Exercise 4: Non-linear models

Goal and Background

The goal of this analysis is to fit a common non-linear model (Beverton-Holt stock-recruit function) with varying parameters. This exercise will be done in order to 1) fit stock-recruit functions to data-poor groups (that otherwise may not be fit using maximum likelihood), and to 2) make inferences about whether or not different Atlantic cod (*Gadus morhua*) stocks in the North Atlantic have different stock recruit functions (i.e., different parameter estimates for the same overall function, the commonly-applied Beverton-Holt function). The basic data include spawning stock biomass, recruits, and location (stock), which all comes from the RAM Legacy Stock Assessment Database. Again, R code is provided and you will focus on the JAGS coding of the hierarchical model.

Importing data and data manipulations

- Use the `cod_sr.csv` file to read the stock-recruit data into R.
- Scale both spawning stock biomass (`SSB`) and recruit (`R`) numbers by 1000.
- The `LOCATION` variable is the spatial grouping of interest for these stocks.

The model we are fitting is a Beverton-Holt stock-recruit function

$$y_i = \frac{\alpha_{j(i)} \times SSB_i}{1 + (\beta_{j(i)} \times SSB_i)} + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma^2)$$

Where y_i is number of recruits and SSB_i is the number of spawners (in 1000s), α_j are stock-specific estimates of the initial slope (maximum R/S as S approaches 0), and β_j are stock-specific estimates of the asymptote (maximum number of recruits at high S). σ^2 is the within-stock variance, μ_α is the population-average initial slope, μ_β is the population-average asymptote. ϵ_i is the residual error, assumed to be independently and identically distributed as $N(0, \sigma^2)$

Questions

1. Plot the rescaled `SSB` and `R` data with group identifiers (e.g., colors or symbols—you can decide how to do this, just make it an effective plot).
2. Fit the whole dateset (ignoring groups) to the model using the `nls()` function. Note that the model fitting may be sensitive to starting values, so you may have to experiment with different starting values. Generate a plot that includes the data and this overall model fit with `nls()`.
3. Fit the data using `nlme()`, which is a non-linear modeling fitting funtion that includes random effects. (Hint: One way to do this is to use the `groupedData()` function first to identify the separate stocks. Then, write a function using `function()` for the SSB-R equation. Finally, use the `nlme()` function in which you provide it your pre-coded function and the grouped data.)
4. Add to or replot the data as in Q1, but include separate stock-specific fits of the Beverton-Holt model to the data. Do you trust these results? How do they compare to the overall fit produced in `nls()`.
5. Now fit the model in JAGS. There is an addiitonal complexity to this model as we estimate the parameters in log space, which done purely to help with convergence. I have provided a script below, where **the comment `#CODE` indicates a place you need to add code** to make the model run.

6. Replot the data. To that, add the stock-specific model fits. (Note you will need to use the estimates in the `BB` matrix, but convert them back to the original scale using the `exp()` function—recall these are estimated on the log scale!) Also add the overall (grand) model fit, and a legend indicating which lines are which.

7. Create a 2-panel plot, where each panel will contain parameter estimates for one parameter in the model. Plot the posterior mean and 95% credible interval. (This can be done as easily as using the `segments()` function.) Then, simply add a point for the `nlme` group-specific estimates. How close were the `nlme` estimates to the JAGS estimates?

This assigmnet is due as a digital, knitted version of a markdown file by 5pm CT Wednesday December 4, 2019. ### Code

```
library(R2jags)
library(lme4)
library(MCMCpack)
library(Hmisc)
library(FSA)
library(nlme)
library(dplyr)
library(lattice)

dat <- read.csv("cod_sr.csv",header = T,sep = ",")

# Define the model in the BUGS language and write a text file
sink("model.txt")
cat("
model {

# Likelihood:
# Level-1 of the model
for (i in 1:n){
   y[i] ~ dnorm(y.hat[i], tau.y)
   y.hat[i] <-                          #CODE
   }

    tau.y <- pow(sigma.y,-2)
        sigma.y ~ dunif(0,100)

# Level-2 of the model
for(j in 1:J){
      alpha[j] <- exp(BB[j,1])
      beta[j] <- exp(BB[j,2])

      BB[j,1:K] ~ dmnorm(BB.hat[j,], Tau.B[,]) # bivariate normal

      BB.hat[j,1] <- mu.alpha
      BB.hat[j,2] <- mu.beta
}

# Priors and derived quantities
    mu.alpha ~ dnorm(0, 0.0001)
    mu.beta ~ dnorm(0, 0.0001)
```

```
# Get grand mean on untransformed scale
    mu.alpha.raw <- exp(mu.alpha)
    mu.beta.raw <- exp(mu.beta)

### Model variance-covariance
  Tau.B[1:K,1:K] ~ dwish(W[,], df)
    df <- K+1
    Sigma.B[1:K,1:K] <- inverse(Tau.B[,])
    for (k in 1:K){
    for (k.prime in 1:K){
    rho.B[k,k.prime] <- Sigma.B[k,k.prime]/
    sqrt(Sigma.B[k,k]*Sigma.B[k.prime,k.prime])
    }
    sigma.B[k] <- sqrt(Sigma.B[k,k])
    }

} # end model
",fill = TRUE)
sink()

# Number of stocks
J <-                            #CODE

# Number of varying parameters
K <-                            #CODE

W <- diag(K)

# Load data
data <- list(y = ,                                  #CODE
             group = as.numeric(dat$LOCATION),
             n = dim(dat)[1],
             J = length(unique(dat$LOCATION)),
             SSB = ,                                #CODE
             K = K,
             W = W )


# Initial values
inits <- function (){
  list (mu.alpha = rnorm(1), mu.beta=rnorm(1), sigma.y=runif(1),
     BB=matrix(rnorm(J*K),nrow=J,ncol=K),Tau.B=rwish(K+1,diag(K)) )
}


# Parameters monitored
parameters <- c("mu.alpha","mu.beta","BB","sigma.y", "Sigma.B",
                "mu.alpha.raw", "mu.beta.raw")

# MCMC settings
ni <- 50000
nt <- 1
nb <- 25000
```

```
nc <- 3

out <- jags(data, inits, parameters, "model.txt", n.chains = nc,
n.thin = nt, n.iter = ni, n.burnin = nb)
```