ESS 575: Bayesian Regression Lab

Team England

17 October, 2022

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

Setup

In this lab you will practice building regression models that are appropriate for a given data set.

- Coexistance data
- Abundance data

Load the data

```r
coexist_pth <- "https://nthobbs50.github.io/ESS575/content/labs/coexist.csv"
coverage_pth <- "https://nthobbs50.github.io/ESS575/content/labs/hesp_coverage.csv"

coexist <- read.csv(coexist_pth)
coverage <- read.csv(coverage_pth)
```

Problem A.

Hein et. al (2012) investigated how environmental conditions influence coexistence between Arctic char (*Salmo trutta*) and pike (*Esox lucius*) in Swedish lakes. Pike were introduced to 151 lakes containing brown trout. Coexistence of the two species was recorded, as $yi = 1$ if both species were found in lake i and 0 otherwise. For each lake, five environmental conditions deemed relevant to coexistence patterns were observed:

- elevation
- upstream catchment area

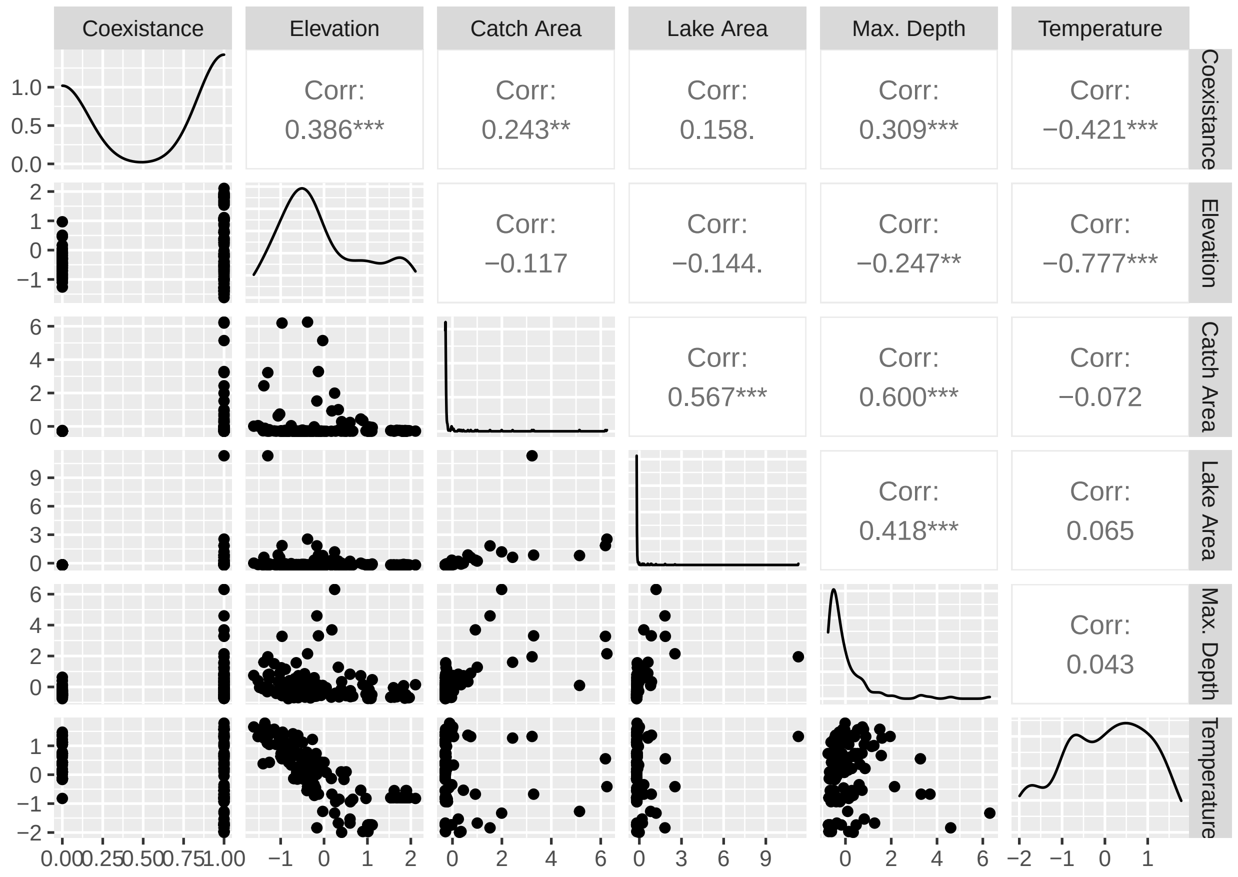
- maximum area
- maximum depth
- mean annual air temperature at outlet

The predictors have been standardized.

Question 1

Plot the data using `pairs` or `GGally::ggpairs`. What do you notice? How might this impact your modeling choices?

```r
GGally::ggpairs(
  data = coexist
  , columns = c("coexist", "elev", "catcharea", "lakearea", "maxdepth", "temp1")
  , columnLabels = c("Coexistance", "Elevation", "Catch Area", "Lake Area", "Max. Depth", "Temperature"
)
```



This grid of plots given by `GGally` shows the empirical density (a.k.a. marginal distribution) of each variable on the diagonal, the scatterplot of points for pairs of variables on the lower triangle, and the Pearson correlation between variables in the upper right triangle.

We notice that the coexist outcome variable takes on two values (0 and 1). We also notice that there is one relatively large lake in the data set. A binary data model is appropriate given the support of the data.

Question 2

We seek to understand the relationship between the probability of coexistence and the five environmental covariates. Write out a reasonable data model.

$$y_i \sim \textsf{Bernoulli}\big(p_i\big)$$

where:

$$p_i = g(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, x_i) = \text{inverse logit}\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_5 x_{5i}\big) = \frac{\exp\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_5 x_{5i}\big)}{1 + \exp\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_5 x_{5i}\big)}$$

Question 3

Assume we know very little about the impact of the environmental covariates on coexistence. Specify appropriate prior distributions for all unknown parameters. Explain your choice of priors.

Assuming vague priors on the intercept and slope can be accomplished by setting the variance $\sigma^2 = 2.7$ normally distributed with a mean of 0; e.g., $\beta_0 \sim \textsf{normal}(0, 2.7),\ \beta_1 \sim \textsf{normal}(0, 2.7), \cdots,\ \beta_1 \sim \textsf{normal}(0, 2.7)$. Such that:

$$\big[\beta_0, \beta_1, \cdots, \beta_5 \mid \mathbf{y}\big] \propto \prod_{i=1}^{n} \textsf{Bernoulli}\big(y_i \mid g(\beta_0, \beta_1, \cdots, \beta_5, x_i)\big) \times \textsf{normal}\big(\beta_0 \mid 0, 2.7\big) \times \big(\beta_1 \mid 0, 2.7\big) \times \cdots \times \textsf{normal}\big(\beta_5 \mid 0, 2.7\big)$$

Question 4

Write the expression for the posterior in terms of the joint distribution of the parameters and data.

$$\big[\beta_0, \beta_1, \cdots, \beta_5 \mid \mathbf{y}\big] \propto \prod_{i=1}^{n} \textsf{Bernoulli}\big(y_i \mid g(\beta_0, \beta_1, \cdots, \beta_5, x_i)\big) \times \textsf{normal}\big(\beta_0 \mid 0, 2.7\big) \times \textsf{normal}\big(\beta_1 \mid 0, 2.7\big) \times \cdots \times \textsf{normal}\big(\beta_5 \mid 0, 2.7\big)$$

where:

$$p_i = g(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, x_i) = \text{inverse logit}\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_5 x_{5i}\big) = \frac{\exp\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_5 x_{5i}\big)}{1 + \exp\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_5 x_{5i}\big)}$$

Question 5

Write out the JAGS code for the model.

```r
# define explanatory variables
exp_vars <- c("elev", "catcharea", "lakearea", "maxdepth", "temp1")
# list of data
data = list(
  n = nrow(coexist) # n is required in the JAGS program to index the for structure
  , y = as.double(coexist$coexist)
  , x_mtrx = as.matrix(
        coexist %>%
          dplyr::mutate(intercept = 1) %>%
          dplyr::select(c("intercept", exp_vars)) %>%
          #the execution of JAGS is about 5 times faster on double precision than on integers.
          dplyr::mutate_if(is.numeric, as.double)
      )
)

## JAGS Model
model{
  # priors
  b0 ~ dnorm(0, (1/2.7))
  b1 ~ dnorm(0, (1/2.7))
  b2 ~ dnorm(0, (1/2.7))
  b3 ~ dnorm(0, (1/2.7))
  b4 ~ dnorm(0, (1/2.7))
  b5 ~ dnorm(0, (1/2.7))
  # likelihood
  for (i in 1:n) {
    p[i] <- ilogit(
       b0
       + b1*x_mtrx[i,2] # or b1*x_mtrx[i,"elev"]
       + b2*x_mtrx[i,3] # or b2*x_mtrx[i,"catcharea"]
       + b3*x_mtrx[i,4]
       + b4*x_mtrx[i,5]
       + b5*x_mtrx[i,6]
    )
    y[i] ~ dbern(p[i])
  }
}
```

Problem B

Kembel and Cahill Jr. (2011) collected data from temperate grassland plant communities in Alberta, Canada. Twenty-seven plots are established, and the slope, slope position, aspect, and relative moisture of each plot is recorded. For each plot, the abundance of several species is recorded, interpreted as the proportion of land area covered by the species. The land coverage by needle-and-thread grass (*Hesperostipa comata* ssp. *comata*) is considered here.

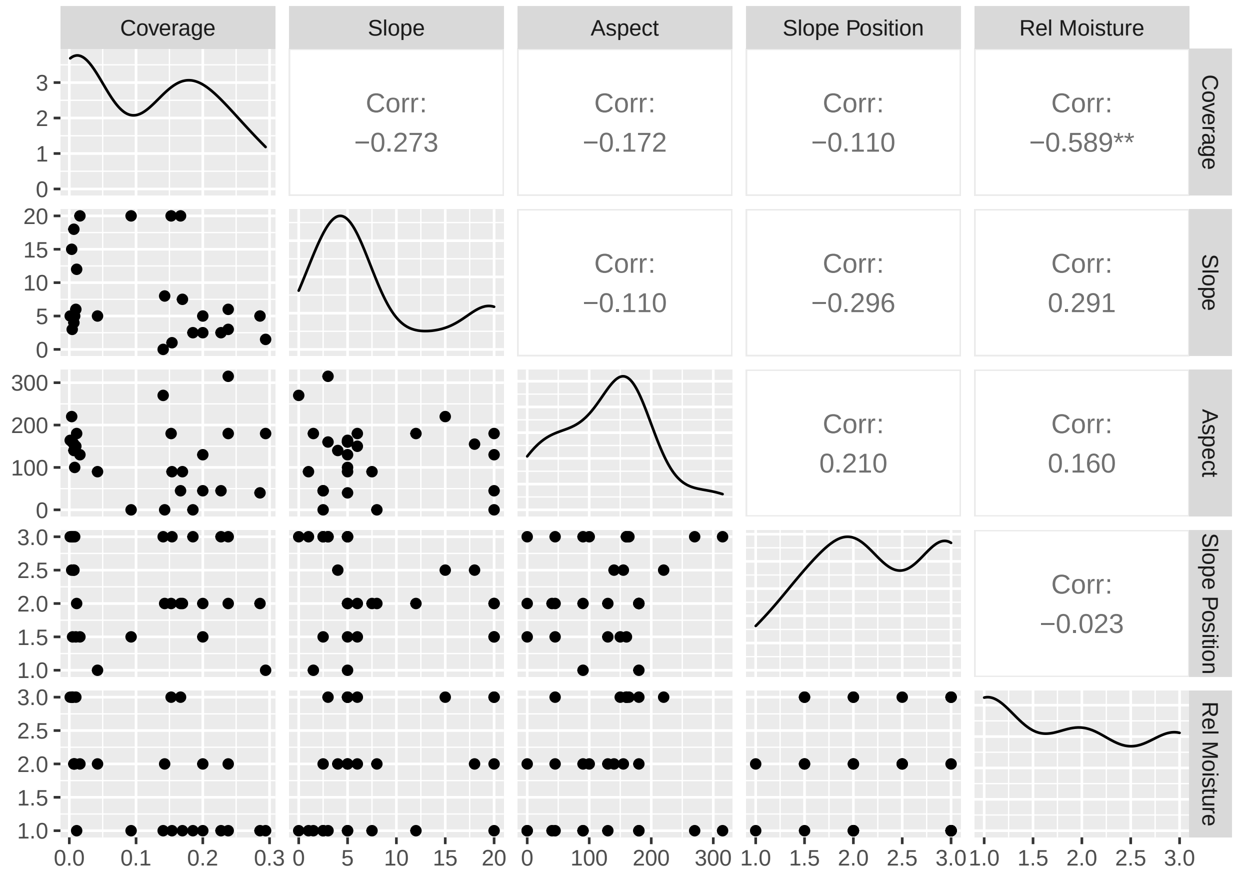
Question 1

Plot the data using `pairs` or `GGally::ggpairs`. What do you notice? How might this impact your modeling choices?

```
my_vars <- c("coverage", "slope", "aspect", "slope_position", "rel_moisture")
# plot
coverage %>% names
```

```
## [1] "coverage"       "slope"          "aspect"         "slope_position"
## [5] "rel_moisture"
```

```
GGally::ggpairs(
  data = coverage
  , columns = my_vars
  , columnLabels = my_vars %>% stringr::str_replace_all("[[:punct:]]", " ") %>% stringr::str_to_title()
)
```



Slope position and relative moisture appear to be discrete data (not continuous). Slope should be costrainted to values between and including 0-180 while aspect should be constrained to values between and including 0-360. Coverage is a proportion and can take on any value between and including 0 and 1.

Question 2

Let y_i represent the proportion of land covered by *Hesperostipa comata* ssp. *comata*. What deterministic model would you use to predict the mean of y_i as a function of four covariates $x_1 \cdots x_4$?

$$y \sim \text{beta}\big(\alpha, \beta\big)$$

deterministic model of μ:

$$\mu_i = g(\beta_0, \beta_1, \cdots, \beta_4, x_i) = \text{inverse logit}\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_4 x_{4i}\big) = \frac{\exp\left(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_4 x_{4i}\right)}{1 + \exp\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_4 x_{4i}\big)}$$

Question 3

Write the likelihood for an observation, y_i.

The likelihood function for y_i is:

$$y_i \sim \mathsf{beta}\big(a_i, b_i\big)$$

where:

$$a_i = \frac{\mu_i^2 - \mu_i^3 - \mu_i\sigma^2}{\sigma^2}$$
$$b_i = \frac{\mu_i - 2\mu_i^2 + \mu_i^3 - \sigma^2 + \mu_i\sigma^2}{\sigma^2}$$

Question 4

What would you use for vague priors on the coefficients to assure that the prior on μ_i is vague?

Assuming vague priors on the intercept and slope can be accomplished by setting the variance $\sigma^2 = 2.7$ normally distributed with a mean of 0; e.g., $\beta_0 \sim \mathsf{normal}(0, 2.7),\ \beta_1 \sim \mathsf{normal}(0, 2.7), \cdots,\ \beta_1 \sim \mathsf{normal}(0, 2.7)$. Such that:

$$\big[\beta_0, \beta_1, \cdots, \beta_4, \sigma^2 \mid \mathbf{y}\big] \propto \prod_{i=1}^{n} \mathsf{beta}\big(y_i \mid a_i, b_i\big) \times \mathsf{normal}\big(\beta_0 \mid 0, 2.7\big) \times \cdots \times \mathsf{normal}\big(\beta_4 \mid 0, 2.7\big) \times \mathsf{uniform}\big(\sigma \mid 0, 100\big)$$

Question 5

Express the posterior distribution as proportional to joint distribution for your model.

$$y_i \sim \mathsf{beta}\big(a_i, b_i\big)$$
$$a_i = \frac{\mu_i^2 - \mu_i^3 - \mu_i\sigma^2}{\sigma^2}$$
$$b_i = \frac{\mu_i - 2\mu_i^2 + \mu_i^3 - \sigma^2 + \mu_i\sigma^2}{\sigma^2}$$
$$\mu_i = g(\beta_0, \beta_1, , \cdots, \beta_4, x_i) = \text{inverse logit}\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_4 x_{4i}\big) = \frac{\exp\left(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_4 x_{4i}\right)}{1 + \exp\big(\beta_0 + \beta_1 x_{1i} + \cdots + \beta_4 x_{4i}\big)}$$
$$\big[\beta_0, \beta_1, \cdots, \beta_4, \sigma^2 \mid \mathbf{y}\big] \propto \prod_{i=1}^{n} \mathsf{beta}\big(y_i \mid a_i, b_i\big) \times \mathsf{normal}\big(\beta_0 \mid 0, 2.7\big) \times \cdots \times \mathsf{normal}\big(\beta_4 \mid 0, 2.7\big) \times \mathsf{uniform}\big(\sigma \mid 0, 100\big)$$

Question 6

Write the JAGS code for the model. Fit the model to the data with predictor variables standardized. Do a posterior predictive check for the mean and the standard deviation. Interpret the coefficients. Which covariate has the greatest effect?

JAGS Model

```
# define explanatory variables
exp_vars <- c("slope", "aspect", "slope_position", "rel_moisture")
# list of data
data = list(
  n = nrow(coverage) # n is required in the JAGS program to index the for structure
  , y = as.double(coverage$coverage)
  , x_mtrx = as.matrix(
        coverage %>%
          dplyr::mutate(dplyr::across(exp_vars, scale)) %>%
          dplyr::mutate(intercept = 1) %>%
          dplyr::select(c("intercept", exp_vars)) %>%
          #the execution of JAGS is about 5 times faster on double precision than on integers.
          dplyr::mutate_if(is.numeric, as.double)
      )
)

## JAGS Model
model{
  # priors
  sigma ~ dunif(0, 1) # dunif(alpha = lower limit, beta = upper limit)
  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))
  b4 ~ dnorm(0, (1/2.7))
  # likelihood
  for (i in 1:n) {
    # deterministic model
    mu[i] <- ilogit(
       b0
       + b1*x_mtrx[i,2] # or b1*x_mtrx[i,"slope"]
       + b2*x_mtrx[i,3] # or b2*x_mtrx[i,"aspect"]
       + b3*x_mtrx[i,4]
       + b4*x_mtrx[i,5]
    )
    # likelihood parameters
    a[i] <- (mu[i]^2 - mu[i]^3 - mu[i]*sigma^2) / sigma^2
    b[i] <- (mu[i] - 2*mu[i]^2 + mu[i]^3 - sigma^2 + mu[i]*sigma^2) / sigma^2
    # likelihood
      # returns density (for continuous) because l.h.s. is data (deterministic b/c defined in data)
      y[i] ~ dbeta(a[i], b[i])
    # posterior predictive distribution of y.new (for model checking)
      # returns random number generator because l.h.s. is not data (i.e. it is unknown "stochastic node
      y_sim[i]  ~ dbeta(a[i], b[i])
  }
  # Derived quantities
  #posterior predictive checks
    mean_y = mean(y)
    sd_y = sd(y)
    mean_y_sim = mean(y_sim)
    sd_y_sim = sd(y_sim)
    p_val_mean = step(mean_y_sim - mean_y)
    p_val_sd = step(sd_y_sim - sd_y)
}
```

Implement JAGS Model

```
#################################################################
# insert JAGS model code into an R script
#################################################################
{ # Extra bracket needed only for R markdown files - see answers
  sink("GrassCovJAGS.R") # This is the file name for the jags code
  cat("
  model{
    # priors
    sigma ~ dunif(0, 1) # dunif(alpha = lower limit, beta = upper limit)
    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))
    b4 ~ dnorm(0, (1/2.7))
    # likelihood
    for (i in 1:n) {
      # deterministic model
      mu[i] <- ilogit(
         b0
         + b1*x_mtrx[i,2]
         + b2*x_mtrx[i,3]
         + b3*x_mtrx[i,4]
         + b4*x_mtrx[i,5]
      )
      # likelihood parameters
      a[i] <- (mu[i]^2 - mu[i]^3 - mu[i]*sigma^2) / sigma^2
      b[i] <- (mu[i] - 2*mu[i]^2 + mu[i]^3 - sigma^2 + mu[i]*sigma^2) / sigma^2
      # likelihood
        # returns density (for continuous) because l.h.s. is data (deterministic b/c defined in data)
        y[i] ~ dbeta(a[i], b[i])
      # posterior predictive distribution of y.new (for model checking)
        # returns random number generator because l.h.s. is not data (i.e. it is unknown/stochastic nod
        y_sim[i]  ~ dbeta(a[i], b[i])
    }
    # Derived quantities
    #posterior predictive checks
      mean_y = mean(y)
      sd_y = sd(y)
      mean_y_sim = mean(y_sim)
      sd_y_sim = sd(y_sim)
      p_val_mean = step(mean_y_sim - mean_y)
      p_val_sd = step(sd_y_sim - sd_y)
  }
  ", fill = TRUE)
  sink()
}
################################################################
# implement model
```

```
##################################################################
# specify the initial conditions for the MCMC chain
inits = list(
  list(
    sigma = runif(n = 1, min = 0.0001, max = 0.001)
    , b0 = runif(n = 1, min = 0.0001, max = 0.001)
    , b1 = runif(n = 1, min = 0.0001, max = 0.001)
    , b2 = runif(n = 1, min = 0.0001, max = 0.001)
    , b3 = runif(n = 1, min = 0.0001, max = 0.001)
    , b4 = runif(n = 1, min = 0.0001, max = 0.001)
  )
  , list(
    sigma = runif(n = 1, min = 0.0001, max = 0.001)
    , b0 = runif(n = 1, min = 0.0001, max = 0.001)
    , b1 = runif(n = 1, min = 0.0001, max = 0.001)
    , b2 = runif(n = 1, min = 0.0001, max = 0.001)
    , b3 = runif(n = 1, min = 0.0001, max = 0.001)
    , b4 = runif(n = 1, min = 0.0001, max = 0.001)
  )
  , list(
    sigma = runif(n = 1, min = 0.0001, max = 0.001)
    , b0 = runif(n = 1, min = 0.0001, max = 0.001)
    , b1 = runif(n = 1, min = 0.0001, max = 0.001)
    , b2 = runif(n = 1, min = 0.0001, max = 0.001)
    , b3 = runif(n = 1, min = 0.0001, max = 0.001)
    , b4 = runif(n = 1, min = 0.0001, max = 0.001)
  )
)
# specify the data that will be used by your JAGS program
  #the execution of JAGS is about 5 times faster on double precision than on integers.
  # define explanatory variables
  exp_vars <- c("slope", "aspect", "slope_position", "rel_moisture")
  # list of data
  hey_data = list(
    n = nrow(coverage) # n is required in the JAGS program to index the for structure
    , y = as.double(coverage$coverage)
    , x_mtrx = as.matrix(
        coverage %>%
          dplyr::mutate(dplyr::across(exp_vars, scale)) %>%
          dplyr::mutate(intercept = 1) %>%
          dplyr::select(c("intercept", exp_vars)) %>%
          #the execution of JAGS is about 5 times faster on double precision than on integers.
          dplyr::mutate_if(is.numeric, as.double)
      )
  )

# 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
# 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
```

```r
# 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 = "GrassCovJAGS.R"
  , data = hey_data
  , inits = inits
  , n.chains = length(inits)
  , n.adapt = n.adapt
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 27
##    Unobserved stochastic nodes: 33
##    Total graph size: 628
##
## 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(
      "sigma"
      , "b0"
      , "b1"
      , "b2"
      , "b3"
      , "b4"
      , "mean_y"
      , "sd_y"
      , "mean_y_sim"
      , "sd_y_sim"
      , "p_val_mean"
      , "p_val_sd"
    )
  , n.iter = n.iter
  , n.thin = 1
  , progress.bar = "none"
)
#####################
# check output
#####################
# summary
MCMCvis::MCMCsummary(zm, params = c("b4"))[[1]]
```

```
## [1] -0.4404762
```

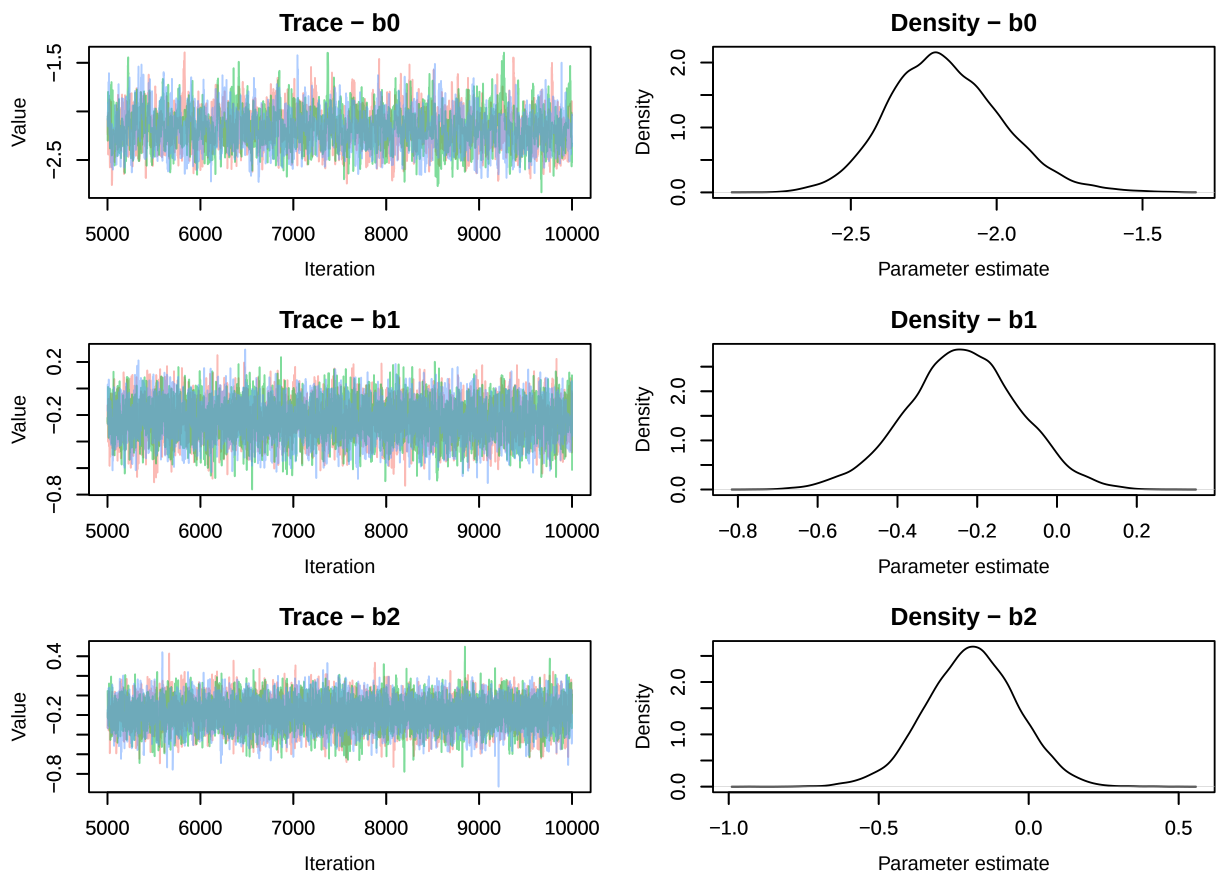
```
MCMCvis::MCMCsummary(zm, params = c(
      "sigma"
      , "b0"
      , "b1"
      , "b2"
      , "b3"
      , "b4"
      , "mean_y"
      , "sd_y"
      , "mean_y_sim"
      , "sd_y_sim"
      , "p_val_mean"
      , "p_val_sd"
    )
  )
```

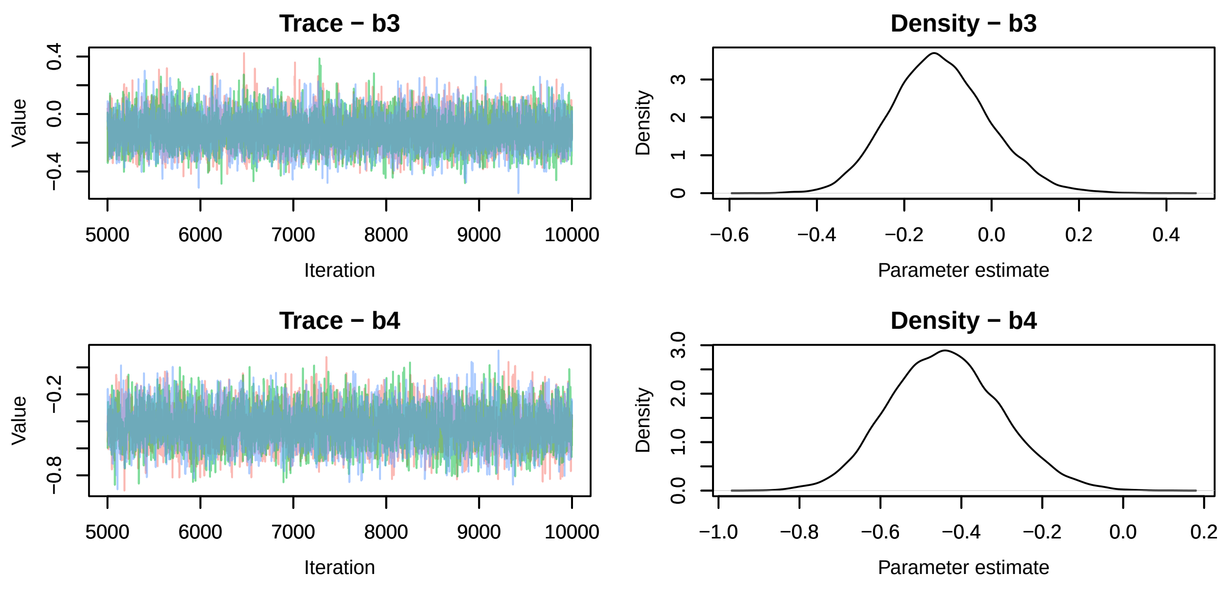
```
##                   mean         sd        2.5%         50%       97.5% Rhat
## sigma       0.09194429 0.01999844  0.06170970  0.08899181  0.13984961    1
## b0         -2.17617890 0.19349386 -2.52891946 -2.18708507 -1.76888005    1
## b1         -0.23690026 0.13935015 -0.51199695 -0.23658458  0.03375589    1
## b2         -0.18651117 0.15128801 -0.48788954 -0.18495551  0.10560784    1
## b3         -0.11908528 0.11242582 -0.32672151 -0.12436107  0.11264978    1
## b4         -0.44047620 0.13606185 -0.69685483 -0.44368284 -0.16379833    1
## mean_y      0.11133836 0.00000000  0.11133836  0.11133836  0.11133836  NaN
## sd_y        0.10142293 0.00000000  0.10142293  0.10142293  0.10142293  NaN
## mean_y_sim  0.11774256 0.02510212  0.07627898  0.11515439  0.17456826    1
## sd_y_sim    0.10963205 0.02508929  0.07061052  0.10611759  0.16919041    1
## p_val_mean  0.56446667 0.49583490  0.00000000  1.00000000  1.00000000    1
## p_val_sd    0.58503333 0.49272449  0.00000000  1.00000000  1.00000000    1
##            n.eff
## sigma       2287
## b0          2477
## b1          7210
## b2         10466
## b3         11657
## b4          6365
## mean_y         0
## sd_y           0
## mean_y_sim  5638
## sd_y_sim    4844
## p_val_mean 10057
## p_val_sd    9588
```

```
# chain 1 first 6 iterations and specific columns
zm[[1]][1:6, c(
      "b0"
      , "b1"
      , "b2"
      , "b3"
      , "b4"
    )]
```

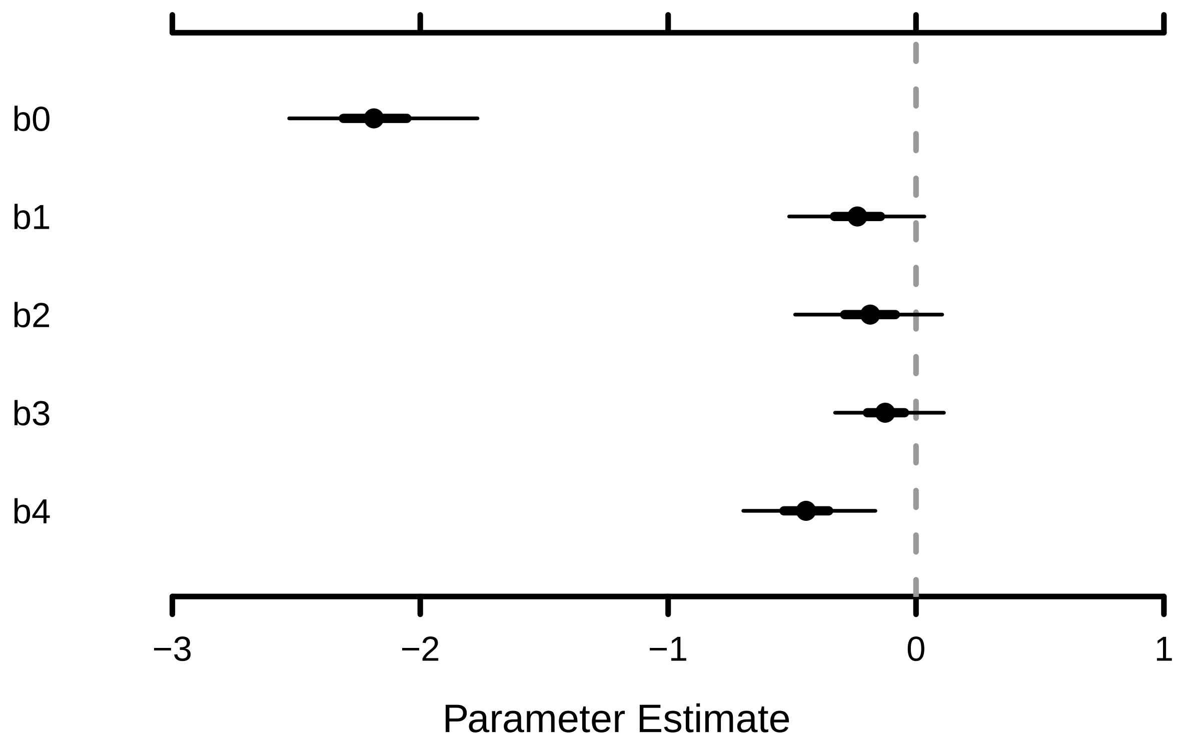
```
##              b0          b1          b2           b3          b4
## [1,] -1.846420 -0.22719723 -0.11250844 -0.084480018 -0.4997900
## [2,] -2.104090 -0.17626404  0.03527275 -0.176017570 -0.4161494
## [3,] -2.150024 -0.19081000 -0.30528918  0.076824077 -0.3913865
## [4,] -2.252855 -0.12105441 -0.22524643 -0.109211181 -0.4421346
## [5,] -2.309889 -0.06897838 -0.13900073  0.004513708 -0.5138481
## [6,] -1.974588 -0.24337515 -0.29263033 -0.025463816 -0.4397708
```

```
# trace plot
MCMCvis::MCMCtrace(zm, params = c(
      "b0"
      , "b1"
      , "b2"
      , "b3"
      , "b4"
    )
    , pdf = FALSE
  )
```





```
# Caterpillar plots
MCMCvis::MCMCplot(zm, params = c(
      "b0"
      , "b1"
      , "b2"
      , "b3"
      , "b4"
    ))
```



```r
exp_vars <- c("slope", "aspect", "slope_position", "rel_moisture")
```

Which covariate has the greatest effect?

The covariate with the greatest effect is relative moisture (β_4 = -0.440). Relative effect can be inferred by the absolute value of the coefficient estimates because the predictor data was standardized.

Posterior predictive check summary

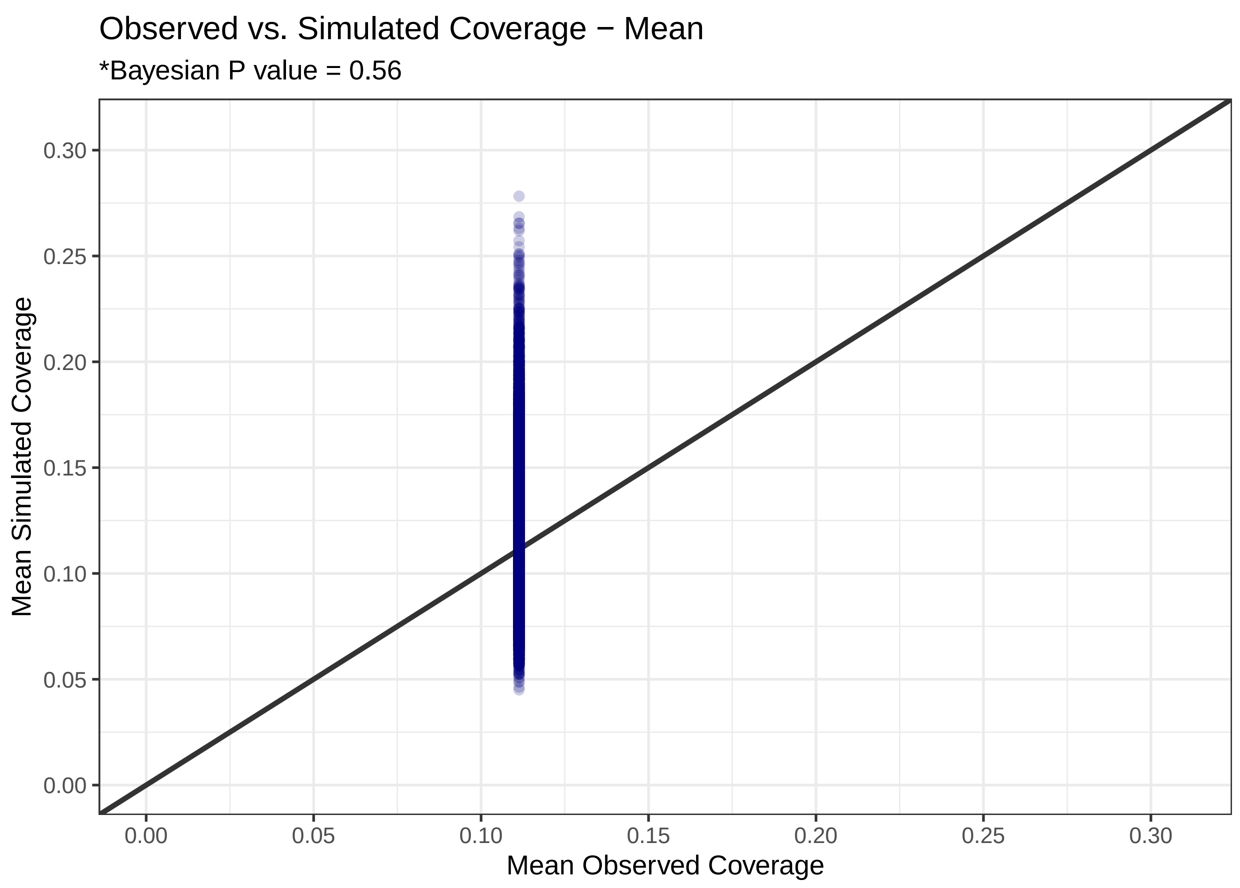
Do a posterior predictive check for the mean and the standard deviation.

```r
# summary
MCMCvis::MCMCsummary(zm, params = c(
      "mean_y"
      , "mean_y_sim"
      , "p_val_mean"
      , "sd_y"
      , "sd_y_sim"
      , "p_val_sd"
    )
  )
```

```
##                   mean         sd       2.5%       50%     97.5% Rhat n.eff
## mean_y      0.1113384 0.00000000 0.11133836 0.1113384 0.1113384  NaN     0
## mean_y_sim 0.1177426 0.02510212 0.07627898 0.1151544 0.1745683    1  5638
## p_val_mean 0.5644667 0.49583490 0.00000000 1.0000000 1.0000000    1 10057
## sd_y       0.1014229 0.00000000 0.10142293 0.1014229 0.1014229  NaN     0
## sd_y_sim   0.1096321 0.02508929 0.07061052 0.1061176 0.1691904    1  4844
## p_val_sd   0.5850333 0.49272449 0.00000000 1.0000000 1.0000000    1  9588
```

Posterior predictive check - mean

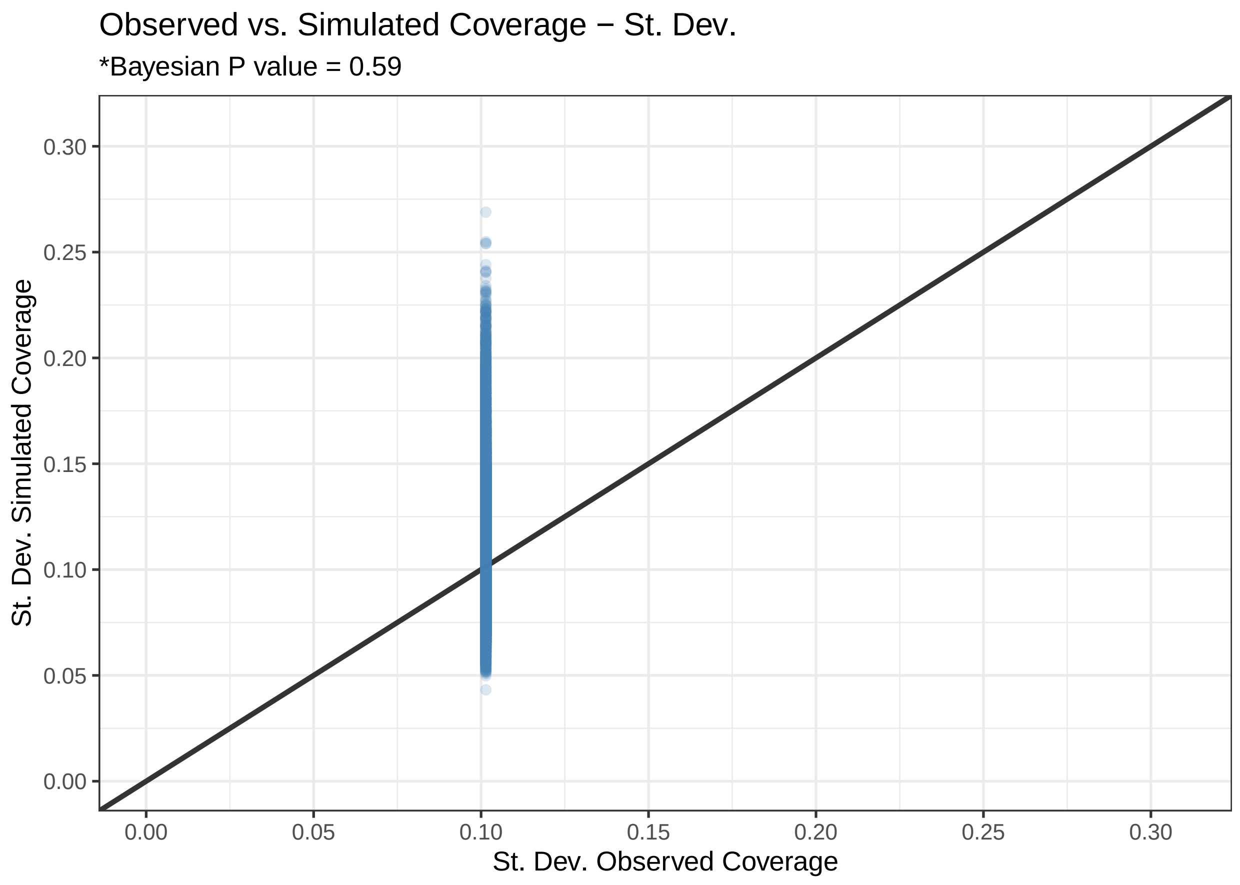
```
# PLOT
MCMCvis::MCMCchains(zm, params = c("mean_y", "mean_y_sim")) %>%
  data.frame() %>%
ggplot(data = .) +
  geom_abline(intercept = 0, slope = 1, lwd = 1, color = "gray20") +
  geom_point(
    mapping  = aes(x = mean_y, y = mean_y_sim)
    , color = "navy"
    , alpha = 0.2
  ) +
  scale_y_continuous(
    limits = c(
      min(coverage$coverage)*.95
      , max(coverage$coverage)*1.05
    )
    , breaks = scales::extended_breaks(n=10)
  ) +
  scale_x_continuous(
    limits = c(
      min(coverage$coverage)*.95
      , max(coverage$coverage)*1.05
    )
    , breaks = scales::extended_breaks(n=10)
  ) +
  xlab("Mean Observed Coverage") +
  ylab("Mean Simulated Coverage") +
  labs(
    title = "Observed vs. Simulated Coverage - Mean"
    , subtitle = paste0(
        "*Bayesian P value = "
        , MCMCvis::MCMCchains(zm, params = c("p_val_mean")) %>% mean() %>% scales::comma(accuracy = 0.0
      )
  ) +
  theme_bw()
```



The Bayesian p-value of 0.56 indicates good fit for the mean grass coverage level. If the p-value were very large or very small (i.e., close to 1 or 0), it would indicate lack of fit.

Posterior predictive check - SD

```
# PLOT
MCMCvis::MCMCchains(zm, params = c("sd_y", "sd_y_sim")) %>%
  data.frame() %>%
ggplot(data = .) +
  geom_abline(intercept = 0, slope = 1, lwd = 1, color = "gray20") +
  geom_point(
    mapping  = aes(x = sd_y, y = sd_y_sim)
    , color = "steelblue"
    , alpha = 0.2
  ) +
  scale_y_continuous(
    limits = c(
      min(coverage$coverage)*.95
      , max(coverage$coverage)*1.05
    )
    , breaks = scales::extended_breaks(n=10)
  ) +
  scale_x_continuous(
    limits = c(
      min(coverage$coverage)*.95
      , max(coverage$coverage)*1.05
    )
    , breaks = scales::extended_breaks(n=10)
  ) +
  xlab("St. Dev. Observed Coverage") +
  ylab("St. Dev. Simulated Coverage") +
  labs(
    title = "Observed vs. Simulated Coverage - St. Dev."
    , subtitle = paste0(
        "*Bayesian P value = "
        , MCMCvis::MCMCchains(zm, params = c("p_val_sd")) %>% mean() %>% scales::comma(accuracy = 0.01)
      )
  ) +
  theme_bw()
```



The Bayesian p-value of 0.59 indicates good fit for the standard deviaion of grass coverage. If the p-value were very large or very small (i.e., close to 1 or 0), it would indicate lack of fit.