CHAPTER

5

State-Space Models for Population Counts

5.1 INTRODUCTION

State-space models are hierarchical models that decompose an observed time series of counts or other observed responses into a process variation and an observation error component. They are suitable for the description of Markovian, that is, autoregressive, processes that are latent or hidden, because they are observed imperfectly (Harvey, 1989). Originally developed in industry (Kalman, 1960), the use of state-space models in population dynamics has only recently begun (De Valpine and Hastings, 2002; Buckland et al., 2004; Clark and Björnstad, 2004). The change in

population size over time is a Markovian process because population size in year $t + 1$ depends on population size in year t. Perhaps the simplest population dynamics model is the exponential:

$$N_{t+1} = N_t \lambda_t,$$

where λ_t is the population growth rate, defined for $t = 1, \ldots, T - 1$; T being the number of years with observed counts. A typical goal of a population dynamics analysis is to estimate the population growth rate and to study factors affecting it. This is relatively straightforward if the exact population size in each year is known. Regression-like models can then be used to estimate temporal variability, the strength of density-dependence, and the impact of environmental factors (see, e.g., Dennis et al., 1991; Dennis and Taper, 1994; Lande et al., 2003). However, only in exceptional cases do we know the exact size of a population. More typically, we only have a count with some unknown observation error associated. This observation error must be accounted for when analyzing population dynamics data. Otherwise, the power to detect factors affecting population dynamics is reduced and density-dependence may be detected spuriously (Freckleton et al., 2006).

The state-space models of this chapter allow one to deal with the challenge of observation error in the analysis of population dynamics, at least partially (see Section 5.3). The state-space model adopted here consists of two sets of equations. The state-process equations describe the true, but unknown development of a state, that is, the population dynamics free of observation error. For the dynamics of an exponentially growing population, the unknown state is population size N_t at time t:

$$N_{t+1} = N_t \lambda_t$$

$$\lambda_t \sim \text{Normal}(\overline{\lambda}, \sigma_\lambda^2).$$

In this model, we assume that the growth rates (λ_t) are realizations of a normal random process with mean $\overline{\lambda}$ and variance σ_λ^2. The mean is the long-term growth rate of the population, and the variance is a measure of the environmental variability affecting this growth rate, that is, environmental stochasticity. Note that the population size in the first year is not defined by the equations above. We therefore need to define this initial state by another model. In the Bayesian context, we can either specify a prior distribution for N_1, or fix it to the observed count, making the assumption that the first count is error-free.

The second set of equations maps the true state of the process on the observed data. These observation equations are conditional on the process equations. Assume we have a population count for each of a series of years.

We link these observations (y_t) with the true population size using the following equations:

$$y_t = N_t + \varepsilon_t$$

$$\varepsilon_t \sim \text{Normal}(0, \sigma_y^2).$$

They represent the assumption that in year t, we make an observation error of magnitude ε_t and that these errors in the counts around the true population size can be described by a normal distribution with mean zero and variance σ_y^2; the latter is the observation error. In this model, the true population size is as likely to be over- as it is to be undercounted. For instance, in one year, we may miss more individuals than we double-count, resulting in a negative value of ε_t, and in another year, it may be the other way round. The observation error is, in fact, a residual error and therefore incorporates not only observation errors but also the lack of fit of the state equations. Figure 5.1 represents the model graphically.

The model assumes that the two sets of random terms, λ_t and ε_t, are serially independent, independent from each other and that each set is identically distributed. Formulations of the likelihood of the model can be found, for example, in De Valpine and Hastings (2002). The model likelihood is composed of the likelihoods for the initial state, for the state process, and for the observation process.

The state-space modeling framework is very flexible and can be extended in various ways. We could, of course, define different state-processes (e.g., include survival and fecundity, see Chapter 11, or use a logistic growth model). Also, different descriptions of the observation process are possible, for instance, the assumption of binomial, Poisson, or lognormal errors. As we will see later, state-space models can be used for purposes other than estimating the true trajectories of population size. Examples we will see later include probabilities of survival (Chapters 7, 8, and 10), state-transition (Chapter 9),
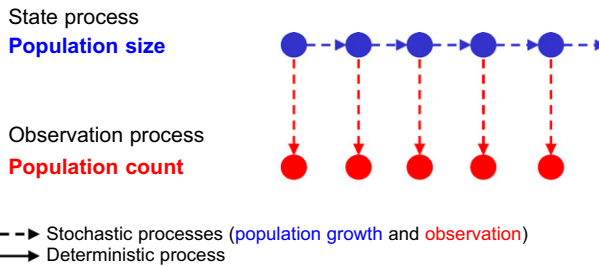
FIGURE 5.1 State process of a population and the conditional observation process. The state process describes the dynamics of population size over time (N_t), whereas the observation process describes the error-prone population counts (y_t). Note that this figure is essentially a version of Fig. 2.1.

and species occurrence and site occupancy (Chapter 12). They are also at the heart of integrated population models (Chapter 11). Because the observation process is conditional on the state process, a state-space model is also a hierarchical model (Royle and Dorazio, 2008).

5.2 A SIMPLE MODEL

We will first simulate and analyze count data in a very simple context. Let's assume that a population of ibex (Fig. 5.2) with initial size of 30 individuals was monitored during 25 years and grew annually by 2% on average. Shortly after the start of snow-melt, ibexes are counted by the aid of a telescope from the slope opposite to the mountain ridge which constitutes the core area of the population. The population survey is not perfect, and we assume that the variance of the observation error is 20. To simulate the data, we first define the underlying parameters.

```
n.years <- 25          # Number of years
N1 <- 30               # Initial population size
mean.lambda <- 1.02    # Mean annual population growth rate
sigma2.lambda <- 0.02  # Process (temporal) variation of the growth rate
sigma2.y <- 20         # Variance of the observation error
```

Next we simulate the population sizes under the assumption of exponential growth.

FIGURE 5.2   Ibexes (*Capra ibex*), Switzerland (Photograph by F. Labhardt).

```
y <- N <- numeric(n.years)
N[1] <- N1
lambda <- rnorm(n.years-1, mean.lambda, sqrt(sigma2.lambda))
for (t in 1:(n.years-1)){
    N[t+1] <- N[t] * lambda[t]
    }
```

Finally, we generate the observed data conditional on the true population sizes.

```
for (t in 1:n.years){
    y[t] <- rnorm(1, N[t], sqrt(sigma2.y))
    }
```

You may have noticed that neither population size nor the counts are integers, as might be expected for a biological population. However, our goal here is to simulate and analyze the data in exactly the same way as the classical state-space models of this chapter are formulated and to explore the performance of state-space models. In Section 5.3, we will simulate a data set under more mechanistic assumptions which lead to integers.

Now, we analyze the simulated data. We first write the model in the BUGS language.

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

# Priors and constraints
N.est[1] ~ dunif(0, 500)            # Prior for initial population size
mean.lambda ~ dunif(0, 10)          # Prior for mean growth rate
sigma.proc ~ dunif(0, 10)           # Prior for sd of state process
sigma2.proc <- pow(sigma.proc, 2)
tau.proc <- pow(sigma.proc, -2)
sigma.obs ~ dunif(0, 100)           # Prior for sd of observation process
sigma2.obs <- pow(sigma.obs, 2)
tau.obs <- pow(sigma.obs, -2)

# Likelihood
# State process
for (t in 1:(T-1)){
    lambda[t] ~ dnorm(mean.lambda, tau.proc)
    N.est[t+1] <- N.est[t] * lambda[t]
    }

# Observation process
for (t in 1:T) {
    y[t] ~ dnorm(N.est[t], tau.obs)
    }
}
",fill = TRUE)
sink()
```

We have chosen uniform priors for the standard deviations of the process and observation errors. This is to be preferred over a gamma prior distribution of the variances because they are less informative (Gelman, 2006). Next, we bundle the data, define initial values and MCMC settings, and analyze the model. The initial value for the population size in the first year needs to be chosen with care; if it is too far from the true value, the model may not run. A practical option is to use a random value close to the population counts. Problems with updating in the MCMC algorithm may also occur if the initial values for the process and observation standard deviations are too large. Again, choosing "good" initial values helps to avoid problems.

```
# Bundle data
bugs.data <- list(y = y, T = n.years)

# Initial values
inits <- function(){list(sigma.proc = runif(1, 0, 5), mean.lambda =
   runif(1, 0.1, 2), sigma.obs = runif(1, 0, 10), N.est = c(runif(1, 20,
   40), rep(NA, (n.years-1))))}

# Parameters monitored
parameters <- c("lambda", "mean.lambda", "sigma2.obs", "sigma2.proc",
   "N.est")

# MCMC settings
ni <- 25000
nt <- 3
nb <- 10000
nc <- 3

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

How quickly the Markov chains converge depends greatly on the data. With some data sets, convergence is obtained much more swiftly than with others. Now we draw a plot (Fig. 5.3) to compare estimated, true, and observed population sizes. We notice that the estimated population sizes are smoothed and closer to the true population sizes than the raw counts.

```
# Define function to draw a graph to summarize results
graph.ssm <- function(ssm, N, y){
   fitted <- lower <- upper <- numeric()
   n.years <- length(y)
   for (i in 1:n.years){
      fitted[i] <- mean(ssm$sims.list$N.est[,i])
      lower[i] <- quantile(ssm$sims.list$N.est[,i], 0.025)
      upper[i] <- quantile(ssm$sims.list$N.est[,i], 0.975)}
```
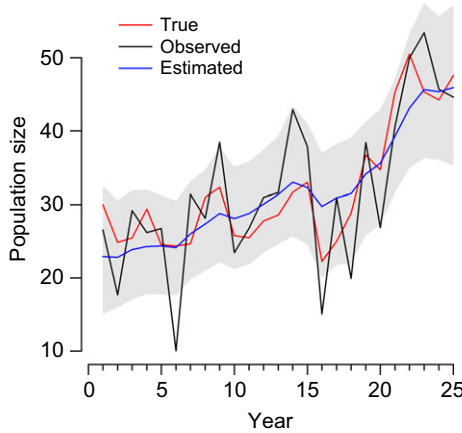
FIGURE 5.3 Analysis of ibex population dynamics: trajectory of true population size (red), observed counts (black), and posterior means of estimated population size (blue; with 95% CRI shaded).

```
    m1 <- min(c(y, fitted, N, lower))
    m2 <- max(c(y, fitted, N, upper))
    par(mar = c(4.5, 4, 1, 1), cex = 1.2)
    plot(0, 0, ylim = c(m1, m2), xlim = c(0.5, n.years), ylab = "Population
        size", xlab = "Year", las = 1, col = "black", type = "l", lwd = 2,
        frame = FALSE, axes = FALSE)
    axis(2, las = 1)
    axis(1, at = seq(0, n.years, 5), labels = seq(0, n.years, 5))
    axis(1, at = 0:n.years, labels = rep("", n.years + 1), tcl = -0.25)
    polygon(x = c(1:n.years, n.years:1), y = c(lower, upper[n.years:1]),
        col = "gray90", border = "gray90")
    points(N, type = "l", col = "red", lwd = 2)
    points(y, type = "l", col = "black", lwd = 2)
    points(fitted, type = "l", col = "blue", lwd = 2)
    legend(x = 1, y = m2, legend = c("True", "Observed", "Estimated"),
        lty = c(1, 1, 1), lwd = c(2, 2, 2), col = c("red", "black", "blue"),
        bty = "n", cex = 1)
}
# Execute function: Produce figure
graph.ssm(ssm, N, y)
```

5.3 SYSTEMATIC BIAS IN THE OBSERVATION PROCESS

In the previous example we have seen that the model performs well in the presence of "random" observation errors. By this we mean that on average, false positives (i.e., double-counting) and false negatives (i.e., nondetection)

cancel out. In this sense, the state-space model is able to "correct" noisy counts for observation error. However, the model cannot recover unbiased estimates of true population size, when false negative and positive errors don't cancel out on average. With the data considered in this chapter, there is no direct information about the observation process, that is, about detection probability or double-counting rates. Thus, in the absence of false-positive errors, the "state" modeled in the state-space models of this chapter is the product of population size N and detection probability p, and not population size N, as one might think. We illustrate this next.

We start with a simple example where we assume that the ibex population remains stable at 50 individuals over 25 years. Further, we assume that detection probability is constant at 0.7, thus on average we only count 70% of the population. Each individual can be seen or missed, so the annual count is a binomial random variable, and we write $y_t \sim \text{Bin}(N_t, p)$. We start by defining the true population sizes.

```
n.years <- 25 # Number of years
N <- rep(50, n.years)
```

Then, we simulate the counts using the binomial distribution and inspect the numbers.

```
p <- 0.7
y <- numeric(n.years)
for (t in 1:n.years){
    y[t] <- rbinom(1, N[t], p)
    }
y
[1] 34 34 35 38 29 37 ... 27 35 33 32
```

Clearly, the observed counts are smaller than the true population size (what a surprise!). More importantly, we also notice that the counts vary, although both the true population size and the detection probability were constant across time. As we have seen in Section 1.3, this variation is the binomial sampling variation. The magnitude of this variance is known from statistical theory to be on average $Np(1 - p) = 10.5$.

We next use WinBUGS to fit to these data the same state-space model as in Section 5.2. Thus, we partition the observed variation in the counts into process variation and observation error and use a normal approximation for the latter. In large samples, we would expect the estimate of the process variance to be close to zero and that for the observation error to be close to 10.5.

```
# Bundle data
bugs.data <- list(y = y, T = n.years)
```

```
# Initial values
inits <- function(){list(sigma.proc = runif(1, 0, 5), mean.lambda =
    runif(1, 0.1, 2), sigma.obs = runif(1, 0, 10), N.est = c(runif(1, 30,
    60), rep(NA, (n.years-1))))}
```

```
# Parameters monitored
parameters <- c("lambda", "mean.lambda", "sigma2.obs", "sigma2.proc",
    "N.est")
```

```
# MCMC settings
ni <- 25000
nt <- 3
nb <- 10000
nc <- 3
```

```
# Call WinBUGS from R (BRT <1 min)
ssm <- bugs(bugs.data, inits, parameters, "ssm.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(ssm, digits = 3)
              mean     sd   2.5%    25%    50%    75%   97.5%  Rhat n.eff
[...]
mean.lambda  0.999  0.009  0.981  0.994  0.998  1.002  1.020 1.015  4800
sigma2.obs  15.219  5.644  7.157 11.370 14.170 18.020 28.500 1.005   660
sigma2.proc  0.002  0.003  0.000  0.000  0.001  0.002  0.011 1.012   210
[...]
```

As expected, the process variance is estimated to be close to zero and the estimate of the observation variance is large. As a result of sampling variation, the latter estimate is of course not equal to 10.5, although a 95% CRI covers the expected value. With larger population sizes and averaged over many simulation replicates, we would expect to be right on target (see also Section 1.5). The estimate of the average population growth rate is essentially equal to 1, which is also what we would expect from a constant population. Now, let's compare the estimated and true population sizes, as well as the counts (Fig. 5.4).

```
# Produce figure
graph.ssm(ssm, N, y)
```

Figure 5.4 shows that the estimated population sizes are consistently lower than the true sizes, but the temporal pattern of the estimated population sizes matches quite well to that of the true population sizes. Thus, the model is able to correct for the temporal variation of the observation process induced by the binomial nature of the counts. However, the model cannot fully correct for the detection error; the population estimates are all well below the true population size. The mean (over time) of the estimated population sizes is 33.7, which is close to what we would expect, Np ($50*0.7 = 35$). Hence, we obtain an improved estimate of the population index Np with the
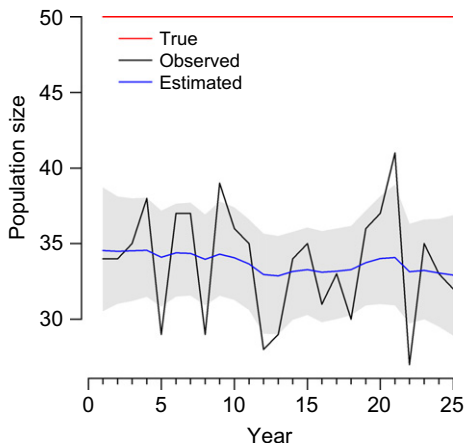
FIGURE 5.4 Effects of systematic observation errors on the state-space model: estimated states represent an unbiased population index. True (red), observed (black), and estimated trajectory of population size (blue; with 95% CRI shaded).

state-space model, by eliminating the effects of annual observation errors, but we are unable to estimate true population size. We feel that the nature of the correction for the observation error in the kind of state-space models considered in this chapter is sometimes misunderstood among ecologists.

Next we conduct a simulation where detection probability is not constant over time, but instead has a positive trend that is linear on the logit scale. What will our simple state-space model tell us about the true trend of the ibex population in this case?

We start by defining the true population sizes.

```
n.years <- 25 # Number of years
N <- rep(50, n.years)
```

Then we simulate the counts using the binomial distribution.

```
lp <- -0.5 + 0.1*(1:n.years)     # Increasing trend of logit p
p <- plogis(lp)
y <- numeric(n.years)
for (t in 1:n.years){
   y[t] <- rbinom(1, N[t], p[t])
   }
```

We analyze the data with the same model as before:

```
# Bundle data
bugs.data <- list(y = y, T = n.years)

# Call WinBUGS from R (BRT <1 min)
ssm <- bugs(bugs.data, inits, parameters, "ssm.bug", n.chains = nc,
   n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
   bugs.dir, working.directory = getwd())
```
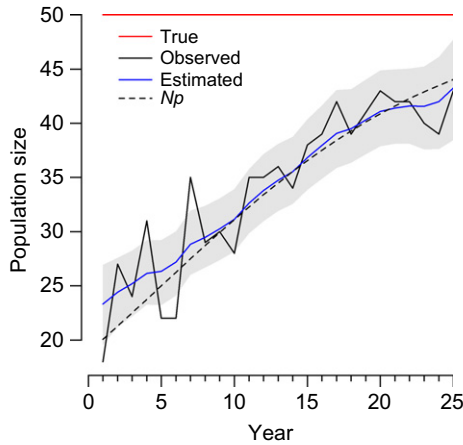
FIGURE 5.5   Effects of temporal change in systematic observation errors on the state-space model: estimated states represent a biased population index. True (red), observed (black), and estimated population size (blue) along with a 95% CRI (shading). The dashed line shows the expectation of the counts under binomial sampling.

The comparison among true, observed, and estimated population size shows that the model succeeds in getting rid of the random sampling variation in the counts that is resulting from binomial sampling (Fig. 5.5). However, the model can't correct the estimated population trajectory for the shift imposed by the deterministic trend in the observation process. To clarify this, we also calculated the expected value of the counts as Np and added this to the plot. Clearly, the estimated population size under the state-space model follows quite closely to this expected value (Np). In other words, our simple state-space model was not able to correct for a systematic pattern in detection, and consequently we would erroneously infer a population increase if we trusted the model result in this case.

```
# Produce figure
graph.ssm(ssm, N, y)
points(N*p, col = "black", type = "l", lwd = 2, lty = 2)
legend(x = 1, y = 45.5, legend = "Np", lwd = 2, col = "black", lty = 2,
   bty = "n")
```

These examples illustrate three important points about the state-space models in this chapter:

1. They yield unbiased estimates of population size only if false-negative (detection probability) and false-positive observations (double-counting) cancel out on average. Counts are then unbiased on average but there is sampling variation, and this is corrected for by the model.
2. They produce unbiased estimates of population indices (i.e., of Np) if detection probability is <1 and has no pattern over time. In that case,

for the example of imperfect detection (and no false positives), the state-space model corrects for binomial sampling variation.

3. They do not yield unbiased estimates of population size nor of indices if there are temporal patterns either in detection probability or in the false-positive rates (unless, of course, the two cancel out).

5.4 REAL EXAMPLE: HOUSE MARTIN POPULATION COUNTS IN THE VILLAGE OF MAGDEN

The population of house martins (Fig. 5.6) in Magden (a small village in Northern Switzerland where MS grew up) has been surveyed by Reto Freuler since 1990. In initial years, Freuler counted all occupied nests himself, but later he sent questionnaires to house owners with known house martin nests asking for the number of occupied nests to be reported to him. The goal of the analysis is to estimate the average stochastic population growth rate, the process variance and to predict population sizes until 2015 with an assessment of uncertainty. Furthermore, we want to estimate the probability that the population size in 2015 is lower than in 2009, the year with the most recent data. In contrast to the earlier examples, we now adopt the exponential growth model on the log scale because it is more appropriate for stochastic environments (Lande et al., 2003).

FIGURE 5.6   House martin (*Delichon urbica*), Finland, 2005 (Photograph by T. Muukkonen).

The state-process model, now, becomes $\log(N_{t+1}) = \log(N_t) + r_t$, with $r_t \sim N(\bar{r}, \sigma_r^2)$; here, r_t is the stochastic population growth rate.

We use a similar state-space model to the one introduced earlier, with the mentioned exception (log scale). We also modify the priors for the initial population size, for the mean growth rate, and for the standard deviation of the state and the observation processes. Sometimes, it can be hard to fit state-space models in WinBUGS, which may fall in a trap or fail to start updating. Often, this has to do with the prior or the initial value for the first-year population size. It is advisable to choose a prior with mean equal to the first-year count and a relatively small variance. The range of the initial value for the first population size should also be relatively small. Although we specifically aim to predict population sizes in the future, the model does not need any changes for this purpose.

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

# Priors and constraints
logN.est[1] ~ dnorm(5.6, 0.01)       # Prior for initial population size
mean.r ~ dnorm(1, 0.001)             # Prior for mean growth rate
sigma.proc ~ dunif(0, 1)             # Prior for sd of state process
sigma2.proc <- pow(sigma.proc, 2)
tau.proc <- pow(sigma.proc, -2)
sigma.obs ~ dunif(0, 1)              # Prior for sd of obs. process
sigma2.obs <- pow(sigma.obs, 2)
tau.obs <- pow(sigma.obs, -2)

# Likelihood
# State process
for (t in 1:(T-1)){
    r[t] ~ dnorm(mean.r, tau.proc)
    logN.est[t+1] <- logN.est[t] + r[t]
    }

# Observation process
for (t in 1:T) {
    y[t] ~ dnorm(logN.est[t], tau.obs)
    }

# Population sizes on real scale
for (t in 1:T) {
    N.est[t] <- exp(logN.est[t])
    }
}
",fill = TRUE)
sink()
```

Next, we load the data. They consist of a vector with the number of observed breeding pairs in each year. Because we want to predict future

population sizes, we add NAs in the data. Why? Well, we saw in Chapter 2 that Bayesians treat all types of unknown quantities in the same way, be they predictions, missing values, or parameters. Hence, it is enough to specify some additional years without observations (regardless of whether they occur within the data series, or only at the end) with NA observations, and include them in the loop for calculating the likelihood. They are then updated (estimated) as part of the model fitting. Here, we would like to predict population size in year 2015, and so we need to add six NAs. Another way to achieve the same would be to extend the loop of the state-process to cover the additional years, and to leave the data as they are (i.e., without NAs).

```
# House martin population data from Magden
pyears <- 6 # Number of future years with predictions
hm <- c(271, 261, 309, 318, 231, 216, 208, 226, 195, 226, 233, 209, 226,
   192, 191, 225, 245, 205, 191, 174, rep(NA, pyears))
year <- 1990:(2009 + pyears)
```

We then bundle the data, specify initial values, list parameters to be estimated, determine the MCMC settings, and run the model.

```
# Bundle data
bugs.data <- list(y = log(hm), T = length(year))

# Initial values
inits <- function(){list(sigma.proc = runif(1, 0, 1), mean.r = rnorm(1),
   sigma.obs = runif(1, 0, 1), logN.est = c(rnorm(1, 5.6, 0.1), rep(NA,
   (length(year)-1))))}

# Parameters monitored
parameters <- c("r", "mean.r", "sigma2.obs", "sigma2.proc", "N.est")

# MCMC settings
ni <- 200000
nt <- 6
nb <- 100000
nc <- 3

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

Convergence is satisfactory after 200,000 iterations after a 100,000 burnin. The least well-mixed parameters are, not surprisingly, the two variances. If WinBUGS does not run properly and produces an undefined real result, just start the run again. The inits function will produce a different set of starting values each time it is called.

```
# Summarize posteriors
print(hm.ssm, digits = 2)

              mean     sd    2.5%     25%     50%     75%   97.5% Rhat n.eff
r[1]         -0.02   0.06   -0.13   -0.05   -0.02    0.01    0.11 1.00 47000
[...]
r[25]        -0.02   0.11   -0.26   -0.08   -0.02    0.04    0.21 1.00 27000
mean.r       -0.02   0.03   -0.08   -0.04   -0.02   -0.01    0.03 1.00 39000
sigma2.obs    0.01   0.01    0.00    0.00    0.00    0.01    0.02 1.00   870
sigma2.proc   0.01   0.01    0.00    0.00    0.01    0.02    0.03 1.01   770
N.est[1]    274.10  15.88  244.50  265.20  272.30  282.10  310.40 1.00  7100
N.est[2]    269.51  14.50  244.60  260.30  267.20  277.70  302.50 1.00  4500
[...]
N.est[26]   164.84  54.42   79.43  131.50  159.30  188.60  292.30 1.00 44000
```

We plot counts and the posterior means of population sizes along with their credible intervals.

```
# Draw figure
fitted <- lower <- upper <- numeric()
year <- 1990:2015
n.years <- length(hm)
for (i in 1:n.years){
   fitted[i] <- mean(hm.ssm$sims.list$N.est[,i])
   lower[i] <- quantile(hm.ssm$sims.list$N.est[,i], 0.025)
   upper[i] <- quantile(hm.ssm$sims.list$N.est[,i], 0.975)}
m1 <- min(c(fitted, hm, lower), na.rm = TRUE)
m2 <- max(c(fitted, hm, upper), na.rm = TRUE)
par(mar = c(4.5, 4, 1, 1))
plot(0, 0, ylim = c(m1, m2), xlim = c(1, n.years), ylab = "Population
   size", xlab = "Year", col = "black", type = "l", lwd = 2, axes = FALSE,
   frame = FALSE)
axis(2, las = 1)
axis(1, at = 1:n.years, labels = year)
polygon(x = c(1:n.years, n.years:1), y = c(lower, upper[n.years:1]),
   col = "gray90", border = "gray90")
points(hm, type = "l", col = "black", lwd = 2)
points(fitted, type = "l", col = "blue", lwd = 2)
legend(x = 1, y = 150, legend = c("Counts", "Estimates"), lty = c(1, 1),
   lwd = c(2, 2), col = c("black", "blue"), bty = "n", cex = 1)
```

Counts and estimated population sizes are quite close (Fig. 5.7), which is reflected in a small estimate of the observation variance. The overall trajectory of the house martin population is negative, but with important annual fluctuations. This can be seen in Fig. 5.7, but is apparently not reflected in the estimate of the process variance, which is only about double of the observation variance. Is there something wrong? No, there is nothing wrong; we just can't compare variances directly if they are measured around different means. A better way to compare variances is to calculate the coefficient of variation (CV), which is the ratio of the
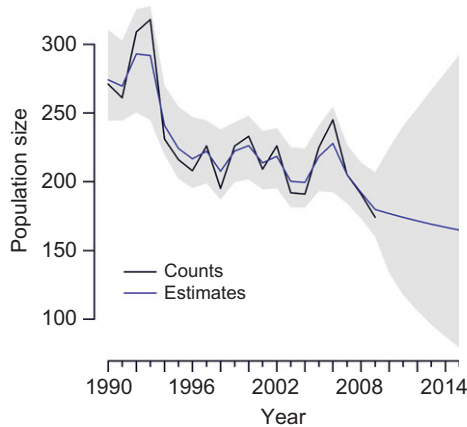
FIGURE 5.7   Counts (black) and estimated population size (blue) of house martins in Magden (with 95% CRI shaded).

standard deviation to the mean. The average population size over the 20 study years is about 230 pairs, thus the CV of the observation error is $\frac{\sqrt{0.006}}{\log(230)} = 0.014$. The CV of the process variation is $\frac{\sqrt{0.012}}{0.022} = 4.98$, and thus about 350 times larger than the CV of the observation error.

The projected population sizes after 2009 are based on the estimates of the population size in 2009, the average growth rate of the population, and the process variance. On average, we predict a steadily declining population, but the uncertainty in the estimated population trajectory is large (as reflected by the wide CRIs). This is mostly because of the large process variance, so it is difficult to make precise predictions. Typically, the uncertainty becomes larger the further we predict into the future. The predicted trajectory is the Bayesian analog of what Lande et al. (2003) call a population prediction interval (PPI). However, it is not based on the bootstrap and the usual approximations involved with maximum likelihood. Such a prediction with full uncertainty assessment is easy to obtain in a Bayesian framework of inference, but is more challenging in the frequentist arena.

We were also asking about the probability that the house martin population in 2015 would be smaller than in 2009. This quantity is again a derived variable and can easily be obtained from the MCMC output. We evaluate the proportion of MCMC samples of the estimated population size in 2015 that is smaller than in 2009. The probability that the population size in 2015 is smaller than in 2009 is 0.689 (code below), the probability that it would be larger is 0.311. Hence, a decline of the house martin population until 2015 is twice as likely as the inverse.

```
# Probability of N(2015) < N(2009)
mean(hm.ssm$sims.list$N.est[,26] < hm.ssm$mean$N.est[20])
```

5.5 SUMMARY AND OUTLOOK

We have introduced classical state-space models of population dynamics, which are used to analyze population counts and to partition the observed variation into a component due to process variation and another due to observation error. We have seen that these models can be useful in some situations, but it must be kept in mind that they usually do not provide an unbiased estimate of the true population size. Instead, they yield a smoothed estimate of a population index (i.e., of Np), which may or may not be unbiased with respect to the true population trajectory. In the absence of temporal patterns in the observation error, the trajectories of the true population sizes and that of the estimated population index will be parallel. The smoothing occurs with respect to the assumed underlying biological processes, and is therefore more mechanistic and perhaps more realistic than the arbitrary smoothing of generalized additive models (GAM; King et al., 2010), which are often used to analyze population counts (Fewster et al., 2000).

We have considered only very simple state-space models to describe state and observation processes. The beauty of these models is that we can readily change them to be more realistic or in a way that specific hypotheses can be tested. An obvious extension is to use a density-dependent model such as the logistic, theta-logistic, or Gompertz model (e.g., Lande et al., 2003; Dennis et al., 2006). However, it appears that models with density-dependence have serious extrinsic identifiability problems. With increasing magnitude of the observation error, the parameters become harder to estimate, especially the parameter which quantifies the strength of density-dependence (Knape, 2008).

If our aim is to get unbiased estimates of the population size, the use of state-space models of the type featured in this chapter is not sufficient. In other words, there is no way around explicitly estimating detection probability. In Chapters 6 (for a single closed population), 10 (for a single open population), and 12 (for metapopulations), we will see how this can be done. This requires additional information to tease apart detection probability and the true levels of the state process (population size). However, when no such additional data are available about the detection process, the state-space models in this chapter represent probably one of the best frameworks for inference about population trajectories. In Chapters 7–10, we will use different, more mechanistic state-space models to estimate survival and other demographic quantities from capture–recapture, mark-recovery, and multistate capture–recapture data, and in Chapter 13 for metapopulation dynamics.

5.6 EXERCISES

1. Random variability in detection probability: quite often we cannot assume that detection probability is constant over time, for example, because of weather factors that affect the counts. Simulate and analyze

data for a population, whose size remains constant at 50 individuals over (a) 25 years and (b) 50 years, but where the annual detection probability varies randomly in the interval between 0.3 and 0.7. Does the state-space model perform well in this situation?

2. Modeling of variance structures: in the house martin data set, we saw that from year $t = 9$ onwards, a different data-collection protocol (questionnaires) was used. Adapt the model to account for possibly different observation errors in the two periods.

3. Unstructured and dynamic hierarchical model for population counts: In Section 3.3.2, we encountered a different two-level hierarchical model for a single time series of population counts. What is the difference to a state-space model in this chapter? Fit the exponential population state-space model to the peregrine data from Section 3.3.2 and compare the inference about the population trajectory under the two models. In addition, construct a model with a linear trend in the population growth rate and another model with a linear trend in the observation error and fit them to the peregrine data.