ESS 575: Occupancy Modeling Lab

Team England

10 November, 2022

Contents

Team England:

- Caroline Blommel
- Carolyn Coyle
- Bryn Crosby
- George Woolsey

cblommel@mail.colostate.edu, carolynm@mail.colostate.edu, brcrosby@rams.colostate.edu, george.woolsey@colostate.edu

Motivation

Modeling presence or absence is a classic problem involving mixture models, specifically random variables that are zero-inflated. Extra zeros are encountered when we model presence or absence because zeros arise from two conditions: truly absent individuals and individuals present but undetected. This means we need a model for the process that controls occupancy, the true state, and model of the data that accounts for detection. This is often our starting point in Bayesian analysis – there is a true, unobserved state we seek to understand using a model of a process. We take imperfect observations on that state and must correct them using a model of the data.

R libraries needed for this lab

You need to load the following libraries. Set the seed to 10 to compare your answers to ours.

```r
# bread-and-butter
library(tidyverse)
library(lubridate)
library(viridis)
library(scales)
library(latex2exp)
# visualization
library(cowplot)
library(kableExtra)
# jags and bayesian
library(rjags)
library(MCMCvis)
library(HDInterval)
library(BayesNSF)
#set seed
set.seed(10)
```

Problem

A fundamental question in landscape ecology seeks to understand how landscape structure shapes variation in habitat use by species. We will use data from the Swiss Survey of Common Breeding Birds, courtesy of Royle and Dorazio (2008), to model habitat occupancy by a resident bird in the Swiss Alps, the willow tit (*Parus montanus*). The data come from annual surveys of one km^2 quadrats distributed across Switzerland (Fig. 1). Surveys are conducted during the breeding season on three separate days, but some quadrats have missing data so that the number of replicate observations is fewer than three.

During each survey, an observer records every visual or acoustic detection of a breeding species (we do not differentiate between these two types of detection in this problem) and marks its location using a global positioning system or, in earlier years, a paper map. We assume that the true state (occupied or unoccupied) does not change among sample dates, an assumption known as closure. This assumption is reasonable because we are observing a resident species during the breeding season.

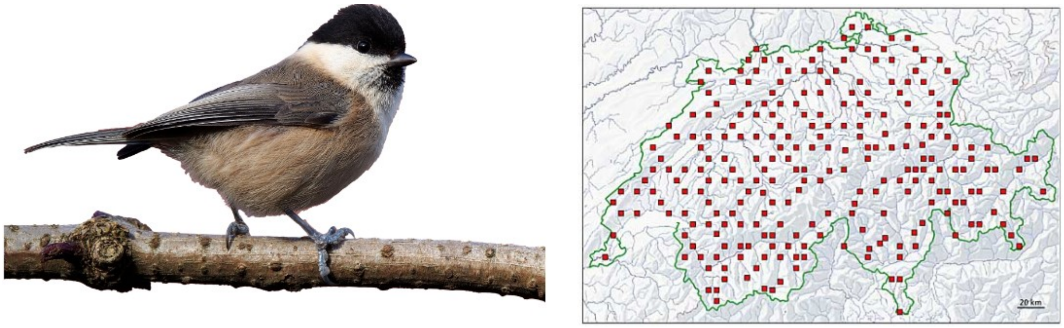


Fig. 1. The willow tit (left, credit: Francis C. Franklin) is one of 70 bird species that are surveyed annually for abundance in 267 1 km^2 sampling units distributed across Switzerland (right, credit: the Swiss Ornithological Institute).

We want to understand the influence of forest cover and elevation on the distribution of the willow tit. The data frame `BayesNSF::SwissBirds` has the number of times a quadrat (`quadrat`) was searched (`numberVisits`) and the number of times willow tits were detected (`numberDetections`). We have covariates on forest canopy cover (`forestCover`) as well as elevation in meters (`elevation`) for each quadrat surveyed. Data on detection each day on each quadrat (0 or 1) are also available. Develop a model of the influence of forest cover and elevation on the distribution of willow tits. Your model should allow estimation of the optimum elevation of willow tit habitat at the mean forest cover, where optimum elevation is defined as the elevation where probability of occupancy is maximum.

```
# data structure
BayesNSF::SwissBirds %>%
  dplyr::glimpse()
```

```
## Rows: 237
## Columns: 5
## $ quadrat          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16~
## $ numberVisits     <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,~
## $ numberDetections <int> 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 1, 2, 0, 3,~
## $ elevation        <int> 420, 450, 1050, 1110, 510, 630, 590, 530, 1140, 770, ~
## $ forestCover      <int> 3, 21, 32, 35, 2, 60, 5, 13, 50, 57, 84, 17, 58, 26, ~
```

```
summary(BayesNSF::SwissBirds$quadrat)
```

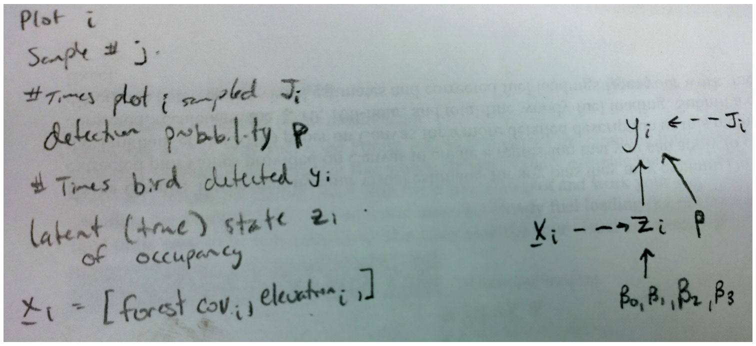
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1      60     119     119     178     237
```

```
summary(BayesNSF::SwissBirds$numberVisits)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.000   3.000   3.000   2.806   3.000   3.000
```

Question 1

Diagram the network of knowns and unknowns.


Plot i
Sample # j.
#Times plot i sampled J_i
detection probability p
# Times bird detected y_i
latent (true) state z_i of occupancy
x_i = [forest cov._i, elevation_i]
y_i
J_i
x_i
z_i
p
β0, β1, β2, β3

Bayesian network for willow tit occupancy

Question 2

Write a mathematical expression for the posterior and the joint distribution.

Stochastic model of y (the observed detections):

$$y_i \sim \mathsf{binomial}(p_i)$$

Deterministic model of z (the true probability of occupancy):

$$z_i \sim \mathsf{Bernoulli}\big(g(\beta_0, \beta_1, \beta_2, \beta_3, x_i)\big)$$

$$g(\beta_0, \beta_1, \beta_2, \beta_3, x_i) = \text{inverse logit}\big(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{1i}^2 + \beta_3 x_{2i}\big) = \frac{\exp\big(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{1i}^2 + \beta_3 x_{2i}\big)}{1 + \exp\big(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{1i}^2 + \beta_3 x_{2i}\big)}$$

$$x_1 = \text{elevation; } x_2 = \text{forest cover}$$

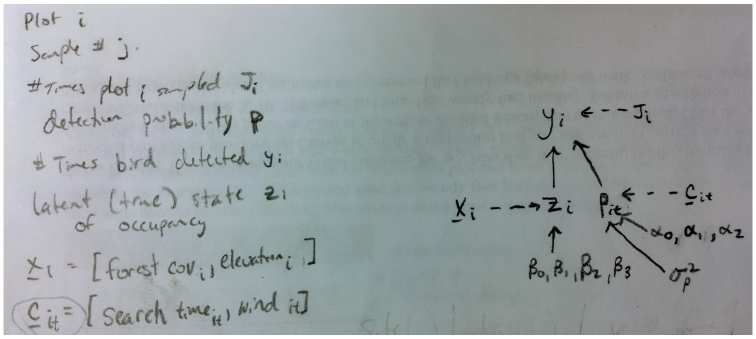
Utilizing a quadratic regression model with the inclusion of x_{1i}^2 in the model allows us to account for a nonlinear (e.g. parabolic) relationship between occupancy and elevation. This model form allows us to estimate the optimum elevation of willow tit habitat at the mean forest cover.

Posterior and Joint:

$$
\begin{aligned}
\big[\boldsymbol{z}, \boldsymbol{\beta}, p \mid \boldsymbol{y}\big] \propto & \prod_{i=1}^{n=237} \mathsf{binomial}\big(y_i \mid J_j,\, z_i \cdot p\big) \cdot \mathsf{Bernoulli}\big(z_i \mid g(\boldsymbol{\beta}, \boldsymbol{x_i})\big) \cdot \mathsf{uniform}\big(p \mid 0, 1\big) \\
& \times \prod_{j=1}^{J=3} \mathsf{normal}\big(\boldsymbol{\beta} \mid 0, 2.7\big)
\end{aligned}
$$

Question 3

Modify your model to include the effect of search time and wind speed (measured at each quadrat on each day) on detection probability. Draw a DAG and write the posterior and joint distributions. In so doing, assume that posterior predictive checks of a preliminary detection model revealed that you need to include an explicit variance term for the detection probability.


Plot i
Sample # j.
#Times plot i sampled J_i
detection probability p
# Times bird detected y_i
latent (true) state z_i of occupancy
$\underline{x}_i$ = [forest cov.$_i$, elevation$_i$]
$\underline{c}_{it}$ = [search time$_{it}$, wind $_{it}$]
y_i
J_i
$\underline{x}_i$
z_i
p_{it}
$\underline{c}_{it}$
$\alpha_0, \alpha_1, \alpha_2$
$\beta_0, \beta_1, \beta_2, \beta_3$
σ_p^2

Bayesian network for willow tit occupancy with detection probability variance

Deterministic model of p (the probability of detection):

$$
\begin{aligned}
p_{it} &\sim \mathsf{beta}\big(\mu, \sigma^2\big) \\
\mu_{it} = g_2(\alpha_0, \alpha_1, \alpha_2, x_{it}) &= \text{inverse logit}\big(\alpha_0 + \alpha_1 x_{1it} + \alpha_2 x_{2it}\big) = \frac{\exp\big(\alpha_0 + \alpha_1 x_{1it} + \alpha_2 x_{2it}\big)}{1 + \exp\big(\alpha_0 + \alpha_1 x_{1it} + \alpha_2 x_{2it}\big)} \\
x_1 &= \text{search time; } x_2 = \text{wind speed}
\end{aligned}
$$

Posterior and Joint:

$$
\begin{aligned}
\big[\boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{\alpha}, p, \sigma^2 \mid \boldsymbol{y}\big] \propto & \prod_{i=1}^{n=237} \prod_{t=1}^{T} \mathsf{Bernoulli}\big(y_i \mid J_j,\, z_i \cdot p\big) \cdot \mathsf{Bernoulli}\big(z_i \mid g(\boldsymbol{\beta}, \boldsymbol{x_i})\big) \cdot \mathsf{beta}\big(p_{it} \mid g_2(\boldsymbol{\alpha}, \boldsymbol{x_{it}})\big), \sigma^2\big) \\
& \times \prod_{j=1}^{J=3} \mathsf{normal}\big(\boldsymbol{\beta} \mid 0, 2.7\big) \\
& \times \prod_{j=1}^{J=3} \mathsf{normal}\big(\boldsymbol{\alpha} \mid 0, 2.7\big) \cdot \mathsf{uniform}\big(\sigma^2 \mid 0, 100\big)
\end{aligned}
$$

Question 4

Approximate the marginal posterior distributions of parameters in the forest and elevation model with constant detection probability (the first one, above) using JAGS. Conduct posterior predictive checks. Some hints: 1) You will need to standardize the covariates by subtracting the mean and dividing by the standard deviation for each observation in the elevation and forest cover data. Use the `scale` function to do this (it will drastically speed convergence). 2) You *must* give initial values of 1 to all unknown 0 or 1 z states.

Data set up

```
# mean and sd of elevation for standardizing
mean_elevation <- mean(BayesNSF::SwissBirds$elevation)
sd_elevation <- sd(BayesNSF::SwissBirds$elevation)
# sequence of elevation for derived quantities
elevation_pred <- seq(
  round(
    max( ## don't let elev go below 0
      min(BayesNSF::SwissBirds$elevation) - (sd(BayesNSF::SwissBirds$elevation) * 2)
      , 0
    )
    , -1
  )
  , round(
    max(BayesNSF::SwissBirds$elevation) + (sd(BayesNSF::SwissBirds$elevation) * 2)
    , -1
  )
  , 10
)
# standardize
elevation_pred_z <- (elevation_pred - mean_elevation) / sd_elevation

# list of data
data = list(
  N = nrow(BayesNSF::SwissBirds) # number sites
  , y = as.double(BayesNSF::SwissBirds$numberDetections)
  , elevation = scale(BayesNSF::SwissBirds$elevation) %>% as.numeric() %>% as.double
  , forestCover = scale(BayesNSF::SwissBirds$forestCover) %>% as.numeric() %>% as.double
  , n = as.double(BayesNSF::SwissBirds$numberVisits)
  , mean_elevation = mean_elevation
  , sd_elevation = sd_elevation
  , mean_forestCover = mean(BayesNSF::SwissBirds$forestCover)
  , elevation_pred_z = elevation_pred_z %>% as.double()
)
```

JAGS Model

Write out the JAGS code for the model.

```
## JAGS Model
model{
  # priors
```

```
b0 ~ dnorm(0, (1/2.7)) # dnorm(mu = mean, tau= precision )
b1 ~ dnorm(0, (1/2.7))
b2 ~ dnorm(0, (1/2.7))
b3 ~ dnorm(0, (1/2.7))
p ~ dunif(0, 1)
# likelihood
for(i in 1:N){
  # Deterministic model of z (the true probability of occupancy)
  psi[i] <- ilogit(
     b0
     + b1*elevation[i]
     + b2*(elevation[i]^2)
     + b3*forestCover[i]
  )
  # likelihood
    z[i] ~ dbern(psi[i])
  # likelihood
    # Stochastic model of y (the observed detections)
    y[i] ~ dbin(z[i] * p, n[i])
    y_sim[i]  ~ dbin(z[i] * p, n[i])
  # sum of squares
    sq[i] <- (y[i]-psi[i])^2
    sq_sim[i] <- (y_sim[i]-psi[i])^2
}
# Derived quantities
  #posterior predictive checks
    # test statistics y
    mean_y <- mean(y)
    sd_y <- sd(y)
    fit_y <- sum(sq)
    # test statistics y_sim
    mean_y_sim <- mean(y_sim)
    sd_y_sim <- sd(y_sim)
    fit_y_sim <- sum(sq_sim)
    # p-values
    p_val_mean <- step(mean_y_sim - mean_y)
    p_val_sd <- step(sd_y_sim - sd_y)
    p_val_fit <- step(fit_y_sim - fit_y)
  # optimum elevation of willow tit habitat at the mean forest cover
    # ... where optimum elevation is defined as the elevation
    # ... where probability of occupancy is maximum.
    # f(x) = ax^2 + bx + c
    # x-value of the vertex of the parabola = -b/2a
    optimum_elev_z = -b1/(2*b2)
    optimum_elev = (optimum_elev_z * sd_elevation) + mean_elevation
    # The predicted probability of occupancy
    for(j in 1:length(elevation_pred_z)){
      p_est[j] <- ilogit(
           b0
           + b1*elevation_pred_z[j]
           + b2*(elevation_pred_z[j]^2)
           # + b3*mean_forestCover # i don't know why including this produces incorrect estimates
        )
      }
}
```

Implement JAGS Model

```
##################################################################
# insert JAGS model code into an R script
##################################################################
{ # Extra bracket needed only for R markdown files - see answers
  sink("BirdOccupancyJAGS.R") # This is the file name for the jags code
  cat("
  model{
    # priors
    b0 ~ dnorm(0, (1/2.7)) # dnorm(mu = mean, tau= precision )
    b1 ~ dnorm(0, (1/2.7))
    b2 ~ dnorm(0, (1/2.7))
    b3 ~ dnorm(0, (1/2.7))
    p ~ dunif(0, 1)
    # likelihood
    for(i in 1:N){
      # Deterministic model of z (the true probability of occupancy)
      psi[i] <- ilogit(
         b0
         + b1*elevation[i]
         + b2*(elevation[i]^2)
         + b3*forestCover[i]
      )
      # likelihood
        z[i] ~ dbern(psi[i])
      # likelihood
        # Stochastic model of y (the observed detections)
        y[i] ~ dbin(z[i] * p, n[i])
        y_sim[i]  ~ dbin(z[i] * p, n[i])
      # sum of squares
        sq[i] <- (y[i]-psi[i])^2
        sq_sim[i] <- (y_sim[i]-psi[i])^2
    }
    # Derived quantities
      #posterior predictive checks
        # test statistics y
        mean_y <- mean(y)
        sd_y <- sd(y)
        fit_y <- sum(sq)
        # test statistics y_sim
        mean_y_sim <- mean(y_sim)
        sd_y_sim <- sd(y_sim)
        fit_y_sim <- sum(sq_sim)
        # p-values
        p_val_mean <- step(mean_y_sim - mean_y)
        p_val_sd <- step(sd_y_sim - sd_y)
        p_val_fit <- step(fit_y_sim - fit_y)
      # optimum elevation of willow tit habitat at the mean forest cover
```

```
        # ... where optimum elevation is defined as the elevation
        # ... where probability of occupancy is maximum.
        # f(x) = ax^2 + bx + c
        # x-value of the vertex of the parabola = -b/2a
        optimum_elev_z = -b1/(2*b2)
        optimum_elev = (optimum_elev_z * sd_elevation) + mean_elevation
        # The predicted probability of occupancy
        for(j in 1:length(elevation_pred_z)){
          p_est[j] <- ilogit(
               b0
               + b1*elevation_pred_z[j]
               + b2*(elevation_pred_z[j]^2)
               # + b3*mean_forestCover # i don't know why including this produces incorrect estimates
            )
        }
  }
  ", fill = TRUE)
  sink()
}
################################################################
# implement model
##################################################################
# specify the initial conditions for the MCMC chain
inits = list(
  list(
    b0 = runif(n = 1, min = -3, max = 3)
    , b1 = runif(n = 1, min = -3, max = 3)
    , b2 = runif(n = 1, min = -3, max = 3)
    , b3 = runif(n = 1, min = -3, max = 3)
    , p = runif(n = 1, min = 0, max = 1)
    , z = rep(1, nrow(BayesNSF::SwissBirds))
  )
  , list(
    b0 = runif(n = 1, min = -3, max = 3)
    , b1 = runif(n = 1, min = -3, max = 3)
    , b2 = runif(n = 1, min = -3, max = 3)
    , b3 = runif(n = 1, min = -3, max = 3)
    , p = runif(n = 1, min = 0, max = 1)
    , z = rep(1, nrow(BayesNSF::SwissBirds))
  )
  , list(
    b0 = runif(n = 1, min = -3, max = 3)
    , b1 = runif(n = 1, min = -3, max = 3)
    , b2 = runif(n = 1, min = -3, max = 3)
    , b3 = runif(n = 1, min = -3, max = 3)
    , p = runif(n = 1, min = 0, max = 1)
    , z = rep(1, nrow(BayesNSF::SwissBirds))
  )
)
# specify 3 scalars, n.adapt, n.update, and n.iter
# n.adapt = number of iterations that JAGS will use to choose the sampler
  # and to assure optimum mixing of the MCMC chain
n.adapt = 1000
```

```r
# n.update = number of iterations that will be discarded to allow the chain to
#   converge before iterations are stored (aka, burn-in)
n.update = 10000
# n.iter = number of iterations that will be stored in the
  # final chain as samples from the posterior distribution
n.iter = 10000
######################
# Call to JAGS
######################
jm = rjags::jags.model(
  file = "BirdOccupancyJAGS.R"
  , data = data
  , inits = inits
  , n.chains = length(inits)
  , n.adapt = n.adapt
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 237
##    Unobserved stochastic nodes: 479
##    Total graph size: 5587
##
## Initializing model
```

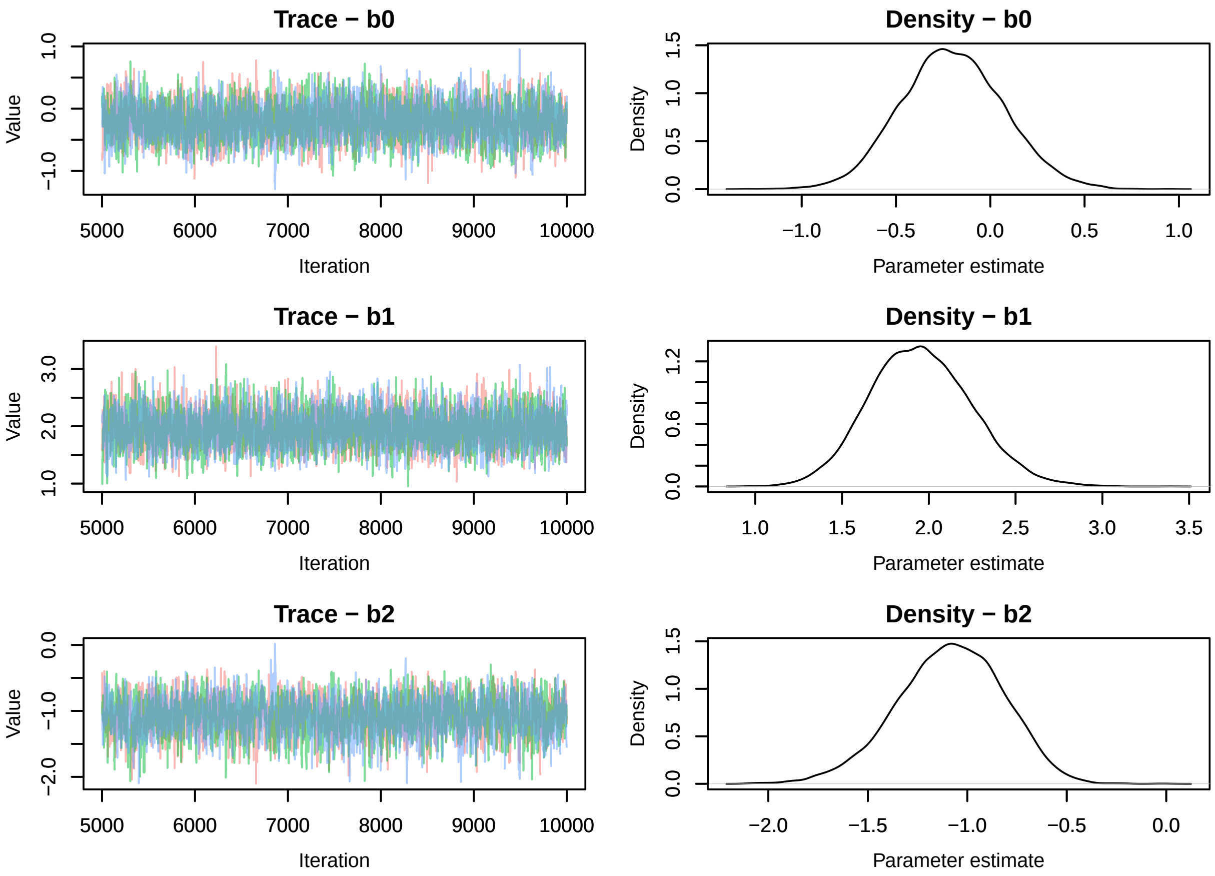
```r
stats::update(jm, n.iter = n.update, progress.bar = "none")
# save the coda object (more precisely, an mcmc.list object) to R as "zm"
zm = rjags::coda.samples(
  model = jm
  , variable.names = c(
      # parameters
      "b0"
      , "b1"
      , "b2"
      , "b3"
      , "p"
      # test statistics
      , "mean_y"
      , "sd_y"
      , "fit_y"
      , "mean_y_sim"
      , "sd_y_sim"
      , "fit_y_sim"
      # p-values
      , "p_val_mean"
      , "p_val_sd"
      , "p_val_fit"
      # derived quantities
      , "optimum_elev"
      , "p_est"
    )
  , n.iter = n.iter
  , n.thin = 1
  , progress.bar = "none"
)
```

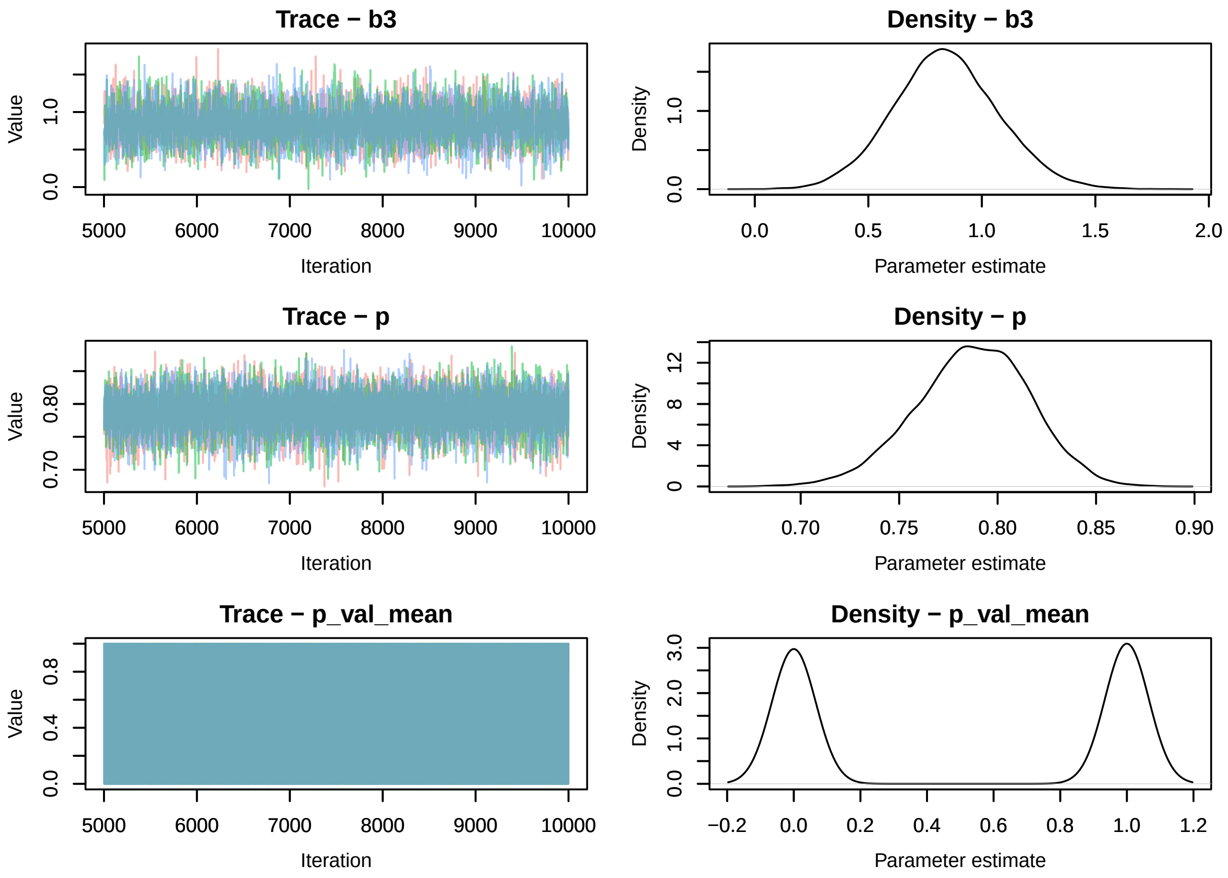
Question 5

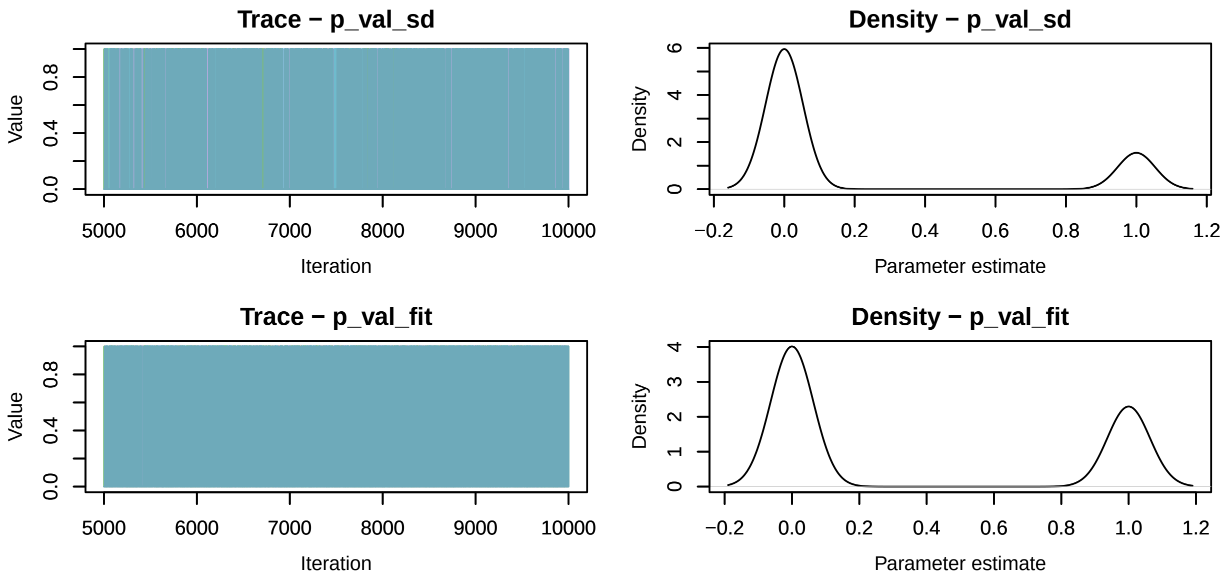
Summarize the parameters and check chains for convergence. Exclude the predictions of ψ from the summary if they were included in the coda object. What can you conclude about model fit?

Trace plots

```
# trace plot
MCMCvis::MCMCtrace(zm, params = c(
    # parameters
    "b0"
    , "b1"
    , "b2"
    , "b3"
    , "p"
    # p-values
    , "p_val_mean"
    , "p_val_sd"
    , "p_val_fit"
  )
  , pdf = FALSE
)
```


Trace − b0
Value
1.0
0.0
−1.0
5000
6000
7000
8000
9000
10000
Iteration
Density − b0
Density
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
Parameter estimate
Trace − b1
Value
3.0
2.0
1.0
5000
6000
7000
8000
9000
10000
Iteration
Density − b1
Density
1.2
0.6
0.0
1.0
1.5
2.0
2.5
3.0
3.5
Parameter estimate
Trace − b2
Value
0.0
−1.0
−2.0
5000
6000
7000
8000
9000
10000
Iteration
Density − b2
Density
1.5
1.0
0.5
0.0
−2.0
−1.5
−1.0
−0.5
0.0
Parameter estimate


Trace − b3
Value
0.0
1.0
5000
6000
7000
8000
9000
10000
Iteration
Density − b3
Density
0.0
1.0
0.0
0.5
1.0
1.5
2.0
Parameter estimate
Trace − p
Value
0.70
0.80
5000
6000
7000
8000
9000
10000
Iteration
Density − p
Density
0
4
8
12
0.70
0.75
0.80
0.85
0.90
Parameter estimate
Trace − p_val_mean
Value
0.0
0.4
0.8
5000
6000
7000
8000
9000
10000
Iteration
Density − p_val_mean
Density
0.0
1.0
2.0
3.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Parameter estimate



Posterior predictive check - Test Statistics

```
# summary
MCMCvis::MCMCsummary(zm, params = c(
      # test statistics
      "mean_y"
      , "mean_y_sim"
      , "sd_y"
      , "sd_y_sim"
      , "fit_y"
      , "fit_y_sim"
    )
    , n.eff = FALSE
  )
```

```
##                    mean          sd        2.5%         50%       97.5% Rhat
## mean_y        0.7679325  0.00000000   0.7679325   0.7679325   0.7679325  NaN
## mean_y_sim    0.7658039  0.03688750   0.6919831   0.7679325   0.8354430    1
## sd_y          1.1900653  0.00000000   1.1900653   1.1900653   1.1900653  NaN
## sd_y_sim      1.1543579  0.04194753   1.0681558   1.1561396   1.2314852    1
## fit_y       291.9980004  8.67781361 276.5597869 291.5437034 310.4772590    1
## fit_y_sim   282.8945616 27.00793663 230.1036839 282.9214919 335.9084340    1
```

Posterior predictive check - p-values

```
# summary
MCMCvis::MCMCsummary(zm, params = c(
      # p-values
      "p_val_mean"
      , "p_val_sd"
      , "p_val_fit"
    )
  )
```

```
##                  mean        sd 2.5% 50% 97.5% Rhat n.eff
## p_val_mean 0.5107333 0.4998931    0   1     1    1 26024
## p_val_sd   0.2030667 0.4022884    0   0     1    1 24098
## p_val_fit  0.3652000 0.4814942    0   0     1    1 24156
```

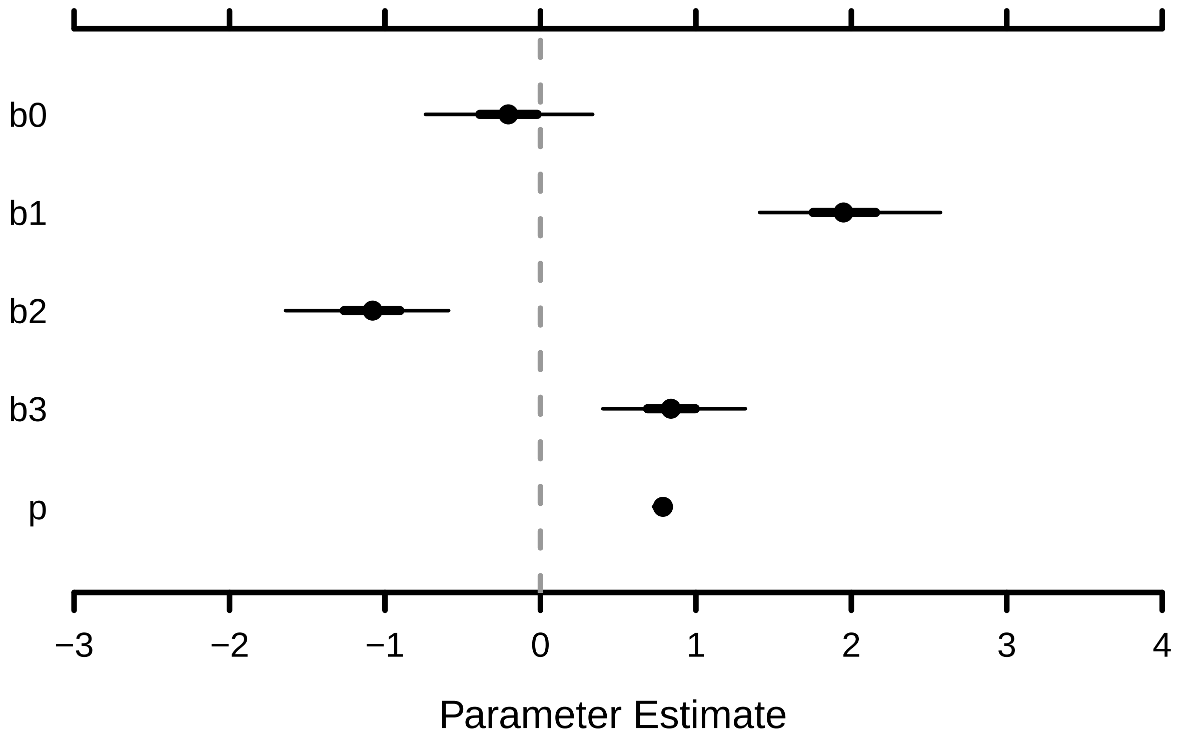
Estimates of the β and detection probability p

```
# summary
MCMCvis::MCMCsummary(zm, params = c("b0", "b1", "b2", "b3", "p"))
```

```
##          mean         sd       2.5%         50%      97.5% Rhat n.eff
## b0 -0.2050746 0.27492939 -0.7391919 -0.2065485  0.3358938    1  4147
## b1  1.9605920 0.29671505  1.4108448  1.9496562  2.5733944    1  4642
## b2 -1.0883756 0.26756690 -1.6390175 -1.0795142 -0.5905856    1  2919
## b3  0.8454065 0.23232915  0.4020145  0.8395416  1.3181528    1  6761
## p   0.7873659 0.02880759  0.7275522  0.7884421  0.8411666    1 14501
```

Caterpillar plot for the β and detection probability p

```
# Caterpillar plots
MCMCvis::MCMCplot(
  zm
  , params = c("b0", "b1", "b2", "b3", "p")
  , horiz = TRUE
)
```



Gelman-Rubin diagnostic

```
# Heidelberger and Welch diagnostic
coda::heidel.diag(zm)
```

What can you conclude about model fit?

Based on visual inspection of the trace plot the chains have converged. There is evidence of sufficient convergence of the β parameters. The values for Rhat near 1 in the summary table suggest convergence while the effective sample sizes are not too small. The Heidelberger and Welch diagnostic yields a "passed" half-width test.

Question 6

What can you conclude about the relative importance of elevation and forest cover in controlling the bird's distribution? Plot the median probability of occupancy and the 95% highest posterior density interval as a function of elevation at the mean of forest cover. Find the optimum elevation of willow tit habitat at the mean forest cover, where optimum elevation is defined as the elevation where probability of occupancy is maximum. Plot a normalized histogram of MCMC output for the optimum elevation at the average forest cover. Overlay 0.95 highest posterior density limits on the optimum elevation.

Conclusions

What can you conclude about the relative importance of elevation and forest cover in controlling the bird's distribution?

The covariate with the greatest effect on occupancy is elevation ($\beta_1 = 1.961$). The coefficient estimate of the impact of forest cover on occupancy ($\beta_3 = 0.845$) is approximately half as large in absolute value as the coefficent on elevation. Relative effect can be inferred by the absolute value of the coefficient estimates because the predictor data was standardized.

Plot the median probability of occupancy

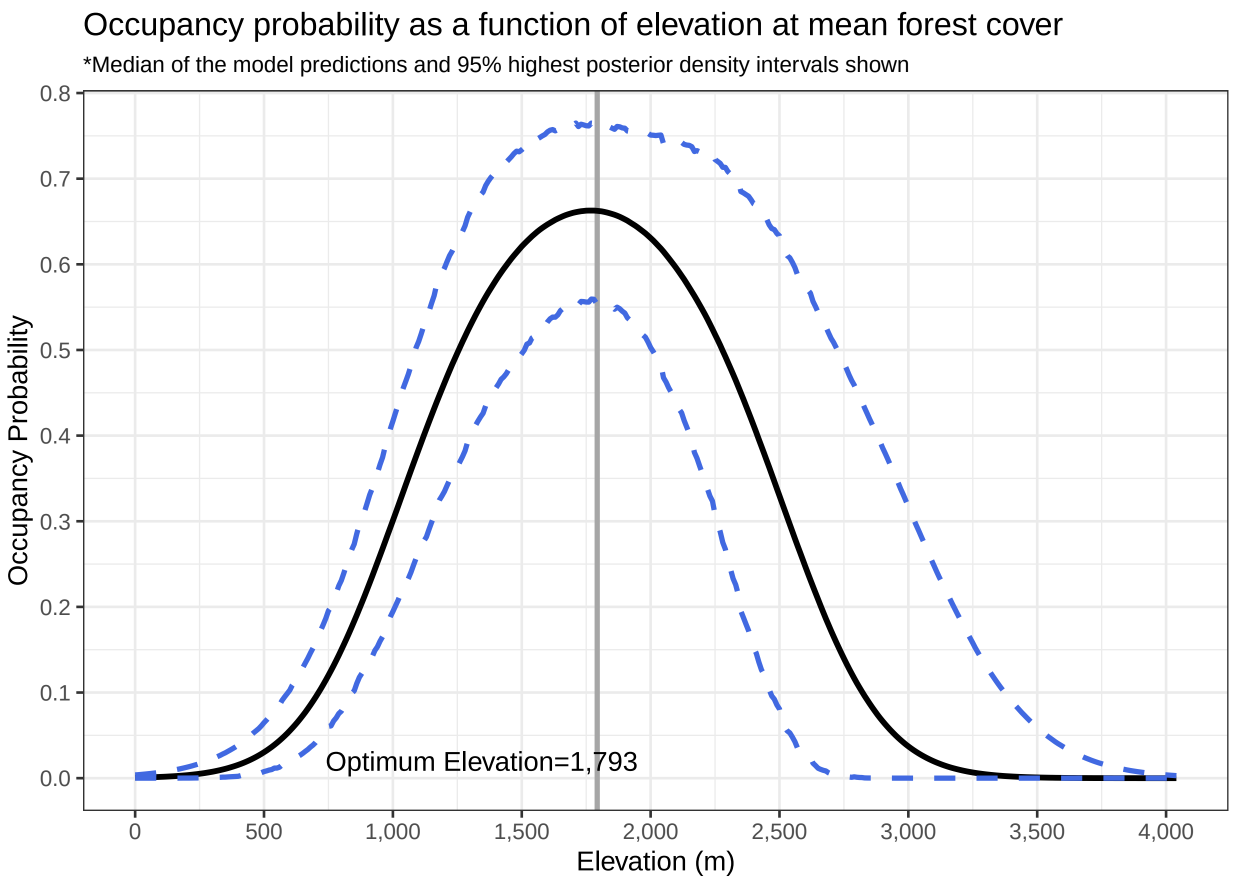
```
# View the data
# The estimated rate of occupancy
MCMCvis::MCMCpstr(zm, params = "p_est", func = function(x) quantile(x, c(0.025, 0.5, 0.975))) %>%
  as.data.frame() %>%
  dplyr::bind_cols(elevation_pred_z = elevation_pred_z) %>%
  dplyr::slice_head(n = 8)
```

```
##             p_est.2.5.    p_est.50. p_est.97.5. elevation_pred_z
## p_est[1] 5.146581e-05 0.0006202829 0.005141737        -1.829667
## p_est[2] 5.865969e-05 0.0006791418 0.005464212        -1.814195
## p_est[3] 6.644312e-05 0.0007438328 0.005789351        -1.798723
## p_est[4] 7.514366e-05 0.0008132628 0.006137498        -1.783251
## p_est[5] 8.518264e-05 0.0008889911 0.006516070        -1.767779
## p_est[6] 9.618715e-05 0.0009707971 0.006902504        -1.752307
## p_est[7] 1.091605e-04 0.0010599801 0.007314567        -1.736835
## p_est[8] 1.233629e-04 0.0011579026 0.007757877        -1.721363
```

Plot the median probability of occupancy and the 95% highest posterior density interval as a function of elevation at the mean of forest cover.

```
# optimum elevation mean
optimum_elevation_mean_temp <- MCMCvis::MCMCsummary(zm, params = c("optimum_elev"))[["mean"]]
# data
dplyr::bind_cols(
  elevation_pred = elevation_pred
  , median_p_est = MCMCvis::MCMCpstr(zm, params = "p_est", func = median) %>% unlist()
  , MCMCvis::MCMCpstr(zm, params = "p_est", func = function(x) HDInterval::hdi(x, credMass = 0.95)) %>%
      as.data.frame()
) %>%
# plot
ggplot(data = .) +
  geom_vline(
    mapping = aes(xintercept = optimum_elevation_mean_temp)
    , lwd = 1
    , color = "gray65"
  ) +
  annotate(
    "text"
    , x = optimum_elevation_mean_temp*0.75
    , y = 0.02
```

```r
    , label = paste0("Optimum Elevation=", scales::comma(optimum_elevation_mean_temp, accuracy=1))
    , size = 4
  ) +
  geom_line(
    mapping = aes(x = elevation_pred, y = median_p_est)
    , color = "black"
    , lwd = 1.1
  ) +
  geom_line(
    mapping = aes(x = elevation_pred, y = p_est.upper)
    , color = "royalblue"
    , lwd = 1
    , linetype = "dashed"
  ) +
  geom_line(
    mapping = aes(x = elevation_pred, y = p_est.lower)
    , color = "royalblue"
    , lwd = 1
    , linetype = "dashed"
  ) +
  scale_y_continuous(breaks = scales::extended_breaks(n=10)) +
  scale_x_continuous(breaks = scales::extended_breaks(n=10), labels = scales::comma) +
  xlab("Elevation (m)") +
  ylab("Occupancy Probability") +
  labs(
    title = "Occupancy probability as a function of elevation at mean forest cover"
    , subtitle = "*Median of the model predictions and 95% highest posterior density intervals shown"
  ) +
  theme_bw() +
  theme(
    plot.subtitle = element_text(size = 9)
  )
```



Estimate of optimum elevation

Find the optimum elevation of willow tit habitat at the mean forest cover, where optimum elevation is defined as the elevation where probability of occupancy is maximum.

```
# summary
MCMCvis::MCMCsummary(zm, params = c("optimum_elev"))
```

```
##                  mean       sd     2.5%      50%    97.5% Rhat n.eff
## optimum_elev 1792.744 287.2808 1589.182 1767.026 2143.317  1.2 11335
```

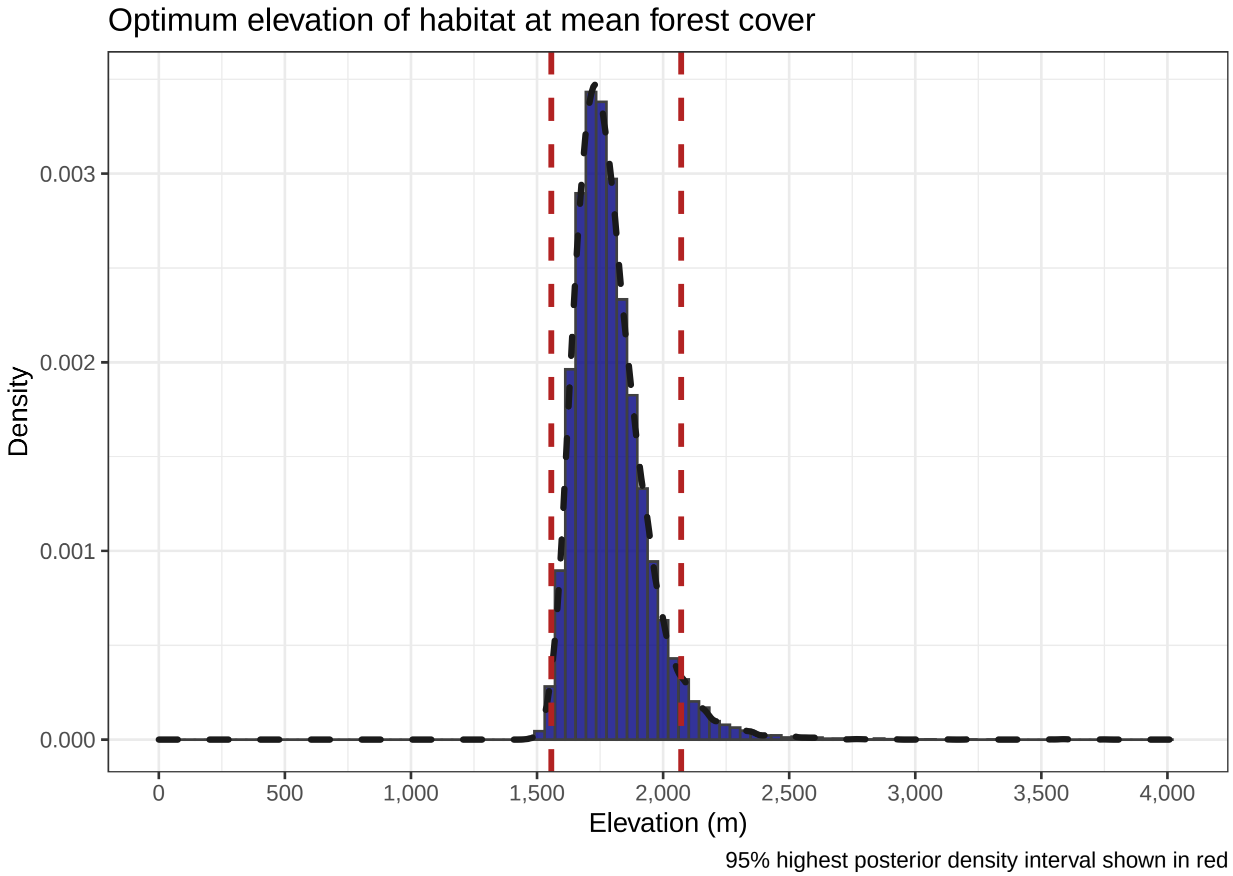
Plot a normalized histogram for the optimum elevation

Plot a normalized histogram of MCMC output for the optimum elevation at the average forest cover. Overlay 0.95 highest posterior density limits on the optimum elevation.

```
# extract data
dta_temp <- MCMCvis::MCMCchains(zm, params = c("optimum_elev")) %>%
  as.data.frame()
hdi_temp <- HDInterval::hdi(dta_temp$optimum_elev, credMass = 0.95)

# the marginal posterior density of occupancy
  # plot
```

```
ggplot(data = dta_temp, mapping = aes(x = optimum_elev)) +
geom_histogram(
  aes(y = ..density..)
  , bins = 100
  , fill = "navy"
  , alpha = 0.8
  , color = "gray25"
) +
geom_density(
  aes(y = ..density..)
  , linetype = 2
  , lwd = 1.2
  , color = "gray10"
) +
geom_vline(
  xintercept = hdi_temp
  , color = "firebrick"
  , linetype = "dashed"
  , lwd = 1.1
) +
scale_x_continuous(
  breaks = scales::extended_breaks(n=9)
  , limits = c(min(elevation_pred), max(elevation_pred))
  , labels = scales::comma
) +
xlab("Elevation (m)") +
ylab("Density") +
labs(
  title = "Optimum elevation of habitat at mean forest cover"
  , caption = "95% highest posterior density interval shown in red"
) +
theme_bw()
```



Optimum elevation of habitat at mean forest cover

95% highest posterior density interval shown in red