Hierchical Logistic Regression

In this case study we perform a Bayesian hierarchical logistic regression. This type of regression model exploits some kind of grouping structure that is present in the data. Group parameters are pooled together to share information, much like we saw in Case Study 5. At the same time we can also us the other predictor variables in a similar way to the regression models we have looked at in Case Study 6 and Case Study 7. So we are combining ideas that we have seen previously into a single model framework.

Just like previous case studies, the models we will use can be fit using the `brms` package. We keep the same structure as before, focusing on:

- Performing an initial assessment of the data;
- Fitting a hierarchical logistic regression model using the `brm` function;
- Assessing the model output and transforming the model parameters appropriately to facilitate interpretation and forming conclusions in the context of the data problem. We do this in two stages:
    - Looking at the output for predictor variables;
    - Looking at the output for the group parameters.

This case study will closely follow the accompanying lecture slides, so make sure to check these again if you would like any further context for what we are doing. Don't forget to go through the previous case studies if you haven't already.

Data exploration

We are going to investigate the book banning data set we considered in class. Briefly, the data set consists of a record of book challenges in the USA between 2000 and 2010, including information regarding the nature of the challenge, the state that challenge was made in, and whether or not the challenge was successful.

This data set as been taken from the R package `bayesrules` , and is available to download from https://www.scss.tcd.ie/~arwhite/Teaching/CS7DS3/book_banning.csv (https://www.scss.tcd.ie/~arwhite/Teaching/CS7DS3/book_banning.csv). Let's read the dataset into R. For convenience we're going to remove data points where the year of challenge is not available, and we're going to focus on a subset of the available variables. Let's briefly explore its structure:

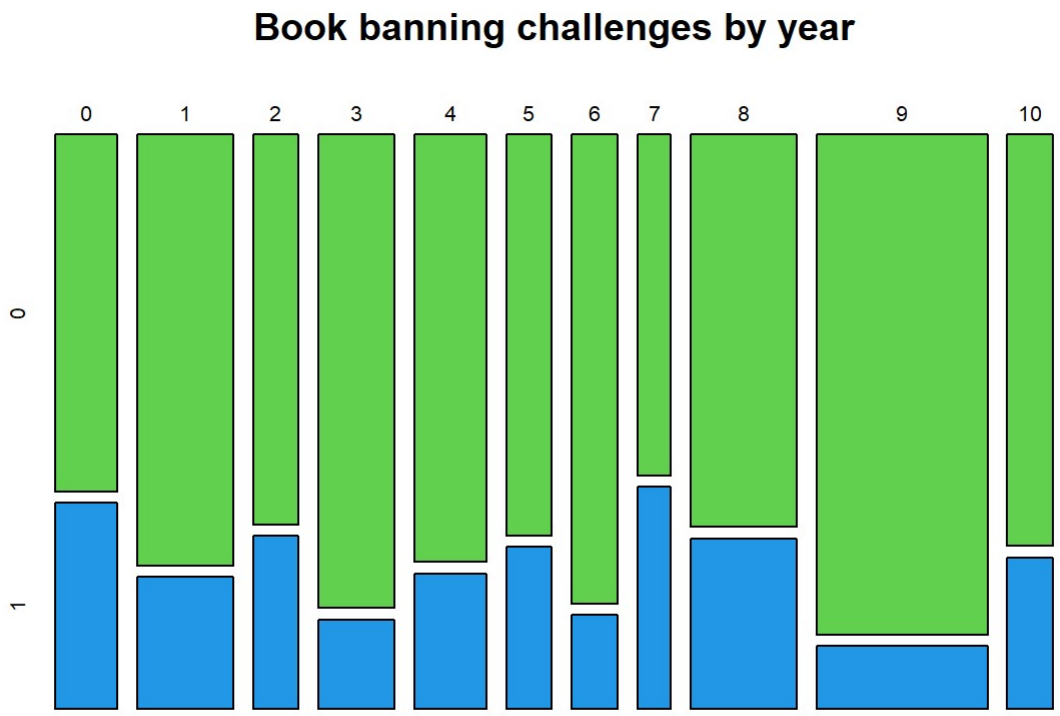
```
book_banning <- read.csv("https://www.scss.tcd.ie/~arwhite/Teaching/CS7DS3/book_banning.csv")
book_banning <- book_banning[!is.na(book_banning$year),]
book_banning$year_c <- book_banning$year - 2000
book_banning <- book_banning[, c(6:13, 18)]
book_banning$state <- as.factor(book_banning$state)
book_banning[sample(1:920, 6), ]
```

```
##     removed explicit antifamily occult language lgbtq violent state year_c
## 290       0        0          0      1        0     0       0    IN      8
## 688       0        1          0      0        1     0       0    PA      5
## 745       0        0          0      0        0     0       0    PA      9
## 259       0        0          0      0        0     0       0    IL      1
## 47        0        0          0      0        0     0       0    CA      4
## 564       1        0          0      0        0     1       0    OR      1
```

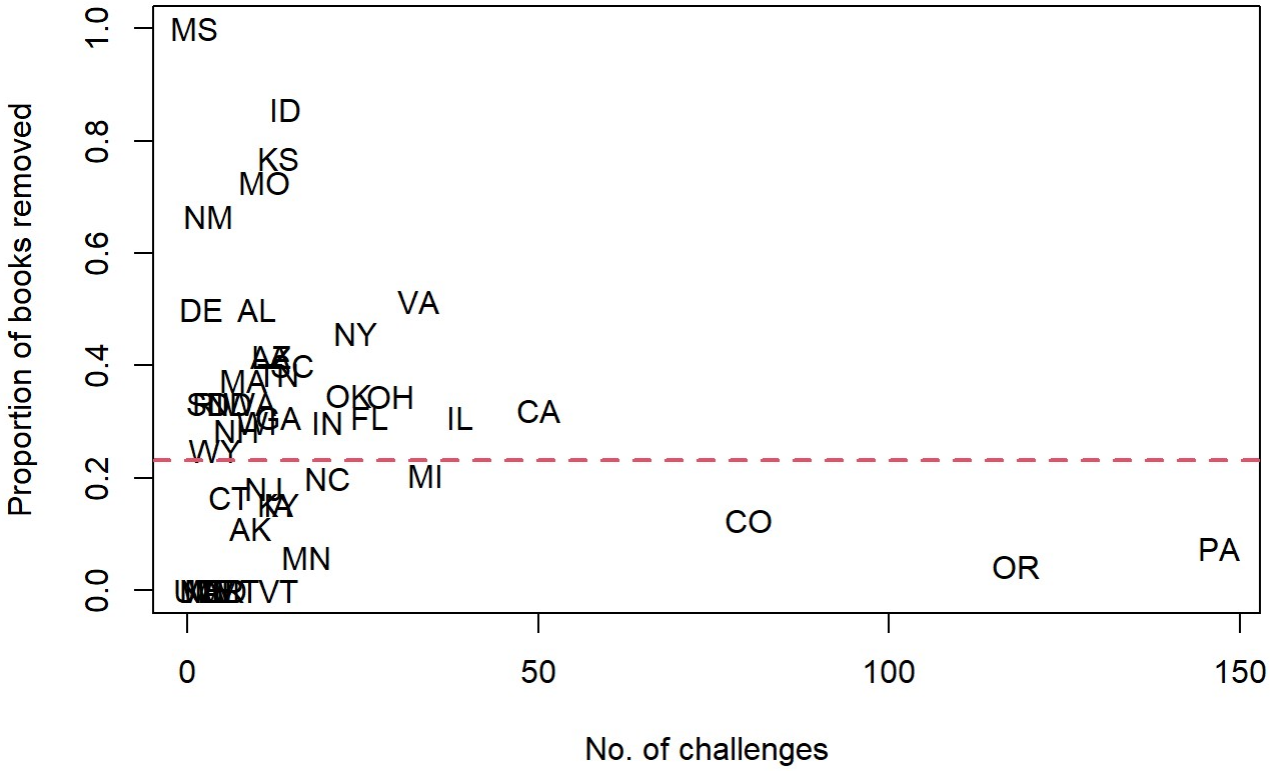
Our response variable is going to be `removed`, which indicates whether or not a challenge was successful. We are going to use variables `year_c`, `explicit`, `antifamily`, `occult`, `language`, `lgbtq`, and `violent` as predictor variables, and `state` as a grouping variable. Variables `explicit`, `antifamily`, `occult`, `language`, `lgbtq`, and `violent` are indicator variables, i.e., they take values of 0 or 1. In each case they indicate whether or not the nature of the challenge was related to the predictor variable name. FOr example, if the challenge was related to sexually explicit (i.e., `explicit = 1`) material. For ease of interpretation, `year_c` indicates how many years after 2000 the challenge took place, i.e. a value of 0 indicates that the challenge took place in 2000, a value of `indicates 2001, etc.

Below are some attempts summarise and visualise the data, focusing on the state level. Try to make sense of the output below:

```
mosaicplot(table(book_banning$year, book_banning$removed), col = c(3, 4), main = "Book bannin
g challenges by year")
```




```
state_n <- tapply(book_banning$removed, book_banning$state, length)
state_mean <- tapply(book_banning$removed, book_banning$state, mean)
plot(state_n, state_mean, xlab = "No. of challenges", ylab = "Proportion of books removed", c
ol = 0)
text(state_n, state_mean, levels(as.factor(book_banning$state)))
abline(h = mean(book_banning$removed=="1"), col = 2, lwd = 2, lty = 2)
```



Exercises

- Do you notice any trends in terms of the number of challenges being made or changes in proportion of successful challenges over time?
- Which states have the most challenges? Do you notice any differences in which states are likely to uphold a challenge?
- Using a `mosiacplot` or other summary measure, investigate whether any of the challenge terms such as `explicit` or `violent` appear to have any effect on the success of the challenge in question.

Logistic Regression modelling

We're going to fit a hierarchical logistic regression model to the data. The model we are going to run has several terms, including an intercept term β_0, independent predictor variables for each indicator variable term, and `year_c`. It will also have a grouping parameter for each state k. For this model, this has the form of a random intercept

$$\beta_{0,k} \sim \mathcal{N}(0, \sigma_b^2).$$

As this is a logistic regression model, we will be modelling the log-odds of a successful challenge as a linear function of the predictor variables and grouping term:

$$\log\left(\frac{\theta_{ik}}{1-\theta_{ik}}\right) = \beta_0 + \beta_{0,k} + \beta_1 x_{ik1} + \cdots + \beta_p x_{ikp}.$$

Here θ_{ik} is the probability that $y_{ik} = 1$, i.e., that the challenge to book i in state k is upheld and the book is removed from the state education system.

To fit the model in R, we use a similar `formula` syntax to previous case studies that used the `brm` function in the `brms` package, although we do not specify any interactions in this model. A key difference is the inclusion of the term `(1|state)` in the formula. This specifies that a random hierarchical intercept term should included in the model, and that the intercepts should be grouped at the state level.

Several additional arguments have the same or similar specification to the previous case study on logistic regression, e.g., specifying `family = bernoulli()`. In this model I have also specified a $\mathcal{N}(0, 10)$ prior for the regression parameters for the predictor variable. This is a weakly informative prior that should ensure some additional stability for the model estimates.

We can assess the model output using the default `plot` and `summary` functions, as before. When plotting the data I am first focusing on the predictor variables. we will look at the random intercepts later.

```
library("brms")
fit2 <- brm(removed ~ explicit + antifamily + occult + language + lgbtq + violent + year_c +
(1|state), data = book_banning, family = bernoulli(), prior = prior(normal(0,10), class = b))
```

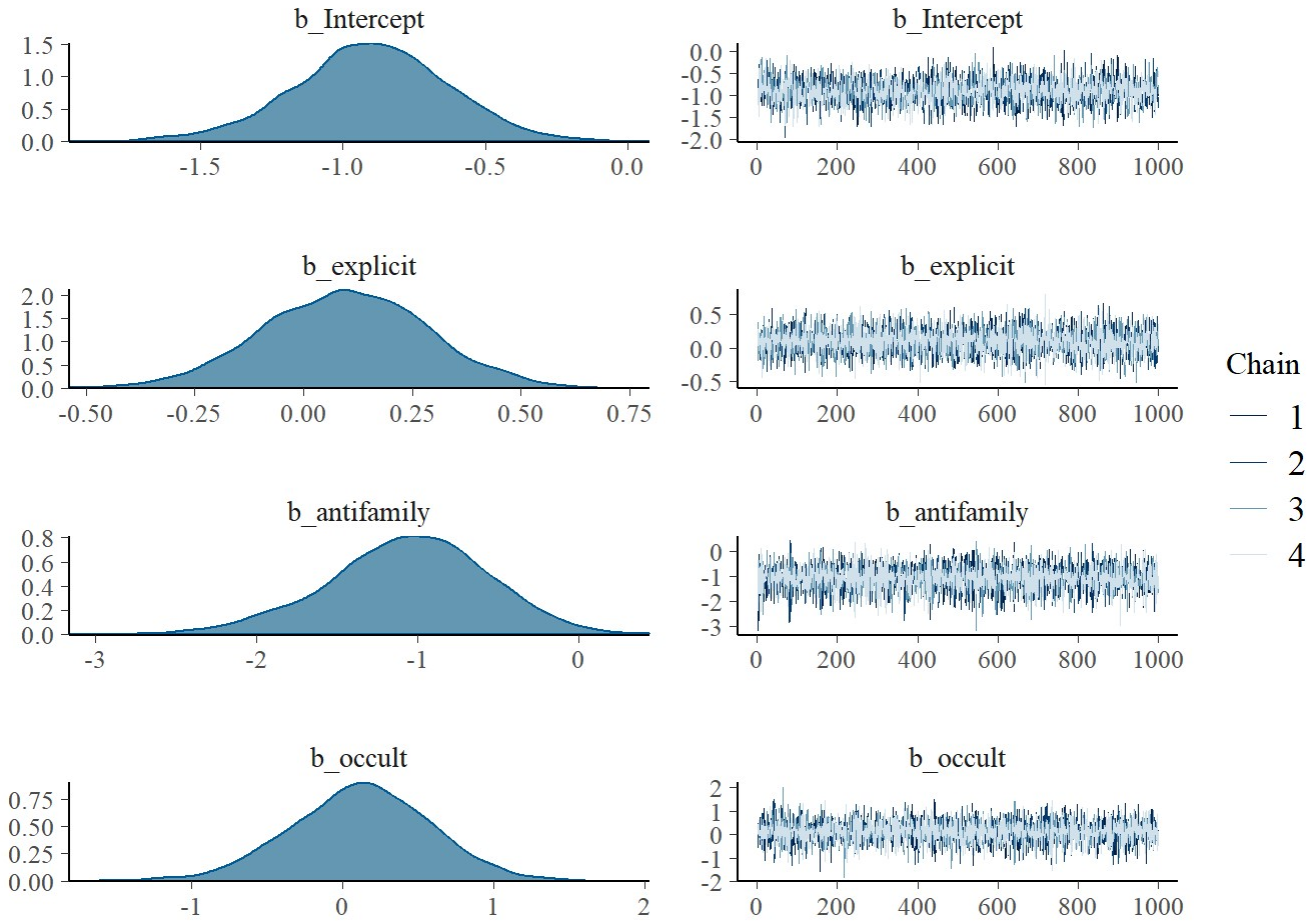
```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000253 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.53 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 3.256 seconds (Warm-up)
## Chain 1:                2.104 seconds (Sampling)
## Chain 1:                5.36 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.000234 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 2.34 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 3.209 seconds (Warm-up)
## Chain 2:                2.152 seconds (Sampling)
## Chain 2:                5.361 seconds (Total)
## Chain 2:
##
```

```
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.000243 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 2.43 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 3.332 seconds (Warm-up)
## Chain 3:                1.928 seconds (Sampling)
## Chain 3:                5.26 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.000251 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 2.51 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 3.998 seconds (Warm-up)
## Chain 4:                2.306 seconds (Sampling)
## Chain 4:                6.304 seconds (Total)
## Chain 4:
```

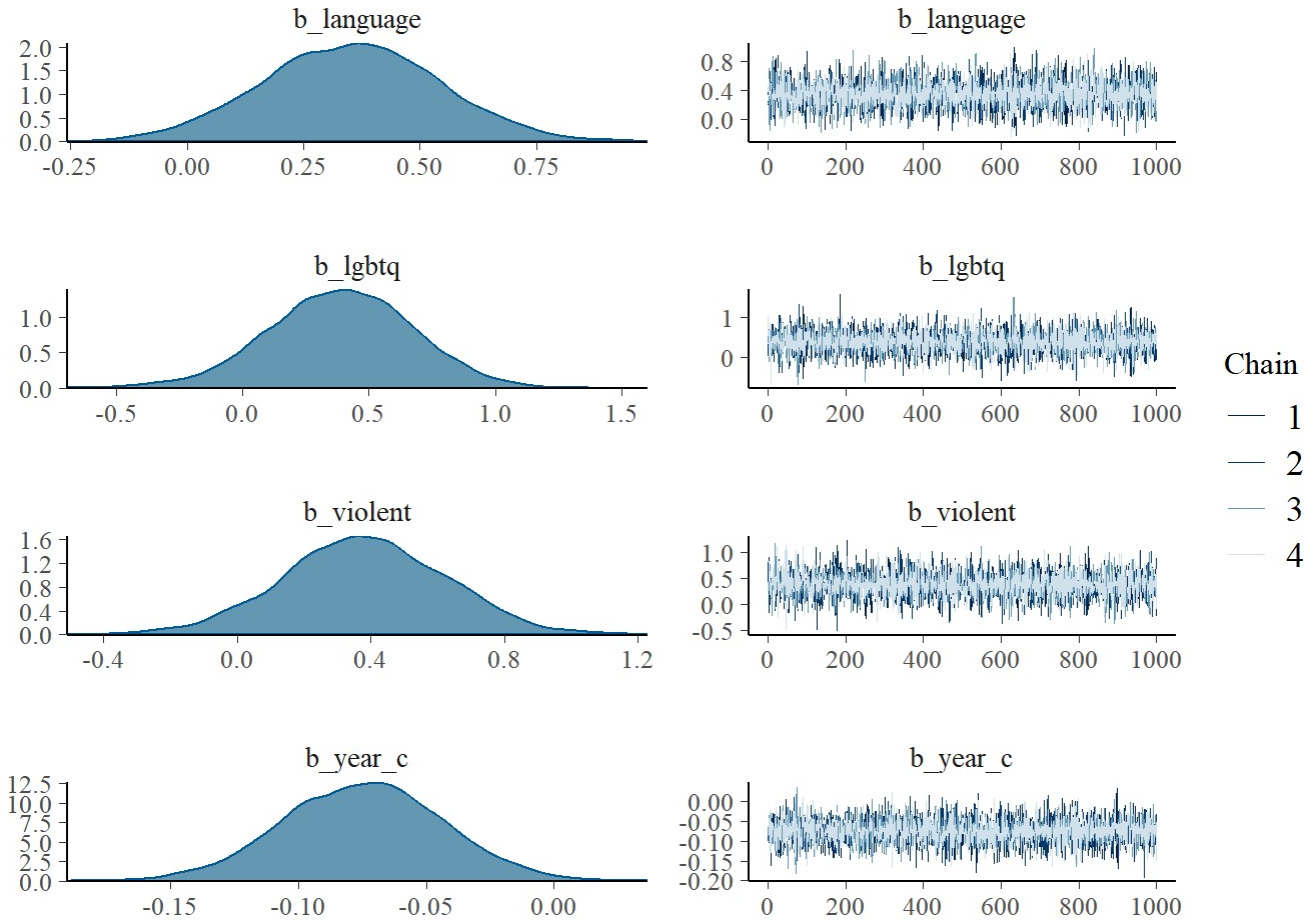
```
summary(fit2)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: removed ~ explicit + antifamily + occult + language + lgbtq + violent + year_c +
(1 | state)
##    Data: book_banning (Number of observations: 920)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Group-Level Effects:
## ~state (Number of levels: 47)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.01      0.18     0.70     1.41 1.00     1393     1991
##
## Population-Level Effects:
##            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     -0.90      0.27    -1.46    -0.37 1.00     3622     3222
## explicit       0.10      0.19    -0.27     0.46 1.00     6765     3241
## antifamily    -1.06      0.50    -2.12    -0.15 1.00     6559     3068
## occult         0.12      0.46    -0.81     1.01 1.00     6403     2836
## language       0.35      0.19    -0.01     0.72 1.00     6964     3197
## lgbtq          0.39      0.29    -0.20     0.93 1.00     7158     2979
## violent        0.38      0.24    -0.09     0.85 1.00     7381     3068
## year_c        -0.07      0.03    -0.14    -0.01 1.00     6235     3083
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(fit2, variable =c("b_Intercept", "b_explicit", "b_antifamily", "b_occult"))
```




```
plot(fit2, variable =c("b_language", "b_lgbtq", "b_violent", "b_year_c"))
```



Exercises

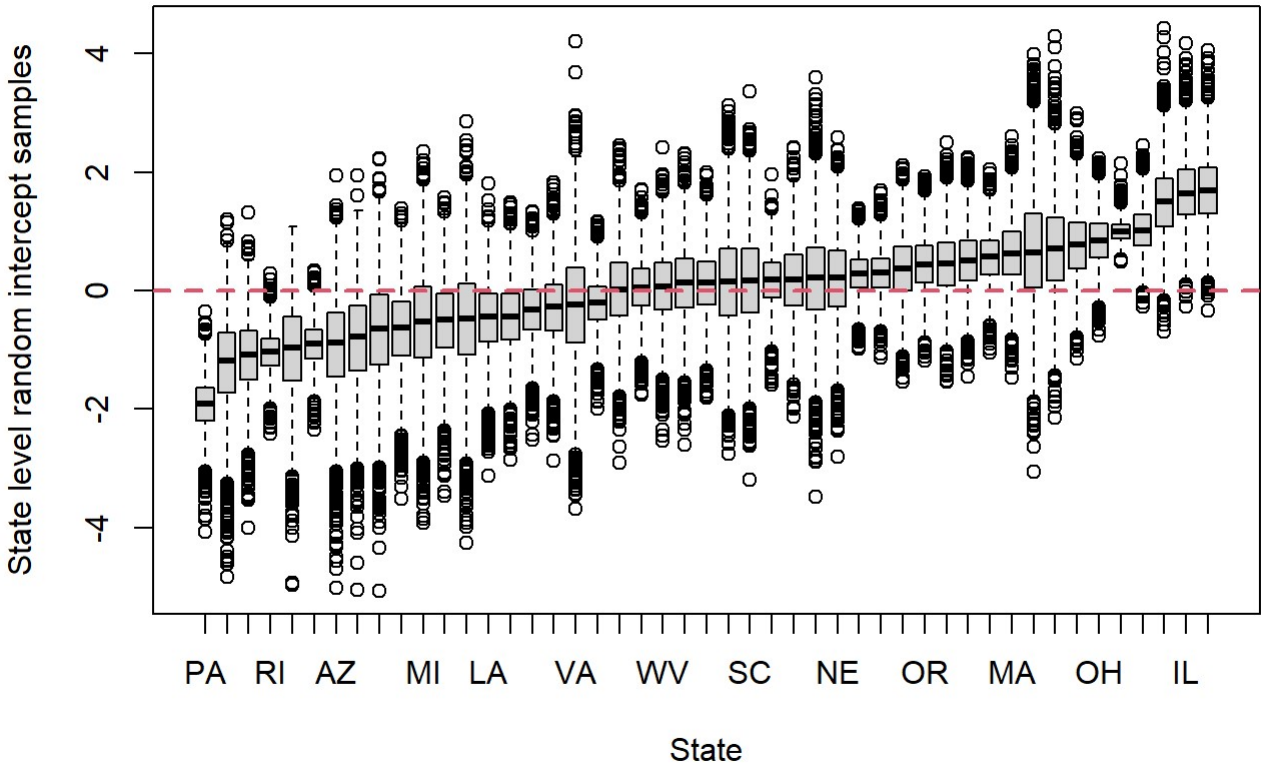
- Assess MCMC performance and confirm it is satisfactory.
- Use the output to interpret the estimated model parameters. What is the evidence that individual challenge categories will effect the probability of a book being banned, or that the probability of a challenge being successful has changed over time? Briefly summarise this output in layman's terms.
- Can you use the `conditional_effects` function to investigate how the probability of a successful challenge might change over time, accounting for different categories of challenge?

Assessing differences between states – assessing random intercepts

Next let's look at the random intercepts term. We'll extract the samples from the `fit2` object in a matrix format and inspect these more closely. We can analyse these terms in a similar way to the parameter estimates in the hierarchical modelling case study. Below we extract the terms, convert the matrix to a data frame with long format and add a second index variable. Then we use a boxplot to assess the samples, with states sorted by median estimate.

```
mat2 <- brms::as_draws_matrix(fit2)
state_r_effect <- data.frame(sample = as.numeric(mat2[, 9:55]), state = rep(unique(book_banni
ng$state), each = 4000))

state_ordered <- with(state_r_effect, reorder(state, sample, median))
boxplot(state_r_effect$sample ~ state_ordered, ylab = "State level random intercept samples",
xlab = "State")
abline(h = 0, col = 2, lwd = 2, lty = 2)
```



We see a range of values for each state here. Note that the coefficient estimates range from about -2 to 2 for median estimates and as far as -4 to 4 in the tails of the distributions. This is much larger than the coefficient ranges that we saw for individual predictors. This suggests that knowing the state in which the challenge took place is more influential on the probability than the year or the specific nature of the challenge itself.

Which states are especially influential on challenge result? Below we construct a matrix which computes quantiles at 20%, 50% and 80% levels for the state level coefficients.These values are exponentiated, so values < 1 indicate a negative impact on probabilities (directly a reduction of the baseline odds), while values > 1 indicate a positive impact on probabilities (an increase to the baseline odds.) We subset the matrix to identify which states have the strongest negative and positive effects buy highlighting the states exclude 0 from their upper and lower bounds respectively.

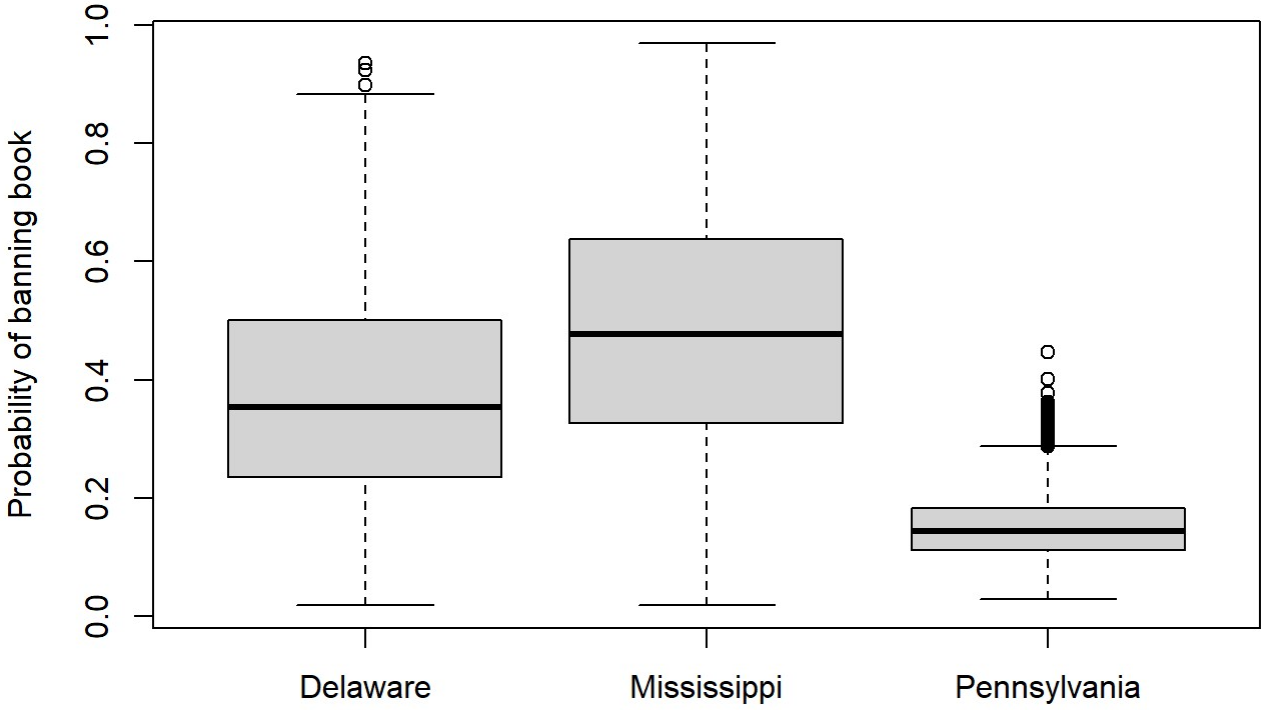
```
mat1 <- matrix(unlist(tapply(state_r_effect$sample, state_r_effect$state, quantile, prob = c
(0.2, 0.5, 0.8))), nrow = 3, ncol = 47, dimnames = list( c("0.2", "0.5", "0.8"), levels(as.fa
ctor(book_banning$state))))
mat1 <- mat1[, order(mat1[2, ])]
round(exp(mat1[, which(mat1[3, ] < 0)]), 2)
```

```
##       PA   WA   MO   RI   NC   CT   AZ   ME   AL
## 0.2 0.10 0.15 0.20 0.26 0.19 0.30 0.20 0.23 0.30
## 0.5 0.15 0.31 0.34 0.36 0.38 0.41 0.41 0.46 0.53
## 0.8 0.21 0.56 0.56 0.47 0.72 0.55 0.77 0.88 0.92
```

```
round(exp(mat1[, which(mat1[1, ] > 0)]), 2)
```

```
##       OR   SD   OK   MA   NY   AR   OH   AK   VT   MS   IL   KY
## 0.2 1.06 1.08 1.22 1.20 1.05 1.32 1.62 2.35 1.99 2.68 3.27 3.35
## 0.5 1.55 1.67 1.78 1.88 2.03 2.16 2.32 2.69 2.77 4.47 5.17 5.42
## 0.8 2.27 2.51 2.50 2.94 3.95 3.47 3.31 3.17 3.87 7.40 8.64 8.93
```

Finally, let’s compare the impact of two different states on the chances of a book being banned. Suppose a book challenge was raised on the grounds of explicit material, inappropriate language, and violent content, in 2009. This corresponds to observation 173, and the challenge took place in Delaware (DE). Let’s ask three questions: 1) What is the model’s predicted probability that the challenge was successful? 2) What is the model’s predicted probability that the challenge was successful if the challenge took place in Mississippi (MS) and not Delaware? 3) What is the model’s predicted probability that the challenge was successful if the challenge took place in Pennsylvania (PA) and not Delaware?



We can see that the probability of the challenge being upheld is affected by the state that is involved. The chances of the challenge being successful are low (about 35%) for Delaware, increase to about even (50%) for Mississippi, and reduce considerably to about 15% for Pennsylvania. We can also see that there is a considerable degree of uncertainty in the model's estimated probabilities for Delaware and Mississippi, although comparably less so for Pennsylvania.

Exercises

- Which states appear to have the strongest positive and negative effects? Repeat the quantile analysis but at the 5% level (i.e., 2.5%, 50% and 97.5%). Which states, if any, remain notable?
- Compare the estimated random intercept values to the raw estimates (`state_mean`) you obtained earlier with the mean or median random intercept values. (Bear in mind this is not quite a like for like comparison, for a few different reasons.) How similar/different are these estimates, e.g., ranking states ? * Why do you that there is less uncertainty for the probability of the book being banned when the state in question is Pennsylvania?
- Using the `predict` function, assess the predictive performance of the `fit2` model.