Models in Ecology

Lab 3 (Graduate Students)

Mark Lewis, Maria Kuruvilla

Oct 2025

## Modelling background and equations

In this lab we use mixed effects models, containing both fixed and random effects, to assess the impacts of sea lice on salmon returns. The approach we will take follows the class reading of Krkošek and Hilborn (2011). However, there are some differences. We will have an updated data set, which include later years with lower sea lice levels. Also, so as to keep things manageable we will only fit a subset of the models given in Krkošek and Hilborn (2011).

We will draw on the theory of Lecture 10 of the BIOL 470/550B class (Mixed Effects Models). Please review these notes. Pay particular attention to the pages on “Which random effects should we include?”, “Testing random effects structure” and “Crossed vs. nested random effects”. They will all be helpful here.

Please also have the Krkošek and Hilborn (2011) handy as we will be referring to it throughout the lab. Finally, it is easier to read the equations in this lab if you download the file to your computer and read it using Word.

## Background: Fitting mixed-effects models in R

We start by loading packages already installed in the Graduate Lab 3 Preparation:

#load packages   
  
library(here)  
library(tidyverse)

We will use the R package lme4 to fit linear mixed-effects models. You may need to install and load this package if you haven’t already.

install.packages("lme4")  
library(lme4)

It is also good to install this package

install.packages("Rcpp")  
library(Rcpp)

and one for plotting

install.packages("gplots")  
library(gplots)

Assuming that you are starting from a fresh RStudio session you can select “no” to restarting R and you can also select “no” to compiling the source code from scratch when prompted by R.

The function lmer is much the same as the function lm that we used in Lab 2, but gives us the option to include random effects. When fitting a model using lmer, the fixed effects are specified as in lm, and the random effects are included in parentheses (Table 1). For example, a model fit to response data y, with covariate x in which we want to account for a random effect of group, b, on the intercept would be coded as:

lmer(y~x+(1|b))

There are many online resources for fitting mixed effects models in R, so if you have a question, do a quick Google search. Someone else probably had that question too. And of course, you can always ask me. Here are some ways to specify random effects in lmer for grouping factors b and c with covariate x.

| Code | Random effect on: |
| --- | --- |
| (1|b) | Intercept |
| (0+x|b) | Slope |
| (1+x|b) | Intercept & Slope |
| (x|b) | Intercept & Slope |
| (1|b) + (1|c) | Crossed random effects on intercept |
| (1|b/c) | Nested random effects on intercept |

## Mixed-effects models for salmon populations

### Ricker model

The Ricker model is population model that is widely used to describe the population dynamics of salmon. It predicts the expected number of individuals in the next time generation (known as Recruits in fisheries) based on the current population size. The model is given by the following equation:

where is the number of recruits, is the number of spawners, is the growth rate and is the strength of the density dependence. The linear form of the equation is given by:

which can be fit using linear regression.

Read Krkosek and Hilborn (2011) (henceforth referred to as KH), focusing on the section Stock recruit model, in which they develop a hierarchical version of the Ricker model that includes random effects to account for spatial and temporal co-variation among populations. The full model they present (equation 7) for the log ratio of recruits to spawners is:

Here

* the index indicates which of the up to 128 pink salmon populations (64 Rivers x 2 options for Odd\_Even years) is being considered,
* the index enumerates the exposure category (in KH there are 6 and in our data set there are 8),
* the index gives the return year (Return\_Yr) and
* the index describes which of the fisheries management areas (Area) the population is found in.

Note that equation (1) indicates that the population dynamics are analyzed at the level of the individual populations: the index i appears in each of the variables. Equation (1) indicates that the analysis will occur at the level of each population. The spawners and recruits are measured for each population: the index I appears in the formulas. The density dependence b\_i will also be different for each population, as will the noise term . What we are really interested in from an ecological perspective is how the growth rate is structured across exposure categories, indexed by , (rather than across populations, indexed by ). However, to deal with the hierarchy of different factors effecting the growth rate, random effects at the levels of return years (indexed by ), populations (indexed by ) and management areas (indexed by ) are also included in the growth rate analysis.

The six exposure classes in KH were given by 3 time periods relative to when infestations occurred (before, during, fallow) x two exposure to sea lice options (exposed and unexposed to sea lice from farmed salmon). Here exposure to the sea lice came from populations being in management area 12 where the fish farms were present.

However, the data we will use for this lab come from a newer dataset that includes return years to 2009. Therefore, we will include a fourth time period, “after” infestations (see Peacock et al. (2013) for more on this, where it is shown that changes in farm management greatly reduced the impact of sea lice in later years). Thus, the eight exposure classes we have are unexposed-before, unexposed-fallow, unexposed-during, unexposed-after, exposed-before, exposed-fallow, exposed-during, exposed-after. Of these 8 classes, only one of them (exposed-during) give both the location and time interval where the wild salmon returns were hypothesized by KH to be influenced by the sea lice on salmon farms that juveniles were exposed to on their out-migration to the ocean. Download the spawner-recruit “Pink S-R data” as a .csv file. Read those data into R.

data <- read.csv(here("lab3","Pink\_S-R\_data.csv"))  
  
# get a glimpse of the data  
glimpse(data)

## Rows: 1,762  
## Columns: 6  
## $ Return\_Yr <int> 1972, 1978, 1980, 1982, 1984, 1986, 1988, 1990, 1992, 1972, …  
## $ Odd\_Even <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ River <chr> "Pine", "Pine", "Pine", "Pine", "Pine", "Pine", "Pine", "Pin…  
## $ Area <int> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, …  
## $ Spawners <int> 9000, 4500, 1000, 600, 2000, 700, 550, 25, 130, 12000, 5000,…  
## $ Recruits <int> 5891, 2143, 1702, 4014, 1200, 2661, 61, 260, 306, 29457, 853…

The Return\_Yr is the year that the Recruits came back to spawn, meaning that the Spawners column refers to the spawners that started in year Return\_Yr-2. Area refers to the Department of Fisheries and Oceans statistical management area, which is the spatial scale at which catch data are reported (see <http://www.pac.dfo-mpo.gc.ca/fm-gp/maps-cartes/areas-secteurs/index-eng.html>).

Note that because odd and even year returns do not mix and are treated as separate populations, a population is given by a combination of River and Odd\_Even. Create a column for population (by combining river and Odd\_Even). Then calculate survival as log(Recruits/Spawners). Create a column for exposure category (by combining exposure and time\_period). Finally, create a column for time series length, start row and end row for each population. We will need this to differentiate the data from different populations in stan.

data\_new <- data %>% # the pipe operator (%>%) allows you to write code sequentially without having to save multiple objects  
 # you can read the pipe operator as "then"  
 mutate(survival = log(Recruits/Spawners), #mutate() adds new variables that are functions of existing variables  
 population = paste(River,Odd\_Even), #paste() joins River and Odd\_Even to create a population column  
 time\_period = case\_when(Return\_Yr < 2002 ~ factor("before"), # case\_when() is like ifelse() but allows for multiple conditions  
 Return\_Yr == 2004 ~ factor("fallow"),  
 Return\_Yr >= 2002 & Return\_Yr <= 2006 ~ factor("during"),  
 Return\_Yr > 2006 ~ factor("after")),  
 exposure = as.factor(ifelse(Area==12, "exposed","unexposed"))) %>% # ifelse() is like a simple if-then-else statement  
 mutate(exposure\_category = paste(exposure, time\_period),  
 row\_number = row\_number()) %>% # row\_number() gives the row number, which we will need for the following steps  
 # add a column with the start row number and another with the end row number for the time series of each population  
 group\_by(population) %>% # group\_by() is like the "by" statement in other programs, the next few commands will be done by population  
 mutate(time\_series\_length = n(), # n() gives the number of rows in each group  
 start\_row = first(row\_number), # first() gives the first row number in each group  
 end\_row = last(row\_number)) %>% # last() gives the last row number in each group  
 ungroup() # ungroup() removes the grouping structure from the data frame  
   
   
  
glimpse(data\_new) # check the new data frame

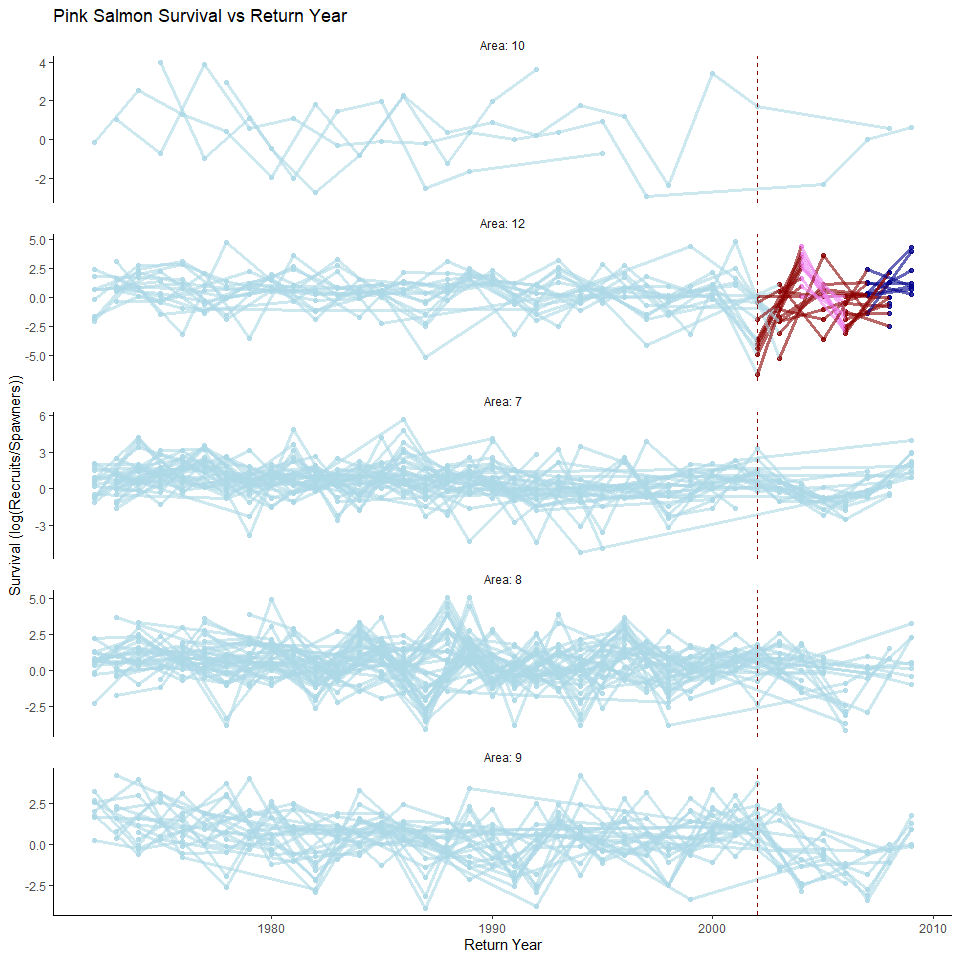
## Rows: 1,762  
## Columns: 15  
## $ Return\_Yr <int> 1972, 1978, 1980, 1982, 1984, 1986, 1988, 1990, 199…  
## $ Odd\_Even <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
## $ River <chr> "Pine", "Pine", "Pine", "Pine", "Pine", "Pine", "Pi…  
## $ Area <int> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, …  
## $ Spawners <int> 9000, 4500, 1000, 600, 2000, 700, 550, 25, 130, 120…  
## $ Recruits <int> 5891, 2143, 1702, 4014, 1200, 2661, 61, 260, 306, 2…  
## $ survival <dbl> -0.423798815, -0.741870680, 0.531804030, 1.90061387…  
## $ population <chr> "Pine 2", "Pine 2", "Pine 2", "Pine 2", "Pine 2", "…  
## $ time\_period <fct> before, before, before, before, before, before, bef…  
## $ exposure <fct> unexposed, unexposed, unexposed, unexposed, unexpos…  
## $ exposure\_category <chr> "unexposed before", "unexposed before", "unexposed …  
## $ row\_number <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, …  
## $ time\_series\_length <int> 9, 9, 9, 9, 9, 9, 9, 9, 9, 16, 16, 16, 16, 16, 16, …  
## $ start\_row <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 10, 10, 10, 10, 10, 10, …  
## $ end\_row <int> 9, 9, 9, 9, 9, 9, 9, 9, 9, 25, 25, 25, 25, 25, 25, …

Now use the unique and the length functions in R to check how many return years, management areas and populations there are in the data frame.

Now you should have all the data you need to fit the models described in KH on page 21: all possible combinations of random effects, each with and without the growth rate structured by exposure category. However, in this lab we focus on four categories of the random effects models in KH, which we call models 1 (no random effects), 3 (random effect for intercept on Return\_Yr), 4 (random effect for intercept on Area), and 7 (random effect for intercept on Area nested within Return\_Yr). These are given in rows 1, 3, 4, and 7 of KH Table 2. For each of these models we are interested in the effect of the exposure category on the population growth rate. Thus, we fit each of the models with no exposure categories (model a) and with the 8 exposure categories given above (model b). Note that the population growth rate alpha is the intercept term in equation (1).

You can visualize the data using the following code. Since there are many populations, we can group the plots by area. ggplot2 is a powerful plotting package in R which allows you to add layers to your plots. Here we use geom\_point() to add points, geom\_line() to add lines, facet\_wrap() to create a separate panel for each area, scale\_color\_manual() to specify colors for the exposure categories, and theme() to modify the look of the plot.

ggplot(data\_new, aes(x = Return\_Yr, y = survival, group = population))+  
 geom\_point(alpha = 0.8, aes(color = exposure\_category))+  
 geom\_line(aes(color = exposure\_category), alpha = 0.6, size = 1.2)+  
 geom\_vline(xintercept = 2002, linetype = "dashed", color = "darkred")+  
 facet\_wrap(~paste("Area:",Area), ncol = 1, scales = "free\_y") +  
 scale\_color\_manual(values = c("exposed before" = "lightblue",  
 "exposed during" = "darkred",   
 "exposed after" = "darkblue",   
 "exposed fallow" = "violet",  
 "unexposed before" = "lightblue",   
 "unexposed during" = "lightblue",  
 "unexposed after" = "lightblue",   
 "unexposed fallow" = "lightblue")) +  
 theme\_classic()+  
 theme(strip.background = element\_blank(),  
 legend.position = "none") +   
 labs(title = "Pink Salmon Survival vs Return Year",  
 x = "Return Year",  
 y = "Survival (log(Recruits/Spawners))",  
 color = "Exposure Category")



In summary we will have models

* 1a: no random effect, no exposure categories,
* 1b: no random effects, exposure categories,
* 3a: random effect for intercept on Return\_Yr, no exposure categories,
* 3b random effect for intercept on Return\_Yr, exposure categories,
* 4a random effect for intercept on Area, no exposure categories,
* 4b: random effect for intercept on Area, exposure categories,
* 7a: random effect for intercept on Area nested within Return\_Yr, no exposure categories, and
* 7b: random effect for intercept on Area nested within Return\_Yr, exposure categories.

Our goal in this lab is create a table with columns as given in Table 2 of KH, but with eight rows, describing the eight models given above. Return to Table 1 and determine how to write the random effects in the last six models above using R.  
Recall from our earlier discussion that each the models are run for each of the populations (indexed by i) in equation (1). To start we consider the case with no random effects. In this case the R function lm suffices to fit the model.

fit1a <- lm(Survival~Population:Spawners,data=data)

Why do we use Population:Spawners in the above command? Try looking at the results from the analyses using

summary(fit1a)

Note that the intercept is given. This is the estimate for the growth rate. Note also that the slope of the line b\_i is also given for each of the populations. Now include the exposure category

fit1b <- lm(Survival~0+exposure\_category+Population:Spawners, data=data)

Again, try looking at the results from the analyses, from fit1b. We can plot these results with a bar chart:

# make a dataframe with the estimates of coefficients of exposure category  
summary1b\_df <- data.frame(summary(fit1b)$coefficients[1:8,]) %>%   
 mutate(exposure\_category = str\_replace\_all(rownames(.), "exposure\_category", "")) %>% #make new column for exposure category  
 mutate(exposure = str\_split(exposure\_category," ",simplify=TRUE)[,1], # make new column for exposure  
 time\_period = str\_split(exposure\_category," ",simplify=TRUE)[,2]) # make new column for time period  
  
ggplot(summary1b\_df) +  
 geom\_bar(aes(x=exposure\_category, y=Estimate, fill=exposure), stat="identity", position=position\_dodge()) +  
 geom\_errorbar(aes(x=exposure\_category, ymin=Estimate-1.96\*Std..Error, ymax=Estimate+1.96\*Std..Error, group=exposure),   
 position=position\_dodge(width=0.9), width=0.2) +  
 labs(y="Population growth rate") +  
 scale\_fill\_grey(name="Exposure status", labels=c("Exposed","Unexposed")) +  
 theme\_classic() +  
 theme(axis.title.x = element\_blank(),  
 axis.text.x = element\_text(size=12, angle=45, hjust=1),  
 axis.title = element\_text(size=14),  
 legend.text = element\_text(size=12),  
 legend.title = element\_text(size=14))

What kind of error bars have been added to these plots? The log likelihoods and AICs of each of the fits can be calculated using:

lL1a <- logLik(fit1a)  
lL1b <- logLik(fit1b)  
AIC1a <- AIC(fit1a)  
AIC1b <- AIC(fit1b)

Check the values for these quantities. Note that model 1a is nested within model 1b. This allows for the use of the likelihood ratio test (LRT) to see if there is sufficient evidence to reject the null hypothesis that exposure category does not alter the growth rate significantly. Here the test statistic

teststat1 <- -2 \* (as.numeric(lL1a)-as.numeric(lL1b))

has an asymptotically Chi-Squared distribution with the degrees of freedom equal to the difference in the number of degrees of freedom of the more complicated model (fit1b). minus the degrees of freedom from the simpler model (fit1a). Work out the number of degrees of freedom for each (hint: use the summary function) and use the pchisq function to evaluate whether the null hypothesis can be rejected.

p.val1 <- pchisq(teststat1, df = …, lower.tail = false)

When lower.tail = false in the pchisq function we are asking for the probability the value teststat1 or larger could have been observed from the Chi-Squared distribution with the appropriate degrees of freedom and this is what we want for a p value. Record the p value. Are we able to reject the null hypothesis?

Now consider the cases with random effects included. We use the lmer function for fitting these cases. Recall from the class notes that when the models are initially fitted to decide which random effect structure is best the fit uses restricted ML (REML=TRUE) (stage 1). However, when fixed effect structure is tested the restricted ML is no longer used (REML=FALSE) (stage 2).  
To review the notes from class (Lecture 10), restricted maximum likelihood (REML) is an alternative to ML that estimates the random-effect parameters (i.e. standard deviations) averaged over the values of the fixed-effect parameters; REML estimates of standard deviations are generally less biased than corresponding ML estimates. In other words, ML estimation (REML=FALSE), estimates the standard deviations of the random effects assuming that the fixed-effect estimates are precisely known whereas restricted maximum likelihood (REML=TRUE) averages over some of the uncertainty in the fixed-effect parameters. LR tests on fixed effects, or any comparison of models with different fixed effects, also require ML, rather than REML, estimates. Testing (from class notes in Lecture 10) takes the following form:

Thus in the lab when initially comparing the random effects structure (Table 1 in the lab) use Method 1 above, when comparing the fixed effects in the models (Table 2 in the lab) use Method 2, and when plotting the bar chart for estimates from the “best” model use Method 3. Our first step is to compare models 3a, 3b, 4a, 4b, 7a, and 7b using REML=TRUE (stage 1). Here the models need to be fit and AIC values are compared and the model with the lowest AIC is selected as having the best random effects structure. Make a table for this comparison with columns given by “Model”, “AIC” and “AIC”. Fill in the table by calculating the appropriate fits of the models (refer to page 4 of this lab for a description of the models). By way of example the code for model 7 would be

fit7a <- lmer(Survival~Population:Spawners+(1|Return\_Yr/Area), REML=TRUE, data=data)  
fit7b <- lmer(Survival~0+exposure\_category+Population:Spawners+(1|Return\_Yr/Area), REML=TRUE, data=data)  
AIC7a <- AIC(fit7a)  
AIC7b <- AIC(fit7b)

Create similar code for models 3 and 4. Based on the AIC values select which model has the best random effects structure from the information theory perspective. Make a bar chart for the best model as you did for model 1. Now we are in a position move to stage 2 (test the fixed effects structure using LRT for models 3, 4 and 7). We have already undertaken the test for model 1. However for models 3, 4 and 5 lmer must be invoked again, but with REML=FALSE:

fit7a\_fixed<-lmer(Survival~Population:Spawners+(1|Return\_Yr/Area), REML=FALSE, data=data)  
fit7b\_fixed<-lmer(Survival~0+exposure\_category+Population:Spawners+(1|Return\_Yr/Area), REML=FALSE, data=data)  
AIC7a <- AIC(fit7a\_fixed)  
AIC7b <- AIC(fit7b\_fixed)  
ll7a\_fixed<-logLik(fit7a\_fixed)  
ll7b\_fixed<-logLik(fit7b\_fixed)   
teststat7\_fixed <- -2 \* (as.numeric(ll7a\_fixed)-as.numeric(ll7b\_fixed))

Work out the number of degrees of freedom and use the pchisq function to evaluate whether the null hypothesis can be rejected. Record the p value. Then repeat this procedure for models 3 and 4.

p.val7 <- pchisq(…, df = …, lower.tail = false)

(You fill in the …). Finally, create a Table with columns the same as Table 2 in KH and rows that are suitable for models 1, 3, 4 and 7 and populate the table with what you have found. You will use this Table for Assignment 3.

## Fitting Bayesian models in Stan

This section is optional for undergraduate students, but required for graduate students

Let’s start by loading the additional packages we need to use.

#load packages   
library(rstan)  
rstan\_options("auto\_write" = TRUE)  
options(mc.cores = parallel::detectCores())  
library(tidybayes)  
library(bayesplot)

Let us now write the model 7b from the KH paper in stan. You can make a new stan file in the same way as you made a new R script. Click on ‘+ file’ in the bottom right files window and then select ‘Stan File’. You can name the file ‘ricker\_7b.stan’. This file will open up in the top left portion of the RStudio console. Copy and paste the following code in it. You can check if the code is syntactically correct by clicking on the ‘check’ button on the top right of the window. If you see this warning “Warning message: In readLines(file, warn = TRUE) :incomplete final line found on ‘ricker\_7b.stan’”, then just add an empty line to the end of your code by clicking enter. In order to run this code, we will have to write some lines of code in an R Script.

data{  
 int<lower=0> N; //number of observations  
 int J; //number of populations  
 int K; // number of exposure categories  
 int A; // number of areas  
 int years; // numer of years  
 array[N] int ii; //index of brood years  
 array[N] int aa; //index of areas  
 array[N] real spawners;   
 array[N] real survival; //log(Recruits/spawner)  
 matrix[N,K] exposure\_category\_matrix; // dummy variable with 1,0  
 array[J] int start\_row; //start of observation for that population  
 array[J] int end\_row; //end of observations for that population  
   
}  
  
  
  
parameters {  
   
   
 vector[years] theta\_year; // random effect of year  
 real<lower=0> sigma\_theta\_area; // standard deviation for the distribution of random effects theta\_year\_area   
 vector[K] r; //effect of exposure on survival  
   
 //variance components  
 real<lower=0> mu\_sigma; // standard deviation for survival  
 vector[years] z; // parameter that affects theta\_year\_area (also called non centered parameterization)  
 vector<lower=0>[J] b; //per capita density dependence term, slope  
}  
  
transformed parameters {  
   
 vector[N] exposure; //total exposure effect for each observation  
 vector[N] mu; //expectation at each time for each stock  
 matrix[A,years] theta\_year\_area; //nested random effect of area within year  
   
 for(a in 1:A){ # loop over areas  
 for(year in 1:years){ # loop over years  
 theta\_year\_area[a,year] = theta\_year[year] + z[year]\*sigma\_theta\_area; //nested random effect of area within year  
 //distribution of theta\_year\_area effect has a mean which is theta\_year  
 }  
   
 }  
   
   
 for(j in 1:J){ // for every population  
   
 for(t in start\_row[j]:end\_row[j]){ //for every year  
 exposure[t] = 0; // set exposure to 0  
 for(k in 1:K){// for every level in the exposure category  
 exposure[t] = exposure[t] + exposure\_category\_matrix[t,k]\*r[k];   
 // sum over the exposures. exposure category matrix has 1,0. r is the effect of exposure on survival  
   
 }  
   
 mu[t] = exposure[t] + theta\_year\_area[aa[t],ii[t]] - b[j]\*spawners[t]; // survival = r + theta -bS  
 }  
   
 }  
   
}  
  
  
model {  
   
 theta\_year ~ normal(0,1); // prior for random effect of year   
 b ~ normal(0,0.1); // prior for slope  
 r ~ normal (1.2,1); // prior for growth rate (from literature)  
 //variance terms  
 mu\_sigma ~ normal(1,1); // prior for standard deviation  
 sigma\_theta\_area ~ normal(1,1); // prior for standard deviation for random effect  
   
 //likelihood  
 for(j in 1:J){ //for every river  
 survival ~ normal(mu, mu\_sigma); //likelihood for all observations   
 }  
   
   
}

Now we have to call call this model from RStudio and fit the model to data. First we need to make a dummy variable for exposure category. Dummy variables or indicator variables take on values of 0 or 1 to indicate the absence or presence of a categorical effect. If there are more than two levels in the categorical variable, multiple columns are created with the number of columns equivalent to the number of levels. For each row, only one of the columns would take on the value of 1, while the others would be 0.

# model 7b ----------------------------------------------------------------  
  
  
  
# first make dummy variable for exposure category  
  
exposure\_category\_matrix <- model.matrix(~0 + exposure\_category,   
 data = data\_new)   
  
# first remove "exposure\_category" from the colnames  
  
colnames(exposure\_category\_matrix) <- str\_replace\_all(colnames(exposure\_category\_matrix), "exposure\_category","")  
  
years = max(data\_new$Return\_Yr) - min(data\_new$Return\_Yr) + 1 # number of years  
  
areas = length(unique(data\_new$Area)) # number of areas  
  
  
data\_list\_7b <- list(  
 N = nrow(data\_new), # number of rows in the data frame  
 J = length(unique(data\_new$population)), # number of populations  
 K = ncol(exposure\_category\_matrix), # number of exposure categories  
 A = areas, # number of areas  
 years = years, # number of years  
 ii = as.numeric(factor(data\_new$Return\_Yr)), #index of year  
 aa = as.numeric(factor(data\_new$Area)), #index of areas  
 spawners = data\_new$Spawners,   
 survival = data\_new$survival,  
 exposure\_category\_matrix = exposure\_category\_matrix, # dummy variable for exposure  
 start\_row = unique(data\_new$start\_row), # start of observation for that population  
 end\_row = unique(data\_new$end\_row) # end of observations for that population  
)  
  
head(exposure\_category\_matrix)  
  
head(data\_new)  
  
print(exposure\_category\_matrix[1699,])  
print(data\_new[1699,])

You can use “print(exposure\_category\_matrix)” to make sure you understand it:

If we look at the first row of the exposure category matrix, there is a 1 in the “unexposed before” column and 0 in every other column. This corresponds to the first row of our dataset which is from Area 7 in year 1972. This area is in the “unexposed” category and this year is in the time period “before” exposure. Therefore there is a 1 in the column corresponding to the appropriate exposure and time period category. However, if we look at row 1699, which corresponds to data from Area 12 and year 2003, the exposure category has 1 in the “exposed during” column and 0 in every other column. Note that dummy variables are also created by lm() and lmer() functions when we use categorical variables.

This model takes ~5 minutes to run.

# call model  
  
# ricker\_7b\_model2 <- stan\_model(file = here("lab3","ricker\_7b\_moment\_matching.stan"))  
  
ricker\_7b\_model <- stan\_model(file = "ricker\_7b.stan")  
  
# sample from model  
  
ricker\_7b\_model\_sampling <- rstan::sampling(ricker\_7b\_model,  
 data = data\_list\_7b,  
 chains = 4,  
 iter = 2000,  
 warmup = 1000)

However, be sure to run at least 4 chains with 2000 iter, and 1000 warmup, before you draw inferences from the model results. This will take ~1 hour to run.

You can use the following code to plot the posterior distributions of the r parameter (the effect of exposure on survival).

#let's look at the mean and the 95% credible intervals of the growth rate  
model\_7b\_bayesian\_summary <- data.frame(summary(ricker\_7b\_model\_sampling)$summary) %>%   
 filter(startsWith(rownames(summary(ricker\_7b\_model\_sampling)$summary), "r")) %>%   
 select(mean, X2.5., X97.5.)  
  
#write it to csv  
  
write.csv(model\_7b\_bayesian\_summary, file ="model\_7b\_bayesian\_summary.csv")  
  
exposure\_effects\_list <- paste0("r[",1:8,"]")  
  
#we can also plot it  
bayesplot::mcmc\_intervals(ricker\_7b\_model\_sampling,  
 pars = exposure\_effects\_list,  
 prob = 0.80, #80% credible interval, thick line  
 prob\_outer = 0.95) + #95% credible interval, thin line  
 labs(title = "Posterior Intervals of Growth Rate",  
 x = "Growth Rate",  
 y = "Density")  
  
  
#rename the r to colnames of exposure\_category\_matrix  
  
  
bayesplot::mcmc\_areas(ricker\_7b\_model\_sampling,  
 pars = exposure\_effects\_list,  
 prob = 0.95,  
 prob\_outer = 0.99) +  
 scale\_y\_discrete(labels = colnames(exposure\_category\_matrix)) +  
 labs(title = "Posterior Distributions of Growth Rate",  
 x = "Growth rate",  
 y = "Density")  
  
#save the figure  
  
ggsave("model\_7b\_bayesian\_growth\_rate.png", width = 6, height = 8)

You are now ready to move on to Assignment 3!

## Literature Cited

Krkošek, M. & Hilborn, R., 2011. Sea lice (Lepeophtheirus salmonis) infestations and the productivity of pink salmon (Oncorhynchus gorbuscha) in the Broughton Archipelago, British Columbia, Canada. Canadian Journal of Fisheries and Aquatic Sciences, 68(1), pp.17–29. Available at: <http://www.nrcresearchpress.com/doi/abs/10.1139/F10-137>.

Peacock, S.J. et al., 2013. Cessation of a salmon decline with control of parasites. Ecological Applications, 23(3), pp.606–620. Available at: <http://www.esajournals.org/doi/abs/10.1890/12-0519.1>.