

JAGS logistic regression

Becky's code

Data

Hein et. al (2013) investigate how environmental conditions influence coexistence between brown trout (*Salmo trutta*) and pike (*Esox lucius*) in Swedish lakes. One-hundred and fifty one (151) lakes consisting of brown trout are introduced to pike by human intervention, and whether or not the two species coexist was recorded. For each lake, five environmental conditions deemed relevant to coexistence patterns are considered:

- **elev**: elevation
- **catcharea**: upstream catchment area
- **lakearea**: maximum area
- **maxdepth**: maximum depth
- **temp**: mean annual air temperature at outlet

For each lake, we also record whether or not the two species coexisted in the variable called **coexist** (1 = coexisted, 0 = did not coexist).

The data are stored in **coexist.csv**. Note that the explanatory variables have already been standardized to have 0 mean and standard deviation of 1.

Your task

```
library(rjags)
library(MCMCvis)
# change file path as necessary
coexist <- readr::read_csv("coexist.csv")
```

Write a JAGS script named **coexist_mod.R** that fits a logistic regression model for the coexistence of these two species, using all five explanatory variables as predictors (plus an intercept). Use independent $N(0, 3)$ priors for each of the coefficients.

- **dbern()** will be useful to you; it accepts the probability of success as its input
- You can make derived quantities in JAGS like we do in regular R

- Indexing of matrices works in JAGS just like it does in regular R

Then run your model on the data using two chains, each burning in for 5000 iterations and then retaining 10000 samples. Using the appropriate function from `MCMCvis`, assess convergence.

Then, because the predictors are on the same scale we can directly compare their coefficients. Using the appropriate function from the `MCMCvis` package, which explanatory variable appears to be the most important for understanding coexistence of these two species, and why? **Come to class prepared to answer this question!**

```
# least reproducible/flexible
{
  sink("coexist_mod.R")
  cat("
model {
  ## sampling model
  for (i in 1:n) {
    y[i] ~ dbern(p[i])
    p[i] = 1/(1 + exp(-(beta0 + beta1*X[i,2] + beta2*X[i,3] + beta3*X[i,4] +
      beta4*X[i,5] + beta5*X[i,6])))
  }

  ## priors
  beta0 ~ dnorm(0, 1/3)
  beta1 ~ dnorm(0, 1/3)
  beta2 ~ dnorm(0, 1/3)
  beta3 ~ dnorm(0, 1/3)
  beta4 ~ dnorm(0, 1/3)
  beta5 ~ dnorm(0, 1/3)
}
", fill = T)
  sink()
}
```

```
y <- coexist$coexist
X <- model.matrix(coexist ~ ., data= coexist)
n <- length(y)
data_ls <- list("y" = y, "X" = X, "n" = n)
n_chain <- 2
jm <- jags.model("coexist_mod.R",
  data = data_ls,
  n.chains = n_chain)
```

```

Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 151
  Unobserved stochastic nodes: 6
  Total graph size: 2388

```

```

Initializing model

```

```

update(jm, n.iter = 5000)
jags_out <- coda.samples(jm,
                        variable.names = c("beta0", "beta1", "beta2", "beta3", "beta4", "beta5"),
                        n.iter = 10000)
# MCMCtrace(jags_out, pdf = F)
MCMCsummary(jags_out)

```

	mean	sd	2.5%	50%	97.5%	Rhat	n.eff
beta0	2.302537	0.5013381	1.3578551	2.284620	3.3283533	1	1824
beta1	1.178742	0.4204820	0.3927634	1.161743	2.0326527	1	3093
beta2	3.106179	1.4821943	0.2397541	3.094687	6.0153687	1	2276
beta3	1.279643	1.4674442	-1.2703649	1.203833	4.3585357	1	3983
beta4	3.812682	0.7189025	2.4766147	3.782420	5.3090612	1	3593
beta5	-1.226740	0.4641932	-2.1607448	-1.216053	-0.3469669	1	2792

```

# much more reproducible!
{
sink("coexist_mod2.R")
cat("
model {
## sampling model
for (i in 1:n) {
  y[i] ~ dbern(p[i])
  p[i] = 1/(1 + exp(-(beta %*% X[i, ])))
}

## priors
for(k in 1:K){
  beta[k] ~ dnorm(0, 1/s2_beta)
}
}
", fill = T)

```

```
sink()
}
```

```
K <- ncol(X)
data_ls <- list("y" = y, "X" = X, "n" = n, "K" = K, "s2_beta" = 3)
n_chain <- 2
jm <- jags.model("coexist_mod2.R",
                 data = data_ls,
                 n.chains = n_chain)
```

```
Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 151
  Unobserved stochastic nodes: 6
  Total graph size: 1976
```

```
Initializing model
```

```
update(jm, n.iter = 5000)
jags_out <- coda.samples(jm,
                         variable.names = c("beta"),
                         n.iter = 10000)
# MCMCtrace(jags_out, pdf = F)
MCMCsummary(jags_out)
```

	mean	sd	2.5%	50%	97.5%	Rhat	n.eff
beta[1]	2.289694	0.5000677	1.3628297	2.270619	3.3279910	1	1850
beta[2]	1.172307	0.4215813	0.3794084	1.155905	2.0348932	1	2995
beta[3]	3.046512	1.4598793	0.2132135	3.032155	5.9207261	1	2529
beta[4]	1.330792	1.4595143	-1.1863451	1.250653	4.4039685	1	4190
beta[5]	3.793822	0.7151111	2.4729888	3.768620	5.2578488	1	3844
beta[6]	-1.225601	0.4657207	-2.1887669	-1.212458	-0.3479633	1	2699