

11

Estimation of Demographic Rates, Population Size, and Projection Matrices from Multiple Data Types Using Integrated Population Models

OUTLINE

11.1 Introduction	348
11.2 Developing an Integrated Population Model (IPM)	350
11.2.1 First Step: Define the Link between Changes in Population Size and Demographic Rates	350
11.2.2 Second Step: Define the Likelihoods of Each Individual Data Set	352
11.2.3 Third Step: Formulate the Joint Likelihood	354
11.3 Example of a Simple IPM (Counts, Capture–Recapture, Reproduction)	357
11.3.1 Load Data	357
11.3.2 Analysis of the Model	358
11.4 Another Example of an IPM: Estimating Productivity without Explicit Productivity Data	363
11.5 IPMs for Population Viability Analysis	366
11.6 Real Data Example: Hoopoe Population Dynamics	371
11.7 Summary and Outlook	379
11.8 Exercises	380

11.1 INTRODUCTION

Chapters 6–10 introduced models to separately estimate population size, recruitment, survival, or movement probabilities from various types of data. Often, studies focus on population dynamics, that is, changes in abundance over time and demographic causes of those changes. In that case, the obvious thing to do is to combine different available data sets to get deeper insights into population dynamics and better estimates of the demographic quantities. The link between population size and demographic, or vital, rates is straightforward because the change in population size over time is the direct result of these demographic rates. More formally, we have

$$N_{t+1} = N_t \times g(s, f)$$

where N_t is population size in year t , s is survival, f is productivity, and g is some function. When a study population is geographically open, the argument of g also includes terms for immigration and emigration. Looking at this link between population size and demographic rates, an important point becomes evident: time-series data on the size of a population contain information about the underlying demographic processes. When data on population size and demographic rates are analyzed jointly, three benefits accrue:

1. There is information about demographic rates both from explicit data on demographic rates (e.g., mark-recoveries for survival) and from data on population size. As a result, demographic rates are estimated with increased precision in a combined analysis. At the same time, changes in population size are the result of four demographic rates, and incorporation of data on demographic rates will likely improve population size estimates.
2. When explicit data about a demographic rate (such as productivity) is lacking, it may be possible to estimate the demographic rate by exploiting the information about it from the data on population size.
3. A combined analysis of data on demographic rates and population size allows the simultaneous study of population processes (demographic rates) and the result of these processes (population size). This can be important as we are often interested in the demographic causes for population change in population ecology, conservation biology, or wildlife management. Thus, a joint analysis allows a comprehensive assessment of the state and the dynamics of a population (Baillie, 1991).

A relatively new modeling framework, which uses different types of data to simultaneously estimate trajectories of population size and demographic

parameters, is usually called an integrated population model (Besbeas et al., 2002; reviewed by Schaub and Abadi, 2011). This analytical framework uses a population model for the time-series data on population size and combines it with additional data to inform certain elements of the population model, such as survival, fecundity, or dispersal. The use of integrated population models in animal population ecology is relatively recent (Besbeas et al., 2002, 2003; Brooks et al., 2004; Thomas et al., 2005; Schaub et al., 2007; Baillie et al., 2009; Borysiewicz et al., 2009; King et al., 2010) although some varieties of these models have been used in fisheries (e.g., Elliott and Little, 2000; Maunder, 2004). Here, we focus on the Bayesian analysis of these models, although frequentist analyses using maximum likelihood are also possible and historically came before Bayesian analyses (see, e.g., Besbeas et al., 2002, 2003, 2005; Besbeas and Freeman, 2006; Gauthier et al., 2007; Tavecchia et al., 2009; Péron et al., 2010; De Valpine, 2011). Most frequentist analyses use Kalman filter techniques (Harvey, 1989) with the advantage that numerical optimization of the likelihood function is faster than posterior sampling by MCMC. Furthermore, model selection using AIC is straightforward when using maximum likelihood; however, this comes at the cost of stronger assumptions about distributional forms and linearity of the model (Brooks et al., 2004; De Valpine, 2011).

Developing an integrated population model involves three basic steps (Schaub and Abadi, 2011). First, we need to develop a population model that links the demographic rates with changes in population size. Typically, an age- or stage-classified matrix population model (Caswell, 2001) is used. Second, we write down the likelihood of all data sets available. One data set that is always required for a classical integrated population model is a time series of estimated population sizes or of population indices (counts). To separate process variability from observation errors in these data, we can use a state-space model as used in Chapter 5. Other data sets may be capture–recapture data, for which we may adopt the CJS model, or ring-recovery data. Third, we construct the joint likelihood—the likelihood of the complete model—and make inferences. Under the assumption of independence of all data sets, the joint likelihood is the product of the likelihoods of the individual data sets. In the frequentist framework, the joint likelihood would be maximized, while in a Bayesian analysis of the model, we combine it with prior distributions for all unknowns and use MCMC to sample the joint posterior distribution. To see more clearly how such a model is constructed and analyzed, we will walk through an example and comment each step.

We first use a simulated data set. We will not show or explain the simulation code, as this would take too much room. However, you can find R code for the simulation of our data in Web appendix 2, and comments on simulating these data are given in the study by Abadi et al. (2010a). Our data mimic the dynamics of a population of ortolan buntings (Fig. 11.1), a small passerine



FIGURE 11.1 Singing male ortolan bunting (*Emberiza hortulana*), Switzerland (Photograph by P. Keusch).

species. We assume a 10-year study in a large study area, where the population is closely surveyed. Each year we record the number of singing males and the number of fledglings produced in nests that are found. Both nestlings and adults are caught and ringed. Thus, we have three data sets that contain information about the dynamics of this population: population size (the number of singing males), fecundity (the total number of fledglings produced in the surveyed broods), and survival (capture–recapture data from juveniles and adults). We chose the following parameter values for the simulation of the data sets: 0.26 for juvenile survival, 0.5 for adult survival, 0.6 for recapture, and 4.0 for productivity. We assumed constant parameters over time, which results in a population growth rate of 1.02.

11.2 DEVELOPING AN INTEGRATED POPULATION MODEL (IPM)

11.2.1 First Step: Define the Link between Changes in Population Size and Demographic Rates

First, we need to link changes in population size with the demographic parameters. We use a simple female-based, age-classified population projection matrix model (Caswell, 2001). We assume that all individuals start

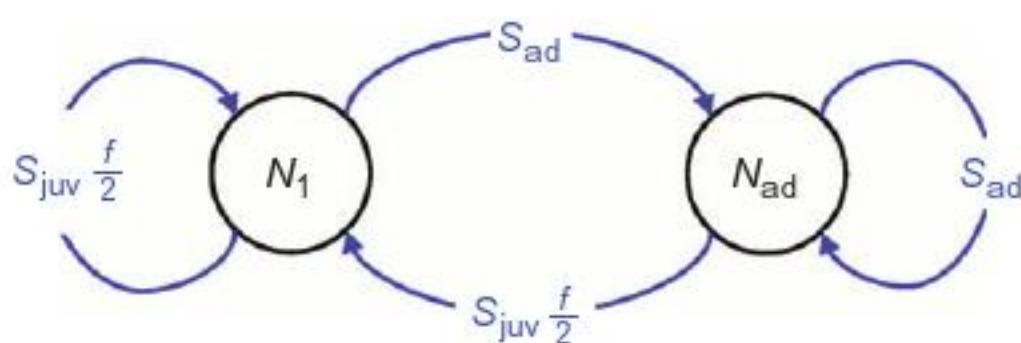


FIGURE 11.2 Life-cycle graph of an ortolan bunting population. The nodes show two age classes (N_1 : 1-year-old females; N_{ad} : females older than 1 year) and the arrows the transition probabilities based on the vital rates (S_{juv} : juvenile survival; S_{ad} : adult survival; f : productivity).

to reproduce at the age of 1 year and distinguish two age classes: 1-year-old individuals (1) and individuals older than one year (ad). Each year we survey the population immediately before the young are born, hence, we assume a prebreeding census. A life-cycle graph of the model is shown in Fig. 11.2. A mathematical description of a deterministic version of the model is

$$\begin{bmatrix} N_1 \\ N_{ad} \end{bmatrix}_{t+1} = \begin{bmatrix} S_{juv} \frac{f}{2} & S_{juv} \frac{f}{2} \\ S_{ad} & S_{ad} \end{bmatrix}_t \begin{bmatrix} N_1 \\ N_{ad} \end{bmatrix}_t,$$

where S_{juv} and S_{ad} are the annual survival probabilities of juvenile and adult females, respectively, f is the number of offspring produced per female, and N_1 and N_{ad} are the number of 1-year-old and adult females in the population. Fecundity is divided by 2, reflecting our assumption of an even sex ratio and because our model only keeps track of females. Care must be taken with the time indices of the demographic rates. When using a prebreeding census, as here, the projection matrix is parameterized with survival from year t to year $t + 1$ and with fecundity in year t . With a postbreeding census, the matrix would still contain survival from year t to $t + 1$ but fecundity in year $t + 1$ and also a different parameterization of the projection matrix.

Another way to write the above model is in terms of the expected numbers of 1-year-old and adult individuals, respectively:

$$E(N_{1,t+1} | N_{1,t}, N_{ad,t}) = N_{1,t} S_{juv,t} \frac{f_t}{2} + N_{ad,t} S_{ad,t} \frac{f_t}{2}$$

$$E(N_{ad,t+1} | N_{ad,t}) = N_{1,t} S_{ad,t} + N_{ad,t} S_{ad,t}$$

These are the expected numbers; now we want to write these equations in such a way that all relevant sources of uncertainty are included. One form of variability that is always present is demographic stochasticity. Demographic stochasticity is particularly important when population size is small (Lande, 2002). To include demographic stochasticity, we use

appropriate distributions to describe the number of individuals in year $t + 1$. To estimate the number of 1-year-old individuals, we must find a distribution that yields an integer value between 0 (if reproduction is 0 or all individuals die) and some big number (if reproduction is great and many survive) and whose expected value is $N_{1,t}S_{\text{juv},t}f_t/2 + N_{\text{ad},t}S_{\text{ad},t}f_t/2$. The Poisson distribution is an appropriate candidate:

$$N_{1,t+1} \sim \text{Poisson}\left(N_{1,t}S_{\text{juv},t}\frac{f_t}{2} + N_{\text{ad},t}S_{\text{ad},t}\frac{f_t}{2}\right).$$

For the adults, we must find a distribution that generates an integer value between 0 (if no individual survives) and $N_{\text{ad},t}$ (if all individuals survive) and whose expected value is $N_{1,t}S_{\text{ad},t} + N_{\text{ad},t}S_{\text{ad},t}$. Here, the binomial distribution is appropriate for modeling such bounded counts:

$$N_{\text{ad},t+1} \sim \text{Binomial}(N_{1,t} + N_{\text{ad},t}, S_{\text{ad},t}).$$

By relating the adult and juvenile population sizes between successive years among each other in a stochastic manner, we account for demographic stochasticity.

11.2.2 Second Step: Define the Likelihoods of Each Individual Data Set

Now we have defined the link between the population size and the demographic parameters. The next step is to write the likelihood of all available data sets. Here we have three different data sets, and we start with the likelihood of population counts.

Likelihood of the Population Count Data

A powerful way to model population counts is a state-space model (Chapter 5). The state process describes the true but unknown population trajectory under the model defined in step 1, and the observation process links the observed population counts to the true population sizes by allowing for observation error. Thus, the state-process model is defined by the two equations developed in Section 11.2.1:

$$N_{1,t+1} \sim \text{Poisson}\left(N_{1,t}S_{\text{juv},t}\frac{f_t}{2} + N_{\text{ad},t}S_{\text{ad},t}\frac{f_t}{2}\right)$$

$$N_{\text{ad},t+1} \sim \text{Binomial}(N_{1,t} + N_{\text{ad},t}, S_{\text{ad},t}).$$

The age-specific number of individuals present in the first study year must be estimated from the data (see below). There are different possibilities to write the observation model that differ in the assumed distribution of the

observation errors. Often, a normal error assumption is made; thus, the count data (y) are modeled as

$$\begin{aligned}y_t &= (N_{1,t} + N_{ad,t}) + \eta_t \\ \eta_t &\sim \text{Normal}(0, \sigma_y^2)\end{aligned}$$

where σ_y^2 is the observation error. Another possibility is to adopt a Poisson distribution for the observation error in population counts:

$$y_t \sim \text{Poisson}(N_{1,t} + N_{ad,t}).$$

A feature of the Poisson distribution is that the variance is equal to its mean, implying that the observation error increases with the population size. A third possibility is a log-normal distribution:

$$\begin{aligned}\log(y_t) &= \log(N_{1,t} + N_{ad,t}) + \eta_t \\ \eta_t &\sim \text{Normal}(0, \sigma_y^2).\end{aligned}$$

Modeling the log population counts ensures that the observation error increases with population size and allows for counts that are skewed on the arithmetic scale.

Some comments are in order on the observation model. First, in our experience, the choice of the observation model (i.e., between the three variants given above) often has no strong effect on the parameter estimates (but see Knape et al., 2011). Second, all the models can only adjust for random observation errors, that is, some sort of binomial sampling error as shown in Section 5.3. Any systematic patterns such as a trend (e.g., when observers become better over time) cannot be properly accounted for by this type of model. The general discussion about state-space models for inference about population size (see Chapter 5) also applies for this component of integrated models. Third, none of the above-mentioned observation models allows estimation of detection probability, and, consequently, the true population size remains unknown. So if the average detection probability of an individual is 0.9, then the estimated population size (N) will on average be 0.9 times the true population size. If we want to estimate detection probability, different survey protocols must be used (see Chapters 6, 10, 12, and 13). If detection probability is stationary (i.e., fluctuates around a constant mean), this caveat does not apply for conclusions about population dynamics and for the estimation of the demographic parameters. Fourth, the survey may be restricted to certain age classes or other segments of the population or it may not differentiate between age classes, and yet, we are usually still able to estimate age-specific population sizes (see also Link et al., 2003, for an impressive example of this, although not in the context of an integrated population model). Finally, it may be possible to get separate counts of different age classes, with the advantage of being able to

extract more detailed information from the counts (Tavecchia et al., 2009). The observation equation then needs a slight adaptation.

In what follows, we will use the normal approximation for the observation error. The likelihood of the state-space model is the product of the likelihood of the observation and the process equations:

$$L_{\text{SS}}(\mathbf{y} | \mathbf{N}, \mathbf{S}_{\text{juv}}, \mathbf{S}_{\text{ad}}, \mathbf{f}, \sigma_y^2) = L_O(\mathbf{y} | \mathbf{N}, \sigma_y^2) \times L_S(\mathbf{N} | \mathbf{S}_{\text{juv}}, \mathbf{S}_{\text{ad}}, \mathbf{f}).$$

The state-space likelihood already contains all parameters that we would like to estimate. So one might wonder why not just use the state-space model alone to estimate these parameters; that is, why aren't the count data enough? The reason is that the parameters in this model would not be identifiable, unless we impose strong constraints on the parameters (King et al., 2010), use informative prior distributions (Thomas et al., 2005; Newman et al., 2006; Buckland et al., 2007), or have counts from several stage/age classes (Link et al., 2003; David et al., 2010). Otherwise, the parameters would not be identifiable. However, if we add more (independent) information about some or all of the parameters, the model typically becomes identifiable. Therefore, we have to go on with the definition of likelihoods of other kinds of data.

Likelihood of the Capture–Recapture Data

For the capture–recapture data, we can use the likelihood of the CJS model introduced in Section 7.9, that is, the multinomial likelihood, $L_{\text{CJS}}(\mathbf{m} | \mathbf{S}_{\text{juv}}, \mathbf{S}_{\text{ad}}, \mathbf{p})$. Whenever possible, we recommend the multinomial likelihood to estimate survival, because of the resulting computational benefits compared with the state-space likelihood. It requires that the capture–recapture data are summarized in the m-array format (\mathbf{m}).

Likelihood of Reproductive Success Data

We use a simple Poisson regression to model productivity (see Chapters 3 and 4). We assume that the total number of nestlings counted in year t , (J_t), follows a Poisson distribution with parameters that are the product of the number of surveyed broods (R_t) and productivity (f_t), hence, $J_t \sim \text{Poisson}(R_t f_t)$. Thus, the likelihood is $L_P(\mathbf{J}, \mathbf{R} | \mathbf{f})$.

11.2.3 Third Step: Formulate the Joint Likelihood

The last step is to formulate the likelihood of the complete model. This is in fact simple, if we can assume independence among the component data sets of the integrated analysis. Then, the joint likelihood is just the product of the component likelihoods:

$$\begin{aligned} L_{\text{IPM}}(\mathbf{y}, \mathbf{m}, \mathbf{J}, \mathbf{R} | \mathbf{N}, \mathbf{S}_{\text{juv}}, \mathbf{S}_{\text{ad}}, \mathbf{f}, \mathbf{p}, \sigma_y^2) &= L_O(\mathbf{y} | \mathbf{N}, \sigma_y^2) \times L_S(\mathbf{N} | \mathbf{S}_{\text{juv}}, \mathbf{S}_{\text{ad}}, \mathbf{f}) \\ &\quad \times L_{\text{CJS}}(\mathbf{m} | \mathbf{S}_{\text{juv}}, \mathbf{S}_{\text{ad}}, \mathbf{p}) \times L_P(\mathbf{J}, \mathbf{R} | \mathbf{f}) \end{aligned}$$

A graph of this model is shown in Fig. 11.3a. The independence of the data sets is a crucial assumption. Under one—restrictive—view, this would mean that no animal in the population count data must occur in the capture–recapture data or in the productivity data. This could be achieved if different (sub)populations are sampled: in one population we collect population count data, in another the capture–recapture data, and in the third productivity data. Alternatively, we could sample a single very large population (e.g., living in a whole country) where it is unlikely that the same individual appears in more than one data set. Although the independence assumption may then hold, we would need a new assumption that the dynamics as well as the link between population size and demography is identical in all sampled (sub)populations or across the large spatial scale. This may well be an unrealistic assumption too.

In practice, we often have different kinds of demographic data from a single and rather small population. After all, it is for small populations with limited data where the combination of multiple data sets is particularly fruitful in terms of the increased precision of parameter estimates (Schaub et al., 2007). Therefore, it is important to understand whether the violation of the independence assumption has a strong impact on the parameter estimates. Abadi et al. (2010a) have recently studied this issue. In principle, if the data sets are not independent, we would not expect biased parameter estimates, but spuriously high precision of the estimates. This is a classical result in statistics when nonindependent data are analyzed as if they were independent: the information from one individual is used multiple times, and hence, the genuine sample size is not as large as it seems. For example, if an individual is included in the population count data, it contributes to the estimation of survival in the state-space likelihood. The same individual might be included in the capture–recapture data and thus again provide information about survival via the multinomial likelihood of the CJS model. The result of this is a kind of overdispersion due to lack of independence.

Abadi et al. (2010a) simulated data types similar to the ones we use here with different degrees of nonindependence. They found that violation of the independence assumption had almost no effect on the accuracy of the parameter estimates. In contrast, Besbeas et al. (2009) found that the violation of the independence assumption had an effect when population count and mark-recovery data were combined. These divergent conclusions likely stem from the fact that different data sets contribute differently to the joint likelihood. In fact, in these classical applications, violation of independence can only occur between the state-space likelihood and another likelihood because in the state-space likelihood all demographic parameters are included. Generally, the amount of information about demographic parameters in the state-space likelihood is small. If it is combined with a likelihood that contains plenty of information about

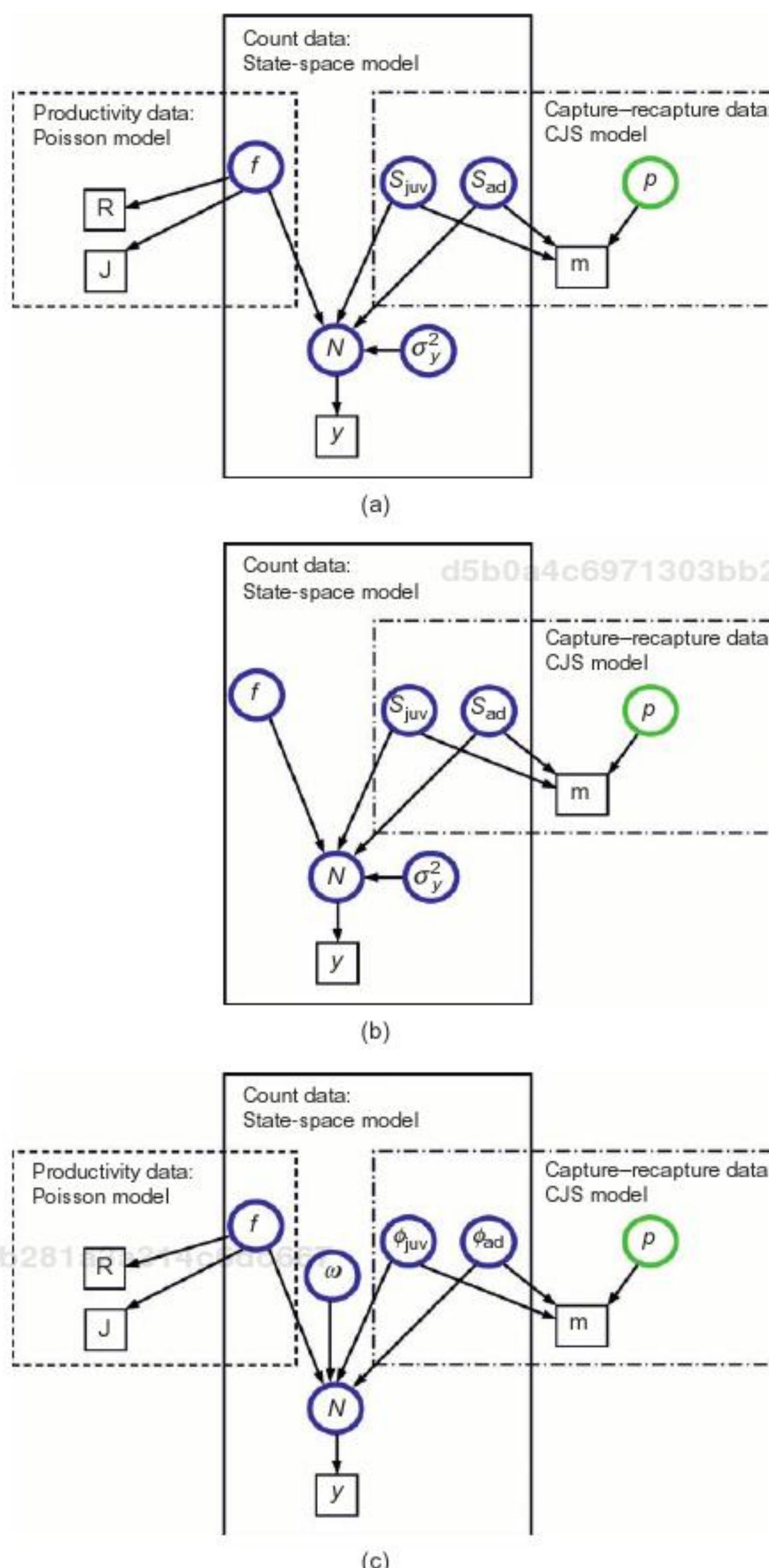


FIGURE 11.3 Graphical representation of different integrated population models. This graph is similar to an acyclic directed graph (DAG) without the priors. Small squares represent the data, circles the parameters (blue: target parameters, green: nuisance parameters), large squares the individual submodels, and arrows the flux of information. Circles appearing in two submodels indicate that they are informed from two data sources. (a) Model of Section 11.3; (b) same model as (a), but without the availability of productivity data (Section 11.4); (c) model of Section 11.6. For the notation of the parameters and data see text.

a demographic parameter, as is the case with the capture–recapture data and survival, the joint likelihood in terms of survival is dominated by the capture–recapture likelihood. Consequently, nonindependence plays a minor role. In contrast, if the state-space likelihood is combined with a likelihood that contains little information about a demographic parameter, as is the case with the mark-recovery likelihood and survival, the joint likelihood with respect to survival contains similar amounts of information from both the population count and the mark-recovery data. In that case, violation of the independence assumption can have a more serious effect on parameter accuracy. Thus, whether the violation of the independence assumption has an effect on parameter accuracy needs to be evaluated for each model separately.

11.3 EXAMPLE OF A SIMPLE IPM (COUNTS, CAPTURE–RECAPTURE, REPRODUCTION)

11.3.1 Load Data

We will load one data set from the ortolan bunting population that was simulated using the code in Web appendix 2.

```
# Population counts (from years 1 to 10)
y <- c(45, 48, 44, 59, 62, 62, 55, 51, 46, 42)

# Capture-recapture data (in m-array format, from years 1 to 10)
m <- matrix(c(11, 0, 0, 0, 0, 0, 0, 0, 0, 70,
             0, 12, 0, 1, 0, 0, 0, 0, 0, 52,
             0, 0, 15, 5, 1, 0, 0, 0, 0, 42,
             0, 0, 0, 8, 3, 0, 0, 0, 0, 51,
             0, 0, 0, 0, 4, 3, 0, 0, 0, 61,
             0, 0, 0, 0, 0, 12, 2, 3, 0, 66,
             0, 0, 0, 0, 0, 0, 16, 5, 0, 44,
             0, 0, 0, 0, 0, 0, 0, 12, 0, 46,
             0, 0, 0, 0, 0, 0, 0, 0, 11, 71,
             10, 2, 0, 0, 0, 0, 0, 0, 0, 13,
             0, 7, 0, 1, 0, 0, 0, 0, 0, 27,
             0, 0, 13, 2, 1, 1, 0, 0, 0, 14,
             0, 0, 0, 12, 2, 0, 0, 0, 0, 20,
             0, 0, 0, 0, 10, 2, 0, 0, 0, 21,
             0, 0, 0, 0, 0, 11, 2, 1, 1, 14,
             0, 0, 0, 0, 0, 0, 12, 0, 0, 18,
             0, 0, 0, 0, 0, 0, 0, 11, 1, 21,
             0, 0, 0, 0, 0, 0, 0, 0, 10, 26), ncol = 10,
             byrow = TRUE)
```

The last column in matrix **m** contains the number of released individuals that are never recaptured. The top half of the array contains the data on birds marked as juveniles and the bottom half on those marked as adults.

```
# Productivity data (from years 1 to 9)
J <- c(64, 132, 86, 154, 156, 134, 116, 106, 110)
R <- c(21, 28, 26, 38, 35, 33, 31, 30, 33)
```

Vector **J** contains the total number of nestlings recorded, and vector **R** is the annual number of surveyed broods. The numbers recorded in the last year are not considered here because they are not needed in the population model.

11.3.2 Analysis of the Model

We find it useful to highlight three sections in the BUGS code for analyzing the integrated population model. First, we define the priors of the unknowns (parameters, latent effects). This includes the latent population size of each class in the first year, the demographic parameters, and the observation error. As always in this book, we can specify vague priors with little prior information if we wish. We must ensure that the priors have support only for values that are possible at all (e.g., positive numbers for the initial population sizes). In our model, all demographic rates are constant over time; hence, we define priors for their means. When using vague priors for the initial population sizes, BUGS may not even start to update or, if it does, the chains may struggle to converge. Thus, with integrated population models, it is often advisable to use slightly more informative priors. King et al. (2010) suggest normal priors centered on the observed count in the first year and with a variance that is equal to the observation error.

Second, we may compute various derived parameters. As we have seen many times, one of the big assets of an MCMC-based Bayesian analysis is the ease with which derived quantities can be computed along with a full assessment of their uncertainty. In our example, we are interested in the population growth rate.

Third, our code contains likelihoods of the different data sets whose modeling we integrate in our analysis. These likelihoods are more or less identical to the BUGS code given for the different data sets so far, for example, in Chapters 5 and 7. You may ask yourself the following question: where is that ugly joint likelihood that we saw in Section 11.2.3? Does it lurk in some terrifying BUGS code that we have yet to disclose? No, what you see is all that is necessary to define the joint likelihood of the model in the BUGS language. To us, as ecologists, this is another great asset of Bayesian population analyses using BUGS: instead of defining huge likelihoods at once, you decompose them and describe them by defining the quantities in the model and their stochastic and deterministic local relationships. The joint likelihood is then defined

implicitly by using the same names for some parameters that co-occur in different component likelihoods. We believe that a good grasp of reasonably complex statistical models such as an IPM is much more within the reach of most ecologists than it is when you want to use the method of maximum likelihood to fit the same model. Thus, again, BUGS frees the modeler in you.

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

# -----
# Integrated population model
# - Age structured model with 2 age classes:
#     1-year old and adults (at least 2 years old)
# - Age at first breeding = 1 year
# - Prebreeding census, female-based
# - All vital rates assumed to be constant
# -----


# -----
# 1. Define the priors for the parameters
# -----


# Observation error
tauy <- pow(sigma.y, -2)
sigma.y ~ dunif(0, 50)
sigma2.y <- pow(sigma.y, 2)

# Initial population sizes
N1[1] ~ dnorm(100, 0.0001) I(0,)      # 1-year
Nad[1] ~ dnorm(100, 0.0001) I(0,)      # Adults

# Survival and recapture probabilities, as well as productivity
for (t in 1:(nyears-1)){
  sjuv[t] <- mean.sjuv
  sad[t] <- mean.sad
  p[t] <- mean.p
  f[t] <- mean.fec
}

mean.sjuv ~ dunif(0, 1)
mean.sad ~ dunif(0, 1)
mean.p ~ dunif(0, 1)
mean.fec ~ dunif(0, 20)

# -----


# 2. Derived parameters
# -----


# Population growth rate
for (t in 1:(nyears-1)){
  lambda[t] <- Ntot[t+1] / Ntot[t]
}
```

```
# -----
# 3. The likelihoods of the single data sets
# -----
# 3.1. Likelihood for population population count data (state-space
# model)
# 3.1.1 System process
for (t in 2:nyears){
  mean1[t] <- f[t-1] / 2 * sjuv[t-1] * Ntot[t-1]
  N1[t] ~ dpois(mean1[t])
  Nad[t] ~ dbin(sad[t-1], Ntot[t-1])
}
for (t in 1:nyears){
  Ntot[t] <- Nad[t] + N1[t]
}

# 3.1.2 Observation process
for (t in 1:nyears){
  y[t] ~ dnorm(Ntot[t], tauy)
}

# 3.2 Likelihood for capture-recapture data: CJS model (2 age classes)
# Multinomial likelihood
for (t in 1:2*(nyears-1)){
  m[t,1:nyears] ~ dmulti(pr[t,], r[t])
}

# Calculate the number of released individuals
for (t in 1:2*(nyears-1)){
  r[t] <- sum(m[t,])
}

# m-array cell probabilities for juveniles
for (t in 1:(nyears-1)){
  # Main diagonal
  q[t] <- 1-p[t]
  pr[t,t] <- sjuv[t] * p[t]
  # Above main diagonal
  for (j in (t+1):(nyears-1)){
    pr[t,j] <- sjuv[t]*prod(sad[(t+1):j])*prod(q[t:(j-1)])*p[j]
  } #j
  # Below main diagonal
  for (j in 1:(t-1)){
    pr[t,j] <- 0
  } #j
  # Last column: probability of non-recapture
  pr[t,nyears] <- 1-sum(pr[t,1:(nyears-1)])
} #t

# m-array cell probabilities for adults
for (t in 1:(nyears-1)){
  # Main diagonal
  pr[t+nyears-1,t] <- sad[t] * p[t]
  # Above main diagonal
  for (j in (t+1):(nyears-1)) {
```

d5b0a4c6971303bb281a3a314c6dc667
ebrary

```

pr[t+nyears-1,j] <- prod(sad[t:j])*prod(q[t:(j-1)])*p[j]
} #j
# Below main diagonal
for (j in 1:(t-1)){
  pr[t+nyears-1,j] <- 0
} #j
# Last column
pr[t+nyears-1,nyears] <- 1 - sum(pr[t+nyears-1,1:(nyears-1)])
} #t

# 3.3. Likelihood for productivity data: Poisson regression
for (t in 1:(nyears-1)){
  J[t] ~ dpois(rho[t])
  rho[t] <- R[t]*f[t]
}
}

", fill = TRUE)
sink()

# Bundle data
bugs.data <- list(m = m, y = y, J = J, R = R, nyyears = dim(m) [2])

# Initial values
inits <- function(){list(mean.sjuv = runif(1, 0, 1), mean.sad = runif(1, 0, 1), mean.p = runif(1, 0, 1), mean.fec = runif(1, 0, 10), N1 = rpois(dim(m) [2], 30), Nad = rpois(dim(m) [2], 30), sigma.y = runif(1, 0, 10))}

# Parameters monitored
parameters <- c("mean.sjuv", "mean.sad", "mean.p", "mean.fec", "N1", "Nad", "Ntot", "lambda", "sigma2.y")

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

# Call WinBUGS from R (BRT 2 min)
ipm <- bugs(bugs.data, inits, parameters, "ipm.bug", n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory = bugs.dir, working.directory = getwd())

```

The chains converge fairly quickly. Here are the posterior summaries:

```

print(ipm, digits = 3)
      mean     sd   2.5%   25%   50%   75%  97.5%   Rhat   n.eff
mean.sjuv  0.257  0.018  0.223  0.245  0.257  0.269  0.294  1.001  7500
mean.sad   0.519  0.026  0.468  0.501  0.519  0.536  0.570  1.001  7500
mean.p     0.619  0.037  0.548  0.594  0.618  0.644  0.691  1.001  7500
mean.fec   3.829  0.115  3.603  3.752  3.826  3.907  4.055  1.003  1000
N1[1]      22.757 13.264  1.127 11.440 22.600 34.052 45.310  1.009   320
[ ... ]
N1[10]    20.237  3.563 13.630 17.880 20.050 22.490 27.565  1.001   4900
Nad[1]     22.955 13.209  1.359 11.330 23.050 34.052 45.180  1.005   540
[ ... ]

```

Nad[10]	22.831	3.230	17.000	21.000	23.000	25.000	29.000	1.002	1400
Ntot[1]	45.711	3.101	39.990	44.260	45.220	46.870	53.137	1.001	5100
[...]									
Ntot[10]	43.068	3.240	37.540	41.480	42.400	44.310	51.241	1.001	3000
lambda[1]	1.047	0.071	0.893	1.008	1.055	1.082	1.192	1.001	7500
[...]									
lambda[9]	0.930	0.068	0.804	0.893	0.920	0.960	1.096	1.001	7500
sigma2.y	14.580	28.382	0.016	1.550	6.087	16.480	79.205	1.010	740

Estimated population sizes are quite close to the counts in our example (Fig. 11.4). As expected, the estimates are less variable than the counts, illustrating the smoothing that results from separating out state and observation processes, the autoregressive nature of the population model, and adding more information (from the other data sets). We could also plot the age-specific population sizes, but in our study example this is not very interesting since both age classes have about the same size. This is because the stable age distribution (obtained as the right eigenvector of the Leslie matrix) in this population is 50% 1-year-old individuals and 50% adults. Yet, generally, this is an advantage of integrated models: one may obtain estimates of the size of population segments (such as age classes) that are never even observed.

```
# Produce Fig. 11.4
par(cex = 1.2)
lower <- upper <- numeric()
for (i in 1:10) {
  lower[i] <- quantile(ipm$sims.list$Ntot[,i], 0.025)
  upper[i] <- quantile(ipm$sims.list$Ntot[,i], 0.975)
}
```

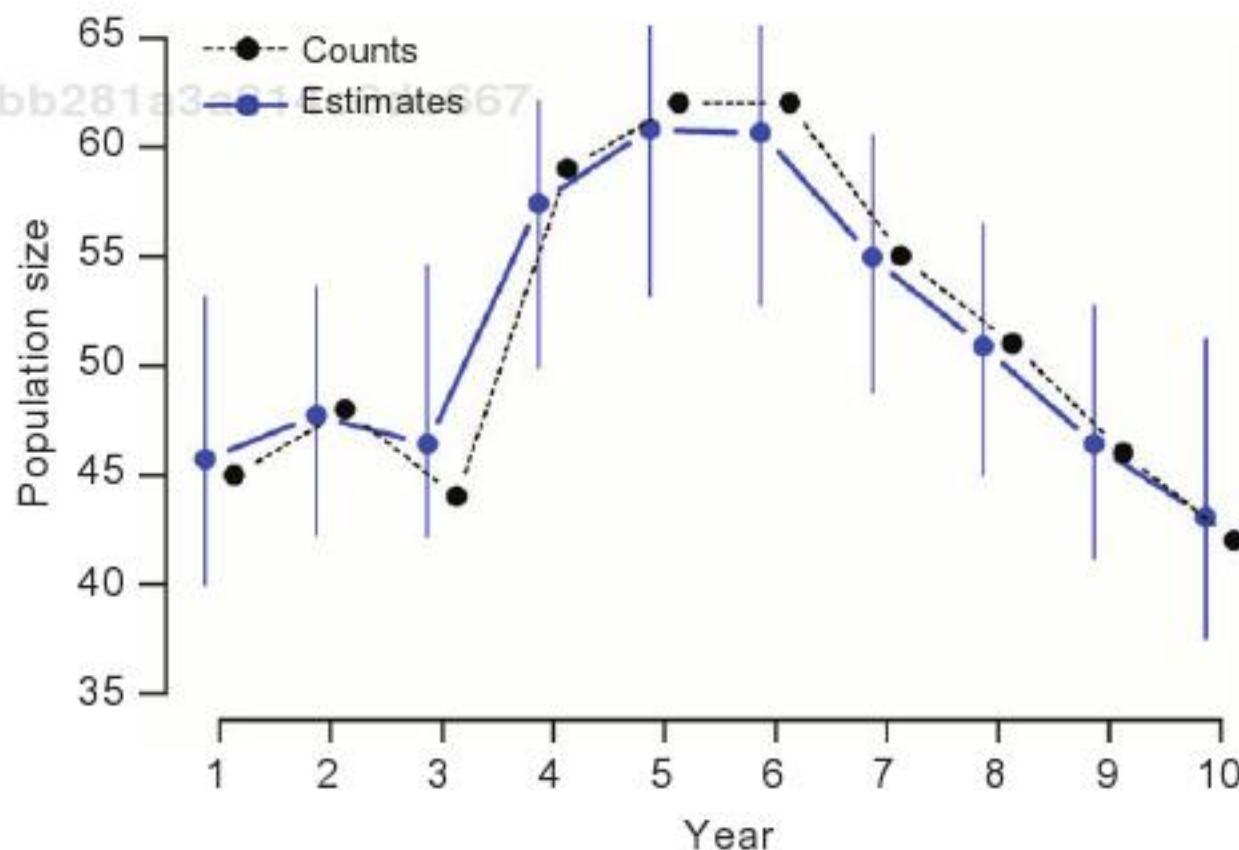


FIGURE 11.4 Observed (black) and estimated population sizes (blue) with 95% CRI under an integrated population model for the simulated ortolan bunting data set.

```
plot(ipm$mean$Ntot, type = "b", ylim = c(35, 65), ylab = "Population size",
      xlab = "Year", las = 1, pch = 16, col = "blue", frame = F, cex = 1.5)
segments(1:10, lower, 1:10, upper, col = "blue")
points(y, type = "b", col = "black", pch = 16, lty = 2, cex = 1.5)
legend(x = 1, y = 65, legend = c("Counts", "Estimates"), pch = c(16, 16),
       col = c("black", "blue"), lty = c(2, 1), bty = "n")
```

11.4 ANOTHER EXAMPLE OF AN IPM: ESTIMATING PRODUCTIVITY WITHOUT EXPLICIT PRODUCTIVITY DATA

One very important advantage of integrated population models is that they typically allow one to estimate demographic parameters for which no explicit data are available (e.g., Besbeas et al., 2002; Schaub et al., 2007). This is possible because the population count data contain information about all demographic parameters, and this information is exploited with the integrated population model. Here, we aim to estimate productivity (f) in a study in which no explicit data on productivity are available. We focus on the same ortolan bunting population as in the previous section, but this time only use the population count data and the capture–recapture data (Fig. 11.3b), and we adopt the same model structure as before (e.g., constant demographic rates). Essentially, the BUGS code requires one simple change: just remove the definition of the likelihood of the productivity data.

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

# -----
# Integrated population model
# - Age structured model with 2 age classes:
#     1-year old and adults (at least 2 years old)
# - Age at first breeding = 1 year
# - Prebreeding census, female-based
# - All vital rates assumed to be constant
# -----
#
# -----
# 1. Define the priors for the parameters
# -----
# Observation error
tauy <- pow(sigma.y, -2)
sigma.y ~ dunif(0, 50)
sigma2.y <- pow(sigma.y, 2)

# Initial population sizes
N1[1] ~ dnorm(100, 0.0001) I(0,)          # 1-year
Nad[1] ~ dnorm(100, 0.0001) I(0,)          # Adults
```

```
# Survival and recapture probabilities, as well as productivity
for (t in 1:(nyears-1)){
  sjuv[t] <- mean.sjuv
  sad[t] <- mean.sad
  p[t] <- mean.p
  f[t] <- mean.fec
}

mean.sjuv ~ dunif(0, 1)
mean.sad ~ dunif(0, 1)
mean.p ~ dunif(0, 1)
mean.fec ~ dunif(0, 20)

# -----
# 2. Derived parameters
# -----
# Population growth rate
for (t in 1:(nyears-1)){
  lambda[t] <- Ntot[t+1] / Ntot[t]
}

# -----
# 3. The likelihoods of the single data sets
# -----
# 3.1. Likelihood for population population count data (state-space
model)
# 3.1.1 System process
for (t in 2:nyears){
  mean1[t] <- f[t-1] / 2 * sjuv[t-1] * Ntot[t-1]
  N1[t] ~ dpois(mean1[t])
  Nad[t] ~ dbin(sad[t-1], Ntot[t-1])
}
for (t in 1:nyears){
  Ntot[t] <- Nad[t] + N1[t]
}

# 3.1.2 Observation process
for (t in 1:nyears){
  y[t] ~ dnorm(Ntot[t], tauy)
}

# 3.2 Likelihood for capture-recapture data: CJS model (2 age classes)
# Multinomial likelihood
for (t in 1:2*(nyears-1)){
  m[t,1:nyears] ~ dmulti(pr[t,], r[t])
}

# Calculate the number of released individuals
for (t in 1:2*(nyears-1)){
  r[t] <- sum(m[t,])
}

# m-array cell probabilities for juveniles
for (t in 1:(nyears-1)){
  # Main diagonal
  q[t] <- 1-p[t]
```

```
pr[t,t] <- sjuv[t] * p[t]
# Above main diagonal
for (j in (t+1):(nyears-1)){
  pr[t,j] <- sjuv[t]*prod(sad[(t+1):j])*prod(q[t:(j-1)])*p[j]
} #j
# Below main diagonal
for (j in 1:(t-1)){
  pr[t,j] <- 0
} #j
# Last column: probability of non-recapture
pr[t,nyears] <- 1-sum(pr[t,1:(nyears-1)])
} #t

# m-array cell probabilities for adults
for (t in 1:(nyears-1)){
  # Main diagonal
  pr[t+nyears-1,t] <- sad[t] * p[t]
  # Above main diagonal
  for (j in (t+1):(nyears-1)){
    pr[t+nyears-1,j] <- prod(sad[t:j])*prod(q[t:(j-1)])*p[j]
  } #j
  # Below main diagonal
  for (j in 1:(t-1)){
    pr[t+nyears-1,j] <- 0
  } #j
  # Last column
  pr[t+nyears-1,nyears] <- 1 - sum(pr[t+nyears-1,1:(nyears-1)])
} #t
}
",fill = TRUE)
sink()

# Bundle data
bugs.data <- list(m = m, y = y, nyyears = dim(m) [2])

# Initial values
d5b0a4c6971303bb281a3a314c6dc667
ebrary
inits<-function(){list(mean.sjuv = runif(1, 0, 1), mean.sad = runif(1,
0, 1), mean.p = runif(1, 0, 1), mean.fec = runif(1, 0, 10), N1 = rpois
(dim(m) [2], 30), Nad = rpois(dim(m) [2], 30), sigma.y = runif(1, 0,
10))}

# Parameters monitored
parameters <- c("mean.sjuv", "mean.sad", "mean.p", "mean.fec", "N1",
"Nad", "Ntot", "lambda", "sigma2.y")

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

# Call WinBUGS from R (BRT 1 min)
ipm.prod <- bugs(bugs.data, inits, parameters, "ipm-prod.bug",
n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
bugs.directory = bugs.dir, working.directory = getwd())
```

```
# Summarize posteriors
print(ipm.prod, digits = 3)
      mean     sd   2.5%   25%   50%   75% 97.5% Rhat n.eff
mean.sjuv 0.266 0.024 0.222 0.250 0.265 0.282 0.317 1.001 7500
mean.sad  0.527 0.029 0.471 0.507 0.527 0.546 0.584 1.002 1900
mean.p    0.611 0.040 0.531 0.584 0.610 0.639 0.689 1.001 3500
mean.fec  3.547 0.487 2.673 3.209 3.515 3.858 4.557 1.001 5400
N1[1]     23.363 13.252 1.184 12.557 23.275 34.402 45.535 1.014 170
[...]
N1[10]    19.499 3.631 12.699 17.090 19.400 21.750 26.870 1.004 680
Nad[1]    22.561 13.274 1.233 11.390 22.590 33.230 45.305 1.022 120
[...]
Nad[10]   23.398 3.313 17.000 21.000 23.000 26.000 30.000 1.002 1500
Ntot[1]   45.925 3.244 40.600 44.450 45.260 46.960 53.905 1.001 7500
[...]
Ntot[10]  42.898 3.130 37.519 41.470 42.290 43.940 50.430 1.002 1600
lambda[1] 1.044 0.069 0.891 1.007 1.054 1.078 1.187 1.001 7500
[...]
lambda[9] 0.928 0.066 0.804 0.894 0.917 0.957 1.091 1.001 7100
sigma2.y 14.000 31.952 0.003 1.268 5.318 15.167 78.404 1.025 210
```

Most parameter estimates are nearly identical in both models. Overall, the precision of the estimates is slightly reduced now since we have ignored part of the available information by not using the productivity data. The most striking difference occurs for the productivity parameter. Under the model without productivity data, the estimate is slightly lower (3.547 vs. 3.829) and much less precise ($sd = 0.487$ vs. 0.115). A simulation study shows that the productivity estimator from both models is unbiased, but if no productivity data are available the precision is understandably lower (Abadi et al., 2010a).

11.5 IPMs FOR POPULATION VIABILITY ANALYSIS

One particularly useful feature of integrated population models is that predictions of the size of “unobserved populations” (e.g., age classes) can be made. We have already mentioned that we may estimate the size of unobserved segments of a population. However, we may just as well estimate the size of (as yet) unobserved populations in the future. In particular, Bayesian population analysis using posterior sampling is strikingly useful for a full assessment of all uncertainties involved in such forecasts. To project the model into the future and forecast population size, we extend the loops of the state process and adapt the prior definitions of the demographic rates to cover the additional years. Within the MCMC samples, the future population sizes are estimated

with full accounting for all uncertainty in the parameter estimates. The posterior distributions of the predicted future population sizes can be directly used to compute population extinction probabilities or population prediction intervals, both typical inferential targets in population viability analyses (Beissinger, 2002). We illustrate predictions with the ortolan bunting example.

```
# Specify model in BUGS language
sink("ipm-pred.bug")
cat(
model {
# -----
# Integrated population model
# - Age structured model with 2 age classes:
#   1-year old and adults (at least 2 years old)
# - Age at first breeding = 1 year
# - Prebreeding census, female-based
# - All vital rates assumed to be constant
# -----
#
# -----
# 1. Define the priors for the parameters
# -----
# Observation error
tauy <- pow(sigma.y, -2)
sigma.y ~ dunif(0, 50)
sigma2.y <- pow(sigma.y, 2)

# Initial population sizes
N1[1] ~ dnorm(100, 0.0001) I(0,)      # 1-year
Nad[1] ~ dnorm(100, 0.0001) I(0,)      # Adults

# Survival and recapture probabilities, as well as productivity
for (t in 1:(nyears-1+t.pred)) {
  sjuv[t] <- mean.sjuv
  sad[t] <- mean.sad
  p[t] <- mean.p
  f[t] <- mean.fec
}

mean.sjuv ~ dunif(0, 1)
mean.sad ~ dunif(0, 1)
mean.p ~ dunif(0, 1)
mean.fec ~ dunif(0, 20)

# -----
# 2. Derived parameters
# -----
# Population growth rate
for (t in 1:(nyears-1+t.pred)) {
  lambda[t] <- Ntot[t+1] / Ntot[t]
}
```

```
# -----
# 3. The likelihoods of the single data sets
# -----
# 3.1. Likelihood for population population count data (state-space
# model)
# 3.1.1 System process
for (t in 2:nyears+t.pred){
  mean1[t] <- f[t-1] / 2 * sjuv[t-1] * Ntot[t-1]
  N1[t] ~ dpois(mean1[t])
  Nad[t] ~ dbin(sad[t-1], Ntot[t-1])
}
for (t in 1:nyears+t.pred){
  Ntot[t] <- Nad[t] + N1[t]
}

# 3.1.2 Observation process
for (t in 1:nyears){
  y[t] ~ dnorm(Ntot[t], tauy)
}

# 3.2 Likelihood for capture-recapture data: CJS model (2 age classes)
# Multinomial likelihood
for (t in 1:2*(nyears-1)){
  m[t,1:nyears] ~ dmulti(pr[t,], r[t])
}

# Calculate the number of released individuals
for (t in 1:2*(nyears-1)){
  r[t] <- sum(m[t,])
}

# m-array cell probabilities for juveniles
for (t in 1:(nyears-1)){
  # Main diagonal
  q[t] <- 1-p[t]
  pr[t,t] <- sjuv[t] * p[t]
  # Above main diagonal
  for (j in (t+1):(nyears-1)){
    pr[t,j] <- sjuv[t]*prod(sad[(t+1):j])*prod(q[t:(j-1)])*p[j]
  } #j
  # Below main diagonal
  for (j in 1:(t-1)){
    pr[t,j] <- 0
  } #j
  # Last column: probability of non-recapture
  pr[t,nyears] <- 1-sum(pr[t,1:(nyears-1)])
} #t

# m-array cell probabilities for adults
for (t in 1:(nyears-1)){
  # Main diagonal
  pr[t+nyears-1,t] <- sad[t] * p[t]
  # Above main diagonal
  for (j in (t+1):(nyears-1)){
    pr[t+nyears-1,j] <- prod(sad[t:j])*prod(q[t:(j-1)])*p[j]
  } #j}
```

d5b0a4c6971303bb281a3a314c6dc667
ebrary

```
# Below main diagonal
for (j in 1:(t-1)){
  pr[t+nyears-1,j] <- 0
} #j
# Last column
pr[t+nyears-1,nyears] <- 1 - sum(pr[t+nyears-1,1:(nyears-1)])
} #t

# 3.3. Likelihood for productivity data: Poisson regression
for (t in 1:(nyears-1)){
  J[t] ~ dpois(rho[t])
  rho[t] <- R[t]*f[t]
}
", fill = TRUE)
sink()

# Give the number of future years for which population size shall be estimated
t.pred <- 5

# Bundle data
bugs.data <- list(m = m, y = y, J = J, R = R, nyears = dim(m) [2],
  t.pred = t.pred)

# Initial values
inits <- function(){list(mean.sjuv = runif(1, 0, 1), mean.sad = runif(1, 0, 1),
  mean.p = runif(1, 0, 1), mean.fec = runif(1, 0, 10), N1 = rpois(dim(m) [2]+ t.pred, 30),
  Nad = rpois(dim(m) [2]+ t.pred, 30), sigma.y = runif(1, 0, 10))}

# Parameters monitored
parameters <- c("mean.sjuv", "mean.sad", "mean.p", "mean.fec", "N1",
  "Nad", "Ntot", "lambda", "sigma2.y")

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

# Call WinBUGS from R (BRT 1 min)
ipm.pred <- bugs(bugs.data, inits, parameters, "ipm-pred.bug",
  n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
  bugs.directory = bugs.dir, working.directory = getwd())
```

The estimated population sizes for the entire 15 years are shown in Fig. 11.5. We notice that 95% CRIs increase over time after year 10, reflecting an increased uncertainty when projecting further ahead. This makes intuitive sense.

```
# Produce Fig. 11-5
par(cex = 1.2)
lower <- upper <- numeric()
for (i in 1:15){
```

d5b0a4c6971303bb281a3a314c6dc667
ebrary

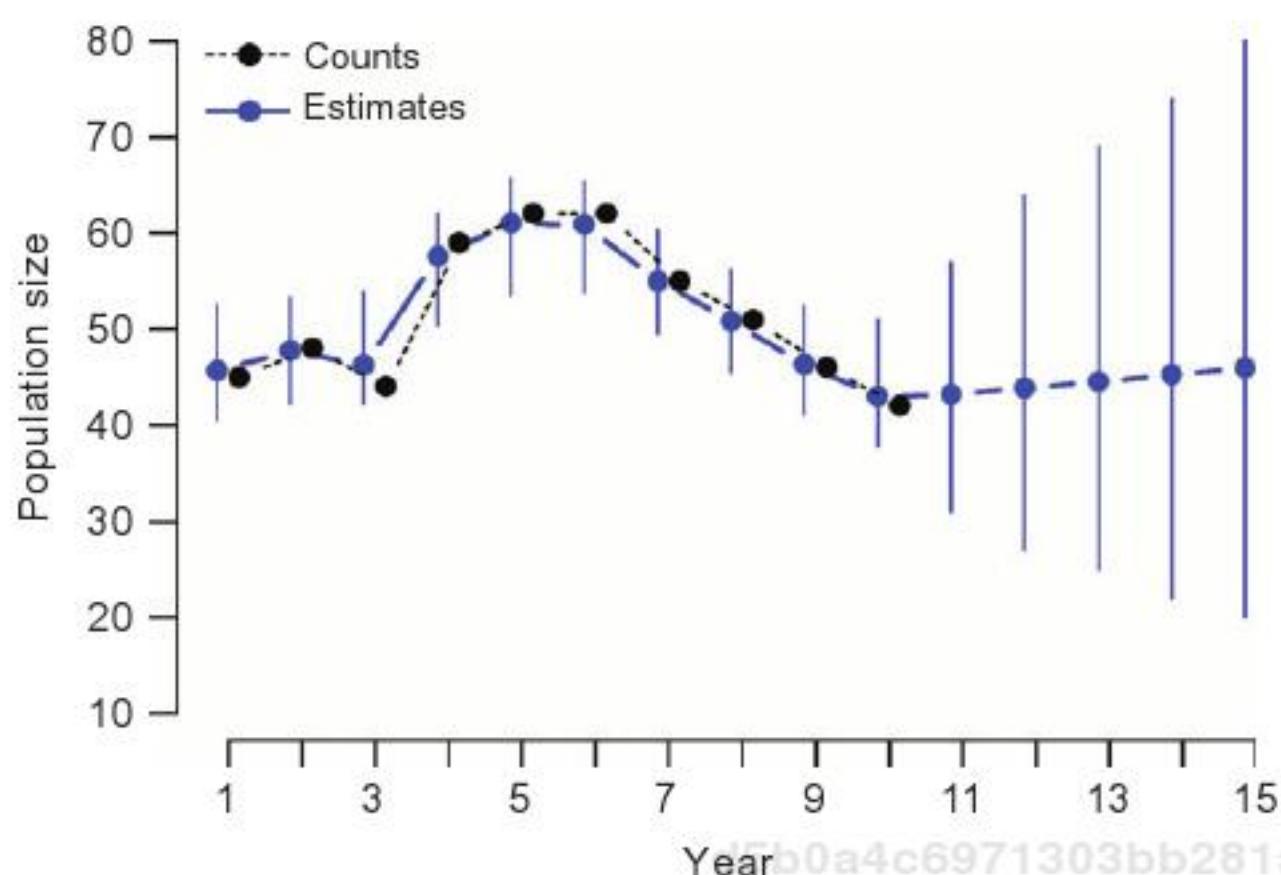


FIGURE 11.5 Observed (black) and estimated population sizes (blue) along with the 95% CRI. For years 11–15, projected population is shown.

```

lower[i] <- quantile(ipm.pred$sims.list$Ntot[,i], 0.025)
upper[i] <- quantile(ipm.pred$sims.list$Ntot[,i], 0.975)
}
plot(ipm.pred$mean$Ntot, type = "b", ylim = c(10, max(upper)), ylab =
  "Population size", xlab = "Year", las = 1, pch = 16, col = "blue",
  frame = F)
segments(1:15, lower, 1:15, upper, col = "blue")
points(y, type = "b", col = "black", lty = 2, pch = 16)
legend(x = 1, y = 80, legend = c("Counts", "Estimates"), pch = c(16, 16),
  col = c("black", "blue"), lty = c(2, 1), bty = "n")

```

The estimation of extinction probabilities is straightforward. In the example, we might be interested to know the probability that population size falls below 30 pairs in 5 years from now. Our model provides the answer to that based on the assumption that the demographic processes modeled remain the same as observed in the past. We only need to evaluate the posterior distribution of population size in year 15:

```
mean(ipm.pred$sims.list$Ntot[,15]<30)
[1] 0.140
```

Forecasting population size is slightly more complicated if the demographic parameters are not constant but vary over time. The best way to achieve this goal is to estimate a mean and a temporal variance of the demographic parameters from the observed data (see Section 11.6), that is, fit random year effects. Future realizations of the demographic rates can then be sampled from these distributions. When specifying random year effects in WinBUGS, process variability is automatically accounted for.

11.6 REAL DATA EXAMPLE: HOOPOE POPULATION DYNAMICS

We look at a real example using data types that are typically collected in demographic population studies. We studied a local population of hoopoes (Fig. 11.6) in the SW Swiss Alps (Valais; Arlettaz et al. 2010). In an area of 62 km², the vast majority of hoopoes use nest boxes for reproduction. From 2002 to 2009, these nest boxes were repeatedly checked every year to record the breeding success, and all the nestlings were ringed. In addition, adults were marked and recaptured. Thus, we obtained the following three types of demographic data: (1) a population count (the maximal number of simultaneously occupied nest boxes in each year), (2) capture–recapture data of adults and young, and (3) data on reproductive output. The goal of our study was to estimate the demographic parameters as well as their temporal variability to understand the dynamics of the hoopoe population (Schaub et al., 2011).

Although the study area is fairly large, the population is not geographically closed. Therefore, emigration and immigration need to be considered. We have already seen in Chapter 7 that capture–recapture data allow estimation of apparent survival probabilities, that is, the product of true survival and site fidelity. By modeling apparent survival in a CJS model as a part of



FIGURE 11.6 Happy hoopoes (*Upupa epops*), Switzerland (Photograph by P. Keusch). Note that both individuals are ringed; they belong to the study population in the Valais (Schaub et al., 2011).

the integrated population model, we automatically account for emigration, even if we cannot estimate it. On the other hand, information on immigration is very elusive. In our study, the population counts contain information about immigration because the annual change in population size is a result of all four demographic processes operating in a population: survival, productivity, immigration, and emigration. We have independent data for the other demographic processes (capture–recapture for apparent survival, reproductive output for productivity); hence, the combined analysis should enable us to obtain an estimate of immigration. This neat idea was brought up by Abadi et al. (2010b).

To write the BUGS code for our model, we take the three steps outlined earlier. First, we describe the link between population change and demographic rates. Hoopoes are short-lived, and we therefore assume two age classes only. We survey the population just before reproduction; that is, we conduct a prebreeding census. Hence, we differentiate 1-year-old individuals from older individuals. We include demographic stochasticity by using appropriate distributions. Since we want to estimate immigration, the population can be divided into three components: surviving philopatric adults (N_{ad}), locally produced young that survived and recruited into our population (N_1), and immigrants (N_{im}). We model the numbers of individuals in these population segments with binomial and Poisson distributions:

$$N_{ad,t+1} \sim \text{Binomial}(N_{1,t} + N_{ad,t} + N_{im,t}, \phi_{ad,t})$$

$$N_{1,t+1} \sim \text{Poisson}\left((N_{1,t} + N_{ad,t} + N_{im,t})\phi_{juv,t} \frac{f_t}{2}\right)$$

$$N_{im,t+1} \sim \text{Poisson}\left((N_{1,t} + N_{ad,t} + N_{im,t})\omega_t\right).$$

Here, $\phi_{juv,t}$ and $\phi_{ad,t}$ are apparent survival probabilities from year t to $t+1$ of young (from fledging to age 1) and adults, respectively, f_t is productivity in year t , and ω_t is the immigration rate. The latter is expressed as a ratio, that is, as the number of immigrants in year $t+1$ per individual in year t . We could estimate the annual number of immigrants using a Poisson distribution, but we prefer the rate, because we are interested in the temporal variation of the demographic rates and in the temporal variation of components of population size.

The second step is to write the likelihood of the individual data sets. As before, we use a state-space model for the population counts, with the state-process equations given above. The assumed observation process is slightly different from the earlier one: we now adopt a Poisson observation error (i.e., $y_t \sim \text{Poisson}(N_{1,t} + N_{ad,t} + N_{im,t})$). Hence, the relative observation error is constant, but the absolute observation error increases with population size, because in the Poisson distribution the variance is equal to the mean. Compared to the log-normal distribution used by Schaub

et al. (2011) for the same data, convergence of Markov chains is obtained faster with a Poisson observation error. Parameter estimates are almost identical regardless of the distribution chosen for the observation error (you can try this out for the lognormal or even the normal). The state-space model further requires prior distributions for the initial sizes of all population components ($N_{1,1}$, $N_{ad,1}$, $N_{im,1}$). For the capture–recapture data, we use the CJS model with a multinomial likelihood and for the productivity data a Poisson regression. We assume independence among the three data sets, and consequently, the joint likelihood, the definition of which is our third step, is the product of the three individual data likelihoods. This model is depicted graphically in Fig. 11.3c.

We are particularly interested in assessing the temporal variability of all demographic parameters, and we formulate a model that includes environmental stochasticity. We specify random time effects for all demographic parameters as discussed in Section 7.4.2. Thus, we imagine that the annual values of all demographic rates (on an appropriately transformed scale) are realizations from normal distributions with means and variances that we estimate. We use the logit link for survival probabilities and the log link for productivity and immigration rates. When adopting a hierarchical model for the demographic parameters in each year, we no longer need priors and initial values for each of them in each year. Rather, we specify a prior and an initial value for their mean and variance, that is, for the hyperparameters. We use vague normal priors for the logit of the mean survival probabilities and the log of mean fecundity and immigration and uniform priors for the variance parameters on the standard deviation scale.

Now, we are ready! We load the data, write the model in BUGS language, and run it.

```
# Load data
nyears <- 9      # Number of years

# Capture-recapture data: m-array of juveniles and adults (these are
# males and females together)
marray.j <- matrix(c(15, 3, 0, 0, 0, 0, 0, 198, 0, 34, 9, 1, 0, 0, 0,
                     287, 0, 0, 56, 8, 1, 0, 0, 0, 455, 0, 0, 0, 48, 3, 1, 0, 0, 518, 0, 0, 0, 0,
                     45, 13, 2, 0, 463, 0, 0, 0, 0, 27, 7, 0, 493, 0, 0, 0, 0, 0, 0, 37, 3,
                     434, 0, 0, 0, 0, 0, 0, 39, 405), nrow = 8, ncol = 9, byrow = TRUE)
marray.a <- matrix(c(14, 2, 0, 0, 0, 0, 0, 0, 43, 0, 22, 4, 0, 0, 0, 0,
                     44, 0, 0, 34, 2, 0, 0, 0, 0, 79, 0, 0, 0, 51, 3, 0, 0, 0, 94, 0, 0, 0, 0, 45,
                     3, 0, 0, 118, 0, 0, 0, 0, 0, 44, 3, 0, 113, 0, 0, 0, 0, 0, 48, 2, 99, 0,
                     0, 0, 0, 0, 0, 51, 90), nrow = 8, ncol = 9, byrow = TRUE)

# Population count data
popcount <- c(32, 42, 64, 85, 82, 78, 73, 69, 79)

# Productivity data
J <- c(189, 274, 398, 538, 520, 476, 463, 438) # Number of offspring
R <- c(28, 36, 57, 77, 81, 83, 77, 72)           # Number of surveyed
                                                    broods
```

```
# Specify model in BUGS language
sink("ipm.hoopoe.bug")
cat("
model {

# -----
# Integrated population model
# - Age structured model with 2 age classes:
#     1-year old and adults (at least 2-years old)
# - Age at first breeding = 1 year
# - Prebreeding census, female-based
# - All vital rates are assumed to be time-dependent (random)
# - Explicit estimation of immigration
# -----


# -----
# 1. Define the priors for the parameters
# -----
# Initial population sizes
N1[1] ~ dnorm(100, 0.0001)I(0,)          # 1-year old individuals
NadSurv[1] ~ dnorm(100, 0.0001)I(0,)       # Adults >= 2 years
Nadimm[1] ~ dnorm(100, 0.0001)I(0,)        # Immigrants

# Mean demographic parameters (on appropriate scale)
l.mphij ~ dnorm(0, 0.0001)I(-10,10)      # Bounded to help with
                                             convergence
l.mphia ~ dnorm(0, 0.0001)I(-10,10)
l.mfec ~ dnorm(0, 0.0001)I(-10,10)
l.mim ~ dnorm(0, 0.0001)I(-10,10)
l.p ~ dnorm(0, 0.0001)I(-10,10)

# Precision of standard deviations of temporal variability
sig.phij ~ dunif(0, 10)
tau.phij <- pow(sig.phij, -2)
sig.phia ~ dunif(0, 10)
tau.phia <- pow(sig.phia, -2)
sig.fec ~ dunif(0, 10)
tau.fec <- pow(sig.fec, -2)
sig.im ~ dunif(0, 10)
tau.im <- pow(sig.im, -2)

# Distribution of error terms (bounded to help with convergence)
for (t in 1:(nyears-1)){
    epsilon.phij[t] ~ dnorm(0, tau.phij)I(-15,15)
    epsilon.phia[t] ~ dnorm(0, tau.phia)I(-15,15)
    epsilon.fec[t] ~ dnorm(0, tau.fec)I(-15,15)
    epsilon.im[t] ~ dnorm(0, tau.im)I(-15,15)
}

# -----
# 2. Constrain parameters
# -----
for (t in 1:(nyears-1)){
    logit(phiij[t]) <- l.mphij + epsilon.phij[t] # Juv. apparent survival
    logit(phia[t]) <- l.mphia + epsilon.phia[t] # Adult apparent
                                                    survival
```

d5b0a4c6971303bb281a3a314c6dc667
ebrary

```
log(f[t]) <- l.mfec + epsilon.fec[t]          # Productivity
log(omega[t]) <- l.mim + epsilon.im[t]        # Immigration
logit(p[t]) <- l.p                            # Recapture probability
}

# -----
# 3. Derived parameters
# -----
mphij <- exp(l.mphij)/(1+exp(l.mphij))      # Mean juvenile survival
                                                probability
mphia <- exp(l.mphia)/(1+exp(l.mphia))      # Mean adult survival
                                                probability
mfec <- exp(l.mfec)                          # Mean productivity
mim <- exp(l.mim)                           # Mean immigration rate

# Population growth rate
for (t in 1:(nyears-1)){
  lambda[t] <- Ntot[t+1] / Ntot[t]
  logla[t] <- log(lambda[t])
}
mlam <- exp((1/(nyears-1))*sum(logla[1:(nyears-1)])) # Geometric
                                                          mean

# -----
# 4. The likelihoods of the single data sets
# -----
# 4.1. Likelihood for population population count data (state-space
# model)
# 4.1.1 System process
for (t in 2:nyears){
  mean1[t] <- 0.5 * f[t-1] * phij[t-1] * Ntot[t-1]
  N1[t] ~ dpois(mean1[t])
  NadSurv[t] ~ dbin(phia[t-1], Ntot[t-1])
  mpo[t] <- Ntot[t-1] * omega[t-1]
  Nadimm[t] ~ dpois(mpo[t])
}

# 4.1.2 Observation process
for (t in 1:nyears){
  Ntot[t] <- NadSurv[t] + Nadimm[t] + N1[t]
  y[t] ~ dpois(Ntot[t])
}

# 4.2 Likelihood for capture-recapture data: CJS model (2 age classes)
# Multinomial likelihood
for (t in 1:(nyears-1)){
  marray.j[t,1:nyears] ~ dmulti(pr.j[t,], r.j[t])
  marray.a[t,1:nyears] ~ dmulti(pr.a[t,], r.a[t])
}

# Calculate number of released individuals
for (t in 1:(nyears-1)){
  r.j[t] <- sum(marray.j[t,])
  r.a[t] <- sum(marray.a[t,])
}
```

```
# m-array cell probabilities for juveniles
for (t in 1:(nyears-1)){
  q[t] <- 1-p[t]
  # Main diagonal
  pr.j[t,t] <- phij[t]*p[t]
  # Above main diagonal
  for (j in (t+1):(nyears-1)){
    pr.j[t,j] <- phij[t]*prod(phia[(t+1):j])*prod(q[t:(j-1)])*p[j]
    } #j
  # Below main diagonal
  for (j in 1:(t-1)){
    pr.j[t,j] <- 0
    } #j
  # Last column
  pr.j[t,nyears] <- 1-sum(pr.j[t,1:(nyears-1)])
} #t

# m-array cell probabilities for adults
for (t in 1:(nyears-1)){
  # Main diagonal
  pr.a[t,t] <- phia[t]*p[t]
  # above main diagonal
  for (j in (t+1):(nyears-1)){
    pr.a[t,j] <- prod(phia[t:j])*prod(q[t:(j-1)])*p[j]
    } #j
  # Below main diagonal
  for (j in 1:(t-1)){
    pr.a[t,j] <- 0
    } #j
  # Last column
  pr.a[t,nyears] <- 1-sum(pr.a[t,1:(nyears-1)])
} #t

# 4.3. Likelihood for productivity data: Poisson regression
for (t in 1:(nyears-1)){
  J[t] ~ dpois(rho[t])
  rho[t] <- R[t]^3 * f[t]
}
",fill = TRUE)
sink()

# Bundle data
bugs.data <- list(nyears = nyyears, marray.j = marray.j, marray.a =
  marray.a, y = popcount, J = J, R = R)

# Initial values
inits <- function(){list(l.mphij = rnorm(1, 0.2, 0.5), l.mphia = rnorm
  (1, 0.2, 0.5), l.mfec = rnorm(1, 0.2, 0.5), l.mim = rnorm(1, 0.2,
  0.5), l.p = rnorm(1, 0.2, 1), sig.phij = runif(1, 0.1, 10), sig.phia =
  runif(1, 0.1, 10), sig.fec = runif(1, 0.1, 10), sig.im = runif(1, 0.1,
  10), N1 = round(runif(nyears, 1, 50), 0), NadSurv = round(runif
  (nyears, 5, 50), 0), Nadimm = round(runif(nyears, 1, 50), 0))}
```

```
# Parameters monitored
parameters <- c("phij", "phia", "f", "omega", "p", "lambda", "mphij",
  "mphia", "mfec", "mim", "mlam", "sig.phij", "sig.phia", "sig.fec",
  "sig.im", "N1", "NadSurv", "Nadimm", "Ntot")

# MCMC settings
ni <- 20000
nt <- 6
nb <- 10000
nc <- 3

# Call WinBUGS from R (BRT 5 min)
ipm.hoopoe <- bugs(bugs.data, inits, parameters, "ipm.hoopoe.bug",
  n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
  bugs.directory = bugs.dir, working.directory = getwd())
```

With the chosen MCMC settings, we achieve convergence, though for a research paper, we may decide to run longer chains. Next, we inspect posterior distributions of the parameters of interest. Figure 11.7 is a graphical summary of what we learn about the dynamics of our hoopoe population (see Web appendix 1 for R code). The estimated population size is close to the population counts; that is, the observation error appears small. This is not surprising because the vast majority of hoopoes in our study use nest boxes, and the count of occupied nest boxes is unaffected by the observation error. Yet, there is still some uncertainty because some hoopoes conduct two broods in a year, and first and second broods may overlap. By modeling the maximal number of simultaneously occupied nest boxes, we strictly model an index of the actual population size, though probably a rather accurate one. The uncertainty around the population size estimates is quite large. This may reflect uncertainty due to the estimation of the demographic parameters and especially the uncertainty introduced by accounting for annual variability in all rates. Survival and productivity are estimated quite precisely; that is, they have a narrow CRI. In contrast, the immigration rate is not estimated very precisely because we have no explicit data on immigration (compare this with the similar situation in Section 11.4, in which we estimated productivity without productivity data). Immigration rate is one of the hardest demographic parameters to estimate in a demographic population analysis.

Specification of annual rates as random effects allowed us to estimate the temporal variance in the demographic rates and to get better estimates of the annual demographic rates (Gelman and Hill, 2007). The annual parameter estimates are shrunk toward the mean of their prior distribution, and the degree of shrinkage depends on the precision and the temporal variance (Burnham and White, 2002). Shrinkage pulls back outlying estimates toward more “sensible” values (see Section 4.1).

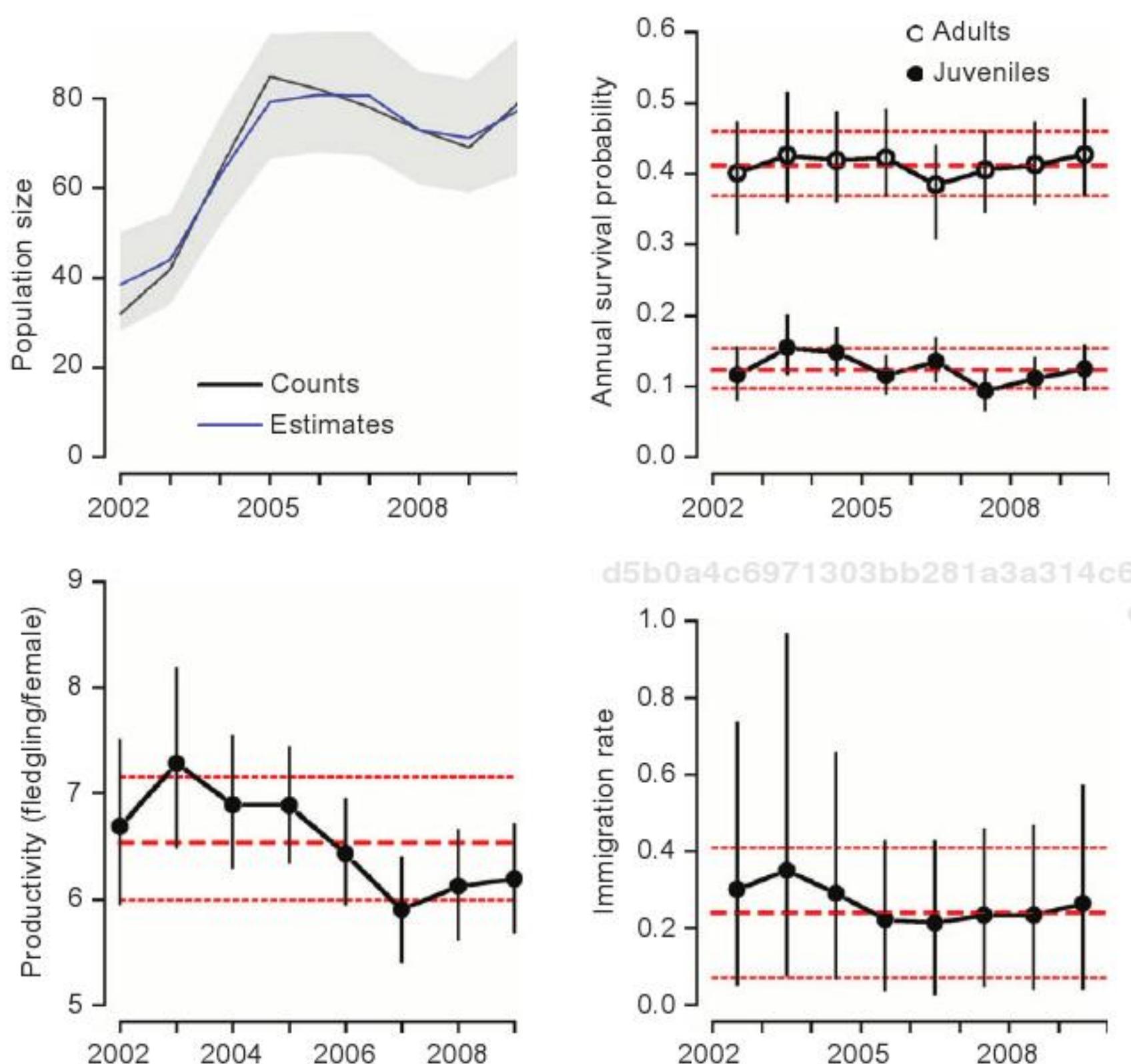


FIGURE 11.7 Posterior means (with 95% CRI) of the demographic parameters in a hoopoe population in SW Switzerland under a population model with random year effects for all demographic rates. Red lines show the mean and the 95% CRI of the mean hyperparameters (see Web appendix 1 for R code to create this figure).

To study the link between demographic rates and population growth, we can correlate the estimates of annual population growth rates with those of each demographic rate. These correlations provide indications about the strength of the contributions of the temporal variation in demographic rates to the temporal variation in the population growth rate (Robinson et al., 2004; Schaub et al., 2011). These correlations were positive for all demographic rates (Fig. 11.8; for R code see Web appendix 1). The correlation was highest for productivity and lowest for adult survival, suggesting that variation in productivity contributed more to the variation in population growth rates than the variation in adult survival. Credible intervals for correlation coefficients and the incidence of a positive correlation can easily be computed as derived quantity from the MCMC output of the analysis (see Web appendix 1).

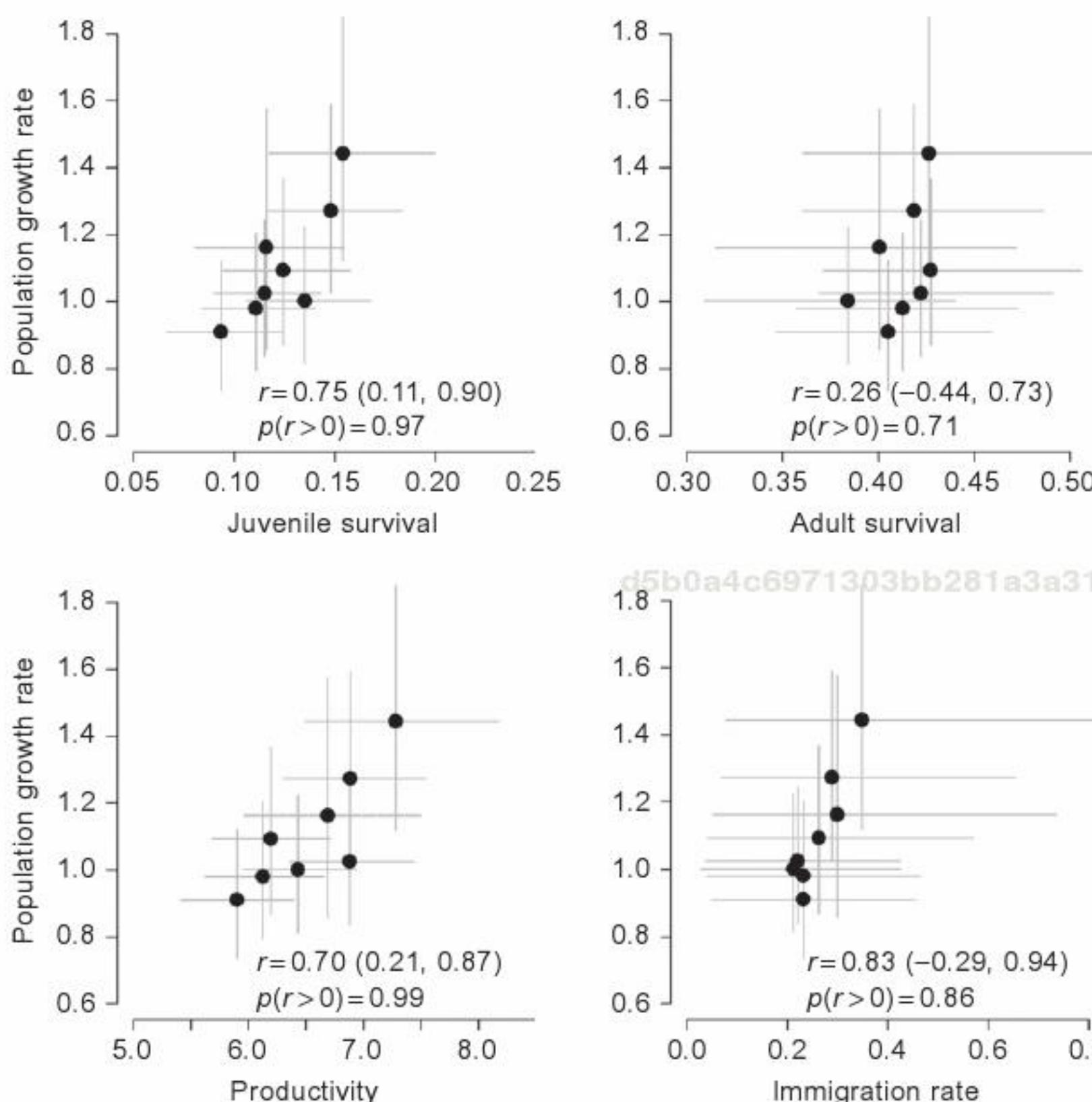


FIGURE 11.8 Estimates of annual demographic rates plotted against the estimates of interannual population growth. Black dots show posterior means and grey lines 95% CRI. Inset we print the posterior mode of the correlation coefficients (r , with 95% CRI) and the probability of a positive correlation ($p(r > 0)$) (see Web appendix 1 for R code to create this figure).

11.7 SUMMARY AND OUTLOOK

In this chapter, we have introduced integrated population models. In this recently developed modeling framework, population counts are combined coherently with data on demographic rates, such as survival or productivity, in a sort of stochastic Leslie matrix model. There are many advantages of using an integrated population model: the precision of parameter estimates is increased and parameters that are not estimable otherwise may become estimable. Thus, an integrated population model is much more powerful than many traditional approaches to population analysis, in which counts and data on demographic rates are analyzed

separately. The core of any integrated model is a population projection matrix model, which provides the explicit link between changes in population size and demographic rates. A state-space formulation for the counts separates the dynamics of the true population size from observation error.

Using integrated population models allows one to make rigorous inference about unobserved quantities, such as the sizes of all age (or stage) classes defined, even if we cannot distinguish them in the field. As one caveat, though, we must not forget that what is called “population size” in state-space models such as those in this chapter and in Chapter 5 is really a smoothed index of population size; see Chapter 5. If our goal is to get unbiased estimates of population size, additional information about the population count data is necessary, combined with a different model for the observation process (i.e., as in Chapters 6, 10, 12, and 13).

So far no goodness-of-fit test is available for integrated population models (Schaub and Abadi, 2011). The best that can be done at the moment is to assess the goodness of fit of individual likelihoods to single data set using established procedures. If these tests are satisfactory, we may assume that the integrated model is also a satisfactory description of the complete data.

Integrated population models are a powerful and flexible framework to understand population dynamics (Schaub and Abadi, 2011). Our examples combined rather simple Leslie matrix state-space, capture–recapture, and Poisson regression models. Yet, it is easy to either make these models more complicated and realistic, consider multistate models or likelihoods of different data types. For example, Brooks et al. (2004) used mark-recovery data of lapwings instead of capture–recapture data to get independent information about true rather than apparent survival (see Exercise 4 in Section 11.8). Schaub et al. (2010) used age-at-death data to inform survival in an eagle owl population. The basic structure of the integrated model remains the same; you just need to replace the different modules in the joint likelihood, and you can make creative use of all the Lego pieces that we have described in this book. Thus, the models featured in this chapter provide some sort of synthesis of all other models, which makes them extremely flexible and powerful!

11.8 EXERCISES

1. Predict the hoopoe population size in 3 years. How large is the extinction probability (assume an extinction threshold of five pairs)?
2. Assume that population count data from years 3 and 5 are missing in the hoopoe example. Use an integrated population model to estimate these missing data. What do you observe?

3. Fit an integrated population model with time-dependent parameters to the ortolan bunting data (Section 11.3). Compare the population size estimates with those from a model with constant demographic parameters and explain.
4. Use an integrated population model to study population dynamics of British lapwings (Besbeas et al., 2002; Brooks et al., 2004). The data consist of a national population index (1965–1998) and of mark recoveries from individuals marked as hatchlings (1963–1997). No data on productivity is available. Construct a model with two age classes for survival and where (1) first breeding occurs at the age of 2 years and (2) where it occurs at the age of 3 years. Further, (3) make a model where survival is a function of the number of frost days. The data file `lapwing.txt` (population index, m-array, normalized number of frost days; from Brooks et al. 2004) can be found on the book website (www.vogelwarte.ch/bpa).

d5b0a4c6971303bb281a3a314c6dc667
ebrary

This page intentionally left blank

d5b0a4c6971303bb281a3a314c6dc667
ebrary

d5b0a4c6971303bb281a3a314c6dc667
ebrary

d5b0a4c6971303bb281a3a314c6dc667
ebrary