:n.occasions])
    }

# Convert entry probs to conditional entry probs
nu[1] <- b[1]
for (t in 2:n.occasions){
    nu[t] <- b[t] / (1-sum(b[1:(t-1)]))
    } #t

# Likelihood
for (i in 1:M){
    # First occasion
    # State process
    w[i] ~ dbern(psi)           # Draw latent inclusion
    z[i,1] ~ dbern(nu[1])
    # Observation process
    mu1[i] <- z[i,1] * p[i,1] * w[i]
    y[i,1] ~ dbern(mu1[i])

    # Subsequent occasions
    for (t in 2:n.occasions){
        # State process
        q[i,t-1] <- 1-z[i,t-1]
        mu2[i,t] <- phi[i,t-1] * z[i,t-1] + nu[t] * prod(q[i,1:(t-1)])
        z[i,t] ~ dbern(mu2[i,t])
        # Observation process
        mu3[i,t] <- z[i,t] * p[i,t] * w[i]
        y[i,t] ~ dbern(mu3[i,t])
        } #t
    } #i

# Calculate derived population parameters
for (i in 1:M){
    for (t in 1:n.occasions){
        u[i,t] <- z[i,t]*w[i]     # Deflated latent state (u)
        }
    }
for (i in 1:M){
    recruit[i,1] <- u[i,1]
    for (t in 2:n.occasions){
        recruit[i,t] <- (1-u[i,t-1]) * u[i,t]
        } #t
    } #i
```

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


```
for (t in 1:n.occasions){
    N[t] <- sum(u[1:M,t])           # Actual population size
    B[t] <- sum(recruit[1:M,t])     # Number of entries
    } #t
for (i in 1:M){
    Nind[i] <- sum(u[i,1:n.occasions])
    Nalive[i] <- 1-equals(Nind[i], 0)
    } #i
Nsuper <- sum(Nalive[])             # Superpopulation size
}
",fill=TRUE)
sink()
```

Under the restricted occupancy or the multistate JS model and assuming $T$ capture occasions and time-dependent parameters, we estimate $T$ $\gamma$ parameters, $T-1$ survival, and $T$ capture parameters. Under the superpopulation approach, we estimate the same number of survival and capture parameters, but only $T-1$ entry parameters are separately estimable (note that $b_T$ is 1 minus the sum of the other $b$) plus the inclusion parameter $\psi$. Thus, the total number of parameters is the same in all three formulations. This illustrates the fact that the different formulations are reparameterizations of the same basic model. In Section 10.6, we provide a summary of the relationships between the quantities in each. The parameterization with the least impact of the priors can then be specified.

Naturally, all model formulations can be extended using the GLM or GLMM framework, as we did for the models in Chapters 6 and 7. Care must be taken, however, with age-dependent models (Brownie et al., 1986; Williams et al., 2002). The age of all individuals at initial capture must be known, and the entry time to the population must be somewhere between the birth of the individual and its initial capture. Capture probabilities depend on age, but the capture probability of the first age class and therefore also the population size for this age class cannot be estimated. The multistate formulation of the JS model seems to be a framework with which age-dependent models can be fitted in the Bayesian paradigm, but the problem remains that population size of the first age class cannot be estimated. Care must also be taken when entry probabilities are modeled because they need to sum to 1, and therefore not all of them can independently be modeled.

## 10.4 MODELS WITH CONSTANT SURVIVAL AND TIME-DEPENDENT ENTRY

We start with a simple example to illustrate the application of all three model formulations. We first simulate data using function `simul.js` and then analyze them. The simulation code needs the parameter of

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



the superpopulation formulation as input, and thus we specify the superpopulation size, entry, survival and capture probabilities, as well as the number of occasions. With $T$ occasions, $T$ entry probabilities must be defined, and they have to sum to 1. In the absence of any individual heterogeneity, there are $T$ capture probabilities and $T-1$ survival probabilities that need to be determined.

We assume a 7-year study of survival and recruitment in black grouse (Fig. 10.3). Each spring black grouse are captured at a lek. They are marked and may be captured again in later years. We assume that annual survival is 0.7, and capture probability is 0.5; both are constant over time. Superpopulation size is 400 individuals. We assume the entry probability to be 0.11 for all occasions but the first one, when it is given by $1-6*0.11=0.34$.

```
# Define parameter values
n.occasions <- 7                          # Number of capture occasions
N <- 400                                  # Superpopulation size
phi <- rep(0.7, n.occasions-1)            # Survival probabilities
b <- c(0.34, rep(0.11, n.occasions-1))    # Entry probabilities
p <- rep(0.5, n.occasions)                # Capture probabilities
```

bba253f30603a496429325e0286b311e
ebrary

FIGURE 10.3 Displaying black grouse cock (*Tetrao tetrix*), Finland, 2005 (Photograph by J. Peltomäki).

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


```
PHI <- matrix(rep(phi, (n.occasions-1)*N), ncol = n.occasions-1,
   nrow = N, byrow = T)
P <- matrix(rep(p, n.occasions*N), ncol = n.occasions, nrow = N,
   byrow = T)

# Function to simulate capture-recapture data under the JS model
simul.js <- function(PHI, P, b, N){
    B <- rmultinom(1, N, b) # Generate no. of entering ind. per occasion
    n.occasions <- dim(PHI)[2] + 1
    CH.sur <- CH.p <- matrix(0, ncol = n.occasions, nrow = N)

    # Define a vector with the occasion of entering the population
    ent.occ <- numeric()
    for (t in 1:n.occasions){
        ent.occ <- c(ent.occ, rep(t, B[t]))
        }

    # Simulate survival
    for (i in 1:N){
        CH.sur[i, ent.occ[i]] <- 1     # Write 1 when ind. enters the pop.
        if (ent.occ[i] == n.occasions) next
        for (t in (ent.occ[i]+1):n.occasions){
            # Bernoulli trial: has individual survived occasion?
            sur <- rbinom(1, 1, PHI[i,t-1])
            ifelse (sur==1, CH.sur[i,t] <- 1, break)
            } #t
        } #i

    # Simulate capture
    for (i in 1:N){
        CH.p[i,] <- rbinom(n.occasions, 1, P[i,])
        } #i

    # Full capture-recapture matrix
    CH <- CH.sur * CH.p

    # Remove individuals never captured
    cap.sum <- rowSums(CH)
    never <- which(cap.sum == 0)
    CH <- CH[-never,]
    Nt <- colSums(CH.sur)      # Actual population size
    return(list(CH=CH, B=B, N=Nt))
    }

# Execute simulation function
sim <- simul.js(PHI, P, b, N)
CH <- sim$CH
```

In our case, the resulting capture–recapture matrix has the dimension of $7 \times 294$, and thus 294 among the 400 individuals were captured. Besides the capture-histories, the function returns the number of individuals that newly entered the population and the actual population size at each occasion. We analyze the simulated data with all three model formulations, thus illustrating the advantages and disadvantages of each.

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



### 10.4.1 Analysis of the JS Model as a Restricted Occupancy Model

We first need to augment the observed capture–recapture data. We must make sure that we augment the data by a large enough number of pseudo-individuals (see Section 6.2.1).

```
# Augment the capture-histories by nz pseudo-individuals
nz <- 500
CH.aug <- rbind(CH, matrix(0, ncol = dim(CH)[2], nrow = nz))
```

Then, we define initial values and parameters we want to monitor, set the MCMC specifications, run the model and print the results.

```
# Bundle data
bugs.data <- list(y = CH.aug, n.occasions = dim(CH.aug)[2],
   M = dim(CH.aug)[1])

# Initial values
inits <- function(){list(mean.phi = runif(1, 0, 1),
   mean.p = runif(1, 0, 1), z = CH.aug)}

# Parameters monitored
parameters <- c("psi", "mean.p", "mean.phi", "b", "Nsuper", "N", "B",
   "gamma")

# MCMC settings
ni <- 5000
nt <- 3
nb <- 2000
nc <- 3

# Call WinBUGS from R (BRT 11 min)
js.occ <- bugs(bugs.data, inits, parameters, "js-rest.occ.bug",
   n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
   bugs.directory = bugs.dir, working.directory = getwd())
print(js.occ, digits = 3)
```

```
            mean     sd    2.5%     25%     50%     75%   97.5%  Rhat n.eff
psi        0.522  0.032   0.465   0.500   0.522   0.543   0.590 1.002  1800
mean.p     0.473  0.035   0.407   0.448   0.472   0.497   0.542 1.003   930
mean.phi   0.718  0.025   0.668   0.701   0.719   0.736   0.768 1.001  2400
b[1]       0.328  0.040   0.252   0.300   0.327   0.354   0.408 1.003   870
[ ... ]
b[7]       0.089  0.029   0.036   0.068   0.088   0.108   0.150 1.004  3000
Nsuper   406.364 21.013 369.975 391.000 405.000 419.000 451.000 1.001  2500
N[1]     134.027 15.971 106.000 123.000 132.000 144.000 169.000 1.003   760
[ ... ]
N[7]     150.457 14.373 126.000 140.000 149.000 159.000 183.000 1.002  1800
B[1]     134.027 15.971 106.000 123.000 132.000 144.000 169.000 1.003   760
[ ... ]
B[7]      35.751 11.451  14.975  28.000  35.000  43.000  59.000 1.001  2100
[ ... ]
```

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

## 10.4.2 Analysis of the JS Model as a Multistate Model

We need to add a dummy occasion before the first real occasion, augment the data set, and recode that data to match the codes of the observed states.

```
# Add dummy occasion
CH.du <- cbind(rep(0, dim(CH)[1]), CH)

# Augment data
nz <- 500
CH.ms <- rbind(CH.du, matrix(0, ncol = dim(CH.du)[2], nrow = nz))

# Recode CH matrix: a 0 is not allowed in WinBUGS!
CH.ms[CH.ms==0] <- 2                    # Not seen = 2, seen = 1
```

Then, we run the analysis.

```
# Bundle data
bugs.data <- list(y = CH.ms, n.occasions = dim(CH.ms)[2],
   M = dim(CH.ms)[1])

# Initial values
inits <- function(){list(mean.phi = runif(1, 0, 1),
   mean.p = runif(1, 0, 1), z = cbind(rep(NA, dim(CH.ms)[1]),
   CH.ms[,-1]))}

# Parameters monitored
parameters <- c("mean.p", "mean.phi", "b", "Nsuper", "N", "B")

# MCMC settings
ni <- 20000
nt <- 3
nb <- 5000
nc <- 3

# Call WinBUGS from R (BRT 32 min)
js.ms <- bugs(bugs.data, inits, parameters, "js-ms.bug", n.chains = nc,
   n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
   bugs.dir, working.directory = getwd())

print(js.ms, digits = 3)
```

```
            mean    sd    2.5%     25%     50%     75%   97.5%  Rhat n.eff
mean.p     0.473 0.036   0.403   0.449   0.473   0.497   0.542 1.009   360
mean.phi   0.720 0.026   0.668   0.702   0.720   0.737   0.770 1.001 15000
b[1]       0.328 0.040   0.254   0.300   0.326   0.354   0.410 1.002  1500
[ ... ]
b[7]       0.088 0.029   0.032   0.068   0.087   0.107   0.146 1.001 11000
Nsuper   405.668 19.807 370.000 392.000 405.000 419.000 447.000 1.015   170
N[1]     133.950 16.146 106.000 122.000 133.000 144.000 169.000 1.010   280
[ ... ]
N[7]     150.550 14.173 126.000 140.000 150.000 160.000 181.000 1.003   970
B[1]     133.950 16.146 106.000 122.000 133.000 144.000 169.000 1.010   280
[ ... ]
B[7]      35.503 11.277  13.000  28.000  35.000  43.000  58.000 1.001 15000
[ ... ]
```

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



### 10.4.3 Analysis of the JS Model Under the Superpopulation Parameterization

This analysis requires the same preparation as the restricted occupancy formulation. We augment the observed capture–recapture data and run the analysis.

```
# Augment capture-histories by nz pseudo-individuals
nz <- 500
CH.aug <- rbind(CH, matrix(0, ncol = dim(CH)[2], nrow = nz))

# Bundle data
bugs.data <- list(y = CH.aug, n.occasions = dim(CH.aug)[2],
   M = dim(CH.aug)[1])

# Initial values
inits <- function(){list(mean.phi = runif(1, 0, 1),
   mean.p = runif(1, 0, 1), psi = runif(1, 0, 1), z = CH.aug)}

# Parameters monitored
parameters <- c("psi", "mean.p", "mean.phi", "b", "Nsuper",
   "N", "B", "nu")

# MCMC settings
ni <- 5000
nt <- 3
nb <- 2000
nc <- 3

# Call WinBUGS from R (BRT 40 min)
js.super <- bugs(bugs.data, inits, parameters, "js-super.bug",
   n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
   bugs.directory = bugs.dir, working.directory = getwd())

print(js.super, digits = 3)
```

```
            mean     sd    2.5%     25%     50%     75%   97.5%  Rhat n.eff
psi        0.504  0.029   0.449   0.484   0.504   0.524   0.563 1.001  3000
mean.p     0.483  0.034   0.417   0.459   0.482   0.506   0.548 1.002  1700
mean.phi   0.720  0.026   0.671   0.702   0.719   0.737   0.773 1.002  1500
b[1]       0.323  0.040   0.251   0.295   0.320   0.348   0.408 1.017   180
[ ... ]
b[7]       0.087  0.029   0.034   0.067   0.087   0.106   0.145 1.007   300
Nsuper   396.991 18.199 364.975 384.000 396.000 408.000 436.000 1.001  3000
N[1]     129.561 15.079 103.000 119.000 128.000 139.000 161.000 1.010   240
[ ... ]
N[7]     147.403 13.196 125.000 138.000 146.000 156.000 176.000 1.001  2300
B[1]     129.561 15.079 103.000 119.000 128.000 139.000 161.000 1.010   240
[ ... ]
B[7]      34.215 10.768  14.000  27.000  34.000  41.000  56.000 1.006   380
[ ... ]
```

### 10.4.4 Comparison of Estimates

Posterior means and standard deviations are nearly identical under all three parameterizations. This is shown graphically for the superpopulation

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



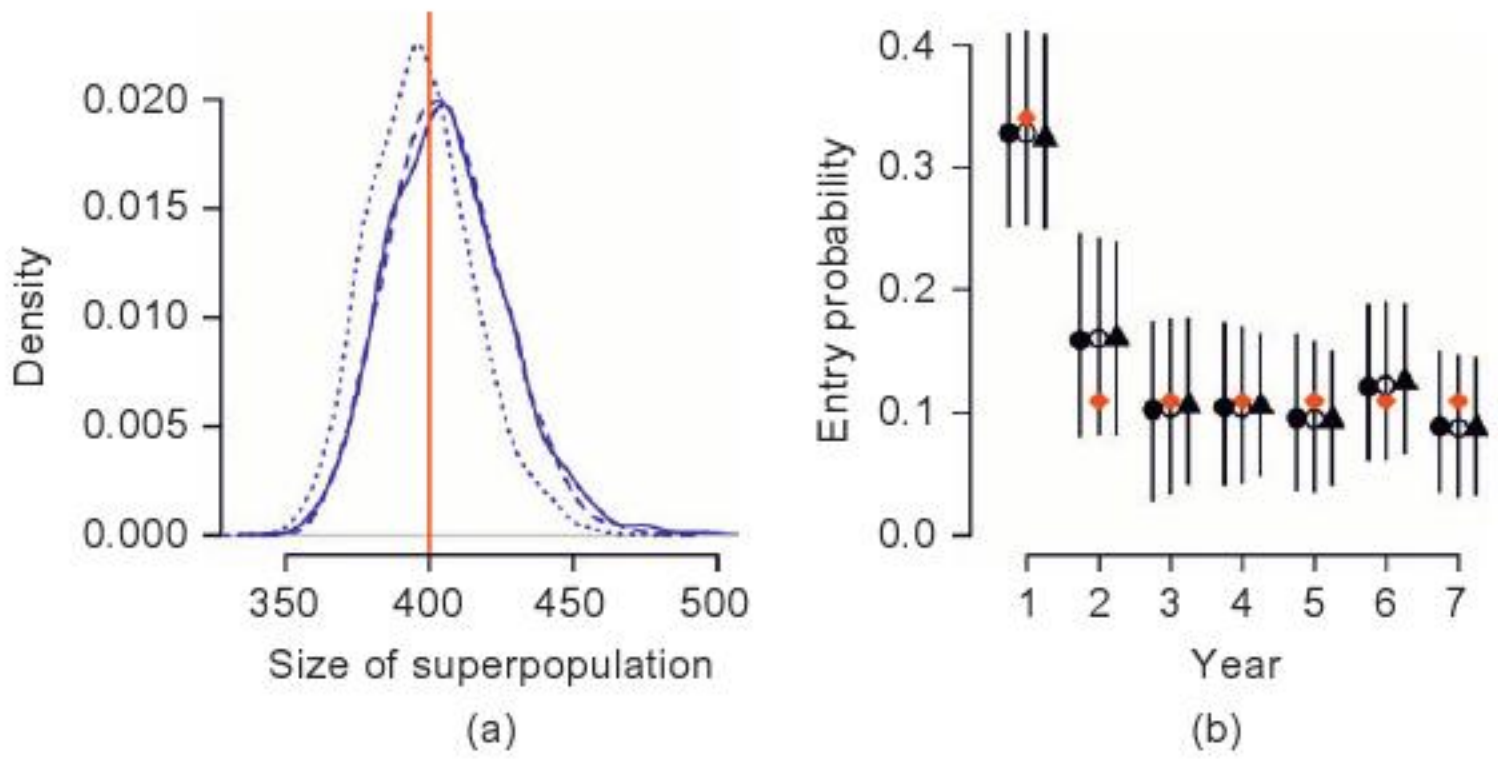


**FIGURE 10.4** (a) Posterior distributions of the size of the number of black grouses ever alive during the study (superpopulation size, $N_s$) using the restricted occupancy (solid), multistate (dashed), and superpopulation parameterizations (dotted). The vertical red line shows the data generating size of the superpopulation. (b) Estimates of annual entry probabilities under the restricted occupancy (closed circle), multistate (open circle), and superpopulation parameterization (closed triangle), as well as the data-generating parameters (red diamond). The vertical lines are 95% CRI.

size and the entry probabilities (Fig. 10.4). We would expect to get exactly the same results for the restricted occupancy and the multistate formulation, but not for the superpopulation formulation. The latter uses different priors than the two former. This may make a difference with a small data set. The three approaches differ in terms of computing time, which is much shorter for the restricted occupancy formulation. The main advantage of the multistate formulation is its flexibility to model different age classes, movement between sites or to integrate other data types such as dead recoveries. The main advantage of the superpopulation formulation is the ability to specify directly a prior for the superpopulation size and to model entry probabilities directly. Entry probabilities are parameters of direct biological interest, whereas removal entry probabilities are mere mathematical constructs. Thus, if the goal of an analysis is to understand the arrival of individuals as a function of covariates, the superpopulation formulation is the best choice. A difficulty could be that the entry probabilities must sum to 1, so not all parameters can be freely modeled. This is the same difficulty as for multistate models with movements among three or more sites (Section 9.6). In order to improve computation speed, it is again possible to provide as data information about the latent state variable $z$ (see Section 7.3.1). We have not done this here explicitly because the specification of them and of the corresponding initial values differs for each approach. However, we make use of this information in the solutions of the exercises (Section 10.9).

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



```
# Code to produce Fig. 10.4
par(mfrow = c(1,2), mar = c(5, 6, 2, 1), mgp=c(3.4, 1, 0), las = 1)
plot(density(js.occ$sims.list$Nsuper), main = "", xlab = "",
   ylab = "Density", frame = FALSE, lwd = 2, ylim=c(0, 0.023),
   col = "blue")
points(density(js.ms$sims.list$Nsuper), type = "l", lty = 2,
   col = "blue", lwd = 2)
points(density(js.super$sims.list$Nsuper), type = "l", lty = 3,
   col = "blue", lwd = 2)
abline(v = N, col = "red", lwd = 2)
mtext("Size of superpopulation", 1, line = 3)

b1.lower <- b2.lower <- b3.lower <- b1.upper <- b2.upper <- b3.upper <-
   numeric()
for (t in 1:n.occasions){
    b1.lower[t] <- quantile(js.occ$sims.list$b[,t], 0.025)
    b2.lower[t] <- quantile(js.ms$sims.list$b[,t], 0.025)
    b3.lower[t] <- quantile(js.super$sims.list$b[,t], 0.025)
    b1.upper[t] <- quantile(js.occ$sims.list$b[,t], 0.975)
    b2.upper[t] <- quantile(js.ms$sims.list$b[,t], 0.975)
    b3.upper[t] <- quantile(js.super$sims.list$b[,t], 0.975)
    }
time <- 1:n.occasions
plot(x = time-0.25, y = js.occ$mean$b, xlab = "", ylab = "Entry
   probability", frame = FALSE, las = 1, xlim = c(0.5, 7.5), pch = 16,
   ylim = c(0, max(c(b1.upper, b2.upper))))
segments(time-0.25, b1.lower, time-0.25, b1.upper)
points(x = time, y = js.ms$mean$b, pch = 1)
segments(time, b2.lower, time, b2.upper)
points(x = time+0.25, y = js.super$mean$b, pch = 17)
segments(time+0.25, b3.lower, time+0.25, b3.upper)
points(x = time, y = b, pch = 18, col = "red")
mtext("Year", 1, line = 3)
```

bba253f30603a496429325e0286b311e
ebrary

## 10.5 MODELS WITH INDIVIDUAL CAPTURE HETEROGENEITY

Here, we show how an individual random effect on capture probability can be included to avoid negative bias in population size estimates (Pledger and Efford, 1998; Link, 2003; Royle and Dorazio, 2008); also see Chapter 6. We assume an 8-year study of grey-headed woodpeckers (Fig. 10.5), where birds are attracted with playback calls and caught in mist nets. There is individual variation in the aggressive reaction to the calls; hence, capture probability differs by individual. We assume a superpopulation size of 300, mean survival probability of 0.75, and annual entry probability of 0.09, except for 0.37 at the first occasion. Mean capture probability is 0.6, with an individual variance of 1 on the logit scale. We first simulate one data set and then analyze it with the superpopulation formulation.

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

FIGURE 10.5 Male grey-headed woodpecker (*Picus canus*), Finland, 2002 (Photograph by T. Muukkonen).

```
# Define parameter values
n.occasions <- 8                          # Number of capture occasions
N <- 300                                  # Size of the superpopulation
phi <- rep(0.75, n.occasions-1)           # Survival probabilities
b <- c(0.37, rep(0.09, n.occasions-1))    # Entry probabilities
mean.p <- 0.6                             # Mean capture probability
var.p <- 1                                # Indv. Variance of capture
                                            prob.
p <- plogis(rnorm(N, qlogis(mean.p), var.p^0.5))

PHI <- matrix(rep(phi, (n.occasions-1)*N), ncol=n.occasions-1,
   nrow=N, byrow=T)
P <- matrix(rep(p, n.occasions), ncol=n.occasions, nrow=N, byrow=F)

# Execute simulation function
sim <- simul.js(PHI, P, b, N)
CH <- sim$CH
```

Here is the BUGS code.

```
# Specify model in BUGS language
sink("js-super-indran.bug")
cat("
model {
```

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


```
# Priors and constraints
for (i in 1:M){
    for (t in 1:(n.occasions-1)){
        phi[i,t] <- mean.phi
        } #t
    for (t in 1:n.occasions){
        logit(p[i,t]) <- mean.lp + epsilon[i]
        } #t
    } #i

mean.phi ~ dunif(0, 1)                  # Prior for mean survival
mean.lp <- log(mean.p / (1-mean.p))
mean.p ~ dunif(0, 1)                    # Prior for mean capture
for (i in 1:M){
    epsilon[i] ~ dnorm(0, tau)I(-15,15)
    }
tau <- pow(sigma, -2)
sigma ~ dunif(0, 5)                     # Prior for sd of indv. variation of p
sigma2 <- pow(sigma, 2)
psi ~ dunif(0, 1)                       # Prior for inclusion probability

# Dirichlet prior for entry probabilities
for (t in 1:n.occasions){
    beta[t] ~ dgamma(1, 1)
    b[t] <- beta[t] / sum(beta[1:n.occasions])
    }

# Convert entry probs to conditional entry probs
nu[1] <- b[1]
for (t in 2:n.occasions){
    nu[t] <- b[t] / (1-sum(b[1:(t-1)]))
    } #t

# Likelihood
for (i in 1:M){
    # First occasion
    # State process
    w[i] ~ dbern(psi)                   # Draw latent inclusion
    z[i,1] ~ dbern(nu[1])
    # Observation process
    mu1[i] <- z[i,1] * p[i,1] * w[i]
    y[i,1] ~ dbern(mu1[i])
    # Subsequent occasions
    for (t in 2:n.occasions){
        # State process
        q[i,t-1] <- 1-z[i,t-1]
        mu2[i,t] <- phi[i,t-1] * z[i,t-1] + nu[t] * prod(q[i,1:(t-1)])
        z[i,t] ~ dbern(mu2[i,t])
        # Observation process
        mu3[i,t] <- z[i,t] * p[i,t] * w[i]
        y[i,t] ~ dbern(mu3[i,t])
        } #t
} #i
```

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


```
# Calculate derived population parameters
for (i in 1:M){
    for (t in 1:n.occasions){
        u[i,t] <- z[i,t]*w[i]       # Deflated latent state (u)
        }
    }
for (i in 1:M){
    recruit[i,1] <- u[i,1]
    for (t in 2:n.occasions){
        recruit[i,t] <- (1-u[i,t-1]) * u[i,t]
        } #t
    } #i
for (t in 1:n.occasions){
    N[t] <- sum(u[1:M,t])           # Actual population size
    B[t] <- sum(recruit[1:M,t])     # Number of entries
    } #t
for (i in 1:M){
    Nind[i] <- sum(u[i,1:n.occasions])
    Nalive[i] <- 1-equals(Nind[i], 0)
    } #i
Nsuper <- sum(Nalive[])             # Superpopulation size
}
",fill=TRUE)
sink()
```

Finally, we augment the data set and run the analysis in BUGS.

```
# Augment the capture-histories by nz pseudo-individuals
nz <- 300
CH.aug <- rbind(CH, matrix(0, ncol = dim(CH)[2], nrow = nz))

# Bundle data
bugs.data <- list(y = CH.aug, n.occasions = dim(CH.aug)[2],
   M = dim(CH.aug)[1])

# Initial values
inits <- function(){list(mean.phi = runif(1, 0, 1),
   mean.p = runif(1, 0, 1), sigma = runif(1, 0, 1), z = CH.aug)}

# Parameters monitored
parameters <- c("sigma2","psi", "mean.p", "mean.phi", "N", "Nsuper",
   "b", "B")

# MCMC settings
ni <- 20000
nt <- 6
nb <- 5000
nc <- 3

# Call WinBUGS from R (BRT 179 min)
js.ran <- bugs(bugs.data, inits, parameters, "js-super-indran.bug",
   n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
   bugs.directory = bugs.dir, working.directory = getwd())
```

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



```
print(js.ran, digits = 3)

            mean     sd    2.5%     25%     50%     75%   97.5%  Rhat n.eff
sigma2     0.191  0.281   0.001   0.019   0.083   0.242   0.980 1.135    23
psi        0.542  0.033   0.483   0.520   0.540   0.562   0.613 1.002  2000
mean.p     0.629  0.037   0.550   0.607   0.631   0.655   0.696 1.003  1100
mean.phi   0.746  0.021   0.704   0.732   0.746   0.760   0.787 1.005   520
N[1]     121.716 10.153 104.000 115.000 121.000 128.000 143.000 1.005   520
[ ... ]
N[8]     113.286  9.498  97.000 107.000 113.000 119.000 134.000 1.005   490
Nsuper   291.239 13.436 270.000 282.000 290.000 298.000 323.000 1.003   780
b[1]       0.410  0.039   0.335   0.383   0.409   0.436   0.485 1.003   770
[ ... ]
b[8]       0.155  0.030   0.099   0.135   0.154   0.175   0.215 1.003   800
B[1]     121.716 10.153 104.000 115.000 121.000 128.000 143.000 1.005   520
[ ... ]
B[8]      45.505  7.246  32.000  40.000  45.000  50.000  61.000 1.002  1200
[ ... ]
```

The posterior means match up quite well with the data-generating parameters. Survival, capture and the entry probabilities are estimated well, whereas the estimate for the individual variation is lower than the data-generating parameter. Yet, as always, in order to study whether the model produces unbiased estimates, the exercise would have to be repeated many times and the average behavior of estimates studied.

## 10.6 CONNECTIONS BETWEEN PARAMETERS, FURTHER QUANTITIES AND SOME REMARKS ON IDENTIFIABILITY

All three approaches to the JS model under data augmentation are related. The restricted occupancy and the multistate formulation are exactly equivalent in terms of parameterization and priors. In contrast, the superpopulation formulation has a different parameterization (in terms of $b$ and $\psi$ instead of $\gamma$) and consequently needs priors for other parameters. Here, we summarize connections between different parameters in the three approaches and also their relation to further quantities of interest.

The expected number of newly entered individuals per occasion is

$$\mathrm{E}(B_1) = M\gamma_1 = N_s b_1$$
$$\mathrm{E}(B_2) = M(1-\gamma_1)\gamma_2 = N_s b_2$$
$$\ldots$$
$$\mathrm{E}(B_t) = M\prod_{i=1}^{t-1}(1-\gamma_i)\gamma_t = N_s b_t$$

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


Let us denote the probability that an "individual" within the augmented data $M$ is a member of the true individuals $N_s$ with $\psi = N_s/M$. After some algebra, we see that

$$\gamma_1 = \psi b_1, \quad \gamma_2 = \psi \frac{b_2}{1-b_1}, \quad \ldots, \quad \gamma_t = \psi \frac{b_t}{1-\sum_{i=1}^{t-1} b_i}$$

Likewise, we can calculate $b$ from $\gamma$ as

$$b_1 = \frac{1}{\psi}\gamma_1, \quad b_2 = \frac{1}{\psi}(1-\gamma_1)\gamma_2, \ldots$$

Because all $b$ sum to 1, $b_T$ at the last occasion $T$ is

$$b_T = \frac{1}{\psi}\gamma_T \prod_{i=1}^{T-1}(1-\gamma_i) = 1 - \frac{1}{\psi}\left(\gamma_1 + \sum_{i=1}^{T-1}\left(\gamma_{i+1}\prod^{i}(1-\gamma_i)\right)\right)$$

bba253f30603a496429325e0286b311e
ebrary

Therefore, we can directly calculate $\psi$ from $\gamma$ as

$$\psi = 1 - \prod_{i=1}^{T}(1-\gamma_i)$$

Pradel (1996) and Link and Barker (2005) used a further parameterization to model the recruitment process. Instead of an entry probability that refers either to the size of the augmented data set ($\gamma$, restricted occupancy parameterization and multistate model) or to the size of the superpopulation ($b$, superpopulation parameterization), they defined a per-capita entry probability ($f$). This quantity is computed as

$$f_t = \frac{B_t}{N_t}$$

and expresses the fraction of new individuals at $t$ per individual alive at $t$. Expressing recruitment in this way results in the biologically most meaningful quantity. For the three models in this chapter, the per-capita entry probability can easily be estimated as a derived parameter, but to model this quantity directly, a different model parameterization is needed (Link and Barker, 2010).

The population growth rate ($\lambda$) is easily computed as a derived quantity from the estimated population sizes or survival and per-capita entry probability:

$$\lambda_t = \frac{N_{t+1}}{N_t} = \phi_t + f_t.$$

As almost always with complex models, not all parameters may be separately identifiable. It is well known that some parameters are not identifiable in a JS model with a time-dependent structure on survival,

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a49642932 5e0286b311e
ebrary



capture, and entry probabilities: the first entry and capture probabilities and the last survival and capture probabilities, respectively, are only estimable as products (i.e., $b_1 p_1$, $\phi_{T-1} p_T$; Schwarz and Arnason, 1996). This confounding occurs for all three parameterizations in this chapter. To verify parameter estimability, we can inspect the prior/posterior overlap (see Section 7.9) or see whether an analysis of simulated data yields estimates that resemble the input parameters.

## 10.7 ANALYSIS OF A REAL DATA SET: SURVIVAL, RECRUITMENT AND POPULATION SIZE OF LEISLER'S BATS

As a real-world example, we reanalyze capture–recapture data from adult female Leisler's bat (Fig. 7.11) from Section 7.11 (see also Schorcht et al., 2009) under the JS model. We only use a homogenous subset of these data consisting of locally born adult females. We are interested in population size and in the variability of local recruitment over time. Our data set consists of 181 individuals. Initial modeling suggested that adult survival was subject to strong temporal variation, whereas recapture probability was constant over time. We therefore fit a model with temporal random effects in survival, fixed time effects in recruitment, and a constant capture rate. We denote this model as $(\phi_t, b_t, p.)$

We first specify the BUGS model. We use the restricted occupancy formulation because of its advantage in terms of computation speed. The model code needs an adaptation: annual survival is now specified as a random effect.

```
# Specify model in BUGS language
sink("js-tempran.bug")
cat("
model {

# Priors and constraints
for (i in 1:M){
    for (t in 1:(n.occasions-1)){
        logit(phi[i,t]) <- mean.lphi + epsilon[t]
        } #t
    for (t in 1:n.occasions){
        p[i,t] <- mean.p
        } #t
    } #i
mean.p ~ dunif(0, 1)                    # Prior for mean capture
mean.phi ~ dunif(0, 1)                  # Prior for mean survival
mean.lphi <- log(mean.phi / (1-mean.phi))
for (t in 1:(n.occasions-1)){
    epsilon[t] ~ dnorm(0, tau)
    }
```

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

```
tau <- pow(sigma, -2)
sigma ~ dunif(0, 5)             # Prior for sd of indv. variation of phi
sigma2 <- pow(sigma, 2)

for (t in 1:n.occasions){
    gamma[t] ~ dunif(0, 1)
    } #t

# Likelihood
for (i in 1:M){
    # First occasion
    # State process
    z[i,1] ~ dbern(gamma[1])
    mu1[i] <- z[i,1] * p[i,1]
    # Observation process
    y[i,1] ~ dbern(mu1[i])

    # Subsequent occasions
    for (t in 2:n.occasions){
        # State process
        q[i,t-1] <- 1-z[i,t-1]
        mu2[i,t] <- phi[i,t-1] * z[i,t-1] + gamma[t] * prod(q[i,1:
           (t-1)])
        z[i,t] ~ dbern(mu2[i,t])
        # Observation process
        mu3[i,t] <- z[i,t] * p[i,t]
        y[i,t] ~ dbern(mu3[i,t])
        } #t
    } #i
# Calculate derived population parameters
for (t in 1:n.occasions){
    qgamma[t] <- 1-gamma[t]
    }
cprob[1] <- gamma[1]
for (t in 2:n.occasions){
    cprob[t] <- gamma[t] * prod(qgamma[1:(t-1)])
    } #t
psi <- sum(cprob[])                 # Inclusion probability
for (t in 1:n.occasions){
    b[t] <- cprob[t] / psi          # Entry probability
    } #t
for (i in 1:M){
    recruit[i,1] <- z[i,1]
    for (t in 2:n.occasions){
        recruit[i,t] <- (1-z[i,t-1]) * z[i,t]
        } #t
    } #i
for (t in 1:n.occasions){
    N[t] <- sum(z[1:M,t])           # Actual population size
    B[t] <- sum(recruit[1:M,t])     # Number of entries
    } #t
for (i in 1:M){
    Nind[i] <- sum(z[i,1:n.occasions])
    Nalive[i] <- 1-equals(Nind[i], 0)
    } #i
```

bba253f30603a496429325e0286b311e ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



```
Nsuper <- sum(Nalive[])        # Size of superpopulation
}
",fill=TRUE)
sink()
```

Next, we load the capture–recapture data and augment the data.

```
leis <- as.matrix(read.table("leisleri.txt", sep = " ",
   header = FALSE))
nz <- 300
CH.aug <- rbind(leis, matrix(0, ncol = dim(leis)[2], nrow = nz))
```

Then, we run the analysis.

```
# Bundle data
bugs.data <- list(y = CH.aug, n.occasions = dim(CH.aug)[2],
   M = dim(CH.aug)[1])

# Initial values
inits <- function(){list(mean.phi = runif(1, 0, 1),
   mean.p = runif(1, 0, 1), sigma = runif(1, 0, 1), z = CH.aug)}

# Parameters monitored
parameters <- c("psi", "mean.p", "sigma2", "mean.phi", "N", "Nsuper",
   "b", "B")

# MCMC settings
ni <- 10000
nt <- 6
nb <- 5000
nc <- 3

# Call WinBUGS from R (BRT 127 min)
n1 <- bugs(bugs.data, inits, parameters, "js-tempran.bug", n.chains =
   nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
   bugs.directory = bugs.dir, working.directory = getwd())

print(n1, digits = 3)
```

```
            mean    sd    2.5%     25%     50%     75%   97.5%  Rhat n.eff
mean.p     0.725 0.030   0.665   0.704   0.726   0.746   0.782 1.001  2500
sigma2     0.477 0.350   0.064   0.247   0.397   0.613   1.363 1.012   260
mean.phi   0.743 0.039   0.669   0.717   0.743   0.768   0.821 1.003   690
[...]
Nsuper   205.896 6.576 194.000 201.000 205.000 210.000 220.000 1.003  2500
[...]
```

The model runs quite slowly, but relatively few MCMC samples are sufficient to obtain convergence. Estimates of mean survival and its temporal variability are close to those obtained under the CJS model (see Section 7.11), and also the capture probabilities differ only very slightly (JS: 0.725 vs. CJS: 0.747). Such a difference is not unusual since the parameters are not exactly the same: in the CJS model, it is the recapture probability, whereas in the JS, it is the capture probability for all occasions, including the first. The estimated population sizes suggest that bat

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary



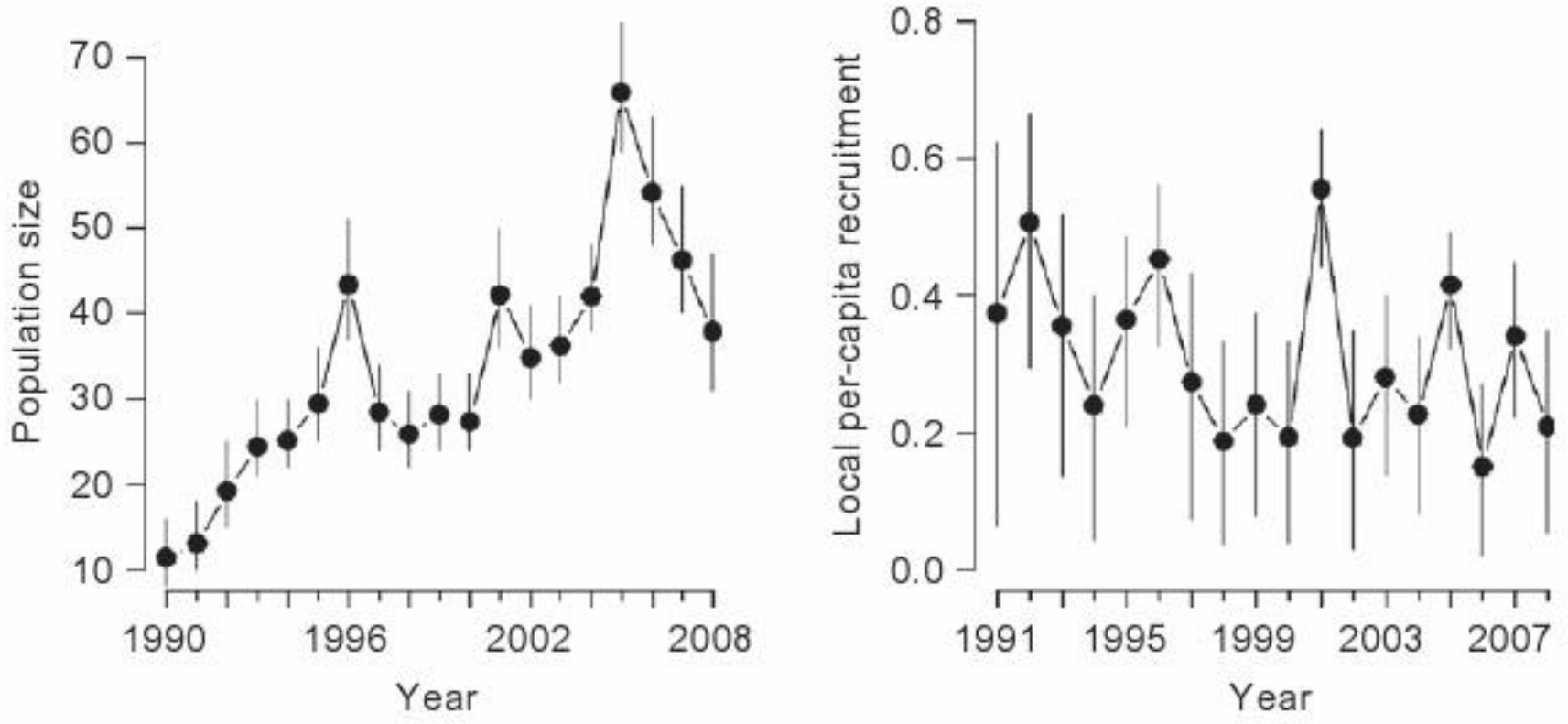


**FIGURE 10.6** Posterior means of population sizes and of per-capita recruitment of adult female Leisler's bats. The vertical lines are 95% CRI.

population increased until 2005 and declined afterwards (Fig. 10.6). The per-capita recruitment seemed to have declined slightly over time, but showed strong annual fluctuations (Fig. 10.6). Note that the per-capita recruitment refers to local recruitment only because the data set consisted only of locally born bats.

```
# Code to produce Fig. 10.6
# Calculate per-capita recruitment
T <- dim(leis)[2]
f <- matrix(NA, ncol = T, nrow = length(nl$sims.list$B[,1]))
for (t in 1:(T-1)){
    f[,t] <- nl$sims.list$B[,t+1] / nl$sims.list$N[,t+1]
    }
n.lower <- n.upper <- f.lower <- f.upper <- f.mean <- numeric()
for (t in 1:T){
    n.lower[t] <- quantile(nl$sims.list$N[,t], 0.025)
    n.upper[t] <- quantile(nl$sims.list$N[,t], 0.975)
    }
for (t in 1:(T-1)){
    f.lower[t] <- quantile(f[,t], 0.025)
    f.upper[t] <- quantile(f[,t], 0.975)
    f.mean[t] <- mean(f[,t])
    }

par(mfrow = c(1, 2))
plot(nl$mean$N, type = "b", pch = 19, ylab = "Population size", xlab = "",
   axes = F, cex = 1.5, ylim = c(10, max(n.upper)))
axis(1, at = seq(1, T, 2), labels = seq(1990, 2008, 2))
axis(1, at = 1:T, labels = rep("", T), tcl = -0.25)
axis(2, las = 1)
segments(1:T, n.lower, 1:T, n.upper)

plot(f.mean, type = "b", pch = 19, ylab = "Local per capita recruitment",
   xlab = "", axes = F, cex = 1.5, ylim = c(0, 0.8))
```

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


```
axis(1, at = seq(1, (T-1), 2), labels = seq(1991, 2008, 2))
axis(1, at = 1:(T-1), labels = rep("", T-1), tcl = -0.25)
axis(2, las = 1)
segments(1:(T-1), f.lower, 1:(T-1), f.upper)
```

## 10.8 SUMMARY AND OUTLOOK

We have described the Jolly-Seber (JS) model to estimate population size, survival and recruitment from capture–recapture data. The main difference to the CJS model (Chapter 7) is that the JS does not condition on first capture, that is, leading zeros before initial capture and the initial capture are modeled as well. There are several variants (parameterizations) of JS models that differ in the way how recruitment-related parameters are defined and estimated. We have illustrated three different parameterizations, showed advantages and disadvantages, as well as the connection between the parameters in them. The formulation of the JS model as a restricted occupancy model has advantages of relatively fast speed and a simple model code. The superpopulation formulation runs relatively slowly, but it is possible to directly model the entry probability and to specify a natural prior for the size of the superpopulation. We also showed that the JS model can be formulated as a multistate model. This formulation is appealing because it allows extensions in many directions. For all three formulations, we use parameter-expanded data augmentation (Royle et al., 2007a; Royle and Dorazio, 2011).

The GLM concept and random effects can be applied to all structural parameters in all three formulations of the JS model. Particular care must be taken, however, with age-dependent models and when the entry process is modeled.

Sometimes capture–recapture data arise under the robust design, that is, under a two-stage temporal sampling scheme, where the population is assumed open between primary occasions but closed between secondary, within-primary, occasions (Pollock, 1982; Kendall et al., 1997). The multiseason metapopulation estimation models in Chapters 12 and 13 also require the robust design data format. The JS model to analyze capture–recapture data sampled under the robust design is a combination of open- and closed-population models and has many advantages. First, it allows the estimation of survival, recruitment-related parameters, and population sizes as does the traditional JS model. Yet, because there is more information about the capture process, the robust design model can deal with certain violations of the basic JS model assumptions, such as permanent trap response. It also provides the basis to decompose entry probability into in situ recruitment and immigration (Nichols and Pollock, 1990) and to estimate population size of all age classes. The restricted occupancy and the superpopulation formulations of the JS model only require a small

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.

bba253f30603a496429325e0286b311e
ebrary


modification in the observation model to fit the robust design model. In its simplest version, we just need to model the observation with the Binomial instead of the Bernoulli distribution

$$Y_{i,t} \mid z_{i,t} \sim \text{Binomial}(K, z_{i,t} p_{i,t})$$

where the binomial total ($K$) is the number of secondary occasions (Royle and Dorazio, 2008; Royle and Dorazio, 2011). The observed data for individual $i$ and each primary occasion $t$ ($Y_{i,t}$) is then the number of times individual $i$ was captured during the primary occasion $t$. Of course, more complex models with binary response require more adaptations.

## 10.9 EXERCISES

1. Simulate capture–recapture data of a species for males and females. The study is conducted for 8 years; the mean survival of males is 0.75 and of females 0.5, and capture is 0.4 for both. The entry probability after the first occasion is 0.1 in both sexes. The size of the superpopulation is 300 in both sexes. Simulate one data set and analyze it with the model ($\phi_{sex}$, $b_t$, $p$).
2. Simulate capture–recapture data of a species collected over 7 years. Mean survival is 0.5, mean capture is 0.6, and entry probability is 0.1 for all but the first occasion. The size of the superpopulation is assumed to be 500. Analyze the data with the model that explicitly uses constant entry probability for all occasions, but the first.
3. Simulate data for a species for which capture–recapture data are sampled and recoveries of dead individuals are available. The study runs for 10 years, mean survival is 0.5, mean capture is 0.6, mean recovery is 0.2, and entry probability is 0.1 for all but the first occasion. The size of the superpopulation is assumed to be 500. Analyze the simulated data with an appropriate model.

bba253f30603a496429325e0286b311e
ebrary

Kery, Marc, and Schaub, Michael. Bayesian Population Analysis using WinBUGS : A Hierarchical Perspective. St. Louis, MO, USA: Academic Press, 2011. ProQuest ebrary. Web. 23 July 2015.
Copyright © 2011. Academic Press. All rights reserved.