

01.2_PhotoID_CMR_Bayes

2024-10-28

Here, we will re-analyze the photo-ID CMR data using a Bayesian hierarchical state-space framework, as described in Kéry and Schaub (2011).

The script was obtained from <https://github.com/oliviergimenez/bayes-multistate-jollyseber>, modified for these data.

Prep the environment:

POPAN Jolly-Seber

Assumptions for Jolly-Seber Mark Recapture models: 1. Animals retain their tags throughout the experiment

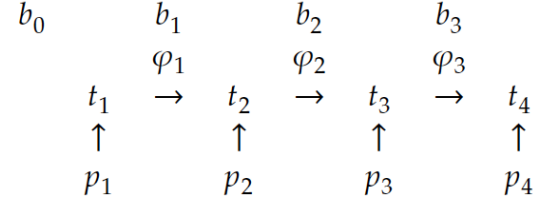
2. Tags are read properly
3. Sampling is instantaneous
4. Survival probabilities are the same for all animals (marked and unmarked) between each pair of sampling occasions (homogenous survival)
5. Catchability is the same for all animals (marked and unmarked) at each sampling occasion (homogenous catchability)
6. The study area is constant

In the case of killer whales, do we meet the assumptions?

1. Yes - most nicks and scars used for ID are retained through the life of the animal.
2. Yes - we can assume that identified individuals are re-identified reliably. However, there are ways to account for identification error - to be explored later.
3. Yes - sampling period is short (1 to a few days)
4. Probably? - while survival probabilities might differ between sex and age classes, being “marked” does not affect an individual’s survival
5. Unlikely? - Equal catchability could be affected by:
 - Behaviour - some individuals/groups may be more likely to approach the boat, and thus we may get more/better photographs
 - Individuals with more distinct markings may be more likely to be identified/re-identified when image quality is lower
 - Cooch and White (2014) describe this as the most critical assumption for JS models
6. Sort of? - The study area includes locations around Northern Baffin Island (mainly Admiralty Inlet and Eclipse Sound) and Cumberland Sound, but we have not consistently sampled in each location each year

Here we estimate population size using a POPAN Jolly-Seber Mark Recapture Model. The POPAN formulation of the JS model is an open population model that implies the existence of a super-population consisting of all animals that would ever be born to the population. Parameter b_i (entry probability) represents the probability that an animal from the hypothetical super-population would enter the population between occasion t and $t+1$. Entry could result from recruitment or immigration.

Figure 12.2: Process model for POPAN parameterization of JS exp capture at occasion i ; φ_i represents the probability of an animal and b_i represents the probability that an animal from the super-population enters the population at occasion i and survive to the next sampling occasion $i+1$ and not to happened, but are easily included.



The parameterization of the POPAN JS Model is as follows:

State-Space JS Model

Notes from Kéry and Schaub (2011)

In this model, the observed capture-recapture data are described as the result of a state process (the ecological process) and the observation process, which depends on the result of the state process.

N_s = superpopulation size (the number of individuals ever alive during the study) b_t = entry probability (the probability that a member of N_s enters the population at occasion t - could result from either birth or immigration) $B_t = N_{sxt}$ = number of individuals entering the population at t - The number of individuals entering at each occasion can be modeled with a multinomial distribution as $\mathbf{B} \sim \text{multinomial}(N_{\text{super}}, \mathbf{b})$.

We denote the latent (hidden) state of individual i at occasion t as $z_{\{i,t\}} = 1$ if it is alive and present in the population, and $z_{\{i,t\}} = 0$ if it is either dead or has not yet entered the population. On entry, the survival process starts. The latent state $z_{\{i,t+1\}}$ at $t + 1$ is determined by a Bernoulli trial (two possible outcomes: success (detection) or failure (no detection)) with success probability $\phi_{\{i,t\}}$ ($t = 1, \dots, T-1$).

The entry and survival process, described above, represent the latent state process. The observation process is defined for individuals that are alive ($z = 1$). The detection of individual i at time t is determined by another Bernoulli trial.

The resulting capture-recapture data consists of the capture histories of n individuals. Typically not all individuals in a population are capture (capture probability < 1), so $n < N_s$. If N_s were known, the capture-recapture data would contain an additional $N_s - n$ all-zero capture histories, but N_s is unknown. Parameters such as entry and capture probabilities refer to N_s , not just n . To deal with this, we use parameter-expanded data augmentation. This fixes the dimension of the parameter space in the analysis by augmenting the observed data with a large number of all-zero capture histories, resulting in a larger data set of fixed dimension M , and to analyze the augmented data set using a reparameterized (zero-inflated) version of the model that would be applied if N_s were known.

Superpopulation parameterization b = entry probability ψ = inclusion parameter

To keep the sequential specification of the state process model, we reexpress the entry probabilities (b) as conditional entry probabilities (η).

The state of individual i at the first occasion is:

$$z_{i,1} \sim \text{Bernoulli}(\eta_1)$$

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

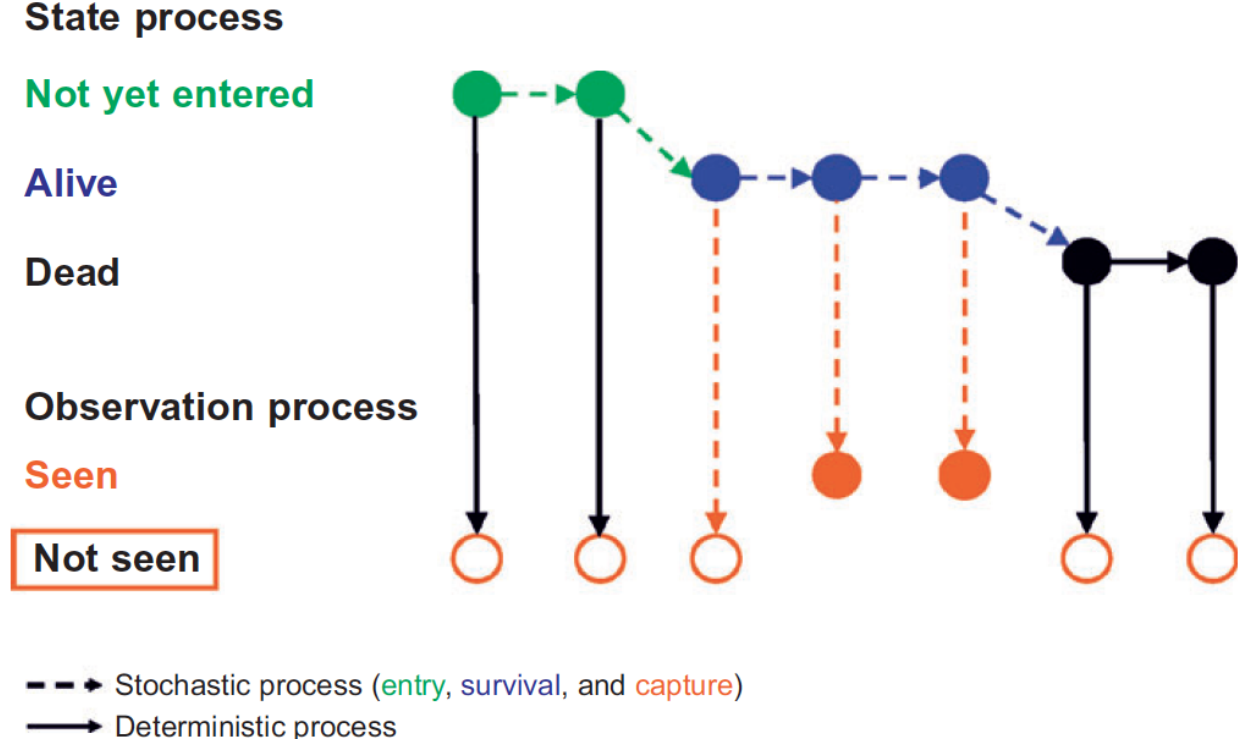


Figure 1: State and Observation Process

$$z_{i,t+1} | z_{i,t}, \dots, z_{i,1} \sim \text{Bernoulli}(z_{i,t}\phi_{i,t} + \eta_{t+1} \prod_{k=1}^t (1 - z_{i,k}))$$

For the observation process, we suppose that each individual of M has an associated latent variable $w_i \sim \text{Bernoulli}(\psi)$. Individuals with $w_i = 1$ are exposed to sampling if alive, while individuals with $w_i = 0$ are not exposed to sampling. The observation model (which admits zero-inflation) is:

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

We need to specify priors for survival, capture, entry, and inclusion probabilities. To express ignorance, specify a uniform prior $U(0,1)$ on all of them. However, for entry probability (b) we use a Dirichlet prior (multinomial generalization of the beta distribution): $b_t \sim \text{Dirichlet}(\alpha)$, where $\alpha_t = 1$ for all t (this allocates the entries of all individuals uniformly over the T occasions). The $U(0,1)$ prior for inclusions probability ψ induces a discrete $U(0, M)$ prior for the superpopulation size N_s .

***** DESCRIBE WHY WE DO THE DATA AUGMENTATION

Application

Now we can build our model using the superpopulation parameterization of JS in a Bayesian framework.

Note that in model described in the book, they use constant survival (ϕ), constant capture (p), and time-dependent entry (b) - this is different than the AIC suggested to be the best fit in the frequentist analysis (constant survival, time-dependent capture, constant entry). The second best model ($(\delta)AIC = 1.3$) had constant survival, time-dependent capture, and time-dependent entry, which is how we will parameterize the model here.

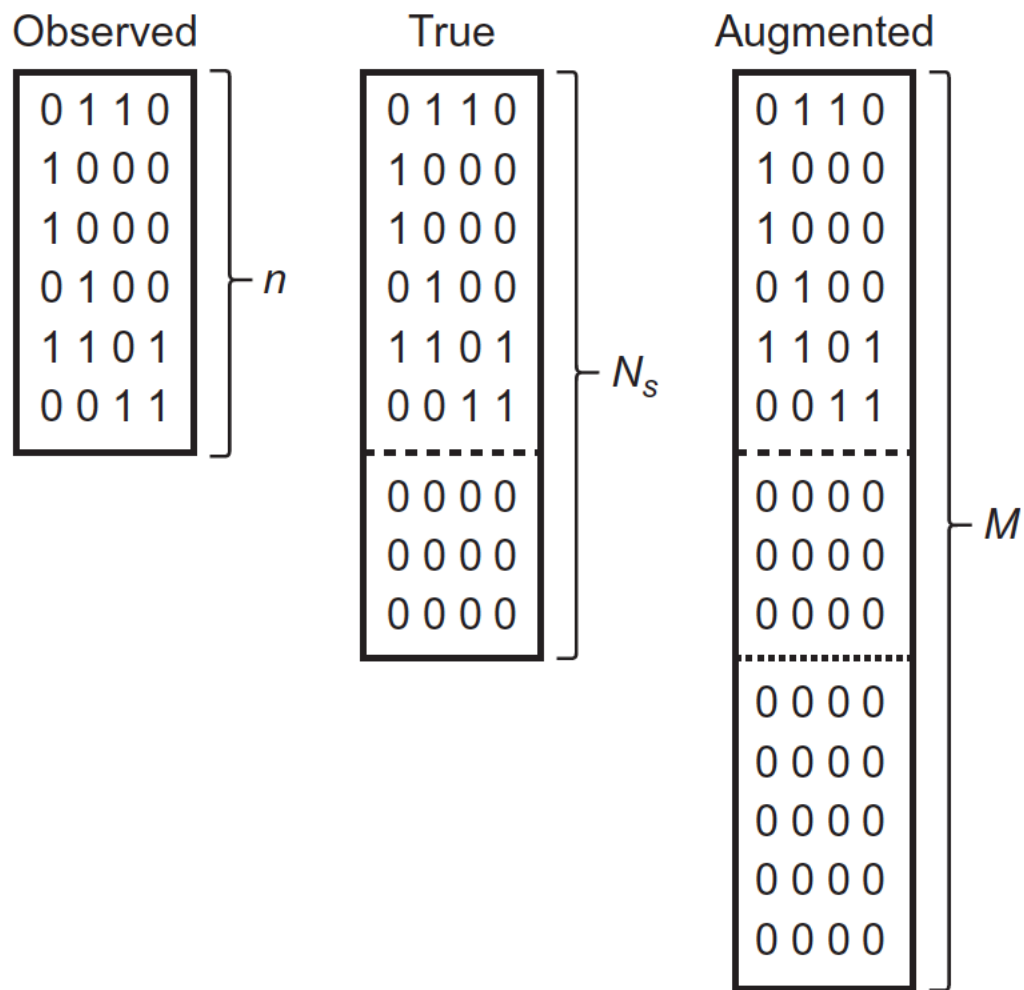


Figure 2: Data augmentation

We have modified the original code obtained from <https://github.com/oliviergimenez/bayes-multistate-jollyseber> with constant survival, time-dependent capture, time-dependent entry. We use vague priors for survival, capture, inclusion probability, and entry, as in Kéry and Schaub (2011).

Further, we have included the population growth rate (λ) computed as a derived quantity from the estimated population sizes or survival and per-capita entry probability:

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

To determine the “average” lambda over the study period, we calculate the geometric mean of the annual lambda, as suggested by Cooch & White (2021; http://www.phidot.org/software/mark/docs/book/pdf/cha_p13.pdf#page=18).

POPAN parameterization of data:

```
popan <- function() {

  # Priors and constraints
  for (i in 1:M){
    for (t in 1:(n.occasions-1)){
      phi[i,t] <- mean.phi # Constant survival
    } ##
    for (t in 1:n.occasions){
      p[i,t] <- p.time[t] # Time-dependent capture
    } ##
  } #i

  mean.phi ~ dunif(0, 1) # Prior for mean survival - uniform distribution with min = 0 and max = 1
  psi ~ dunif(0, 1) # Prior for inclusion probability

  for(t in 1:n.occasions) {
    p.time[t] ~ dunif(0,1) # prior for time-dependent capture
  }

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

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

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

```

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

# Calculate derived population parameters
for (i in 1:M){
  for (t in 1:n.occasions){
    u[i,t] <- z[i,t] * w[i] # Deflated latent state (u)
  }
}
for (i in 1:M){
  recruit[i,1] <- u[i,1]
  for (t in 2:n.occasions){
    recruit[i,t] <- (1 - u[i,t-1]) * u[i,t]
  } #t
} #i
for (t in 1:n.occasions){
  N[t] <- sum(u[1:M,t]) # Actual population size
  B[t] <- sum(recruit[1:M,t]) # Number of entries
} #t
for (i in 1:M){
  Nind[i] <- sum(u[i,1:n.occasions])
  Nalive[i] <- 1 - equals(Nind[i], 0)
} #i
for (t in 1:(n.occasions-1)) {
  lambda[t] <- N[t+1]/N[t] # Lambda realized annual population growth rate
  f[t] <- B[t+1]/N[t] # recruitment (per capita entry) rate
} #t
Nsuper <- sum(Nalive[]) # Superpopulation size
mean.lambda <- prod(lambda[])^(1/(n.occasions-1)) # geometric mean realized growth rate
}

```

Now let's apply it to our data.

Load and format the photo-ID capture history data:

```

# Load data and insert zeros instead of NAs in years where there are no sightings:
CMR_data <- read.csv("photo_CMR_data_final.csv", header = TRUE)
CMR_data[is.na(CMR_data)] <- 0

# Remove the id column and add a new column for CH data:
CMR_data <- CMR_data %>%
  select(-ID) %>%
  mutate(cmr = NA) %>%
  relocate(cmr, "X2009")

```

```

# Creating a column with 1s and 0s for all years observed:
for (i in 1:nrow(CMR_data)){
  CMR_data[i,1] <- paste(CMR_data[i,2], CMR_data[i,3], CMR_data[i,4], CMR_data[i,5], CMR_data[i,6], CMR_data[i,8], CMR_data[i,9], CMR_data[i,10], CMR_data[i,11], CMR_data[i,12], CMR_data[i,14], CMR_data[i,15], CMR_data[i,16], sep = "")}

# Select first column only, rename, and save to txt file
CMR <- CMR_data %>%
  select(cmr) %>%
  dplyr::rename(ch = cmr) %>%
  write.table(file = "_photo_CMR_data.txt", row.names = FALSE, quote = FALSE)

# Re-import and split data
ch_CMR <- import.chdata("_photo_CMR_data.txt")
popan_ch <- splitCH(ch_CMR$ch)

```

Augment the observed capture histories by nz pseudo-individuals, all with capture histories of 0:

```

nz <- 200      # Augmenting the data by 200 pseudo-individuals, approx. equal to population size
CH.aug <- rbind(popan_ch, matrix(0, ncol = dim(popan_ch)[2], nrow = nz))

```

Bundle data.

```

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

```

We use the same initial values as presented by Schaub & Kéry (2011).

Initial values.

```

zinit <- CH.aug
zinit[zinit==0] <- 1

n.occasions.js <- ncol(CH.aug)

inits <- function(){list(mean.phi = runif(1, 0, 1),
  p.time = runif(n.occasions.js, 0, 1),
  psi = runif(1, 0, 1),
  z = zinit)}

```

Parameters monitored.

```

parameters <- c("psi", "p.time", "mean.phi", "b", "Nsuper", "N", "B", "nu", "lambda", "f", "mean.lambda")

```

MCMC settings.

```

n.iter <- 50000      # Number of iterations
n.burnin <- 10000    # Number discarded (burn-in)
n.chains <- 3        # Number of chains

```

Call Jags - run model on bio server.

```

# kw_popan <- jags(data = bugs.data,
#                   inits = inits,
#                   parameters.to.save = parameters,
#                   model.file = popan,
#                   n.chains = n.chains,
#                   n.iter = n.iter,

```

```

#                               n.burnin = n.burnin)
# kw_popan

# Save run:
#saveRDS(kw_popan, "output/kw_popan_results_sept2024_run2_3.rds")

# Load results
kw_popan <- readRDS("output/kw_popan_results_sept2024_run2_3.rds")

```

Model evaluation

Rhat is the potential scale reduction factor, a convergence diagnostic (Gelman-Rubin diagnostic: at convergence, Rhat=1). The general rule is that all Rhat values should be <1.1, which they are above.

```
library(mcmcplots)
```

```

## Registered S3 method overwritten by 'mcmcplots':
##   method      from
##   as.mcmc.rjags R2jags

```

```
library(coda)
```

```
# Convert to mcmc output as object for diagnostic plots
```

```
kw_popan.mcmc <- as.mcmc(kw_popan)
```

```
# save mcmc object
```

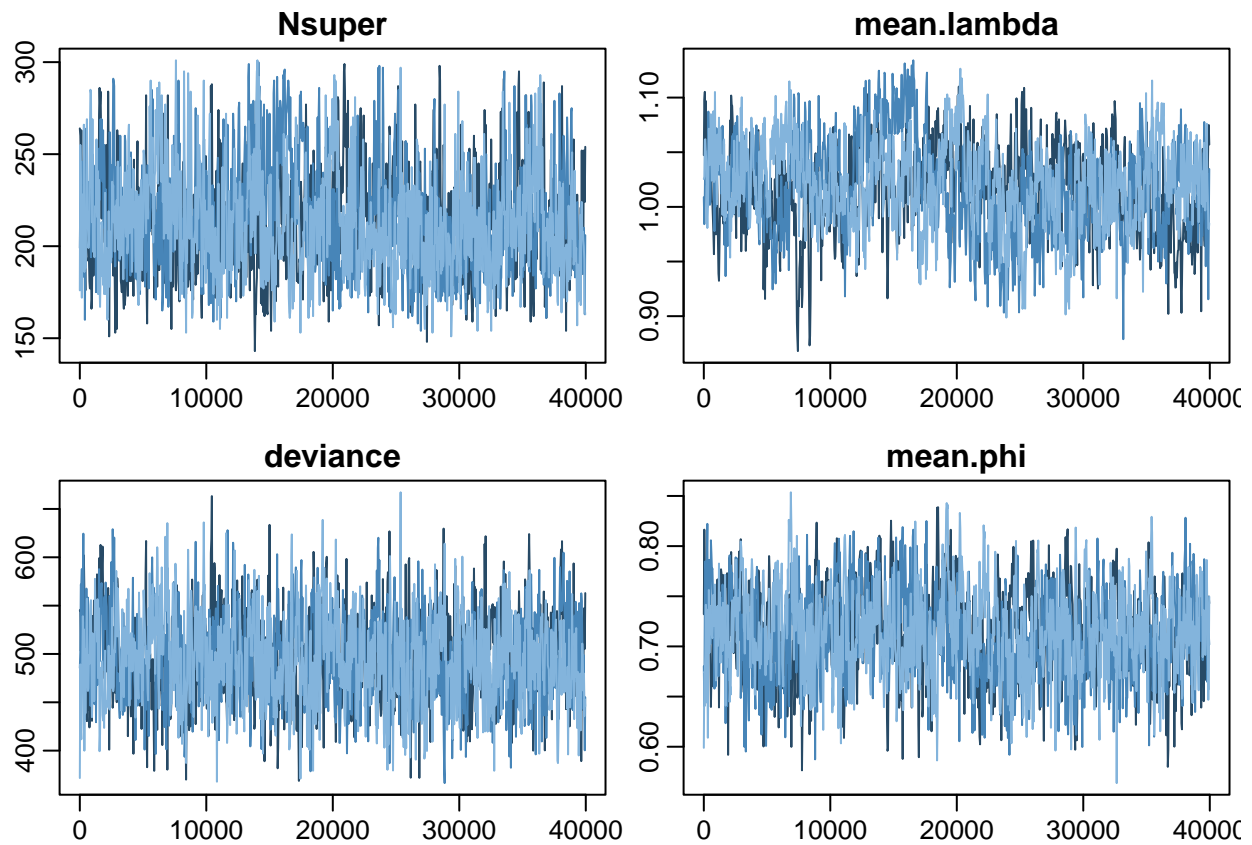
```
#saveRDS(kw_popan.mcmc, "output/kw_popan_mcmc_sept2024_run2_3.rds")
```

```
cols <- c("#274a66", "#4785b8", "#83b4dc")
```

```
# Traceplot - to visually assess convergence between plots
```

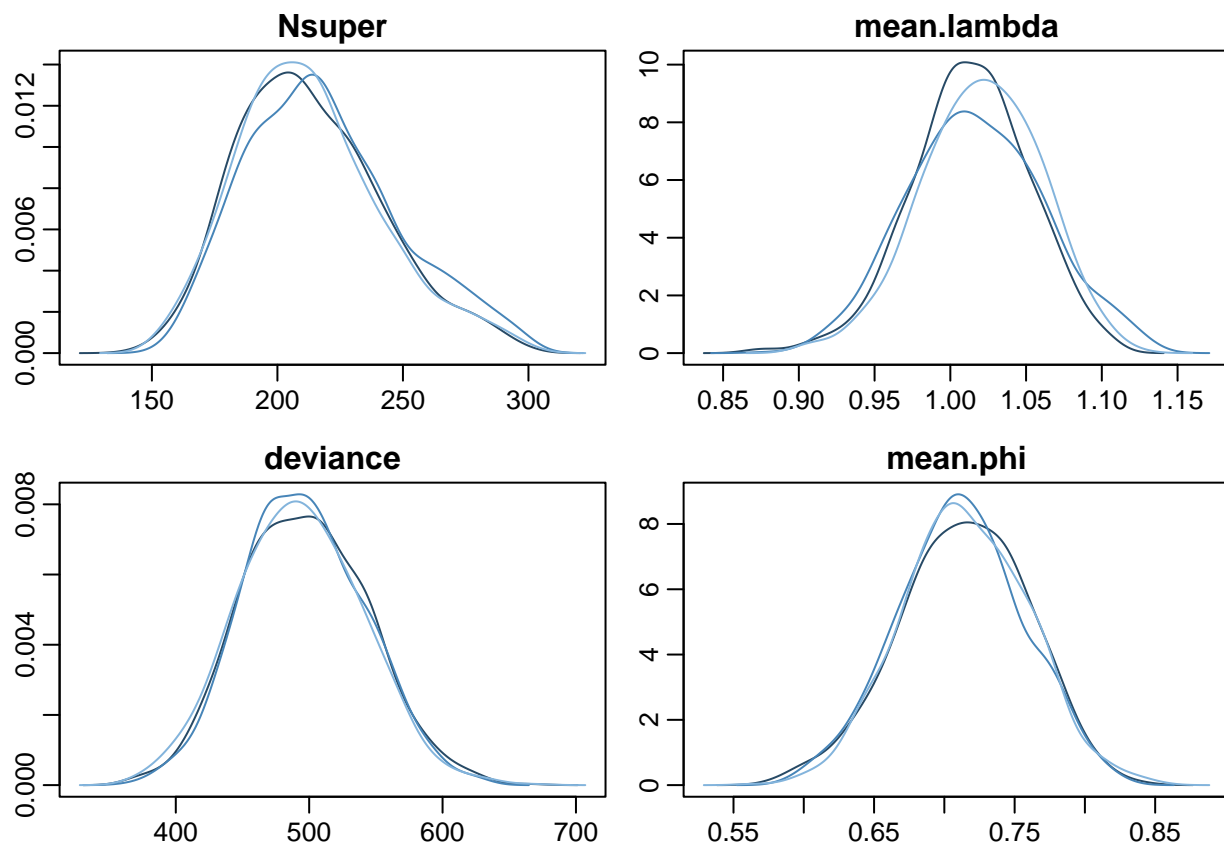
```
#tiff("plots/kw_popan_traceplot.tiff", units="in", width=8, height=5, res=400)
```

```
mcmcplots::traplot(kw_popan.mcmc, parms = c("Nsuper", "mean.lambda", "deviance", "mean.phi"), style = "j")
```

```
#dev.off()

# Density plot - sampling density of the three chains from the posterior distribution
#tiff("plots/kw_popan_densityplot.tiff", units="in", width=8, height=5, res=400)
mcmcplots::denplot(kw_popan.mcmc, parms = c("Nsuper", "mean.lambda", "deviance", "mean.phi"), style = "j")
```



#dev.off()