Lab 11: Swiss Birds Occupancy Modeling Lab

Team USA: Yuting Deng, Kendra Gilbertson, Lily Durkee, Bennett Hardy

Hermione.Deng@colostate.edu, kendra01@colostate.edu, L.Durkee@colostate.edu,Bennett.Hardy@colosta

2022-11-14

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. 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.

We want to understand the influence of forest cover and elevation on the distribution of the willow tit. The data frame 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.

1. Diagram the network of knowns and unknowns.

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

$$\left[p, \mathbf{z}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma^2 \mid \mathbf{y}\right] \propto \prod_{i=1}^{237} \text{binomial}\big(y_i \mid p \cdot z_i, n_i\big) \times \text{Bernoulli}\big(z_i \mid \psi\big) \times \prod_{j=1}^{4} \text{normal}\big(\beta_j \mid 0, 10000\big) \times \text{beta}\big(p \mid 1, 1\big)$$

$$\psi = \text{logit}^{-1}\big(\,\beta_1 + \beta_2 \mathrm{x}_{1i} + \beta_3 \mathrm{x}_{2i} + \beta_4 \mathrm{x}_{3i}^2\big)$$

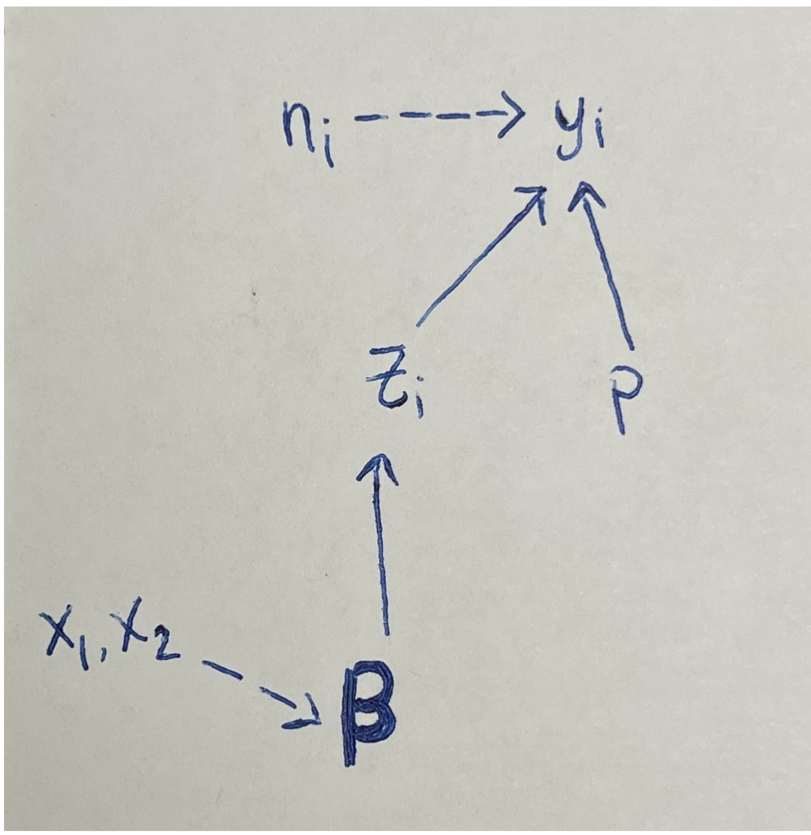


Figure 1: Directed acyclic graph for bird occupancy.

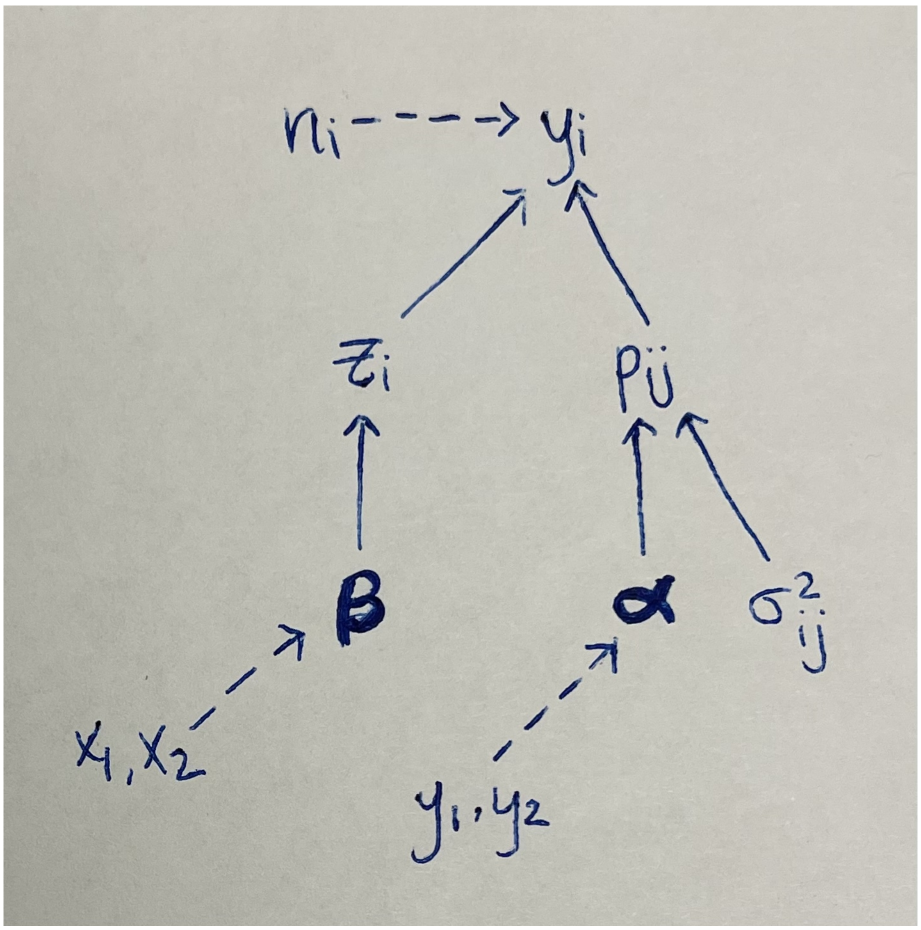


Figure 2: Directed acyclic graph for the joint distribution of bird occupancy including search time and wind speed.

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.

$$
\begin{aligned}
\left[p, \mathbf{z}, \boldsymbol{\beta}, \mu, \sigma^2 \mid \mathbf{y}\right] \propto & \prod_{i=1}^{237} \text{binomial}\big(y_i \mid p \cdot z_i, n_i\big) \times \text{Bernoulli}\big(z_i \mid \psi\big) \times \text{beta}\big(p \mid \mu, \sigma^2\big) \\
& \times \prod_{j=1}^{4} \text{normal}\big(\beta_j \mid 0, 10000\big) \times \prod_{k=1}^{3} \text{normal}\big(\alpha_k \mid 0, 10000\big) \times \text{inverse gamma}\big(\sigma^2 \mid 0.001, 0.001\big) \\
\psi = & \, \text{logit}^{-1}\big(\,\beta_1 + \beta_2 \text{x}_{1i} + \beta_3 \text{x}_{2i} + \beta_4 \text{x}_{3i}^2\big) \\
\mu = & \, \text{logit}^{-1}\big(\alpha_1 + \alpha_2 \text{y}_{1i} + \alpha_3 \text{y}_{2i}\big)
\end{aligned}
$$

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.

```r
# QUESTION 4
df = SwissBirds
elev.pred = seq(250, 2750, 10)
data = list(
 #as.vector help preserving only the numbers
  forest = as.double(as.vector(scale(df$forestCover))),
  elevation = as.double(as.vector(scale(df$elevation))),
  n.visit = as.double(df$numberVisits),
  y = as.double(df$numberDetections),
  sd.elev = sd(df$elevation),
  mu.elev = mean(df$elevation),
  elev.pred = as.double(as.vector(scale(seq(250, 2750, 10)))))
inits = list(
  list(z = rep(1, nrow(SwissBirds)), p = runif(1, 0, 1), beta = runif(4, -2, 2)),
  list(z = rep(1, nrow(SwissBirds)), p = runif(1, 0, 1), beta = runif(4, -2, 2)),
  list(z = rep(1, nrow(SwissBirds)), p = runif(1, 0, 1), beta = runif(4, -2, 2)))
# initial value cannot use p=1 (error:Node inconsistent with parents)
```

```r
{
sink("Lab_11_SwissBirds.R")
cat("
model{
  # priors

  p ~ dbeta(1,1)

  for (i in 1:4){
    beta[i] ~ dnorm(0, 0.0001)
  }
  # likelihood

  for (i in 1:237) {
    z[i] ~ dbern(phi[i])
```

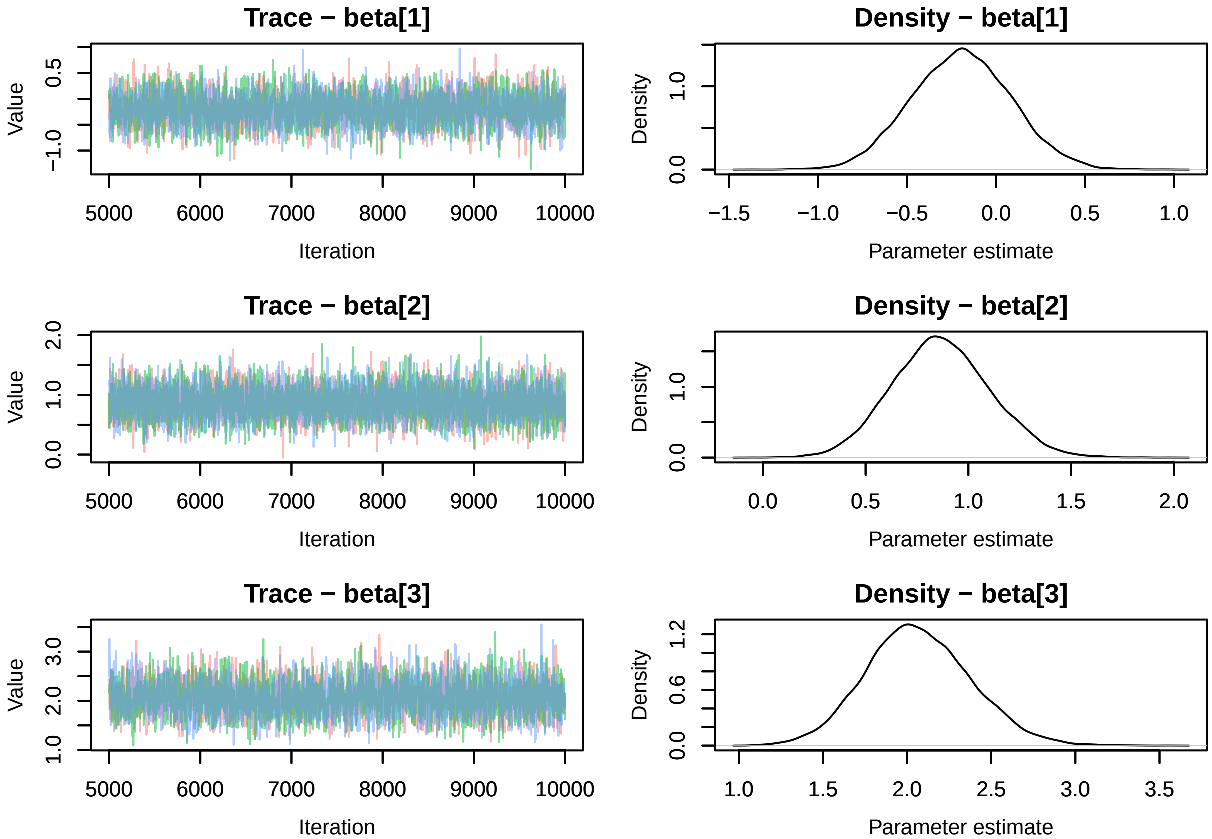
```
    logit(phi[i]) = beta[1] + beta[2]*forest[i] + beta[3]*elevation[i]
                            + beta[4]*elevation[i]^2
    y[i] ~ dbin(p*z[i], n.visit[i])

    # simulated data for posterior predictive checks
    y.sim[i] ~ dbin(p*z[i], n.visit[i])
  }

  # bayesian p values

  sd.data <- sd(y)
  sd.sim <- sd(y.sim)
  p.sd <- step(sd.sim - sd.data)
  mean.data <- mean(y)
  mean.sim  <- mean(y.sim)
  p.mean <- step(mean.sim - mean.data)

  # dedrived quantity

  ele_max = beta[3]/(-2*beta[4]) #second drivative
  ele_max_ori_scale = ele_max*sd.elev + mu.elev

  for (j in 1:length(elev.pred)){
  logit(psi.pred[j]) = beta[1] +  beta[3]*elev.pred[j] + beta[4]*elev.pred[j]^2
  # at the mean of forest cover, since it is scaled, mean = 0
  }

}

",fill = TRUE)
sink()
}
```

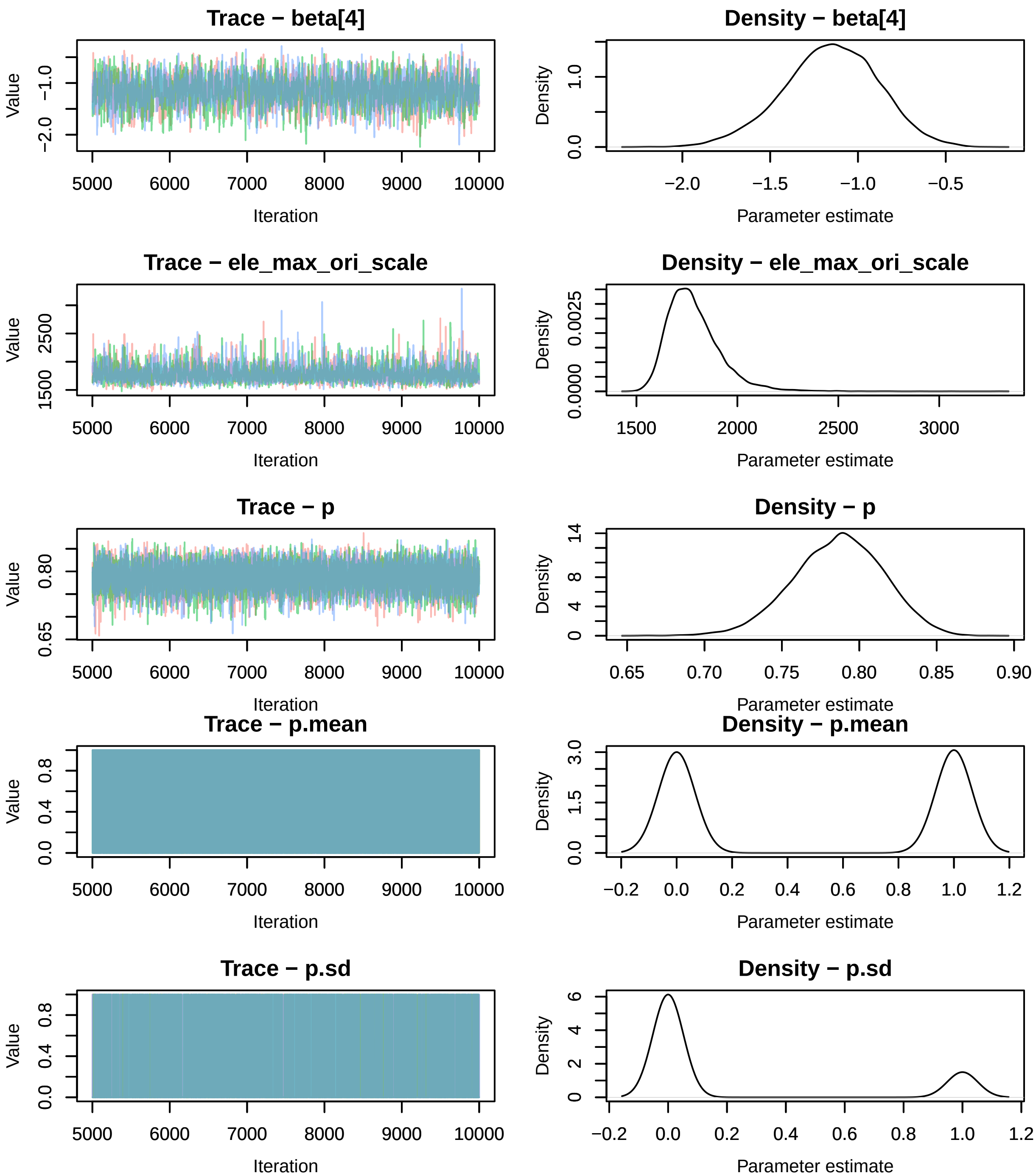
5. Summarize the parameters and check chains for convergence. Exclude the predictions of ψ from the summary if they were included in the coda object. What can you conclude about model fit?

Based on our trace plots and Rhat values, we conclude that our model converged.

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 237
##    Unobserved stochastic nodes: 479
##    Total graph size: 3998
##
## Initializing model
##                        mean      sd     2.5%      50%    97.5% Rhat n.eff
## beta[1]              -0.196  0.2800   -0.748   -0.194    0.349    1  4092
## beta[2]               0.868  0.2370    0.413    0.863    1.340    1  6527
## beta[3]               2.070  0.3160    1.480    2.050    2.720    1  4463
```

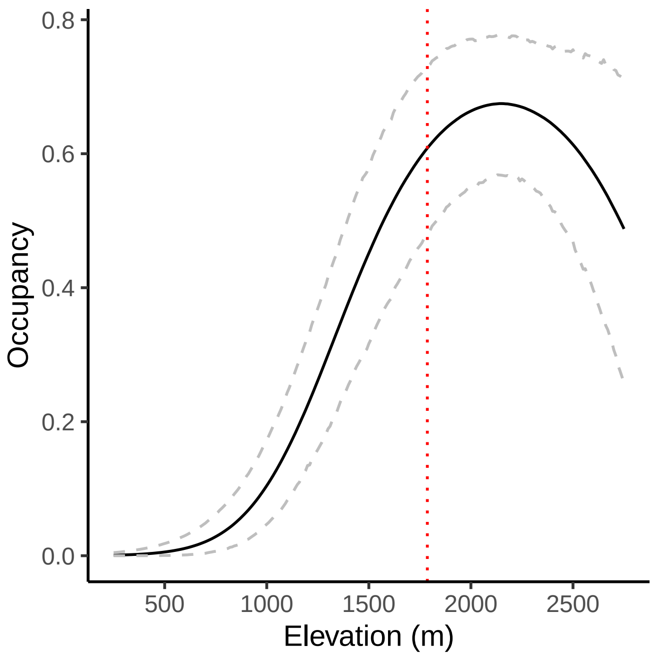
```
## beta[4]             -1.150   0.2760   -1.720   -1.150   -0.631     1  2941
## ele_max_ori_scale 1790.000 140.0000 1590.000 1760.000 2120.000     1  4105
## p                    0.786   0.0292    0.727    0.787    0.841     1 14274
## p.mean               0.505   0.5000    0.000    1.000    1.000     1 23925
## p.sd                 0.198   0.3980    0.000    0.000    1.000     1 24394
```

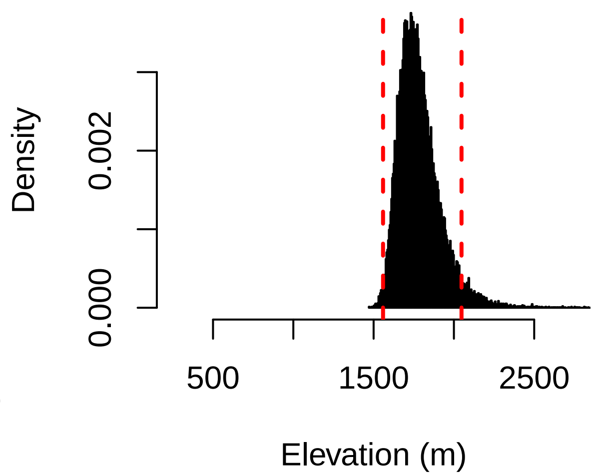



Trace − beta[4]
Value
−2.0
−1.0
5000
6000
7000
8000
9000
10000
Iteration
Density − beta[4]
Density
0.0
1.0
−2.0
−1.5
−1.0
−0.5
Parameter estimate
Trace − ele_max_ori_scale
Value
1500
2500
5000
6000
7000
8000
9000
10000
Iteration
Density − ele_max_ori_scale
Density
0.0000
0.0025
1500
2000
2500
3000
Parameter estimate
Trace − p
Value
0.65
0.80
5000
6000
7000
8000
9000
10000
Iteration
Density − p
Density
0
4
8
14
0.65
0.70
0.75
0.80
0.85
0.90
Parameter estimate
Trace − p.mean
Value
0.0
0.4
0.8
5000
6000
7000
8000
9000
10000
Iteration
Density − p.mean
Density
0.0
1.5
3.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Parameter estimate
Trace − p.sd
Value
0.0
0.4
0.8
5000
6000
7000
8000
9000
10000
Iteration
Density − p.sd
Density
0
2
4
6
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Parameter estimate

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.

Based on our beta estimates from our scaled parameters, elevation was more important than forest cover in controlling the bird's distribution. When we maximize occupancy at mean forest cover, the optimum elevatio is approximately 1800m.





Code

```
knitr::opts_chunk$set(
    echo = FALSE,
    message = FALSE,
    warning = FALSE,
    attr.source = ".numberLines"
)
library(rjags)
library(MCMCvis)
library(HDInterval)
library(BayesNSF)
library(ggplot2)
set.seed(10)
# QUESTION 4
df = SwissBirds
elev.pred = seq(250, 2750, 10)
data = list(
 #as.vector help preserving only the numbers
```

```r
  forest = as.double(as.vector(scale(df$forestCover))),
  elevation = as.double(as.vector(scale(df$elevation))),
  n.visit = as.double(df$numberVisits),
  y = as.double(df$numberDetections),
  sd.elev = sd(df$elevation),
  mu.elev = mean(df$elevation),
  elev.pred = as.double(as.vector(scale(seq(250, 2750, 10)))))
inits = list(
  list(z = rep(1, nrow(SwissBirds)), p = runif(1, 0, 1), beta = runif(4, -2, 2)),
  list(z = rep(1, nrow(SwissBirds)), p = runif(1, 0, 1), beta = runif(4, -2, 2)),
  list(z = rep(1, nrow(SwissBirds)), p = runif(1, 0, 1), beta = runif(4, -2, 2)))
# initial value cannot use p=1 (error:Node inconsistent with parents)
{
sink("Lab_11_SwissBirds.R")
cat("
model{
  # priors

  p ~ dbeta(1,1)

  for (i in 1:4){
    beta[i] ~ dnorm(0, 0.0001)
  }
  # likelihood

  for (i in 1:237) {
    z[i] ~ dbern(phi[i])
    logit(phi[i]) = beta[1] + beta[2]*forest[i] + beta[3]*elevation[i]
                            + beta[4]*elevation[i]^2
    y[i] ~ dbin(p*z[i], n.visit[i])

    # simulated data for posterior predictive checks
    y.sim[i] ~ dbin(p*z[i], n.visit[i])
  }

  # bayesian p values

  sd.data <- sd(y)
  sd.sim <- sd(y.sim)
  p.sd <- step(sd.sim - sd.data)
  mean.data <- mean(y)
  mean.sim  <- mean(y.sim)
  p.mean <- step(mean.sim - mean.data)

  # dedrived quantity

  ele_max = beta[3]/(-2*beta[4]) #second drivative
  ele_max_ori_scale = ele_max*sd.elev + mu.elev

  for (j in 1:length(elev.pred)){
  logit(psi.pred[j]) = beta[1] +  beta[3]*elev.pred[j] + beta[4]*elev.pred[j]^2
  # at the mean of forest cover, since it is scaled, mean = 0
  }
```

```r

}

",fill = TRUE)
sink()
}
# QUESTION 5
n.adapt = 3000
n.update = 10000
n.iter = 10000
jm.check = jags.model(file="Lab_11_SwissBirds.R",
                      data = data,
                      n.adapt = n.adapt,
                      inits = inits,
                      n.chains = length(inits))
update(jm.check, n.iter = n.update)
zc.check = coda.samples(jm.check,
                        variable.names = c("p.sd", "p.mean", "y.sim",
                                           "beta", "ele_max_ori_scale",
                                           "psi.pred", "p"),
                        n.iter = n.iter)
MCMCsummary(zc.check, n.eff = TRUE, excl = c("psi.pred", "y.sim"), digits = 3)
MCMCtrace(zc.check, excl = c("psi.pred", "y.sim"), pdf=FALSE)
# QUESTION 6
#median probability of occupancy and the 95% highest posterior density interval
psi.median <- MCMCpstr(zc.check, params=c("psi.pred"), func = median)
psi.95 <-  MCMCpstr(zc.check, params=c("psi.pred"), func = function(x) hdi(x, .95))
# .95 highest posterior density limits on the optimum elevation
elev.95 = MCMCpstr(zc.check, params=c("ele_max_ori_scale"), func = function(x) hdi(x,.95))
plot.df = data.frame(elev.pred, psi.median, psi.95)
ggplot(plot.df)+
  geom_line(aes(x=elev.pred, y=psi.pred))+
  geom_line(aes(x=elev.pred, y=psi.pred.lower), lty = "dashed", col = "grey")+
  geom_line(aes(x=elev.pred, y=psi.pred.upper), lty = "dashed", col = "grey")+
  geom_vline(xintercept = mean(MCMCchains(zc.check, "ele_max_ori_scale")),
             linetype="dotted", color = "red")+
  labs(x="Elevation (m)", y="Occupancy")+
  scale_x_continuous(n.breaks = 8)+
  theme_classic()
hist(MCMCchains(zc.check, "ele_max_ori_scale"),
     breaks = 500, main = "",
     xlab = "Elevation (m)",
     xlim = c(250, 2750),
     freq = FALSE)
abline(v= elev.95$ele_max_ori_scale[1], lty = "dashed", lwd = 2, col="red")
abline(v = elev.95$ele_max_ori_scale[2], lty = "dashed", lwd = 2, col="red")
# this R markdown chunk generates a code appendix
```