Bayesian Stats - Assignment 2

2024

Models with memory

The primary goal of this assignment is to model some real data that include the kind of groupings that lend themselves to multilevel/hierarchical/partial-pooling models. The data contain counts of the number of birds observed during standardized surveys, and a number of covariates including species, survey-site, year, species-group (*bird_guild*), surrounding landcover, and human footprint. A variable like bird_guild provides groupings of species that may share some characteristics in the way they respond to the environment (e.g, forest birds may have similar relationships with forest cover, so we could use species as the grouping factor to pool/share information among species on their response to variations in forest cover) and there are survey design aspects that may make repeated counts of birds at a given location more similar than counts at another location (e.g., use `RouteName` as the grouping factor to share information among routes, and/or adjust for the repeated measures on the same survey-route).

```r
library(tidyverse)
library(rethinking)
library(kableExtra)
library(patchwork)

birds <- read.csv("bird_data_bayes_stats.csv")
```

Questions to explore

This is a large dataset (there are 17192 observations in total) that lends itself to many possible questions and includes a number of interesting relationships. I will suggest two possible questions to explore, but I am also open to alternatives questions and models that feature multilevel components such as varying intercepts and/or varying effects (treatments, slopes, etc.).

- First example: Are the `counts` of forest birds related to the `proportion_forest` in the surrounding landscape, and are those relationships similar among species in a group, or are they more or less independent among species? Such a question could be answered by comparing the predictions and fit of two models that estimate the relationship between forest amount and counts: **the first** would estimate the relationship for each species independently (no-pooling model) and **the second** would treat each species as a member of a group by sharing information on the effect of forest cover among species within the group (partial-pooling model).
  Multilevel models are especially useful when data are relatively sparse. So, it would also be interesting to explore how the estimates differ between the two models for individual species and whether those differences are related to the amount of data (i.e., the mean counts) for each species? You could also explore how the answer to this question changes depending on how many years of data are included: Are the benefits of multilevel models more apparent with fewer data? It would be reasonable for this assignment to assume that forest amount or bird abundance has not changed too much over time and that annual estimates could be treated as essentially random replicated counts of each species and route (i.e., ignore the `year` column).

Similar kinds of questions could be asked using data for the species in the Urban `bird_guild` and the level of human activity surrounding the route (`human_footprint`)

- Second example: For one species (choose any species or experiment with a few), is the rate of change in the counts over time (e.g., a slope term in a regression model, aka "trend") better estimated by pooling information across routes? Asked a different way, does the trend on one route tell us something about the trend on other routes? For example one could fit three related Poisson or negative binomial regression models to the data for the forest species Ovenbird (a warbler species that builds its nest on the forest floor, and camouflages the nest with a dome made of twigs and leaves that resembles the shape of a brick-oven):
  - no-pooling model: that estimated the effect of `year` on `count` for Ovenbird, separately for each route (no-pooling);
  - patrial-pooling (multilevel) model that estimated the effect of `year` on `count` for Ovenbird, treating each route as a member of a group, so that the model shares information on the species trend across routes.
  - complete-pooling model that estimates the effect of `year` on `count` for Ovenbird and assumes that there is only one trend that is the same across all routes. Note: for these models, you may want to treat the intercepts the same way in each model to keep that aspect of the model constant while varying the trend components. However, you can feel free to explore or choose to estimate the intercepts, either as independent route-level terms (no-pooling approach that assumes mean counts are not related among routes) or as multilevel terms that share information among routes (partial-pooling approach that assumes the mean counts are similar among routes).

The question(s) you ask of these data are up to you. Please choose one of the two ideas above, or develop a custom question/model that uses hierarchical structures.

You'll notice that in the two examples I've given, the causal model is not the primary focus. The goals for this exercise are to explore the multilevel aspects of the modeling: The benefits (or costs) of grouping data and parameters, shrinking parameter estimates towards a mean "hyperparameter", sharing information among members of the groups (routes, species, etc.). These data provide a good opportunity to explore the implications of sharing information that comes with partial-pooling in a multilevel model, and the benefits of building a model that fits the structure of the data. You may want to explore different types of clusters in the same model (varying by route and by species), non-centered parameterizations, posterior predictions for groups, or comparing the fit of different models `ulam(...,log_lik = TRUE)` (FYI, my experience with bird-count data is that Pareto-k warnings may be common, even with robust models, such as those that use a negative binomial distribution). Use the content of Chapter 13 as a guide to what you should focus on, but don't write a mini-thesis. You have many other important things to do.

Note: please fit all models using an MCMC or HMC process (e.g., `ulam()`, `brms`, or writing the models in Stan). You should get familiar with using HMC to fit Bayesian models.

Assessment

Aim to produce a final document that is less than 8-pages in length. Concise is good!. The marking for this assignment will be based on the following:

- Description of the question - 1
- Selecting, preparing and describing/plotting the data (keep it simple) - 2
- Model formulation - 5

  - priors and their rationale (prior predictions), error distributions, multilevel aspects, ulam, brms, and/or Stan code

- Model assessment - 4
  - posterior predictive checks, convergence,

- Model interpretation - 5
  - comparisons of model fit (remember to calculate PSISloo you need `ulam(...,log_lik = TRUE)`), plotting of results to highlight the important conclusions, explaining how the results answer the original question

- General presentation (coherent, logical, clear, reproducible, and concise) - 3

Note: although this whole dataset is much larger than anything we've used in this course to date, your models will only include a subset of the data, and even a complex model fit to the entire dataset would require only 10-20 minutes to fit (at least for the collection of ulam models I've fit so far, that's my experience). So you may want to consider some of the tips on *big* data I've included below, but I don't expect you to be over-heating your computers. If you run into any challenges or questions, please don't hesitate to send an email adam.smith@ec.gc.ca.

The data

These data are **counts** (so distributions such as Poisson `dpois(lambda)` or negative binomial `dgampois(lambda,exp(log_scale))`) of birds observed by expert birders during surveys for the North American Breeding Bird Survey (BBS) [Hudson et al. (2017)](Sauer et al. 2017). See section 12.1.2 in *Statistical Rethinking* for an example of `ulam()` code to fit a negative binomial model. The BBS is a standardized international survey designed to monitor changes over time in bird populations. It has been conducted annually since 1966, during the peak of the breeding season in Canada and the United States (late May through early July). On each of approximately 4000 BBS routes, an expert observer counts all birds seen and heard at 50 pre-determined, road-side locations, one morning each year. The field methods are tightly controlled to limit variation due to changes in observers, time of day, time of the year, and weather conditions. Because of the tight controls on methods and observer skill, the counts on each route are generally considered an accurate index of the relative abundance of birds through time. This dataset is a small subset of the BBS database, including observations of 28 selected species on BBS routes in Southern Ontario, conducted between 2001 and 2021. The `count` column in the dataset represents the sum of all individuals observed on each route and year, for a given species (e.g., the sum of all American Robins observed at all 50 locations that make up a BBS route).

The other columns

The first 6 columns of the dataset demonstrate the basic structure. Each row includes the observed count for each BBS route, year and species, along with a `bird_guild` column that classifies each species as either a species primarily dependent on `Forest` habitat or a species that is commonly found in `Urban` habitats. These categories are, of course, over-simplifications of each species' habitat preferences and or tolerance of human activity, but they should be useful here to select or group species that may have similar relationships with the other variables in the dataset.

```r
kable(head(birds[,c(1:6)]),booktabs = T, row.names = FALSE, digits = 3)
```

RouteName	year	english	french	bird_guild	count
DUNNVILLE	2018	Blue Jay	Geai bleu	Urban	2
DUNNVILLE	2015	Black-capped Chickadee	Mésange à tête noire	Urban	2
DUNNVILLE	2018	Mourning Dove	Tourterelle triste	Urban	43
DUNNVILLE	2017	American Robin	Merle d'Amérique	Urban	94
DUNNVILLE	2015	Warbling Vireo	Viréo mélodieux	Urban	22
DUNNVILLE	2015	Yellow Warbler	Paruline jaune	Urban	18

These are the species included here and their guilds, as well as some simple summaries of their count data.

```r
sp_sum <- birds %>%
  filter(count > 0) %>% #just non-zero observations
  group_by(english,bird_guild) %>%
  summarise(number_routes = length(unique(RouteName)), # number of BBS routes where observed
            number_years = length(unique(year))) # number of years where observed

sp_means <- birds %>% # including zero values
  group_by(english,bird_guild) %>%
  summarise(mean_count = mean(count)) %>% #mean counts of each species over the dataset
  inner_join(.,sp_sum,
             by = c("english","bird_guild")) %>%
  arrange(bird_guild,mean_count)

kable(sp_means,booktabs = T, row.names = FALSE, digits = 3)
```

english	bird_guild	mean_count	number_routes	number_years
Hermit Thrush	Forest	0.311	15	19
Red-breasted Nuthatch	Forest	0.435	25	20
Scarlet Tanager	Forest	0.634	32	20
Nashville Warbler	Forest	0.803	23	20
Yellow-bellied Sapsucker	Forest	0.933	32	20
Black-throated Green Warbler	Forest	1.160	20	20
Chestnut-sided Warbler	Forest	1.581	32	20
Least Flycatcher	Forest	1.611	38	20
Black-and-white Warbler	Forest	1.961	28	20
White-throated Sparrow	Forest	2.303	28	20
Wood Thrush	Forest	2.651	39	20
American Redstart	Forest	2.726	36	20
Veery	Forest	3.199	35	20
Ovenbird	Forest	5.239	36	20
House Finch	Urban	1.503	35	20
Downy Woodpecker	Urban	1.518	42	20
Baltimore Oriole	Urban	6.415	41	20
Northern Cardinal	Urban	6.720	42	20
Black-capped Chickadee	Urban	7.893	41	20
Warbling Vireo	Urban	8.370	41	20
Blue Jay	Urban	8.607	42	20
House Wren	Urban	9.020	42	20
Yellow Warbler	Urban	12.774	42	20
Chipping Sparrow	Urban	19.590	42	20
Mourning Dove	Urban	27.656	42	20

english	bird_guild	mean_count	number_routes	number_years
Song Sparrow	Urban	42.663	42	20
Common Grackle	Urban	49.280	42	20
American Robin	Urban	63.593	42	20

Other useful info and tips

Missing year

You'll see also that even though these data span a 21-year period, there are only data for 20-years for each species. The BBS was cancelled during the COVID travel-restriction of spring 2020.

Covariates on species counts

The next 3 columns are more or less continuous variables that may be related to the observed species counts. These variables are calculated for each BBS route (so the values are repeated for every species and year those routes were surveyed), within a 1 km buffer surrounding the route.

```r
kable(head(birds[c(1:3,70:72),c(1,2,3,7:9)]),booktabs = T, row.names = FALSE, digits = 3)
```

RouteName	year	english	human_footprint	change_human_footprint	proportion_forest
DUNNVILLE	2018	Blue Jay	6.739	3.600	0.177
DUNNVILLE	2015	Black-capped Chickadee	6.739	3.600	0.177
DUNNVILLE	2018	Mourning Dove	6.739	3.600	0.177
AUBURN	2017	Scarlet Tanager	3.216	1.364	0.265
AUBURN	2017	Common Grackle	3.216	1.364	0.265
AUBURN	2015	House Wren	3.216	1.364	0.265

The column `human_footprint` represents a sum of some of the key human pressures on the environment as of 2018, and the column `change_human_footprint` shows an estimate of the change in a similar metric of human pressures between 2000 and 2015. These footprint measures are calculated following the methods in (Venter et al. 2016). The final column is the proportion of the area surrounding each route that is considered forest according to the 2020 Landcover of Canada produced by Natural Resources Canada.

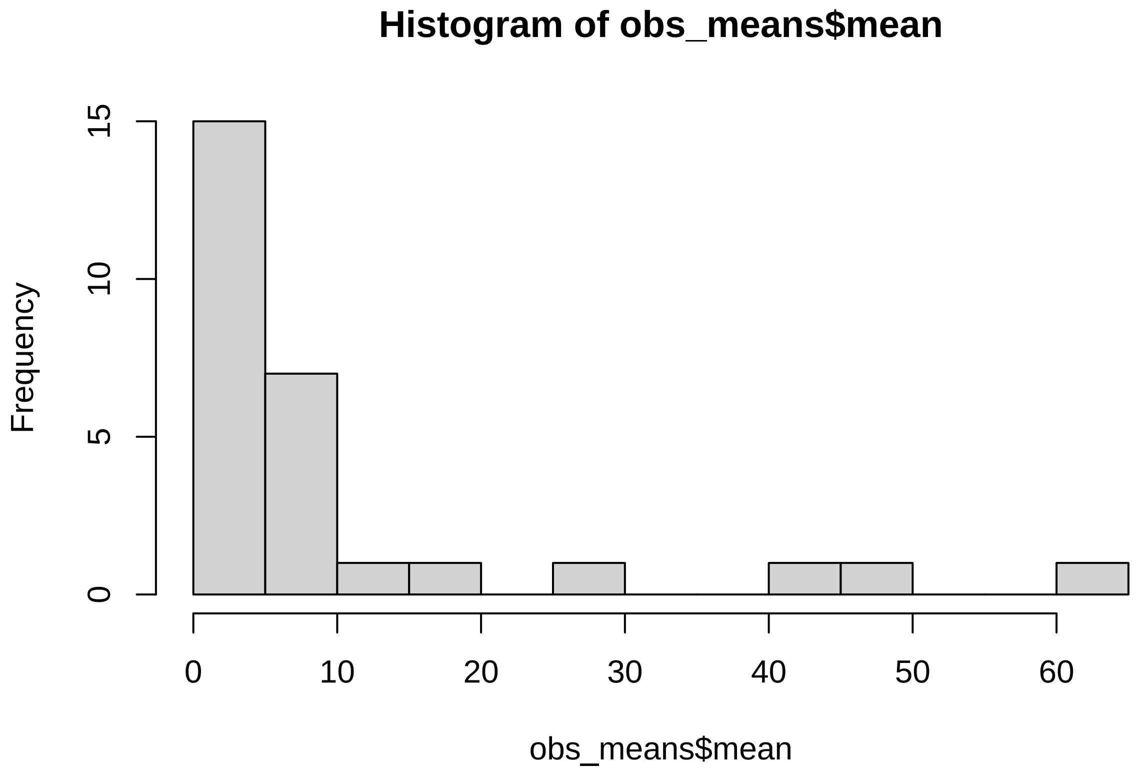
Priors with log-link models

When modeling counts, binomial responses such as presence-absence or other data that are constrained to be non-negative, models often use a log-link. It is important to remember the link function when thinking about appropriate priors for parameters in the model. Prior predictions are your friend.

If the model is relatively simple, you can explore the priors distributions by transforming the priors (which are scaled on the log-scale of the parameters) into the original scale to see how they relate to the data. So first, let's look at the observed mean counts for each species. They range from close to 0 all the way to about 60.

```
# calculating the mean counts for each species for comparison in the plots
obs_means <- birds %>%
  group_by(english) %>%
  summarise(mean = mean(count))

hist(obs_means$mean,breaks = 20)
```



For example, here are 4 alternate priors for the intercept terms in a log-link model estimating the abundance of forest birds. The first, a standard normal distribution.

$$\mathcal{N}(0,1)$$

Second, normal distribution with a positive mean, because the means of the counts are all positive (here we are forgetting the link function).

$$\mathcal{N}(5,1)$$

Third, a normal distribution with a large standard deviation, because the range of mean counts among species is pretty wide (again, we are forgetting the link and thinking about the variation on the original count-scale)

$$\mathcal{N}(0,10)$$

Finally, a normal distribution centered at 0 and with a slightly larger standard deviation than the standard normal.

$$\mathcal{N}(0,2)$$

I've generated random draws from these priors, re-transformed them using the exp() function, then added a rug-plot (dark lines along the x-axis) to indicate the observed means for each species. First the code, then the graphs.

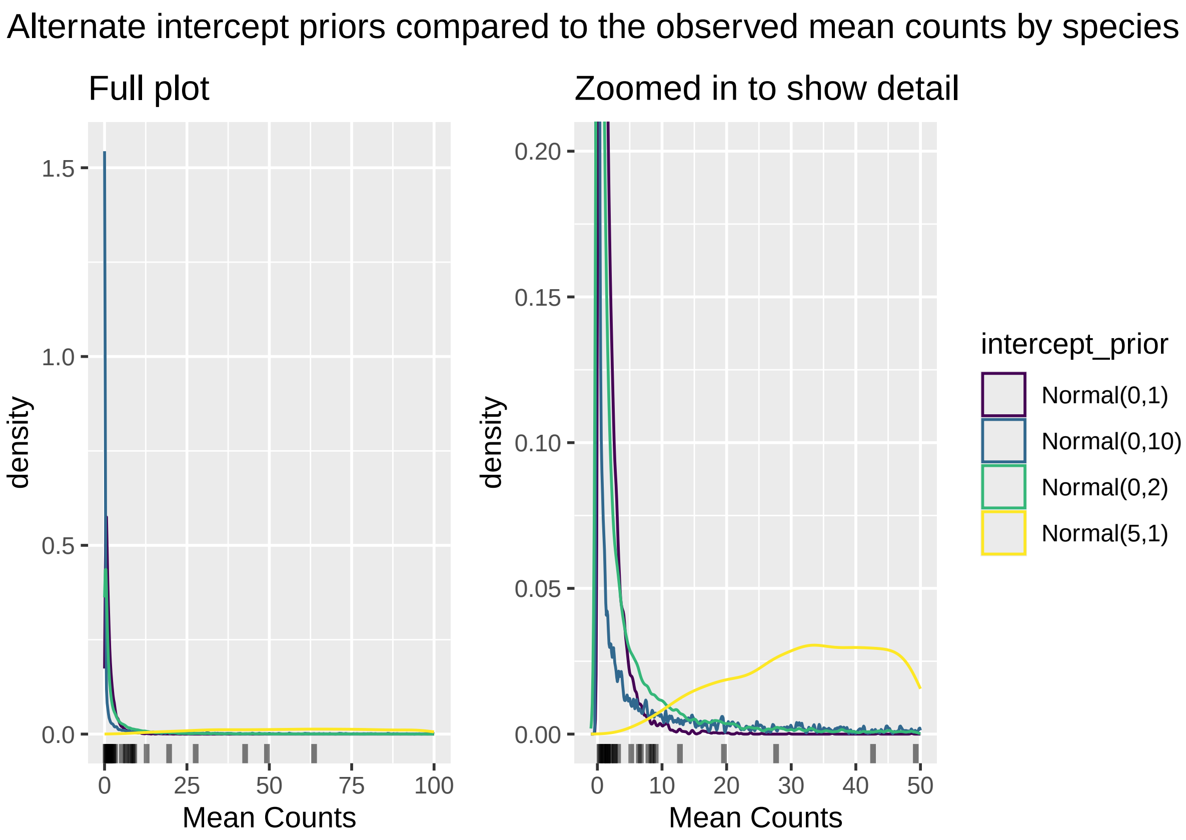
```r
nn <- 10000#number of samples to draw from the prior

# data frame with 10000 samples of log-scale priors transformed to reflect
# their effect on count-scale responses
mean_priors = data.frame(prior = exp(c(rnorm(nn,0,1),
                                   rnorm(nn,5,1),
                                   rnorm(nn,0,10),
                                   rnorm(nn,0,2))),
                         intercept_prior = c(rep("Normal(0,1)",nn),
                                             rep("Normal(5,1)",nn),
                                             rep("Normal(0,10)",nn),
                                             rep("Normal(0,2)",nn)))
bks = seq(0,100,0.5) # plotting set-up

# plotting the priors
prior_obs <- ggplot(data = mean_priors)+
  geom_density(aes(x = prior,colour = intercept_prior))+
  geom_rug(data = obs_means, aes(x = mean),inherit.aes = FALSE,
           size = 1,alpha = 0.5)+
  xlim(c(0,100))+
  scale_colour_viridis_d()+
  xlab("Mean Counts")+
  labs(title = "Full plot")

prior_obs_zoom <- ggplot(data = mean_priors)+
  geom_density(aes(x = prior,colour = intercept_prior))+
  geom_rug(data = obs_means, aes(x = mean),inherit.aes = FALSE,
           size = 1,alpha = 0.5)+
  xlim(c(-1,50))+
  coord_cartesian(ylim = c(0,0.2))+
  scale_colour_viridis_d()+
  xlab("Mean Counts")+
  labs(title = "Zoomed in to show detail")

print(prior_obs + prior_obs_zoom + plot_layout(guides = "collect")
      + plot_annotation(title = "Alternate intercept priors compared to the observed mean counts by spe
```



The plot on the left shows most of the four possible prior distributions, the version on the right is the same plot, but zoomed-in to show detail on the range of most of the observed means. So the priors with mean = 0 seem to capture the large number of observed means that are close to 0 relatively well, most mean counts are pretty small (< 3-5 birds). The prior with a positive mean value (normal(5,1)) doesn't fit the data at all. This positive prior could seem perfectly intuitive if you had forgotten about the log-link in the model (counts are always positive), but it actually has relatively little prior probability mass at the most commonly observed mean counts (< 1) and has most of its prior probability mass at values greater than the highest observed mean count. The standard normal is probably a bit restrictive, the 0-mean normal with a standard deviation of 2 might be close to capturing the likely range the best. The 0-mean normal with a very wide standard deviation (normal(0,10)) may be ok, although it has a very long right tail, 10% of prior is a values greater than 4.4953271×10^5 and the extreme peak at 0 implies that it may create some bias towards low values.

Numerical indicators for categorical variables

A reminder that the categories here are all defined by character variables. But `Stan` and `ulam` will require the groups to be defined as numeric indices. Here is one way to do that that relies on the fact that **R** treats factors as a categorical variable with a set number of possible values, so that `as.integer(factor(x))` always returns a vector of 1:n where n is the number of categories in the original vector `x`. There are also functions in a number of R packages that do this (transform characters to categorical integers).

```
sp_indicators <- birds %>% #creates a data frame that list each species indicator number
  select(english,french) %>% #retaining french names for use in later plotting, but not necessary
  distinct() %>% # selects just the unique rows so only one row per species
  mutate(species_ind = as.integer(factor(english))) %>% #creates a 1:n numeric indicator
  arrange(species_ind)

birds <- birds %>%  # adds the new indicator numbers to the original data frame
  left_join(.,sp_indicators,
            by = c("english","french"))

kable(head(sp_indicators),booktabs = T, row.names = FALSE)
```

english	french	species_ind
American Redstart	Paruline flamboyante	1
American Robin	Merle d'Amérique	2
Baltimore Oriole	Oriole de Baltimore	3
Black-and-white Warbler	Paruline noir et blanc	4
Black-capped Chickadee	Mésange à tête noire	5
Black-throated Green Warbler	Paruline à gorge noire	6

overthinking it a bit...

And here's an example of how to get a numeric indicator for species that is separate within two groups of species (species nested within guild). By using the `group_by(bird_guild)` function then a call to `mutate()`, the numeric indicators for species will be 1:14 for forest birds, and 1:14 for urban birds. This maybe a useful trick in the future, but be warned that this 2-dimensional grouping structure may also be difficult to correctly code into the model description in `ulam()`. Don't invest too much time in figuring out complex coding issues for this assignment. Although it's easy to code this kind of 2-dimensional grouping structure in a Stan model, I haven't gotten this kind of structure to work in `ulam()` yet.

```
sp_indicators_group <- birds %>% #creates a data frame that list each species indicator number
  select(english,french,bird_guild) %>%
  distinct() %>% # selects just the unique rows
  mutate(bird_guild_ind = as.integer(factor(bird_guild))) %>%
  group_by(bird_guild) %>% # separate grouping indicators for each bird_guild
  mutate(species_ind = as.integer(factor(english))) %>%
  arrange(species_ind)

kable(head(sp_indicators_group),booktabs = T, row.names = FALSE)
```

english	french	bird_guild	bird_guild_ind	species_ind
American Robin	Merle d'Amérique	Urban	2	1
American Redstart	Paruline flamboyante	Forest	1	1
Baltimore Oriole	Oriole de Baltimore	Urban	2	2
Black-and-white Warbler	Paruline noir et blanc	Forest	1	2
Black-capped Chickadee	Mésange à tête noire	Urban	2	3
Black-throated Green Warbler	Paruline à gorge noire	Forest	1	3

Hints for working with "big" data and possibly complex models

often when contemplating a large dataset such as this one, fitting each model can take a significant amount of time. The questions I've suggested here don't require the entire dataset. However, if you find yourself in a sitatution where you need to build a Bayesian model for a large dataset, consider some of the following tips to save time (and sanity).

- Start with a subset of the data. For example, if your model is estimating mean abundances by species, select only one or a few year's information to begin with. Similarly if your model is estimating changes through time, start with a few species or a few routes to get a working model, then apply the working model to all of the relvant data.
- Start with simple(r) models and gradually add components. If you're working towards a model with varying intercepts and slopes, start by getting a working model of just the intercepts, then add the slopes.
- Calculating PSIS (i.e., including `ulam(...,log_lik = TRUE)` when running the model) significantly increases the amount of time for the model to finish. Consider building your models with `ulam(...,log_lik = FALSE)` (the default), then add it once you're confident you've settled on a model or manageable set of models.
- Run the models in parallel (assuming your computer has sufficient cores), use the `ulam(..., chains=4, cores = 4)` to run chains in parallel, or try `ulam(..., chains=3, cores = 3)` if your machine has a total of 4 cores, it's best not to leave one free for the operating system, etc.

If you're interested in how the data were created, you can find everything in this GitHub repo: https://github.com/AdamCSmithCWS/Bayes_Stats_Assignment3

References

Hudson, Marie-Anne R., Charles M. Francis, Kate J. Campbell, Constance M. Downes, Adam C. Smith, and Keith L. Pardieck. 2017. "The Role of the North American Breeding Bird Survey in Conservation." *The Condor* 119 (3): 526–45. https://doi.org/10.1650/CONDOR-17-62.1.

Sauer, John R., Keith L. Pardieck, David J. Ziolkowski, Adam C. Smith, Marie-Anne R. Hudson, Vicente Rodriguez, Humberto Berlanga, Daniel K. Niven, and William A. Link. 2017. "The First 50 Years of the North American Breeding Bird Survey." *The Condor* 119 (3): 576–93. https://doi.org/10.1650/CONDOR-17-83.1.

Venter, Oscar, Eric W. Sanderson, Ainhoa Magrach, James R. Allan, Jutta Beher, Kendall R. Jones, Hugh P. Possingham, et al. 2016. "Sixteen Years of Change in the Global Terrestrial Human Footprint and Implications for Biodiversity Conservation." *Nature Communications* 7 (1): 12558. https://doi.org/10.1038/ncomms12558.