ESS 575: JAGS Problems Lab

Team England

06 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

Download the R package BayeNSF ver. 1.1 to your computer.

Run:

```
install.packages("<pathtoBayesNSF>/BayesNSF_1.1.tar.gz", repos = NULL, type = "source")
```

Motivation

JAGS allows you to implement models of high dimension once you master its syntax and logic. It is a great tool for ecological analysis. The problems that follow challenge you to:

- Write joint distributions as a basis for writing JAGS code.
- Write JAGS code to approximate marginal posterior distributions of derived quantities.
- Plot model output in revealing ways.
- Understand the effect of vague priors on parameters and on predictions of non-linear models.

Derived quantities with the logistic

One of the most useful features of MCMC is its equivariance property which means that any quantity that is a function of a random variable in the MCMC algorithm becomes a random variable. Consider two quantities of interest that are functions of our estimates of the random variables r and K:

- The population size where the population growth rate is maximum, $\frac{K}{2}$
- The rate of population growth, $\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right)$

You will now do a series of problems to estimate these quantities of interest. Some hints for the problems below:

- Include expressions for each derived quantity in your JAGS code.
- You will need to give JAGS a vector of N values to plot $\frac{dN}{dt}$ vs N.
- Use a JAGS object for plotting the rate of population growth.
- Look into using the `ecdf()` function on a JAGS object. It is covered in the JAGS Primer.

Question 1

Approximate the marginal posterior distribution of the population size where the population growth rate is maximum and plot its posterior density. You may use the work you have already done in the JAGS Primer to speed this along.

```
####################################################################
# insert JAGS model code into an R script
####################################################################
{ # Extra bracket needed only for R markdown files - see answers
  sink("LogisticJAGS.R") # This is the file name for the jags code
  cat("
  ## Logistic example for Primer
    model{
      # priors
      K ~ dunif(0, 4000) # dunif(alpha = lower limit, beta = upper limit)
      r ~ dunif (0, 2) # dunif(alpha, beta)
      sigma ~ dunif(0, 2) # dunif(alpha, beta)
      tau <- 1/sigma^2
      # likelihood
      for(i in 1:n){
        mu[i] <- r - r/K * x[i]
        y[i] ~ dnorm(mu[i], tau) # dnorm(mu,tau)
      }
      ## quantities of interest
      # population size where the population growth rate is maximum
      N_max_pop_grwth_rt <- K/2
      # The rate of population growth
      for(j in 1:length(N)){
        pop_grwth_rt[j] <- r * N[j] * (1 - ( N[j] / K ))
      }
    }
  ", fill = TRUE)
  sink()
}
##################################################################
# implement model
####################################################################
# SESYNCBayes which has the data frame Logistic, which we then order by PopulationSize
# Logistic = SESYNCBayes::Logistic[order(Logistic$PopulationSize),]
Logistic = BayesNSF::Logistic %>% dplyr::arrange(PopulationSize)
# specify the initial conditions for the MCMC chain
inits = list(
  list(K = 1500, r = .2, sigma = 1),
  list(K = 1000, r = .15, sigma = .1),
  list(K = 900, r = .3, sigma = .01)
)
# set up population size vector
N <- seq(
  0 # does it make sense to estimate the change in pop_grwth_rt for N<2?
  , round(
      max(Logistic$PopulationSize)
        + sd(Logistic$PopulationSize)*2
      , digits = -2 # round to the nearest 100
    )
  , 10
)
# specify the data that will be used by your JAGS program
  #the execution of JAGS is about 5 times faster on double precision than on integers.
hey_data = list(
  n = nrow(BayesNSF::Logistic), # n is required in the JAGS program to index the for structure
  x = as.double(BayesNSF::Logistic$PopulationSize),
  y = as.double(BayesNSF::Logistic$GrowthRate),
  N = as.double(N)
)
# 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
# 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
######################
set.seed(1)
jm = rjags::jags.model(
  file = "LogisticJAGS.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: 50
##    Unobserved stochastic nodes: 3
##    Total graph size: 896
##
## Initializing model
```

```r
stats::update(jm, n.iter = n.update)
# save the coda object (more precisely, an mcmc.list object) to R as "zm"
zm = rjags::coda.samples(
  model = jm
  , variable.names = c("K", "r", "sigma", "tau", "N_max_pop_grwth_rt", "pop_grwth_rt")
  , n.iter = n.iter
  , n.thin = 1
)
#####################
# check output
#####################
# summary
MCMCvis::MCMCsummary(zm, params = c("K", "r", "sigma", "tau", "N_max_pop_grwth_rt"))
```

```
##                            mean           sd         2.5%          50%
## K                  1.237069e+03 6.268353e+01 1.129295e+03 1.231905e+03
## r                  2.008564e-01 9.726328e-03 1.814193e-01 2.010038e-01
## sigma              2.864664e-02 3.056489e-03 2.338619e-02 2.836539e-02
## tau                1.259399e+03 2.620049e+02 7.997790e+02 1.242861e+03
## N_max_pop_grwth_rt 6.185344e+02 3.134177e+01 5.646474e+02 6.159523e+02
##                           97.5% Rhat n.eff
## K                  1.374564e+03    1  7103
## r                  2.196168e-01    1  7451
## sigma              3.536022e-02    1 14985
## tau                1.828441e+03    1 16538
## N_max_pop_grwth_rt 6.872821e+02    1  7103
```

```r
# chain 1 first 6 iterations and specific columns
zm[[1]][1:6, c("K", "r", "sigma", "tau", "N_max_pop_grwth_rt")]
```

```
##             K         r      sigma      tau N_max_pop_grwth_rt
## [1,] 1232.967 0.2012662 0.02914741 1177.064           616.4835
## [2,] 1232.570 0.1921306 0.03039581 1082.362           616.2848
## [3,] 1184.154 0.2170032 0.03118569 1028.228           592.0772
## [4,] 1128.163 0.2083735 0.02773102 1300.374           564.0816
## [5,] 1316.121 0.2072614 0.02691411 1380.511           658.0607
## [6,] 1156.156 0.2109947 0.02529937 1562.358           578.0781
```

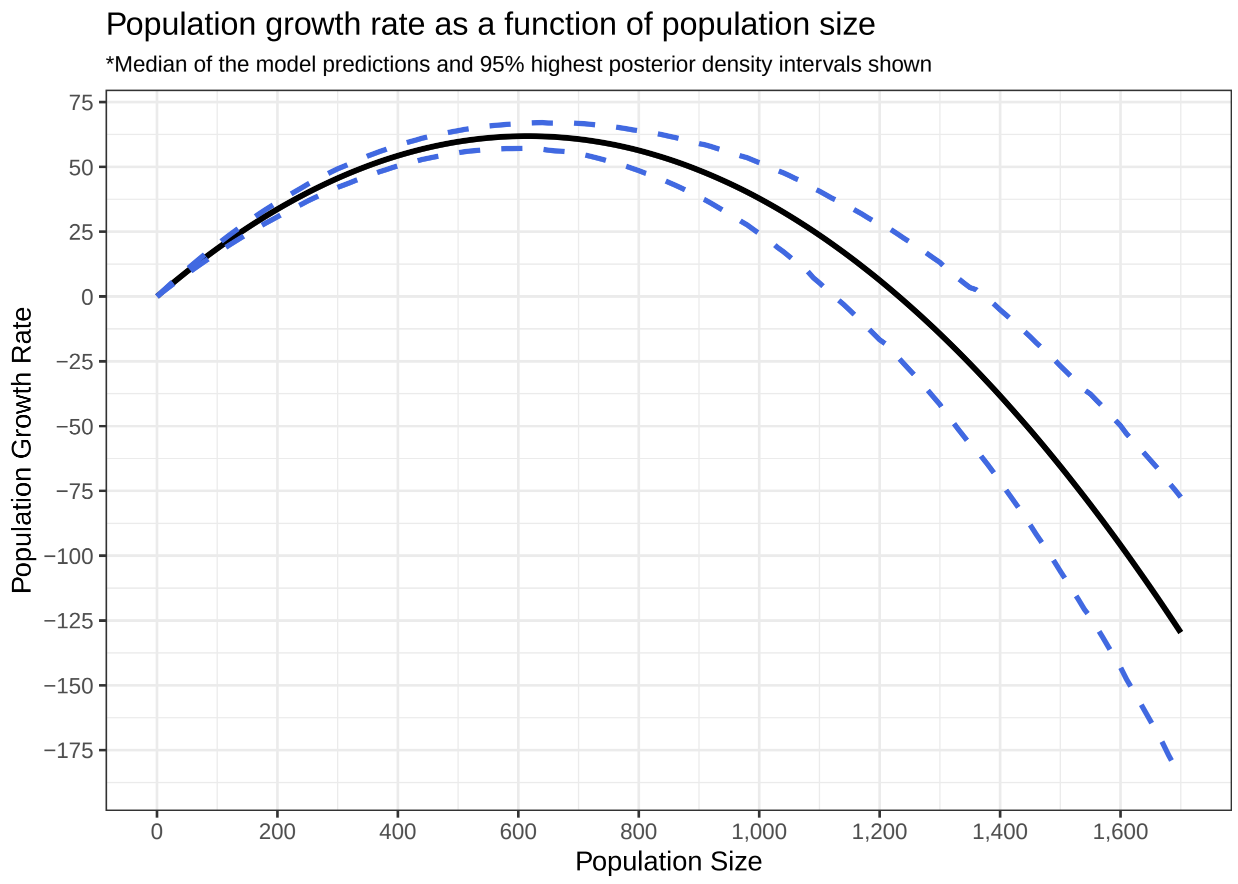
```r
# The rate of population growth
MCMCvis::MCMCpstr(zm, params = "pop_grwth_rt", func = function(x) quantile(x, c(0.025, 0.5, 0.975))) %>
  as.data.frame() %>%
  dplyr::bind_cols(N = N) %>%
  dplyr::slice_head(n = 6)
```

```
##                 pop_grwth_rt.2.5. pop_grwth_rt.50. pop_grwth_rt.97.5.  N
## pop_grwth_rt[1]          0.000000         0.000000           0.000000  0
## pop_grwth_rt[2]          1.800404         1.993564           2.177366 10
## pop_grwth_rt[3]          3.573981         3.954386           4.317707 20
## pop_grwth_rt[4]          5.320287         5.882756           6.419975 30
## pop_grwth_rt[5]          7.039148         7.778152           8.484344 40
## pop_grwth_rt[6]          8.730423         9.640786          10.511418 50
```

Question 2

Plot the median growth rate of the *population* (not the per-capita rate) rate and a 95% highest posterior density interval as a function of N. What does this curve tell you about the difficulty of sustaining harvest of populations?

```
dplyr::bind_cols(
  N = N
  , median_pop_grwth_rt = MCMCvis::MCMCpstr(zm, params = "pop_grwth_rt", func = median) %>% unlist()
  , MCMCvis::MCMCpstr(zm, params = "pop_grwth_rt", func = function(x) HDInterval::hdi(x, credMass = 0.95
) %>%
# plot
ggplot(data = .) +
  geom_line(mapping  = aes(x = N, y = median_pop_grwth_rt), color = "black", lwd = 1.1) +
  geom_line(mapping  = aes(x = N, y = pop_grwth_rt.upper), color = "royalblue", lwd = 1, linetype = "das
  geom_line(mapping  = aes(x = N, y = pop_grwth_rt.lower), color = "royalblue", lwd = 1, linetype = "das
  scale_y_continuous(breaks = scales::extended_breaks(n=10)) +
  scale_x_continuous(breaks = scales::extended_breaks(n=10), labels = scales::comma) +
  xlab("Population Size") +
  ylab("Population Growth Rate") +
  labs(
    title = "Population growth rate as a function of population size"
    , subtitle = "*Median of the model predictions and 95% highest posterior density intervals shown"
  ) +
  theme_bw() +
  theme(
    plot.subtitle = element_text(size = 9)
  )
```



Question 3

What is the probability that the intrinsic rate of increase ($r$) exceeds 0.22? What is the probability that $r$ falls between 0.18 and 0.22?

```
# access data from MCMC list
temp_df <- MCMCvis::MCMCchains(zm, params = c("r")) %>% as.data.frame()
# probability that the intrinsic rate of increase $(r)$ exceeds 0.22
temp_1 <- 1 - stats::ecdf(temp_df$r)(0.22)
# probability that $r$ falls between 0.18 and 0.22
temp_2 <- stats::ecdf(temp_df$r)(0.22) - stats::ecdf(temp_df$r)(0.18)
```

The probability that the intrinsic rate of increase ($r$) exceeds 0.22 is: 2.2%

The probability that $r$ falls between 0.18 and 0.22 is: 96.0%

Lizards on islands

This problem is courtesy of McCarthy (2007). Polis et al. (1998) analyzed the probability of occupancy of islands p by lizards as a function of the ratio of the islands' perimeter to area ratios. The data from this investigation are available in the data frame `BayesNSF::IslandsLizards`. The response data, as you will see, are 0 or 1: 0 if there were no lizards found on the island, 1 if there were 1 or more lizards observed. You are heroically assuming that if you fail to find a lizard, none are present on the island.

Question 1

Construct a simple Bayesian model that represents the probability of occupancy as:

$$g(a,b,x_i) = \frac{e^{a+bx_i}}{1+e^{a+bx_i}}$$

where x_i is the perimeter to area ratio of the i^{th} island. So, now that you have the deterministic model, the challenge is to choose the proper likelihood to link the data to the model. How do the data arise? What likelihood function is needed to represent the data?

The data – occupancy of islands p by lizards – arise from a Bernoulli distribution with the random variable p taking on the values 0 or 1. The likelihood function for the Bernoulli distribution is the inverse logit (i.e. the logistic function) with the form:

$$\text{inverse logit}(\phi_i) = \frac{\exp(\phi)}{1+\exp(\phi)}$$

Question 2

Write the expression for the posterior and joint distribution of the parameters and data, as we have learned how to do in lecture. Use the joint distribution as a basis for JAGS code needed to estimate the posterior distribution of a and b. Assume vague priors on the intercept and slope, e.g., $\beta_0 \sim \mathsf{normal}(0, 10000)$, $\beta_1 \sim \mathsf{normal}(0, 10000)$. Draw a DAG if you like. There doesn't appear to be any variance term in this model. How can that be?

$$\big[a, b \mid \mathbf{y}\big] \propto \prod_{i=1}^{n} \mathsf{Bernoulli}\big(y_i \mid g(a,b,x_i)\big)\ \mathsf{normal}\big(a \mid 0, 10000\big)\ \mathsf{normal}\big(b \mid 0, 10000\big)$$

$$p = g(a,b,x_i) = \text{inverse logit}\big(a+bx_i\big) = \frac{\exp\big(a+bx_i\big)}{1+\exp\big(a+bx_i\big)}$$

There doesn't appear to be any variance term in this model. How can that be?

The Bernoulli distribution is the discrete probability distribution of a random variable which takes the value 1 with probability p and the value 0 with probability $q = 1 - p$ and a variance $\sigma^2 = pq = p(1-p)$. The model above includes p which can be utilized to determine the variance σ^2.