CHAPTER

8

Estimation of Survival Using Mark-Recovery Data

OUTLINE

8.1 INTRODUCTION

Data on marked individuals that are reencountered alive formed the basis for the estimation of survival probabilities in the CJS models in Chapter 7. Here, we deal with another kind of data provided by marked individuals that serve the same goal: mark-recovery data. These data arise from animals that die and whose mark (typically a ring, but other marks are also possible) is recovered. As not all dead individuals will be found,

Bayesian Population Analysis using WinBUGS
DOI: 10.1016/B978-0-12-387020-9.00008-0

the probability that a marked, dead individual is found (the recovery probability) must be estimated in addition to the survival probability. Thus, the dead-recovery probability takes the place of the live detection probability in the CJS models.

The sampling protocol is as follows: a number of individuals are marked, preferably over several years or other defined periods, and information is collected about the time of death from the marked individuals, which typically comes from members of the public, who find and report dead individuals. For each reported individual, we know the place and year of death and thus can compute its longevity. For all marked individuals that are not found or reported, we do not know when they die. However, the number of marked individuals that are never recovered is known and can be expressed as a function of the same survival and recovery probabilities that act on the recovered individuals. Hence, individuals that are not recovered contribute information about survival as well.

The unknown parameters are the survival probability (s) and the recovery probability (r). In theory, dead individuals can be found anywhere, not only in the study area. Hence, this survival probability is the true, rather than the apparent survival probability (ϕ) of the capture–recapture models in Chapter 7, where reencounters are usually restricted to fairly small areas. Apparent and true survival are linked by the fidelity probability (F, probability to remain within the study area), that is, $\phi = sF$. The recovery probability can be decomposed into a series of conditional binary events, such as the probability that an individual is found and the probability that the mark is reported. Sometimes, it is useful to consider them separately, resulting in a different parameterization of the mark-recovery model (Brownie et al., 1985).

Mark-recovery models are sometimes called dead-recovery or band-recovery models. They are primarily used for birds and in particular for hunted species because these have a much higher recovery probability than nongame species. The first treatise on these models was the handbook of Brownie et al. (1985), and a recent overview is provided in Williams et al. (2002). The Bayesian approach to these models is presented by Brooks et al. (2000a, 2000b, 2002), Barry et al. (2003), and Gimenez et al. (2007).

Here, we first show how the mark-recovery model can be fitted with a state-space formulation, the advantage of which is that we can apply exactly the same concepts (modeling along the time and individual axes; fixed and random effects) as introduced in Chapters 6 and 7; thus, we have great flexibility in modeling. However, we will not repeat these concepts here, but we just stress once more that all the key features introduced so far can be combined in a creative way. Second, we show the multinomial likelihood, which again enjoys great benefits in terms of computational efficiency.

8.2 THE MARK-RECOVERY MODEL AS A STATE-SPACE MODEL

Let us assume a newly marked individual at time t. It may survive until time $t+1$ with probability s_t ($t = 1, \ldots, T-1$; T being the number of occasions). Conceptually, we can imagine the individual tossing a coin to determine whether it survives (with probability s_t) or dies (with probability $1 - s_t$). Given that the individual is still alive at time $t+1$, it may again survive until $t+2$ with probability s_{t+1}. This is continued until either the individual is dead or the study ends. Clearly, once an individual is dead, its further fate is no longer a stochastic event, that is, it will remain dead with probability 1. What we have just described is the state process, that is, the possible states of an individual over time: dead or alive. As you may have noticed, this is exactly the same state process as in CJS models. The sole difference is a slightly different definition of the survival parameter: we use true survival in the ring-recovery model, whereas in the CJS model, it is the probability of surviving *and* remaining in the sampling area. The observation process for the mark-recovery model is different from the CJS model because only individuals that have just died can be observed (with probability r_t). Once an individual has been dead for a while, it cannot be recovered anymore because it has typically decayed. Another assumption of the model is that the time of death of a recovered individual is known to the accuracy of the temporal interval of the occasions (typically 1 year). In Fig. 8.1, the state and the observation processes are shown graphically.

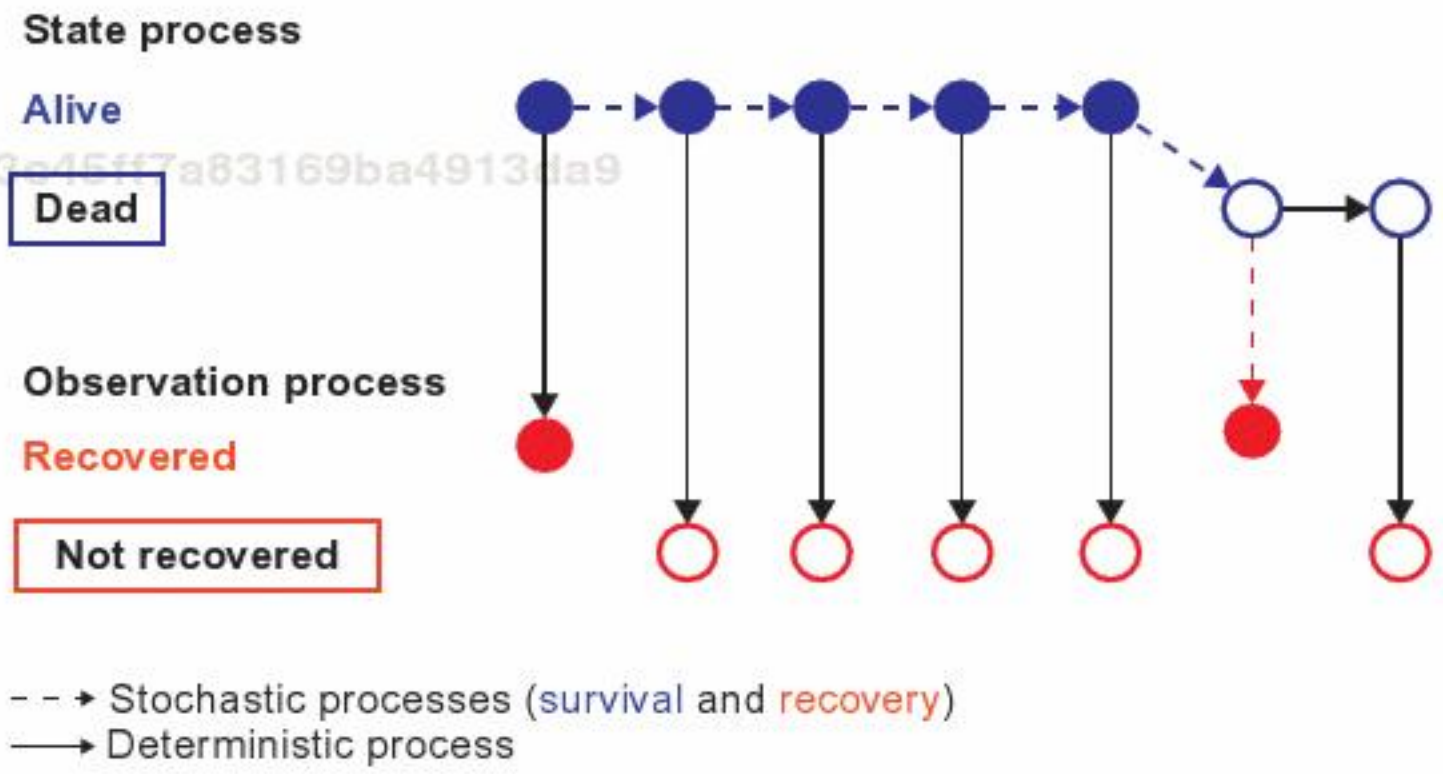

FIGURE 8.1 Example of the state and observation process of a marked individual over time in the mark-recovery model. The sequence of true states in this individual is $z = [1, 1, 1, 1, 1, 0, 0]$, and the observed capture-history is $y = [1, 0, 0, 0, 0, 1, 0]$. The occasion at which the individual is marked produces a "recovery". This is just a convention to ensure that each capture-history starts with a 1.

The analysis of mark-recovery data with the state-space formulation is very similar to that of the CJS model. Again, we define the latent variable $z_{i,t}$, which takes value 1 if individual i is alive at time t and value 0 if it is dead. Thus, $z_{i,t}$ denotes the true state of individual i at time t. We also define the vector f_i that contains the occasion at which individual i is marked (its first encounter). The state of individual i at first encounter z_{i,f_i} is 1 with probability 1, that is, the individual is alive with certainty. The states at subsequent occasions are Bernoulli trials: conditional on being alive at occasion t, individual i survives until occasion $t+1$ with probability $s_{i,t}$. The following two equations define the state process:

$$z_{i,f_i} = 1$$

$$z_{i,t+1} \,|\, z_{i,t} \sim \text{Bernoulli}(z_{i,t} s_{i,t}).$$

The parameter of the Bernoulli trial is the product of the survival probability and the state variable z. The inclusion of z ensures that dead individuals (with $z = 0$) remain dead.

Given that individual i is "recently dead" at occasion t (i.e., has died between $t-1$ and t), it may be recovered with probability $r_{i,t}$ ($t = 2, \ldots, T$). The recovery, or observation, process is modeled as another Bernoulli trial with success probability $r_{i,t}$:

$$y_{i,t} \,|\, z_{i,t}, z_{i,t-1} \sim \text{Bernoulli}((z_{i,t-1} - z_{i,t}) r_{i,t}).$$

The inclusion of the difference between the two successive values of latent variable z ensures that only individuals that are "recently dead" can be recovered. Matrix **y** contains the observed data, that is, the capture-histories. The state and the observation processes are both defined for $t \geq f_i$. The implementation of the model in the BUGS language now only requires the translation of these equations.

The mark-recovery model makes a number of assumptions that must be met to ensure unbiased parameter estimators. They are similar to those made in the CJS model. The survival and recovery probabilities are required to be identical for all individuals in the same group, cohort, or age class, and the fates of individuals must be independent. These assumptions can be tested with appropriate goodness-of-fit tests (Brownie et al., 1985). Furthermore, individuals must be reported without error (no misreading of the marks), and no mark loss is allowed.

8.2.1 Simulation of Mark-Recovery Data

We mimic a study of common terns (Fig. 8.2). Common terns live along rivers or coasts and breed in colonies, preferably on small islands. Adults are captured and marked, and some of them are found dead on migration

FIGURE 8.2 Common terns (*Sterna hirundo*), Finland, 2004 (Photograph by J. Peltomäki).

or at the breeding grounds. We assume that over 14 years, we mark 50 adults every year. We assume annual adult survival of 0.8 and a recovery probability of 0.2. The following R code simulates a mark-recovery matrix. The initial capture and the recovery event are both denoted with 1. The mark-recovery data matrix has 15 columns because common terns can still be recovered after the last year of marking.

```
# Define parameter values
n.occasions <- 14                    # Number of release occasions
marked <- rep(50, n.occasions)       # Annual number of marked individuals
s <- rep(0.8, n.occasions)
r <- rep(0.2, n.occasions)

# Define matrices with survival and recovery probabilities
S <- matrix(s, ncol = n.occasions, nrow = sum(marked))
R <- matrix(r, ncol = n.occasions, nrow = sum(marked))

# Define function to simulate mark-recovery data
simul.mr <- function(S, R, marked){
    n.occasions <- dim(S)[2]
    MR <- matrix(NA, ncol = n.occasions+1, nrow = sum(marked))
    # Define a vector with the occasion of marking
    mark.occ <- rep(1:n.occasions, marked)
    # Fill the CH matrix
    for (i in 1:sum(marked)){
        MR[i, mark.occ[i]] <- 1     # Write an 1 at the release occasion
        for (t in mark.occ[i]:n.occasions){
```

```
            # Bernoulli trial: has individual survived occasion?
            sur <- rbinom(1, 1, S[i,t])
            if (sur==1) next        # If still alive, move to next
                                      occasion
            # Bernoulli trial: has dead individual been recovered?
            rp <- rbinom(1, 1, R[i,t])
            if (rp==0){
              MR[i,t+1] <- 0
              break
              }
            if (rp==1){
              MR[i,t+1] <- 1
              break
              }
        } #t
    } #i
        # Replace the NA in the file by 0
        MR[which(is.na(MR))] <- 0
        return(MR)
    }

# Execute function
MR <- simul.mr(S, R, marked)
```

8.2.2 Analysis of a Model with Constant Parameters

To fit the dead-recovery model, we also need a vector indicating the occasion of marking for each individual.

```
# Create vector with occasion of marking
get.first <- function(x) min(which(x!=0))
f <- apply(MR, 1, get.first)
```

We define the model and run the analysis.

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

# Priors and constraints
for (i in 1:nind){
    for (t in f[i]:(n.occasions-1)){
        s[i,t] <- mean.s
        r[i,t] <- mean.r
        } #t
    } #i
mean.s ~ dunif(0, 1)          # Prior for mean survival
mean.r ~ dunif(0, 1)          # Prior for mean recapture

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

```
    # Define latent state at first capture
    z[i,f[i]] <- 1
    for (t in (f[i]+1):n.occasions){
        # State process
        z[i,t] ~ dbern(mu1[i,t])
        mu1[i,t] <- s[i,t-1] * z[i,t-1]
        # Observation process
        y[i,t] ~ dbern(mu2[i,t])
        mu2[i,t] <- r[i,t-1] * (z[i,t-1] - z[i,t])
        } #t
    } #i
}
",fill = TRUE)
sink()
```

For parameter estimation, we can either use the dead-recovery data matrix only or in addition provide information about the partly known latent state variable z (see Section 7.3.1). The latter results in faster computation and quicker convergence. In the dead-recovery model, the latent state variable is unknown for all individuals that are never recovered dead. However, for those that are recovered dead, the latent state is 1 for all occasions between marking and just prior to the recovery occasion and is 0 afterwards. The following function creates the known state variables from the dead-recovery matrix.

```
# Define function to create a matrix with information about known latent
   state z
known.state.mr <- function(mr){
    state <- matrix(NA, nrow = dim(mr)[1], ncol = dim(mr)[2])
    rec <- which(rowSums(mr)==2)
    for (i in 1:length(rec)){
        n1 <- min(which(mr[rec[i],]==1))
        n2 <- max(which(mr[rec[i],]==1))
        state[rec[i],n1:n2] <- 1
        state[rec[i],n1] <- NA
        state[rec[i],n2:dim(mr)[2]] <- 0
        }
    return(state)
    }

# Bundle data
bugs.data <- list(y = MR, f = f, nind = dim(MR)[1], n.occasions =
   dim(MR)[2], z = known.state.mr(MR))

# Define function to create a matrix of initial values for latent
   state z
mr.init.z <- function(mr){
    ch <- matrix(NA, nrow = dim(mr)[1], ncol = dim(mr)[2])
    rec <- which(rowSums(mr)==1)
    for (i in 1:length(rec)){
```

```
        n1 <- which(mr[rec[i],]==1)
        ch[rec[i],n1:dim(mr)[2]] <- 0
        ch[rec[i],n1] <- NA
        }
    return(ch)
    }

# Initial values
inits <- function(){list(z = mr.init.z(MR), mean.s = runif(1, 0, 1),
   mean.r = runif(1, 0, 1))}

# Parameters monitored
parameters <- c("mean.s", "mean.r")

# MCMC settings
ni <- 5000
nt <- 6
nb <- 2000
nc <- 3

# Call WinBUGS from R (BRT 4 min)
mr.ss <- bugs(bugs.data, inits, parameters, "mr.ss.bug", n.chains =
   nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
   bugs.directory = bugs.dir)
```

The posterior means obtained with the state-space likelihood are nearly identical to those obtained with the multinomial likelihood (see Section 8.3). However, the state-space likelihood is computationally more demanding, and we need longer chains to achieve convergence.

```
print(mr.ss, digits = 3)
         mean     sd   2.5%    25%    50%    75%  97.5%   Rhat  n.eff
mean.s  0.754  0.027  0.699  0.737  0.755  0.773  0.802  1.008    400
mean.r  0.190  0.019  0.157  0.177  0.190  0.202  0.230  1.000   1500
```

As mentioned already, we can apply all the modeling encountered in the CJS within the framework of this basic model (i.e., the likelihood part). The changes required in the BUGS code are restricted to the model section entitled "Priors and constraints".

8.3 THE MARK-RECOVERY MODEL FITTED WITH THE MULTINOMIAL LIKELIHOOD

8.3.1 Constant Parameters

The multinomial likelihood is the classical way to analyze mark-recovery data (Brownie et al., 1985; Williams et al., 2002). For this, the mark-recovery data are first summarized in the m-array (see Section 7.10). In this matrix, rows represent the release (marking) years, and the columns represent recovery years. The only difference between the mark-recovery and the

capture–recapture m-array is that in the former, all individuals that are released in a year (i.e., form a release cohort) remain in the same row of the matrix because they can be reencountered at most once; they are not released after recovery (difficult if they are dead …). In the capture–recapture m-array, some individuals are released again in subsequent occasions and thus appear in more than one row of the matrix.

The following R code produces the necessary m-array for the mark-recovery data.

```
# Define function to create an m-array based for mark-recovery (MR) data
marray.dead <- function(MR){
    nind <- dim(MR)[1]
    n.occasions <- dim(MR)[2]
    m.array <- matrix(data = 0, ncol = n.occasions+1, nrow =
       n.occasions)
    # Create vector with occasion of marking
    get.first <- function(x) min(which(x!=0))
    f <- apply(MR, 1, get.first)
    # Calculate the number of released individuals at each time period
    first <- as.numeric(table(f))
    for (t in 1:n.occasions){
        m.array[t,1] <- first[t]
        }
    # Fill m-array with recovered individuals
    rec.ind <- which(apply(MR, 1, sum)==2)
    rec <- numeric()
    for (i in 1:length(rec.ind)){
        d <- which(MR[rec.ind[i],(f[rec.ind[i]]+1):n.occasions]==1)
        rec[i] <- d+f[rec.ind[i]]
        m.array[f[rec.ind[i]],rec[i]] <- m.array[f[rec.ind[i]],
           rec[i]]+1
        }
    # Calculate the number of individuals that are never recovered
    for (t in 1:n.occasions){
        m.array[t,n.occasions+1] <- m.array[t,1]-sum(m.array[t,2:
           n.occasions])
        }
    out <- m.array[1:(n.occasions-1),2:(n.occasions+1)]
    return(out)
    }
```

We produce the m-array for the simulated data by executing the function:

```
marr <- marray.dead(MR)
```

This m-array contains the observed response. Each row is modeled as a multinomial trial with index equal to the cohort size, that is, the number of individuals released in that year. The multinomial cell probabilities are functions of the parameters for survival and recovery (s and r).

We first fit a model with constant parameters over time, but we have introduced an index of time (subscript) here in order to understand the model better. The cell probabilities are as follows:

Released at Occasion	Recovered at Occasion 2	3	4	Never Recovered
1	$(1-s_1)r_1$	$s_1(1-s_2)r_2$	$s_1s_2(1-s_3)r_3$	$(1-s_1)(1-r_1) + s_1(1-s_2)(1-r_2) + s_1s_2(1-s_3)(1-r_3) + s_1s_2s_3 =$ 1 – Σ (Released at Occasion 1)
2	0	$(1-s_2)r_2$	$s_2(1-s_3)r_3$	$(1-s_2)(1-r_2) + s_2(1-s_3)(1-r_3) + s_2s_3 =$ 1 – Σ (Released at Occasion 2)
3	0	0	$(1-s_3)r_3$	$(1-s_3)(1-r_3) + s_3 =$ 1 – Σ (Released at Occasion 3)

To fit this model in BUGS, we essentially have to define these cell probabilities. The probability for individuals never recovered may look complicated but is simply 1 minus the sum of the other probabilities in each row.

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

# Priors and constraints
for (t in 1:n.occasions){
    s[t] <- mean.s
    r[t] <- mean.r
    }
mean.s ~ dunif(0, 1)            # Prior for mean survival
mean.r ~ dunif(0, 1)            # Prior for mean recovery

# Define the multinomial likelihood
for (t in 1:n.occasions){
    marr[t,1:(n.occasions+1)] ~ dmulti(pr[t,], rel[t])
    }

# Calculate the number of birds released each year
for (t in 1:n.occasions){
    rel[t] <- sum(marr[t,])
    }

# Define the cell probabilities of the m-array
# Main diagonal
for (t in 1:n.occasions){
    pr[t,t] <- (1-s[t])*r[t]
    # Above main diagonal
    for (j in (t+1):n.occasions){
        pr[t,j] <- prod(s[t:(j-1)])*(1-s[j])*r[j]
        } #j
```

```
    # Below main diagonal
    for (j in 1:(t-1)){
        pr[t,j] <- 0
        } #j
    } #t
# Last column: probability of non-recovery
for (t in 1:n.occasions){
    pr[t,n.occasions+1] <- 1-sum(pr[t,1:n.occasions])
    } #t
}
",fill = TRUE)
sink()
```

Now, we run the mark-recovery model using the m-array data.

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

# Initial values
inits<- function(){list(mean.s = runif(1, 0, 1), mean.r = runif(1, 0, 1))}

# Parameters monitored
parameters <- c("mean.s", "mean.r")

# MCMC settings
ni <- 5000
nt <- 6
nb <- 2000
nc <- 3

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

# Summarize posteriors
print(mr, digits = 3)
```



```
          mean    sd   2.5%    25%    50%    75%  97.5%   Rhat  n.eff
mean.s   0.758 0.033  0.691  0.736  0.757  0.782  0.823  1.003    910
mean.r   0.192 0.020  0.156  0.179  0.191  0.204  0.236  1.001   1500
```

Convergence of the chains is not difficult to achieve. The posterior distributions of both parameters include the input values of 0.8 and 0.2 (Fig. 8.3).

```
par(mfrow = c(1, 2), las = 1)
hist(mr$sims.list$mean.s, nclass = 25, col = "gray", main = "", ylab =
   "Frequency", xlab = "Survival probability")
abline(v = 0.8, col = "red", lwd = 2)
hist(mr$sims.list$mean.r, nclass = 25, col = "gray", main = "", ylab =
   "", xlab = "Recovery probability")
abline(v = 0.2, col = "red", lwd = 2)
```

It is straightforward to build more complicated models, for example, with different groups, or fixed or random temporal variation. The BUGS

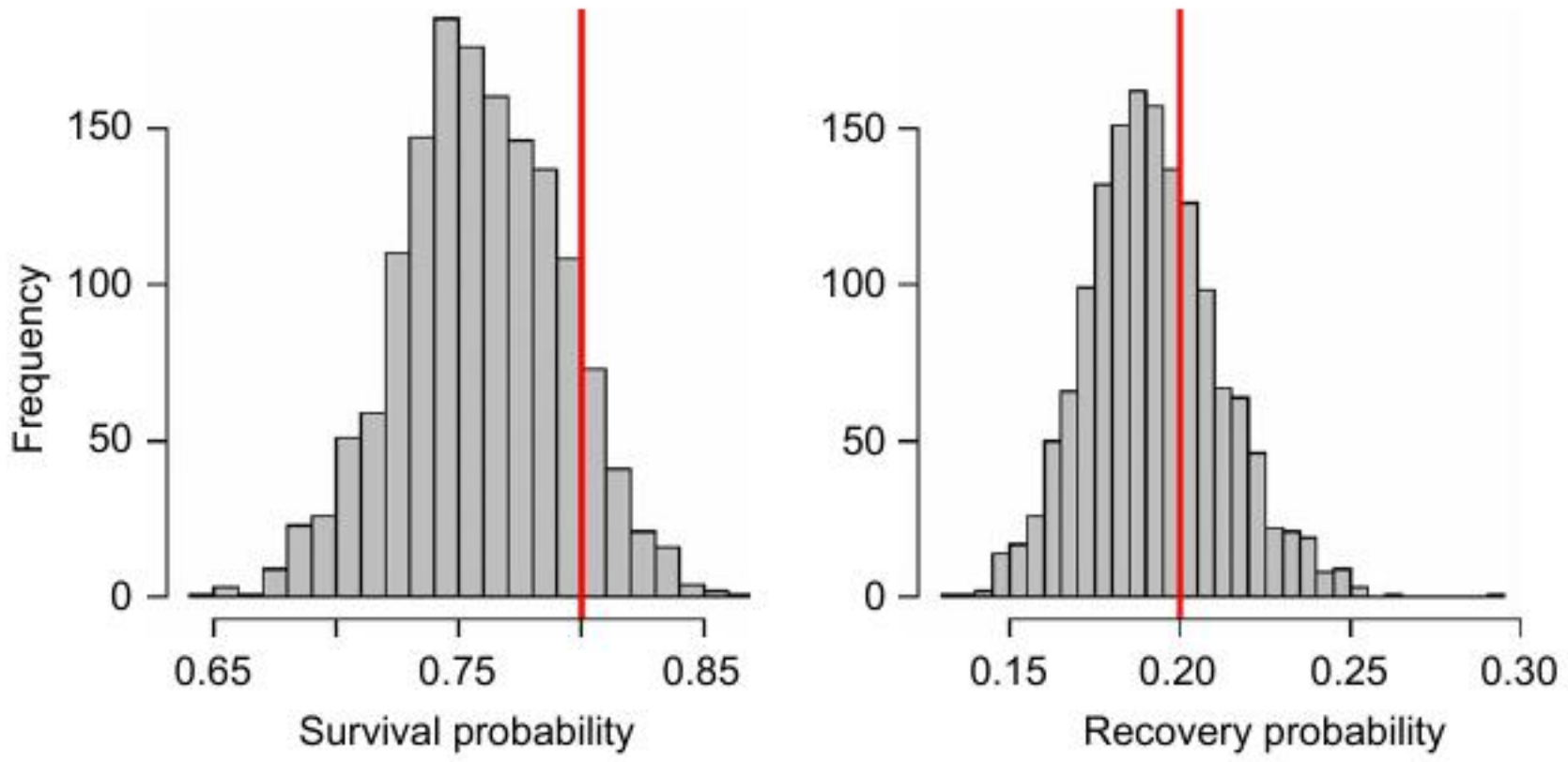

FIGURE 8.3 Posterior distributions of survival and recovery probabilities in the analysis of the simulated data (red line: values used for simulating the data).

code just has to be adapted in exactly the same way as explained for the capture–recapture models (Chapter 7). Basically, the only changes are made in the model section entitled "Priors and constraints".

8.3.2 Age-Dependent Parameters

Survival typically changes with age; hence, age-dependent models are often important. Sometimes, even recovery probability may change with age. Age-dependent recovery probabilities can occur if the main mortality cause changes with age, and different mortality causes are associated with differential recovery probabilities. For example, individuals dying from human-related causes are more likely found by humans than individuals dying from natural causes. As an example, the recovery probability of white storks (*Ciconia ciconia*) dying from power line collisions is significantly higher than for other causes of mortality (Schaub and Pradel, 2004).

We stay with our common tern example, and assume that young (nestlings) and adults are marked. We assume survival to be 0.3 for juveniles and 0.8 for adults. The recovery probabilities for juveniles and adults are 0.25 and 0.15, respectively. We simulate two data sets: one for individuals marked as young and another for individuals marked as adults. We summarize both data sets independently as an m-array.

```
n.occasions <- 15                    # Number of occasions
marked.j <- rep(200, n.occasions)    # Annual number of newly marked
                                       young
marked.a <- rep(20, n.occasions)     # Annual number of newly marked
                                       adults
sjuv <- 0.3                          # Juvenile survival probability
sad <- 0.8                           # Adult survival probability
```

```
rjuv <- 0.25                                  # Juvenile recovery probability
rad <- 0.15                                   # Adult recovery probability
sj <- c(sjuv, rep(sad, n.occasions-1))
rj <- c(rjuv, rep(rad, n.occasions-1))

# Define matrices with survival and recovery probabilities
SJ <- matrix(0, ncol = n.occasions, nrow = sum(marked.j))
for (i in 1:length(marked.j)){
    SJ[(sum(marked.j[1:i])-marked.j[i]+1):sum(marked.j[1:i]),
       i:n.occasions] <- matrix(rep(sj[1:(n.occasions-i+1)],
       marked.j[i]), ncol = n.occasions-i+1, byrow = TRUE)
    }
SA <- matrix(sad, ncol = n.occasions, nrow = sum(marked.a))
RJ <- matrix(0, ncol = n.occasions, nrow = sum(marked.j))
for (i in 1:length(marked.j)){
    RJ[(sum(marked.j[1:i])-marked.j[i]+1):sum(marked.j[1:i]),
       i:n.occasions] <- matrix(rep(rj[1:(n.occasions-i+1)],
       marked.j[i]), ncol = n.occasions-i+1, byrow = TRUE)
    }
RA <- matrix(rad, ncol = n.occasions, nrow = sum(marked.a))

# Execute simulation function
MRj <- simul.mr(SJ, RJ, marked.j)
MRa <- simul.mr(SA, RA, marked.a)

# Summarize data in m-arrays
marr.j <- marray.dead(MRj)
marr.a <- marray.dead(MRa)
```

The cell probabilities in the modeling of the m-array for adults are exactly the same as before except for s_i and r_i, which we now denote sa_i and ra_i, respectively. In contrast, the cell probabilities for the model of individuals marked as nestlings differ: for them, we have to include an age structure for the survival and recovery probabilities. These cell probabilities are as follows:

Released at Occasion	Recovered at Occasion 2	3	4	Never Recovered
1	$(1-sj_1)rj_1$	$sj_1(1-sa_2)ra_2$	$sj_1sa_2(1-sa_3)ra_3$	$1-\Sigma$ (Released at Occasion 1)
2	0	$(1-sj_2)rj_2$	$sj_2(1-sa_3)ra_3$	$1-\Sigma$ (Released at Occasion 2)
3	0	0	$(1-sj_3)rj_3$	$1-\Sigma$ (Released at Occasion 3)

In the model, we define the probabilities of both m-arrays separately, with some parameters (*sa* and *ra*) shared.

```
# Specify model in BUGS language
sink("mr-mnl-age.bug")
cat("
model {
```

```
# Priors and constraints
for (t in 1:n.occasions){
    sj[t] <- mean.sj
    sa[t] <- mean.sa
    rj[t] <- mean.rj
    ra[t] <- mean.ra
    }
mean.sj ~ dunif(0, 1)              # Prior for mean juv.survival
mean.sa ~ dunif(0, 1)              # Prior for mean ad.survival
mean.rj ~ dunif(0, 1)              # Prior for mean juv.recovery
mean.ra ~ dunif(0, 1)              # Prior for mean ad.recovery

# Define the multinomial likelihoods
for (t in 1:n.occasions){
    marr.j[t,1:(n.occasions+1)] ~ dmulti(pr.j[t,], rel.j[t])
    marr.a[t,1:(n.occasions+1)] ~ dmulti(pr.a[t,], rel.a[t])
    }

# Calculate the number of birds released each year
for (t in 1:n.occasions){
    rel.j[t] <- sum(marr.j[t,])
    rel.a[t] <- sum(marr.a[t,])
    }

# Define the cell probabilities of the juvenile m-array
# Main diagonal
for (t in 1:n.occasions){
    pr.j[t,t] <- (1-sj[t])*rj[t]
    # Further above main diagonal
    for (j in (t+2):n.occasions){
        pr.j[t,j] <- sj[t]*prod(sa[(t+1):(j-1)])*(1-sa[j])*ra[j]
        } #j
    # Below main diagonal
    for (j in 1:(t-1)){
        pr.j[t,j] <- 0
        } #j
    } #t
for (t in 1:(n.occasions-1)){
    # One above main diagonal
    pr.j[t,t+1] <- sj[t]*(1-sa[t+1])*ra[t+1]
    } #t
# Last column: probability of non-recovery
for (t in 1:n.occasions){
    pr.j[t,n.occasions+1] <- 1-sum(pr.j[t,1:n.occasions])
    } #t

# Define the cell probabilities of the adult m-array
# Main diagonal
for (t in 1:n.occasions){
    pr.a[t,t] <- (1-sa[t])*ra[t]
    # Above main diagonal
    for (j in (t+1):n.occasions){
        pr.a[t,j] <- prod(sa[t:(j-1)])*(1-sa[j])*ra[j]
        } #j
```

```
    # Below main diagonal
    for (j in 1:(t-1)){
        pr.a[t,j] <- 0
        } #j
    } #t
# Last column: probability of non-recovery
for (t in 1:n.occasions){
    pr.a[t,n.occasions+1] <- 1-sum(pr.a[t,1:n.occasions])
    } #t
}
",fill = TRUE)
sink()
```

We prepare the remainder of the analysis and run the mark-recovery model.

```
# Bundle data
bugs.data <- list(marr.j = marr.j, marr.a = marr.a, n.occasions =
   dim(marr.j)[2]-1)

# Initial values
inits <- function(){list(mean.sj = runif(1, 0, 1), mean.sa =
   runif(1, 0, 1), mean.rj = runif(1, 0, 1), mean.ra = runif(1, 0, 1))}

# Parameters monitored
parameters <- c("mean.sj", "mean.rj", "mean.sa", "mean.ra")

# MCMC settings
ni <- 5000
nt <- 6
nb <- 2000
nc <- 3

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

As before, convergence is achieved fairly quickly.

```
print(mr.age, digits = 3)
          mean     sd   2.5%    25%    50%    75%  97.5%   Rhat  n.eff
mean.sj  0.289  0.051  0.205  0.252  0.283  0.319  0.402  1.000   1500
mean.rj  0.254  0.022  0.219  0.239  0.250  0.265  0.304  1.001   1500
mean.sa  0.797  0.025  0.748  0.779  0.797  0.814  0.848  1.000   1500
mean.ra  0.192  0.029  0.140  0.171  0.190  0.210  0.251  1.000   1500
```

8.4 REAL-DATA EXAMPLE: AGE-DEPENDENT SURVIVAL IN SWISS RED KITES

The red kite (Fig. 8.4) is a large bird of prey that is declining in many parts of Europe but widespread in the Swiss lowlands. During the past 50 years, 1480 nestlings and 152 adults (>2 years old) have been marked,

FIGURE 8.4 Red kite (*Milvus milvus*), Switzerland (Photograph by P. Keusch).

of which 107 individuals were recovered dead. Our interest was to estimate age-specific survival probabilities; we were not interested in any changes over time. Therefore, we considered all individuals marked at the same age as a single release cohort. Here, we have two age classes at marking: juveniles and adults. Since we drop the time index, the resulting m-array of both age classes consists only of a single line, with "columns" referring to the age at which individuals are recovered. Strictly speaking, only in the m-array for individuals marked as juveniles do the columns correspond to true age because in these individuals true age is known. In the m-array of the adults, columns refer to the age since marking (in years) because the age of these individuals at marking is unknown. For example, in the data given below, we see that 1388 of all individuals marked as juveniles are never found, whereas five individuals are found dead in their third year of life. Here are the data for the individuals marked as juveniles and those marked as adults.

```
marray.juv <- c(42, 18, 5, 7, 4, 3, 2, 1, 2, 2, 1, 0, 1, 3, 0, 0, 1, 1388)
marray.ad <- c(3, 1, 1, 3, 0, 2, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 137)
```

In long-lived birds, survival usually varies with age. Here, we fit a survival model with three age classes. The first age class (juveniles) covers one year from fledging, the second age class (subadults) also covers one

year from one to two years of age, and the last age class (adults) contains all ages 2 years and greater. For recovery probability, we only assume two age classes and distinguish the first year after fledging from all ages thereafter. Juvenile survival in mark-recovery models is sometimes difficult to estimate (see discussion below), and therefore, we will also try to utilize a priori information on juvenile survival. Few estimates of juvenile survival in red kites are available, and most of them were obtained using dated methods, that is, ignoring imperfect detection. On average, juvenile survival is likely to be about 0.6 but with a large uncertainty (Aebischer, 2009). We translated this knowledge into a beta prior distribution with parameters 4.2 and 2.8. This ensured a mean of 0.6 and a relatively wide spread of the prior (standard deviation ≅ 0.173). To assess prior sensitivity, we also fitted a model with a flat uniform prior (U(0,1)). The BUGS code above needs some adaptations because we now have three age classes, but only two release cohorts: juveniles and adults.

```
# Specify model in BUGS language
sink("mr-mnl-age3.bug")
cat("
model {

# Priors and constraints
sjuv ~ dbeta(4.2, 2.8)     # Informative prior for juv. survival:
                             Analysis A
#sjuv ~ dunif(0, 1)        # Non-informative for juv. survival prior:
                             Analysis B
ssub ~ dunif(0, 1)         # Prior for subad. survival
sad ~ dunif(0, 1)          # Prior for ad. survival
rjuv ~ dunif(0, 1)         # Prior for juv. recovery
rad ~ dunif(0, 1)          # Prior for ad. recovery

# Define the multinomial likelihoods
marr.j[1:(n.age+1)] ~ dmulti(pr.j[], rel.j)
marr.a[1:(n.age+1)] ~ dmulti(pr.a[], rel.a)

# Calculate the number of birds released each year
rel.j <- sum(marr.j[])
rel.a <- sum(marr.a[])

# Define the cell probabilities of the juvenile m-array
# First element
pr.j[1] <- (1-sjuv)*rjuv
# Second element
pr.j[2] <- sjuv*(1-ssub)*rad
# Third and further elements
for (t in 3:n.age){
    pr.j[t] <- sjuv*ssub*pow(sad,(t-3))*(1-sad)*rad
    }
# Probability of non-recovery
pr.j[n.age+1] <- 1    sum(pr.j[1:n.age])
```

```
# Define the cell probabilities of the adult m-array
# All elements
for (t in 1:n.age){
    pr.a[t] <- pow(sad,(t-1))*(1-sad)*rad
    }
# Probability of non-recovery
pr.a[n.age+1] <- 1-sum(pr.a[1:n.age])
}
",fill = TRUE)
sink()

# Bundle data
bugs.data <- list(marr.j = marray.juv, marr.a = marray.ad, n.age =
   length(marray.juv)-1)

# Initial values
inits <- function(){list(sjuv = runif(1, 0, 1), ssub = runif(1, 0, 1),
   sad = runif(1, 0, 1), rjuv = runif(1, 0, 1), rad = runif(1, 0, 1))}

# Parameters monitored
parameters <- c("sjuv", "ssub", "sad", "rjuv", "rad")

# MCMC settings
ni <- 30000
nt <- 10
nb <- 10000
nc <- 3

# Call WinBUGS from R (BRT <1 min)
rk.ageA <- bugs(bugs.data, inits, parameters, "mr-mnl-age3.bug",
   n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug =
   TRUE, bugs.directory = bugs.dir)
```

Convergence is reached quickly. Here are the posterior summaries for the estimated parameters.

```
print(rk.ageA, digits = 3)
       mean    sd  2.5%   25%   50%   75% 97.5%  Rhat n.eff
sjuv  0.450 0.128 0.248 0.358 0.431 0.525 0.751 1.003  1100
ssub  0.655 0.067 0.521 0.611 0.658 0.702 0.780 1.001  2800
sad   0.841 0.032 0.778 0.820 0.842 0.863 0.904 1.001  4900
rjuv  0.057 0.027 0.033 0.043 0.051 0.062 0.119 1.002  1700
rad   0.090 0.024 0.051 0.073 0.088 0.104 0.143 1.002  1700
```

As shown below, parameter estimates under the model with noninformative priors are similar (for this, we repeat the analysis with one prior statement switched) to those under the model with informative priors. Two exceptions are the posterior mean of juvenile survival (lower) and the standard deviation of the juvenile recovery probability (higher). Thus, the prior information introduced for juvenile survival affected also juvenile recovery probability (and even adult recovery probability). This is not very surprising.

```
print(rk.ageB, digits = 3)
       mean     sd   2.5%    25%    50%    75%  97.5%   Rhat  n.eff
sjuv  0.406  0.139  0.216  0.309  0.378  0.473  0.773  1.003    920
ssub  0.656  0.067  0.518  0.612  0.660  0.703  0.779  1.001   6000
sad   0.842  0.032  0.779  0.820  0.842  0.864  0.906  1.001   6000
rjuv  0.057  0.060  0.031  0.040  0.047  0.057  0.129  1.001   3500
rad   0.099  0.027  0.052  0.080  0.097  0.116  0.159  1.003   1000
```

Inspection of the posterior distributions of the five parameters under the model with noninformative priors shows that subadult and adult survival and adult recovery probability are estimated precisely (Fig. 8.5). By contrast, the posterior distribution of juvenile survival is quite wide and has some of its mass extending to 1, whereas that of juvenile recovery is skewed. This does not mean that we cannot trust the estimates; it simply indicates that there is considerable uncertainty about the values of these parameters. This behavior is due to intrinsic identifiability problems. When survival and recovery probabilities are age dependent and only data of individuals marked as juveniles are available, only adult survival is identifiable (Anderson et al., 1985). Here, we have a different number of age classes for survival and recovery, and we included also data on individuals marked as adults; thus, the parameters in the model are intrinsically identifiable. Yet, from the posterior distributions of juvenile survival in particular, we see that the available information is a bit scarce.

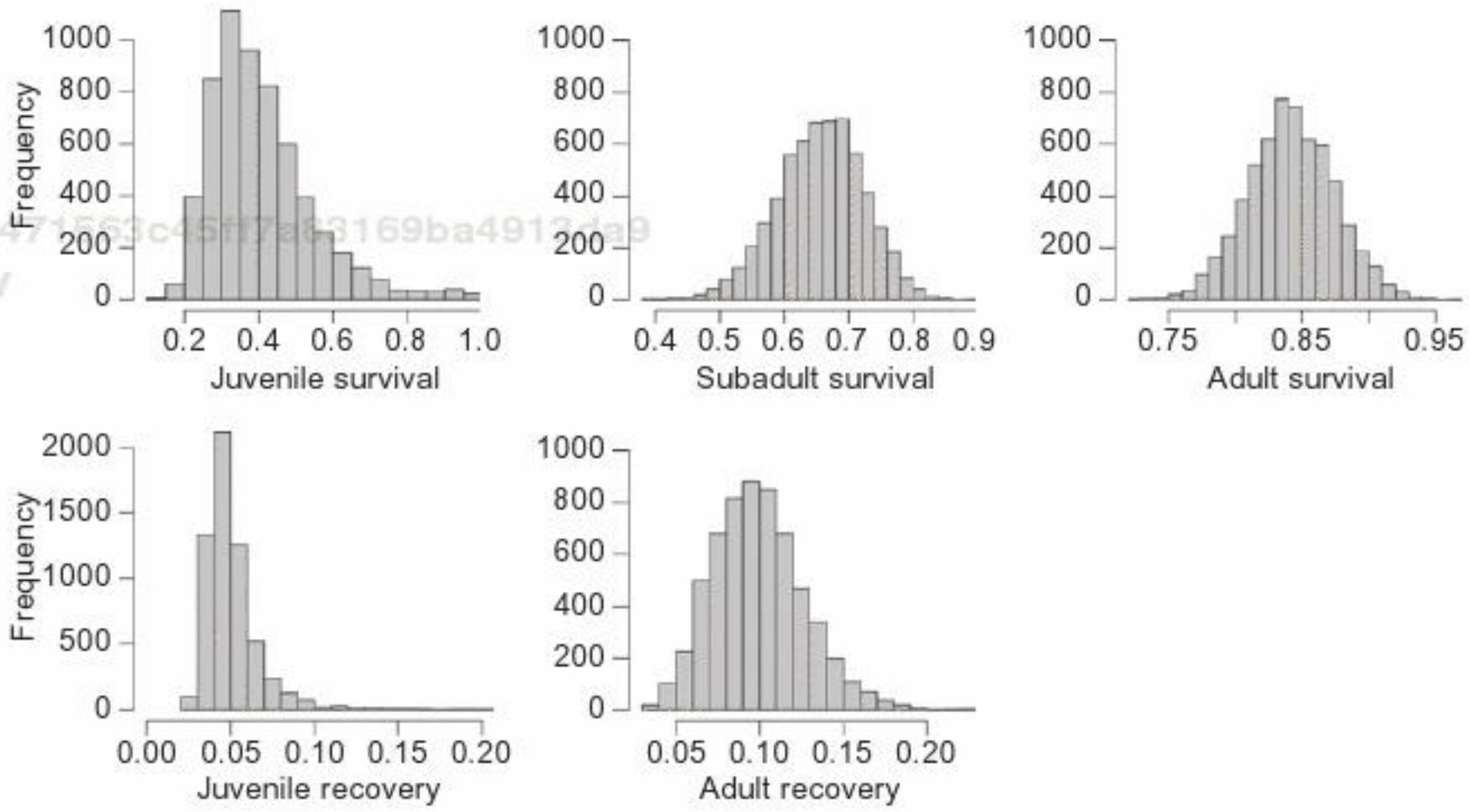

FIGURE 8.5 Posterior distributions of survival and recovery probabilities of Swiss red kites under model B with noninformative priors.

```
par(mfrow = c(2, 3), las = 1)
hist(rk.ageB$sims.list$sjuv, breaks = 20, col = "gray", main = "",
   xlab = "Juvenile survival")
hist(rk.ageB$sims.list$ssub, breaks = 20, col = "gray", main = "",
   xlab = "Subadult survival")
hist(rk.ageB$sims.list$sad, breaks = 20, col = "gray", main = "",
   xlab = "Adult survival")
hist(rk.ageB$sims.list$rjuv, breaks = 20, col = "gray", main = "",
   xlab = "Juvenile recovery", xlim = c(0, 0.2))
hist(rk.ageB$sims.list$rad, breaks = 20, col = "gray", main = "",
   xlab = "Adult recovery")
```

Comparison of the posterior distributions of juvenile survival under the two priors shows that both are similar (Fig. 8.6). Thus, existing information from the literature did not have a very strong impact, and the uncertainty about juvenile survival was not much reduced. We would just like to note that the specification of informative priors can be a challenge, and there may be better ways than the ad hoc approach in our example (e.g., McCarthy and Masters, 2005; Servanthy et al., 2010).

```
plot(density(rk.ageA$sims.list$sjuv), ylim = c(0, 5), lwd = 2,
   main = "", xlab = "Juvenile survival", las = 1)
points(density(rk.ageB$sims.list$sjuv), col = "red", type = "l",
   lwd = 2)
text(x = 0.5, y = 4.8, "Prior distributions", pos = 4, font = 3)
legend(x = 0.6, y = 4.7, legend = c("U(0,1)", "beta(4.2,2.8)"),
   lwd = c(2, 2), col = c("black", "red"), bty = "n")
```

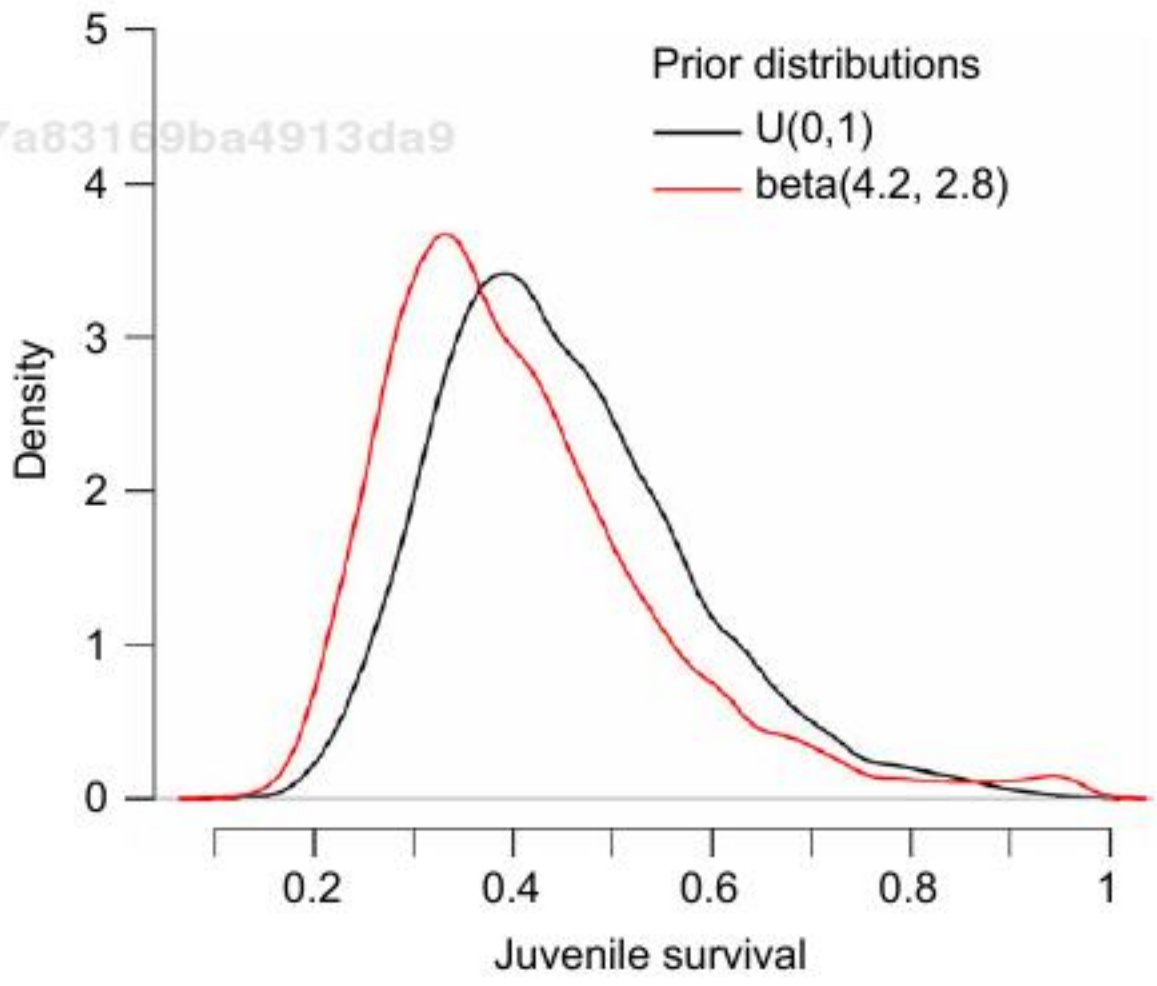

FIGURE 8.6 Posterior distributions of juvenile red kite survival with an informative (beta(4.2, 2.8), red) and a noninformative (U(0, 1), black) prior.

Finally, we reiterate the ease with which derived quantities are estimated in a Bayesian framework with posterior sampling: just compute the derived quantity for every MCMC iteration and summarize that new MCMC sample. For instance, here are the 95% CRI for the differences between juvenile and subadult and between subadult and adult survival:

```
quantile(rk.ageA$sims.list$ssub-rk.ageA$sims.list$sjuv, prob =
  c(0.025, 0.975))
      2.5%       97.5%
-0.1106200   0.4867075

quantile(rk.ageA$sims.list$sad-rk.ageA$sims.list$ssub, prob =
  c(0.025, 0.975))
      2.5%       97.5%
 0.0595925   0.3242150
```


8.5 SUMMARY AND OUTLOOK

We have introduced an important class of models for estimation of survival and recovery probabilities from mark-recovery data, typically for birds. We have shown two different parameterizations of the model. The model based on the multinomial likelihood is computationally cheaper and can be extended to include groups, temporal covariates, as well as age classes. The model based on the state-space likelihood is required if we want to fit individual covariates, but this comes at the expense of much longer computational time. The combination of age-effects and individual covariates is best done with multistate models (Chapter 9).

Mark-recovery models have a long history, and many variants exist. They include models for the estimation of seasonal survival probabilities (Tavecchia et al., 2002), kill rates (Nichols et al., 1991), the proportion of animals dying from different mortality causes (Schaub and Pradel, 2004), and spatial variation of recovery probabilities (Royle and Dubovsky, 2001), to name just a few. Williams et al. (2002) provide a recent review of mark-recovery models and include plenty of additional information, such as different parameterizations, study design, or goodness-of-fit testing.

Mark-recovery data can also be analyzed with multistate models (Lebreton et al., 1999; Gauthier and Lebreton, 2008; see also Section 9.5). Moreover, it is possible to jointly analyze mark recovery and capture–recapture data, although they provide different measures of survival. Multistate capture–recapture models are the best tool to conduct such a joint analysis (see Section 9.5).

8.6 EXERCISES

1. Simulate mark-recovery data of two groups: both groups have a survival probability of 0.5, the first group has a recovery probability of 0.1, and the second group has a recovery probability of 0.2. The study is

conducted for 10 years, and each year, 50 individuals are marked in each group. Fit the model (s, r_g) using (1) the multinomial and (2) the state-space likelihoods.

2. It is quite typical for population studies that only nestlings are marked but no adult individuals. This is because the capture of adults is often much more time consuming than the marking of nestlings, which can be easily marked in the nest. Simulate data from a study on a common tern population in which only nestlings are marked. The study duration is 15 years; in each year, 200 nestlings are marked, and the parameters are $sj = 0.3$, $sa = 0.8$, $rj = 0.25$, and $ra = 0.15$. Analyze these data with (1) the data-generating model and (2) using a model in which the recovery probability is the same in both age classes. Comment on the parameter estimates that you obtain from both models.
3. Simulate mark-recovery data with the following characteristics: one group, during each of the 20 study years 500 individuals are released, the survival probability declines linearly from 0.8 in the first year to 0.6 in the last study year, whereas the recovery probability is constant at 0.05. Analyze these data with the multinomial model.
4. Because of differential behavior, the recovery probability may show strong individual variation. Simulate mark-recovery data for a population with mean survival of 0.7 and a mean recovery probability of 0.2. The variance of the recovery probability among individuals is 0.7 (on the logit scale). Assume that the study lasts 10 years and that each year 100 individuals are released. Analyze the data with (1) the data-generating model and (2) with a model that assume a common recovery probability for all individuals. What is the impact on the estimate of the survival probability?