Models in Ecology

Lab 3

Maria Kuruvilla

Sep 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 Stan

We will use the rstan package to fit a Bayesian mixed-effects (or hierarchical) model in R. You will need the following: - Recent R (> 4.2.0) - Recent Rstudio (or text editor of your choice – but we will be using Rstudio!)

It is recommended that your code for this lab and class is within one folder which is also an Rproject [Why Use Projects?](https://thedavidchen.github.io/post/rstudio-why-use-projects/)

Within R you should install the following packages: - Tidyverse installed in R (we’ll mostly use dplyr and ggplot2) - [Rstan](https://mc-stan.org/rstan/) NOTE: When installing rstan, use the [following instructions.](https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started)

Please go through the above instructions for your operating system. Make sure to run the example! Here is a minimal workflow example:

install.packages("rstan", repos = c('https://stan-dev.r-universe.dev', getOption("repos")))  
example(stan\_model, package = "rstan", run.dontrun = TRUE)

Here are the list of packages we will be using for this lab. YOu can use the following code to check whether you have already installed these packages and to install them, if not. Then load the packages to use them.

list.of.packages <- c("ggplot2", "tidyverse", "here", "tidybayes")  
  
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]  
  
if(length(new.packages)) install.packages(new.packages)  
  
#load packages   
library(rstan)  
rstan\_options("auto\_write" = TRUE)  
options(mc.cores = parallel::detectCores())  
library(tidybayes)  
library(here)  
library(tidyverse)  
library(ggplot2)

### Mixed-effects models for salmon populations

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:

where is the number of recruits from population in year , is the number of spawners, is a random intercept for each population, is a fixed effect for the exposure category (exposed/unexposed and time period), is a random year effect common to all populations in year , is a random effect for year and area, is a random slope for each population, and is the residual error.

* the index river indicates which of the up to 128 pink salmon populations (64 Rivers x 2 options for Odd\_Even years) is being considered,
* the index exposure category enumerates the exposure category (in KH there are 6 and in our data set there are 8),
* the index t gives the return year (Return\_Yr) and
* the index area 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 “river” 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 “river” appears in the formulas. The density dependence b\_river 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 r is structured across exposure categories, indexed by exposure category, (rather than across populations, indexed by river). However, to deal with the hierarchy of different factors effecting the growth rate, random effects at the levels of return years (indexed by t), populations (indexed by river) and management areas (indexed by a) 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. I have downloaded into a folder called “lab3”.

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.

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.

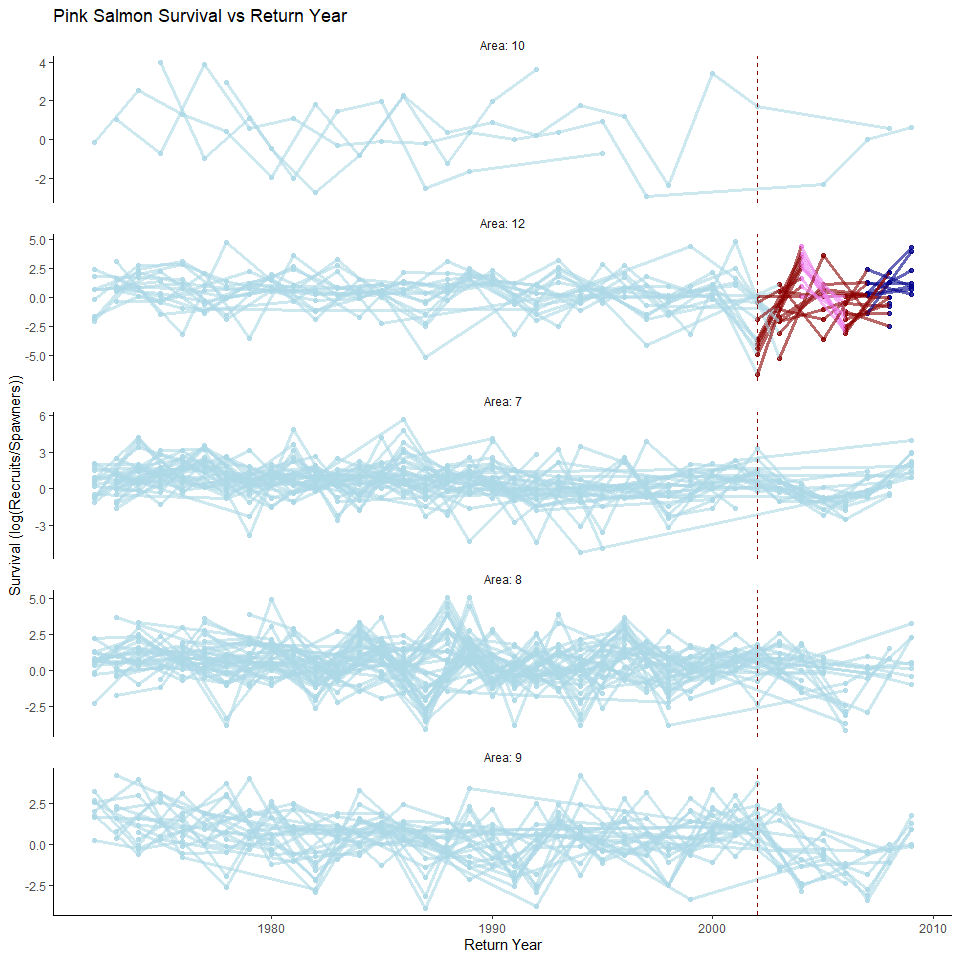
data\_new <- data %>%   
 mutate(survival = log(Recruits/Spawners),  
 population = paste(River,Odd\_Even),  
 time\_period = case\_when(Return\_Yr < 2002 ~ factor("before"),  
 Return\_Yr == 2004 ~ factor("fallow"),  
 Return\_Yr >= 2002 & Return\_Yr <= 2006 ~ factor("during"),  
 Return\_Yr > 2006 ~ factor("after")),  
 exposure = as.factor(ifelse(Area==12, "exposed","unexposed"))) %>%   
 mutate(exposure\_category = paste(exposure, time\_period),  
 row\_number = row\_number()) %>%   
 # 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) %>%  
 mutate(time\_series\_length = n(),  
 start\_row = first(row\_number),  
 end\_row = last(row\_number)) %>%   
 ungroup()  
   
   
  
glimpse(data\_new)

## 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 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.

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.

### Stan

[Stan](https://mc-stan.org/) is a probabilistic programming language that allows us to run Bayesian statistical models - from simple linear regression to more complicated multi-level models and time-series analysis.

A Stan program is organized into a sequence of named blocks:

functions {  
 // ... function declarations and definitions ...  
}  
data {  
 // ... declarations ...  
}  
transformed data {  
 // ... declarations ... statements ...  
}  
parameters {  
 // ... declarations ...  
}  
transformed parameters {  
 // ... declarations ... statements ...  
}  
model {  
 // ... declarations ... statements ...  
}  
generated quantities {  
 // ... declarations ... statements ...  
}

All of the blocks are options. Note that comments start with “//” and all lines end with a semicolon “;”.

Here us an example of a simple linear regression for one salmon population which can be expressed as where is survival in year n, is the growth rate, is the density dependence, and is the spawners at year n:

data {  
 int<lower=0> N; //number of observations or years in this case  
 vector[N] x; //predictor of size N  
 vector[N] y; //response variable of size N  
}  
parameters {  
 real alpha; //intercept to be estimated  
 real beta; //slope to be estimated  
 real<lower=0> sigma; //variance to be estimated  
}  
model {  
  
 alpha ~ normal(1.2, 2); //prior for intercept, based on literature  
 beta ~ normal(0, 10); //prior for slope  
 sigma ~ normal(0, 1); //prior for variance  
 y ~ normal(alpha + beta \* x, sigma); // linear model  
}

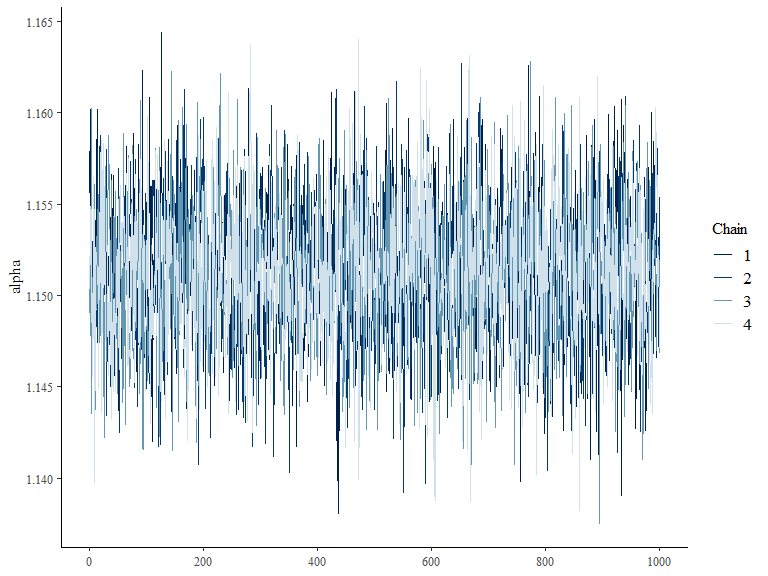
### Model 1a

For the model with no random effects and no exposure categories. Call the following model “ricker\_1a.stan”.

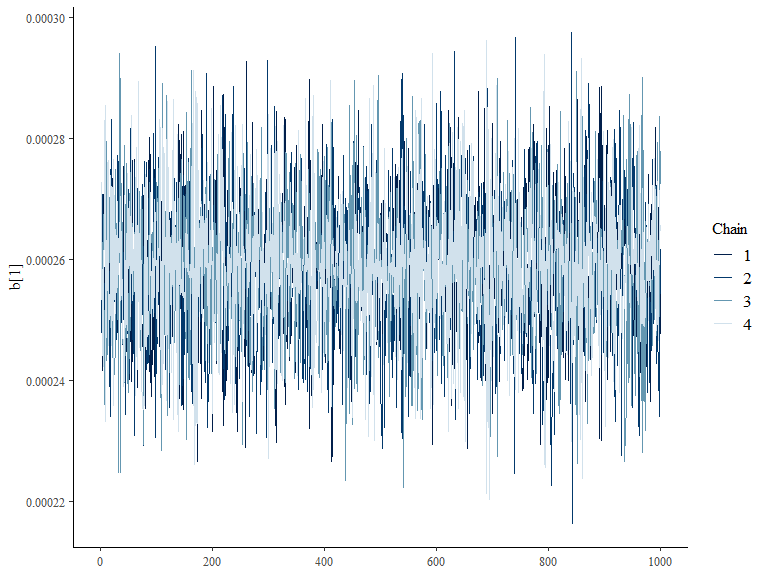
data{  
 int<lower=0> N; //number of observations  
 int J; //number of populations  
 array[N] real spawners;   
 array[N] real survival; //log(Recruits/spawner)  
 array[J] real Smax\_mean; //prior mean of Smax  
 array[J] real Smax\_sigma; // prior sigma of Smax  
 array[J] int start\_row; //start of observation for that population  
 array[J] int end\_row; //end of observations for that population  
   
}  
  
transformed data {  
   
 vector[J] log\_Smax\_mean;  
 vector[J] log\_Smax\_sigma;  
   
 //moment matching  
 for (j in 1:J){  
 log\_Smax\_mean[j] = log(Smax\_mean[j]) - 0.5\*log(1 + (Smax\_sigma[j]^2)/(Smax\_mean[j]^2)); //convert smax prior to per capita slope - transform to log scale with bias correction  
 log\_Smax\_sigma[j] = sqrt(log(1 + (Smax\_sigma[j]^2)/(Smax\_mean[j]^2))); //this converts sigma on the untransformed scale to a log scale  
 }  
   
   
   
}  
  
parameters {  
   
 vector<lower=0>[J] Smax; // spawner level at which recruits are maximized (according to Ricker model)  
 real alpha; //river level intrinsic productivity, intercept  
 //variance components  
 real<lower=0> mu\_sigma; ///mean sigma among all stocks  
}  
  
transformed parameters {  
 vector<lower=0>[J] b; //per capita density dependence term, slope  
 //productivity residuals through time  
 vector[N] e\_t; //stock residual productivity at time t  
 vector[N] mu; //expectation at each time for each stock  
   
 for(j in 1:J){ // for every population  
 b[j] = 1/Smax[j];  
   
 for(t in start\_row[j]:end\_row[j]){ //for every year  
 mu[t] = alpha - b[j]\*spawners[t];  
 e\_t[t] = survival[t] - mu[t]; // no autocorrelation in errors  
 }  
   
 }  
   
}  
  
  
model {  
 alpha ~ normal(1.2,2); //prior for intrinsic productivity for all stocks  
   
 for(j in 1:J){  
 Smax[j] ~ lognormal(log\_Smax\_mean[j], log\_Smax\_sigma[j]); //prior on Smax for each stock  
 }  
 //variance terms  
 mu\_sigma ~ normal(1,1);  
   
 //likelihood  
 for(j in 1:J){  
 survival ~ normal(mu, mu\_sigma); //likelihood for all observations (no autocorrelation)  
 }  
   
   
}

# call model  
  
ricker\_1a\_model <- stan\_model(file = here("lab3","ricker\_1a.stan"))  
  
# first extract max S for priors  
smax\_prior= data\_new %>%  
 group\_by(population) %>%  
 summarize(m.s=Spawners[which.max(Recruits)],m.r=max(Recruits))  
  
# create data list for model   
  
data\_list\_1a <- list(  
 N = nrow(data\_new),  
 J = length(unique(data\_new$population)),  
 spawners = data\_new$Spawners,  
 survival = data\_new$survival,  
 Smax\_mean = smax\_prior$m.s,  
 Smax\_sigma = 2\*smax\_prior$m.s,  
 start\_row = unique(data\_new$start\_row),  
 end\_row = unique(data\_new$end\_row)  
)  
  
# sample from model  
ricker\_1a\_model\_sampling <- rstan::sampling(ricker\_1a\_model,  
 data = data\_list\_1a,  
 chains = 4,  
 iter = 2000,  
 warmup = 1000)

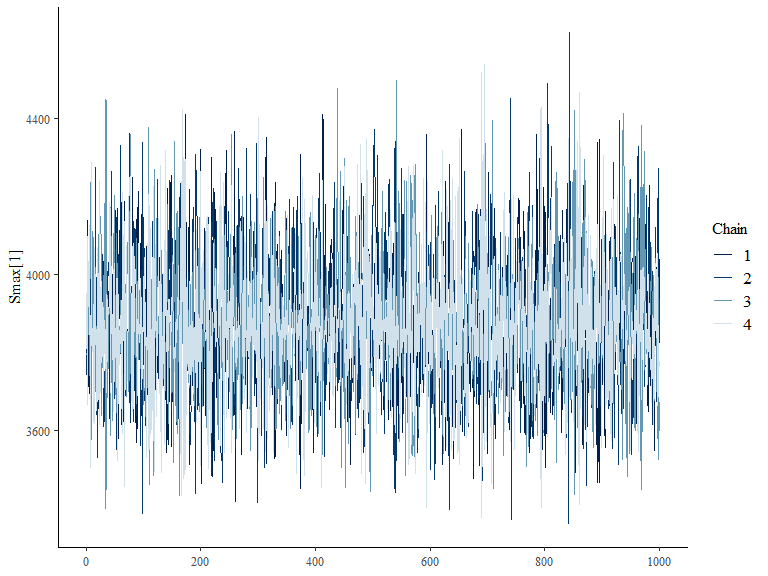
# check trace plots to look for caterpillar like sampling plots  
  
bayesplot::mcmc\_trace(ricker\_1a\_model\_sampling, pars = "alpha")



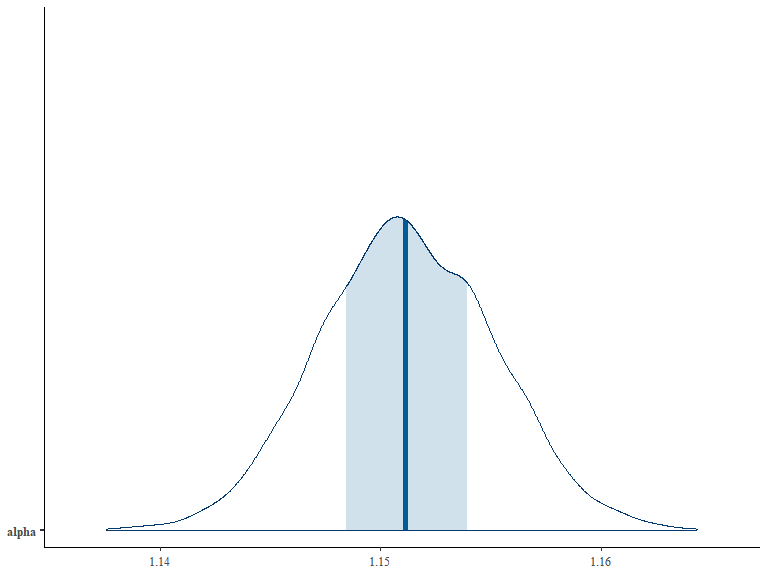
bayesplot::mcmc\_trace(ricker\_1a\_model\_sampling, pars = "b[1]")



bayesplot::mcmc\_trace(ricker\_1a\_model\_sampling, pars = "Smax[1]")



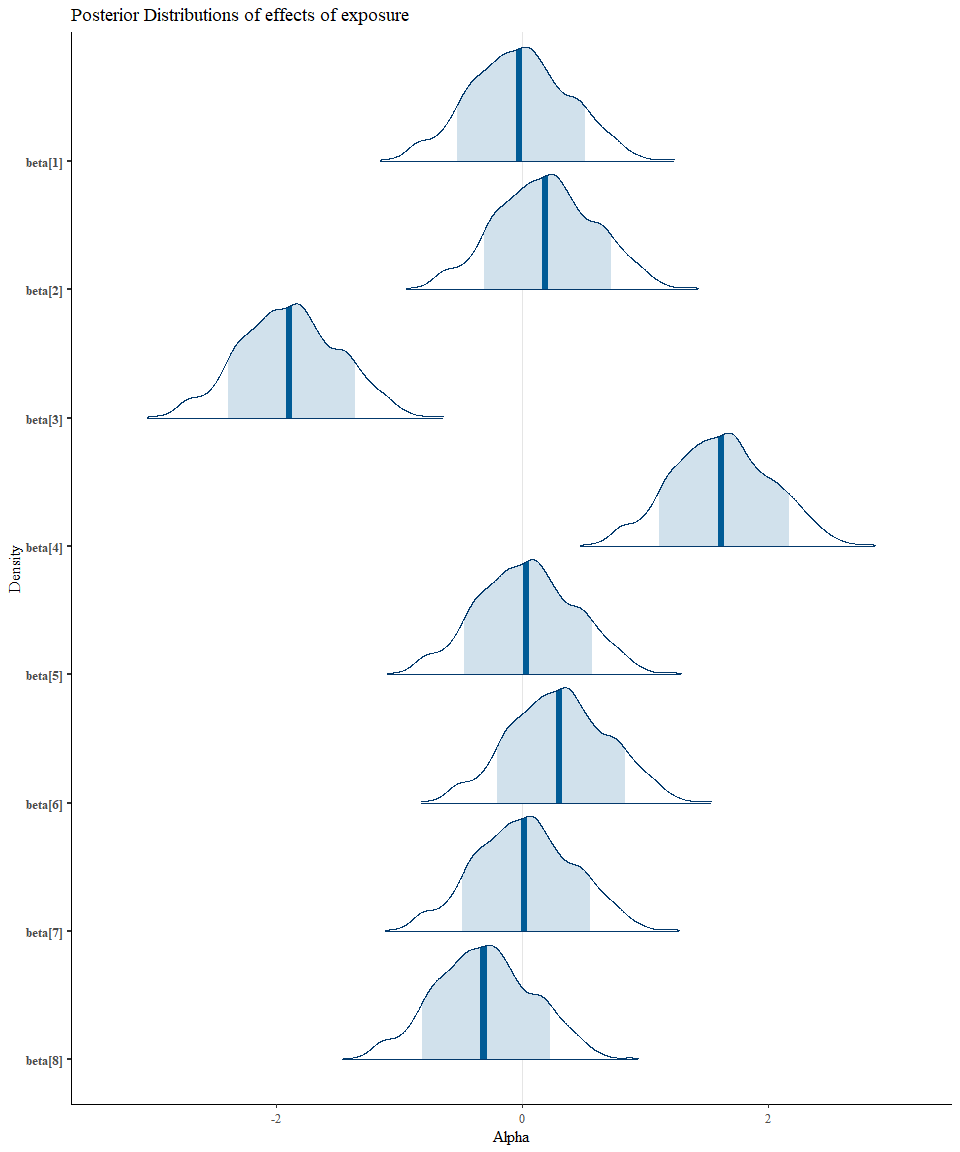
# plot posterior probability distribution  
bayesplot::mcmc\_areas(ricker\_1a\_model\_sampling, pars = c("alpha"))



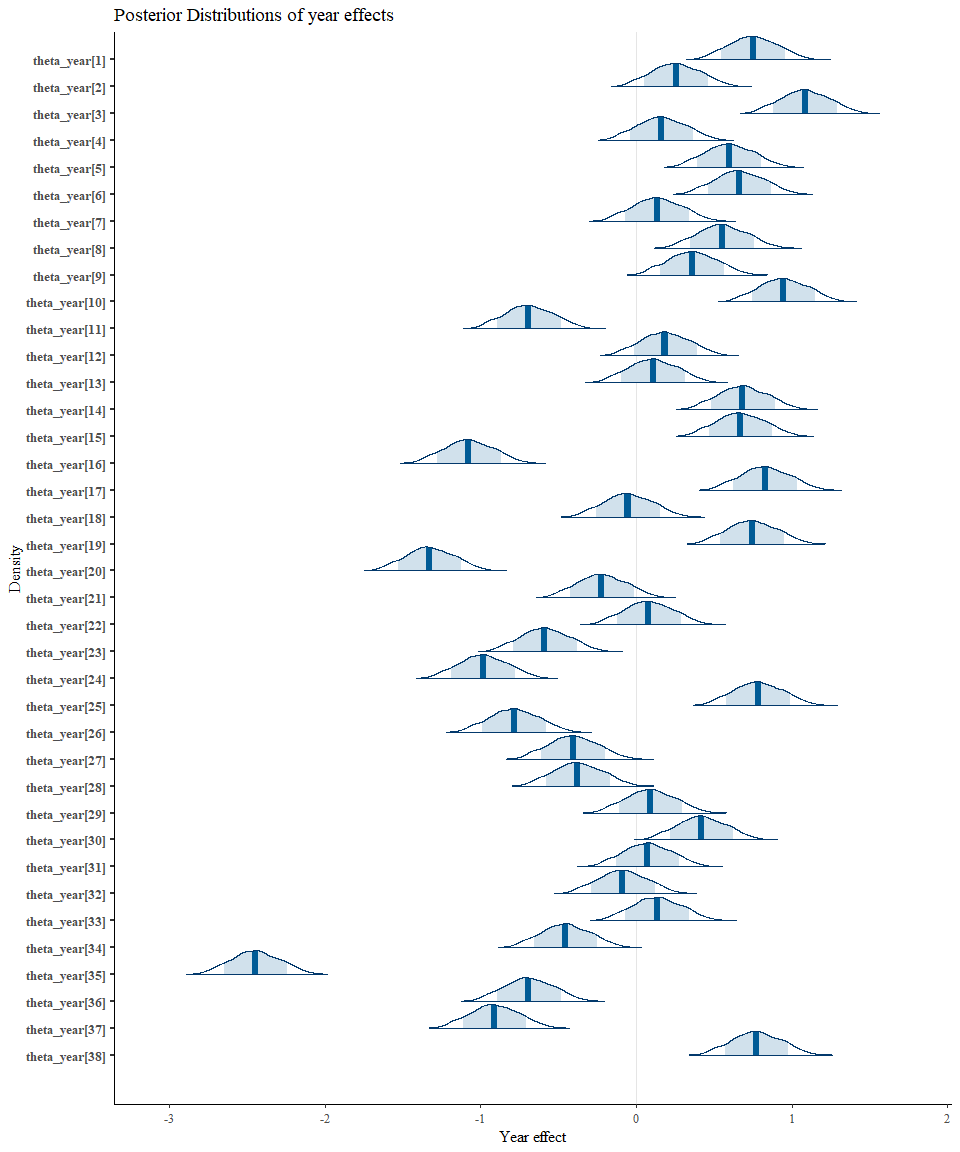
### Model 1b

data{  
 int<lower=0> N; //number of observations  
 int<lower=0> K; //number of observations  
 int J; //number of populations  
 array[N] real spawners;   
 array[N] real survival; //log(Recruits/spawner)  
 matrix[N,K] exposure\_category\_matrix;   
 array[J] real Smax\_mean; //prior mean of Smax  
 array[J] real Smax\_sigma; // prior sigma of Smax  
 array[J] int start\_row; //start of observation for that population  
 array[J] int end\_row; //end of observations for that population  
   
}  
  
transformed data {  
   
 vector[J] log\_Smax\_mean;  
 vector[J] log\_Smax\_sigma;  
   
 //moment matching  
 for (j in 1:J){  
 log\_Smax\_mean[j] = log(Smax\_mean[j]) - 0.5\*log(1 + (Smax\_sigma[j]^2)/(Smax\_mean[j]^2)); //convert smax prior to per capita slope - transform to log scale with bias correction  
 log\_Smax\_sigma[j] = sqrt(log(1 + (Smax\_sigma[j]^2)/(Smax\_mean[j]^2))); //this converts sigma on the untransformed scale to a log scale  
 }  
   
   
   
}  
  
parameters {  
   
 vector<lower=0>[J] Smax; // spawner level at which recruits are maximized (according to Ricker model)  
 real alpha; //river level intrinsic productivity, intercept  
 //variance components  
 real<lower=0> mu\_sigma; ///mean sigma among all stocks  
 // real<lower=0> sd\_sigma; ///variance in sigma   
 vector[K] beta;  
}  
  
transformed parameters {  
 vector<lower=0>[J] b; //per capita density dependence term, slope  
 vector[N] exposure; //exposure effect for each observation  
 //productivity residuals through time  
 vector[N] e\_t; //stock residual productivity at time t  
 vector[N] mu; //expectation at each time for each stock  
   
 for(j in 1:J){ // for every population  
 b[j] = 1/Smax[j];  
   
 for(t in start\_row[j]:end\_row[j]){ //for every year  
 exposure[t] = 0;  
 for(k in 1:K){  
 exposure[t] = exposure[t] + exposure\_category\_matrix[t,k]\*beta[k];  
   
 }  
 mu[t] = alpha - b[j]\*spawners[t] + exposure[t];  
 e\_t[t] = survival[t] - mu[t]; // no autocorrelation in errors  
 }  
   
 }  
   
}  
  
  
model {  
 alpha ~ normal(1.2,2); //prior for intrinsic productivity for all stocks  
   
 for(j in 1:J){  
 Smax[j] ~ lognormal(log\_Smax\_mean[j], log\_Smax\_sigma[j]); //prior on Smax for each stock  
 }  
 //variance terms  
 mu\_sigma ~ normal(1,1);  
 // sd\_sigma ~ normal(0,1);  
 beta ~ normal(0,1); //prior on exposure effects  
 //likelihood  
 for(j in 1:J){  
 survival ~ normal(mu, mu\_sigma); //likelihood for all observations (no autocorrelation)  
 }  
   
   
}  
  
  
generated quantities {  
   
   
   
}

# make dummy variable for exposure category  
  
exposure\_category\_matrix <- model.matrix(~0 + exposure\_category,   
 data = data\_new)   
#make data list for 1b  
  
data\_list\_1b <- list(  
 N = nrow(data\_new),  
 K = ncol(exposure\_category\_matrix),  
 J = length(unique(data\_new$population)),  
 spawners = data\_new$Spawners,  
 survival = data\_new$survival,  
 exposure\_category\_matrix = exposure\_category\_matrix,  
 Smax\_mean = smax\_prior$m.s,  
 Smax\_sigma = 2\*smax\_prior$m.s,  
 start\_row = unique(data\_new$start\_row),  
 end\_row = unique(data\_new$end\_row)  
)  
  
# stan model 1b -----------------------------------------------------------  
  
# call model  
  
ricker\_1b\_model <- stan\_model(file = here("lab3","ricker\_1b.stan"))  
  
# sample from model  
  
ricker\_1b\_model\_sampling <- rstan::sampling(ricker\_1b\_model,  
 data = data\_list\_1b,  
 chains = 4,  
 iter = 2000,  
 warmup = 1000)  
  
#save the samples  
saveRDS(ricker\_1b\_model\_sampling, file = here("lab3",  
 "ricker\_1b\_model\_sampling.RDS"))  
  
# bayesplot::mcmc\_trace(ricker\_1b\_model\_sampling, pars = "alpha")  
  
  
exposure\_effects\_list <- paste0("beta[",1:8,"]")  
  
bayesplot::mcmc\_areas(ricker\_1b\_model\_sampling,  
 pars = exposure\_effects\_list,  
 prob = 0.8) +  
 labs(title = "Posterior Distributions of effects of exposure",  
 x = "Alpha",  
 y = "Density")



# model 3a ----------------------------------------------------------------  
  
# include a random effect of year  
  
#need then number of years  
  
years = max(data\_new$Return\_Yr) - min(data\_new$Return\_Yr) + 1  
  
data\_list\_3a <- list(  
 N = nrow(data\_new),  
 J = length(unique(data\_new$population)),  
 years = years,  
 ii = as.numeric(factor(data\_new$Return\_Yr)), #index of brood years  
 spawners = data\_new$Spawners,  
 survival = data\_new$survival,  
 Smax\_mean = smax\_prior$m.s,  
 Smax\_sigma = 2\*smax\_prior$m.s,  
 start\_row = unique(data\_new$start\_row),  
 end\_row = unique(data\_new$end\_row)  
)  
  
# call model  
  
ricker\_3a\_model <- stan\_model(file = here("lab3","ricker\_3a.stan"))  
  
# sample from model  
  
ricker\_3a\_model\_sampling <- rstan::sampling(ricker\_3a\_model,  
 data = data\_list\_3a,  
 chains = 4,  
 iter = 2000,  
 warmup = 1000)  
  
# bayesplot::mcmc\_trace(ricker\_3a\_model\_sampling, pars = "alpha")  
# bayesplot::mcmc\_trace(ricker\_3a\_model\_sampling, pars = "b[1]")  
# bayesplot::mcmc\_trace(ricker\_3a\_model\_sampling, pars = "Smax[1]")  
#   
# # plot posterior probability distribution  
# bayesplot::mcmc\_areas(ricker\_3a\_model\_sampling, pars = c("alpha"))  
  
#plot theta  
  
year\_effects\_list <- paste0("theta\_year[",1:years,"]")  
  
bayesplot::mcmc\_areas(ricker\_3a\_model\_sampling,  
 pars = year\_effects\_list,  
 prob = 0.8) +  
 labs(title = "Posterior Distributions of year effects",  
 x = "Year effect",  
 y = "Density")



#extract alpha, theta\_year   
  
post\_3a = ricker\_3a\_model\_sampling %>%   
 rstan::extract(pars = c("alpha","theta\_year","Smax"), permuted = TRUE)  
  
  
# plot median theta\_year as a time series with credible intervals  
  
  
  
theta\_year\_df <- data.frame(Return\_Yr = min(data\_new$Return\_Yr):max(data\_new$Return\_Yr),  
 theta\_year = apply(post\_3a$theta\_year, 2, median),  
 theta\_year\_lower = apply(post\_3a$theta\_year, 2, quantile, probs = 0.025),  
 theta\_year\_upper = apply(post\_3a$theta\_year, 2, quantile, probs = 0.975)  
 )  
  
#plot  
  
ggplot(theta\_year\_df, aes(x = Return\_Yr, y = theta\_year))+  
 geom\_line(color = "#4E654D", size = 1.2)+  
 geom\_ribbon(aes(ymin = theta\_year\_lower, ymax = theta\_year\_upper),   
 alpha = 0.5, fill = "#9CC69B")+  
 theme\_classic()+  
 labs(title = "Median Year Effects from Model 3a",  
 x = "Return Year",  
 y = "Theta Year Effect")

