It honestly changed my whole outlook on statistics, so I couldn’t recommend it more . One of the most compelling cases for using Bayesian statistics is with a collection of statistical tools called *linear mixed models* or *multilevel/hierarchical models*. It’s common that data are grouped or clustered in some way. Often in psychology we have repeated observations nested within participants, so we know that data coming from the same participant will share some variance. Linear mixed models are powerful tools for dealing with multilevel data, usually in the form of modeling *random intercepts* and *random slopes*. In this tutorial I assume familiarity with linear regression and some background knowledge in Bayesian inference, such that you should have some familiarity with *priors* and *posterior distributions*.

Why use Bayesian instead of Frequentist statistics? One reason is that *prior* knowledge about a parameter – for example, it’s distribution – can be incorporated in the model. Another reason especially relevant to linear mixed models is that we can easily include multiple random intercepts and slopes without running into the same stringent sample size requirements as with frequentist approaches.

**Setting it All Up**

Installing and running *brms* is a bit more complicated than your run-of-the-mill R packages. Because *brms* uses STAN as its back-end engine to perform Bayesian analysis, you will need to install *rstan*. Carefully follow the instructions and you should have no problem. Once you’ve done that you should be able to install *brms* and load it up.

library(brms)

## Loading required package: Rcpp

## Loading 'brms' package (version 2.10.0). Useful instructions

## can be found by typing help('brms'). A more detailed introduction

## to the package is available through vignette('brms\_overview').

It’s normal to get some warning messages upon loading the package. That’s fine. We can now load up the *tidyverse* for data manipulation and visualization.

library(tidyverse)

The data is available on [here](https://github.com/MultiLevelAnalysis/Datasets-third-edition-Multilevel-book/blob/master/chapter%205/Curran/CurranLong.sav) in a .sav format. You’ll need the *haven* package to read it into R. Make sure you import it from your working directory or specify the path to your downloads folder.

library(haven)

The dataset consists of 405 children in the first two years of elementary school measured over four assessment points on reading recognition and antisocial behaviour. As well, it included time invariant covariates, such as emotional support and cognitive stimulation. For this post, we’ll focus on reading and cognitive stimulation, and we’ll use Bayesian Linear Mixed Models to address a number of questions about children’s reading ability.

For now, we’ll omit assessment periods with missing data, but we’ll return to the issue of missing data at the end of this demonstration.

curran\_dat <- read\_sav("CurranLong.sav") %>%

select(id, occasion, read, homecog) %>%

filter(complete.cases(.))

curran\_dat

## # A tibble: 1,325 x 4

## id occasion read homecog

##

## 1 22 0 2.1 13

## 2 22 1 3.9 13

## 3 34 0 2.1 9

## 4 34 1 2.9 9

## 5 34 2 4.5 9

## 6 34 3 4.5 9

## 7 58 0 2.3 9

## 8 58 1 4.5 9

## 9 58 2 4.2 9

## 10 58 3 4.6 9

## # ... with 1,315 more rows

The data is conveniently in long form already, which means that each row represents a single \(j^{th}\) assessment period for an \(i^{th}\) child. With no other data cleaning to do, we can move ahead with our research questions.

**Model 1: One Random Effect, No Covariates**

We’ll start with a population-level intercept and a random intercept for participants. Our formula looks like this:

\[READ\_{ij} = \beta\_0+u\_{0i}+\epsilon\_{ij}\]

Where \(\beta\_0\) is the population intercept, \(u\_{0i}\) is a random intercept term for an \(i^{th}\) participant, and \(\epsilon\_{ij}\) is a random noise term for an \(i^{th}\) participant and \(j^{th}\) assessment period.

If you find formulae distasteful, that’s OK. They’re not essential to understanding how linear mixed models work. conceptually, think of the \(u\_{0i}\) in the above formula as giving a little *nudge* to the intercept for each participant.

Here I’ll walk carefully through the *brms* code and then each subsequent model will just build one this one. We start by providing our data. The *family* argument is where we supply a distribution for our outcome variable, reading. We’ll use the default *gaussian* to indicate that reading should be normally distributed. The next line is where we specify our model formula. If you’re familiar with mixed effects models in the *lme4* package you’ll be happy to see there are no differences here. I’ve only included the population level intercept ‘1’ after ‘read ~’ to be precise, but this will be included by default. The random intercept is indicated by the ‘(1 | id)’ as in typical mixed effects notation. We then specify some priors. Our first prior is for the population-level *Intercept*. We’ll specify a fairly wide normal distribution for the intercept.

If this whole business of deciding on priors seems strange and arbitrary, just note that a wide prior such as this one will have little bearing on the posterior distribution. If instead we were extremely confident that the intercept was zero, we could use a narrow prior of ‘normal(0, .05)’ which would have a strong effect on the resulting posterior distribution.

The next prior is for the *standard deviation* of the random effects. Unlike an intercept which can be positive or negative, variance (and by association, standard deviation) can only be positive, so we specify a *cauchy* distribution that constrains the sd to be positive. We do the same for *sigma* which is the overall variability.

The rest of the code has to do with how the Markov Chain Monte Carlo (MCMC) algorithm runs. Without going too much into the *whys*, we will run 2,000 chained simulations. The first 1,000 of these will be in the warmup period and we will be rejected. We will run 4 parallel chains on 4 cores (if your computer has fewer cores you will want to reduce this). To help the model converge, I’ve increased the adapt\_delta and max\_treedepth arguments. Finally, an arbitrary seed number for reproducibility.

read1 <- brm(data = curran\_dat,

family = gaussian,

formula = read ~ 1 + (1 | id),

prior = c(prior(normal(0, 10), class = Intercept),

prior(cauchy(0, 1), class = sd),

prior(cauchy(0, 1), class = sigma)),

iter = 2000, warmup = 1000, chains = 4, cores = 4,

control = list(adapt\_delta = .975, max\_treedepth = 20),

seed = 190831)

print(read1)

## Family: gaussian

## Links: mu = identity; sigma = identity

## Formula: read ~ 1 + (1 | id)

## Data: curran\_dat (Number of observations: 1325)

## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;

## total post-warmup samples = 4000

##

## Group-Level Effects:

## ~id (Number of levels: 405)

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sd(Intercept) 0.54 0.08 0.37 0.68 1.00 992 1847

##

## Population-Level Effects:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## Intercept 4.11 0.05 4.01 4.21 1.00 4866 3424

##

## Family Specific Parameters:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sigma 1.55 0.04 1.48 1.62 1.00 2219 2827

##

## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample

## is a crude measure of effective sample size, and Rhat is the potential

## scale reduction factor on split chains (at convergence, Rhat = 1).

What does the output tell us? Right away, we see an estimate of an Intercept of 4.11 within a 95% *credible interval* (you might call it the Bayesian version of a confidence interval, although they’re not quite the same thing) of 4.01 and 4.21. What’s more interesting is the estimate of standard deviation of the random intercepts ‘sd(Intercept)’. So there is person-level variability in reading scores. In turn, *sigma* is the estimate of the overall variability in reading scores.

We can further inspect these estimates by looking at the traceplots and the posterior distributions.

plot(read1)

Each row corresponds with a parameter in the model. The traceplot shows the MCMC samples. These should resemble ‘hairy caterpillars’ – we don’t want to see the chain getting stuck near a particular value.

For a more intuitive understanding of what the mixed effects model is doing, let’s compare a sample of the observed values with the model predicted values. *brms* outputs fitted values using the *fitted* function. We’ll sample 6 participants and see their observed reading scores and model-derived reading scores.

set.seed(25)

curran\_dat %>%

bind\_cols(as\_tibble(fitted(read1))) %>%

group\_by(id) %>%

nest() %>%

sample\_n(6) %>%

unnest() %>%

ggplot() +

geom\_point(aes(x = occasion, y = read), size = 4, alpha = .75, color = "dodgerblue2") +

geom\_point(aes(x = occasion, y = Estimate), shape = 1, size = 4, stroke = 1.5) +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Model 1: One Random Effect, No Covariates",

subtitle = "Blue points are observed values. Black circles are fitted values.") +

scale\_x\_continuous(expand = c(.075, .075), breaks = 0:3) +

facet\_wrap(~id, nrow = 1) +

theme\_bw(base\_size = 14) +

theme(plot.title = element\_text(hjust = .5))

We can see that each estimate is drawn toward the population-level mean and its cluster-level mean. This phenomenon is called *partial-pooling* or *shrinkage*. If we’d run the model without the random effect, all of the estimates (black circles) would be on a single horizontal line. But here we have some variation between individuals. Notice too how points further away from the means (population- and cluster-level) are pulled more strongly. However, there is clearly an increase in reading ability over time that is being ignored by this model.

**Model 2: Two Random Effects, No Covariates**

Let’s add a random intercept for assessment period (labelled ‘occasion’ in this dataset). This model will recognize that observations are nested within participants *and* assessment periods. For now, we’ll consider assessment as a random effect because we are still only interested in the variability in reading scores. Here’s our new formula:

\[READ\_{ij} = \beta\_0+u\_{0i}+w\_{0j}+\epsilon\_{ij}\]

Now we have a term \(w\_{0j}\) for the random intercept for the \(j^{th}\) assessment period.

The only change to our model code is that we’ve added a second random intercept: ‘(1 | occasion)’.

read2 <- brm(data = curran\_dat,

family = gaussian,

read ~ 1 + (1 | id) + (1 | occasion),

prior = c(prior(normal(0, 10), class = Intercept),

prior(cauchy(0, 1), class = sd),

prior(cauchy(0, 1), class = sigma)),

iter = 2000, warmup = 1000, chains = 4, cores = 4,

control = list(adapt\_delta = .975, max\_treedepth = 20),

seed = 190831)

print(read2)

## Family: gaussian

## Links: mu = identity; sigma = identity

## Formula: read ~ 1 + (1 | id) + (1 | occasion)

## Data: curran\_dat (Number of observations: 1325)

## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;

## total post-warmup samples = 4000

##

## Group-Level Effects:

## ~id (Number of levels: 405)

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sd(Intercept) 0.89 0.04 0.82 0.96 1.00 1270 2122

##

## ~occasion (Number of levels: 4)

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sd(Intercept) 1.71 0.86 0.79 3.99 1.00 1701 1878

##

## Population-Level Effects:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## Intercept 4.32 0.89 2.46 6.08 1.00 1426 1848

##

## Family Specific Parameters:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sigma 0.64 0.01 0.61 0.67 1.00 3630 3056

##

## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample

## is a crude measure of effective sample size, and Rhat is the potential

## scale reduction factor on split chains (at convergence, Rhat = 1).

Looking again at our output now, we see variability in the random intercepts for id, but even larger variability across occasions (assessment periods). In fact, the overall variability, sigma, is quite reduced from the previous model. This demonstrates an important lesson about mixed models: that they decompose the overall variance (sigma) into cluster-level variance. If we just ran a linear regression, it wouldn’t differentiate between measurement error and variation between participants and time points.

Let’s again compare the estimates with the observed values. We’ll use the same subsample as before to illustrate the effect of adding a second random effect.

set.seed(25)

curran\_dat %>%

bind\_cols(as\_tibble(fitted(read2))) %>%

group\_by(id) %>%

nest() %>%

sample\_n(6) %>%

unnest() %>%

ggplot() +

geom\_point(aes(x = occasion, y = read), size = 4, alpha = .75, color = "dodgerblue2") +

geom\_point(aes(x = occasion, y = Estimate), shape = 1, size = 4, stroke = 1.5) +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Model 2: Two Random Effects, No Covariates",

subtitle = "Blue points are observed values. Black circles are fitted values.") +

scale\_x\_continuous(expand = c(.075, .075), breaks = 0:3) +

facet\_wrap(~id, nrow = 1) +

theme\_bw(base\_size = 14) +

theme(plot.title = element\_text(hjust = .5))

We can see that the model now adjusts the points based on the assessment-level intercept. The fact that the black circles follow a nearly linear increase is telling us that there is probably an effect of time, such that children get better at reading over time.

**Model 3: One Random Effect, One Covariate**

Now let’s make a conceptual shift. Maybe as researchers we decide that we’re interested in the effect of time (assessment period) on reading ability. Since each assessment was spaced equally apart in time and we have good reason to think that time has a global effect on reading, it makes sense to include it as a fixed effect (this sort of conceptual decision should be made beforehand, but for illustrative purposes I’ve shown it as a random and a fixed effect).

It’s always a good idea to visualize what’s going on so that we know if our assumptions are tenable. we’ll create a lineplot showing the change in reading ability over time, but with each line representing an individual in our sample.

curran\_dat %>%

ggplot(aes(x = occasion, y = read, group = id)) +

geom\_line(size = .75, alpha = .20) +

labs(x = "Assessment Period",

y = "Reading Ability") +

theme\_minimal(base\_size = 16)

There’s a clear upward trend – most children improve in their reading over time. We can now proceed with formally testing the growth in reading:

\[READ\_{ij} = \beta\_0+\beta\_1Time\_j+u\_{0i}+\epsilon\_{ij}\]

Here’s the model code. Note that we are still modeling the random effect of id, but now occasion is a covariate (predictor). We need to specify a new prior, *b* for beta. The beta estimate for the effect of time on reading ability is assumed to derive from a normal distribution with mean 0 and sd 1.

read3 <- brm(data = curran\_dat,

family = gaussian,

read ~ 1 + occasion + (1 | id),

prior = c(prior(normal(0, 10), class = Intercept),

prior(normal(0, 1), class = b),

prior(cauchy(0, 1), class = sd),

prior(cauchy(0, 1), class = sigma)),

iter = 2000, warmup = 1000, chains = 4, cores = 4,

control = list(adapt\_delta = .975, max\_treedepth = 20),

seed = 190831)

print(read3)

## Family: gaussian

## Links: mu = identity; sigma = identity

## Formula: read ~ 1 + occasion + (1 | id)

## Data: curran\_dat (Number of observations: 1325)

## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;

## total post-warmup samples = 4000

##

## Group-Level Effects:

## ~id (Number of levels: 405)

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sd(Intercept) 0.89 0.04 0.82 0.96 1.00 1204 1906

##

## Population-Level Effects:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## Intercept 2.70 0.05 2.60 2.81 1.00 1036 1692

## occasion 1.10 0.02 1.07 1.14 1.00 6696 3112

##

## Family Specific Parameters:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sigma 0.68 0.02 0.65 0.71 1.00 3004 3138

##

## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample

## is a crude measure of effective sample size, and Rhat is the potential

## scale reduction factor on split chains (at convergence, Rhat = 1).

Our output gives a smaller intercept because it’s not telling us the mean any more, instead it’s the y-intercept of the regression line. We have an estimate of beta as 1.10 within a 95% credible interval of 1.07 and 1.14. So there’s strong evidence of a positive effect of time on reading ability.

set.seed(25)

curran\_dat %>%

bind\_cols(as\_tibble(fitted(read3))) %>%

group\_by(id) %>%

nest() %>%

sample\_n(6) %>%

unnest() %>%

ggplot() +

geom\_point(aes(x = occasion, y = read), size = 4, alpha = .75, color = "dodgerblue2") +

geom\_point(aes(x = occasion, y = Estimate), shape = 1, size = 4, stroke = 1.5) +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Model 3: One Random Effect, One Covariate",

subtitle = "Blue points are observed values. Black circles are fitted values.") +

scale\_x\_continuous(expand = c(.075, .075), breaks = 0:3) +

facet\_wrap(~id, nrow = 1) +

theme\_bw(base\_size = 14) +

theme(plot.title = element\_text(hjust = .5))

The estimates aren’t actually that different from the model with two random effects. The main difference now is that the black circles show a strictly linear trend that is the same across participants. But maybe this assumption of equal effects across the board isn’t tenable either.

**Model 4: One Random Slope, One Covariate**

Let’s ramp things up by modeling a random slope for each participant:

\[READ\_{ij} = \beta\_0+u\_{0i}+\beta\_1Time\_j+u\_{1ij}+\epsilon\_{ij}\]

Where \(u\_{1ij}\) is the random slope for the effect of assessment period between participants.

A random slope model also has a random intercept, but now, the slope for time on reading ability will be different for each participant: ‘(1 + occasion | id)’. Another change to our model code is a new prior specifying the correlation between the Intercept and the beta for occasion. But all you need to know here is that the higher above 1 the value is, the more “speculative” the model is of strong correlations.

read4 <- brm(data = curran\_dat,

family = gaussian,

read ~ 1 + occasion + (1 + occasion | id),

prior = c(prior(normal(0, 10), class = Intercept),

prior(normal(0, 1), class = b),

prior(cauchy(0, 1), class = sd),

prior(cauchy(0, 1), class = sigma),

prior(lkj\_corr\_cholesky(1.5), class = cor)),

iter = 2000, warmup = 1000, chains = 4, cores = 4,

control = list(adapt\_delta = .975, max\_treedepth = 20),

seed = 190831)

print(read4)

## Family: gaussian

## Links: mu = identity; sigma = identity

## Formula: read ~ 1 + occasion + (1 + occasion | id)

## Data: curran\_dat (Number of observations: 1325)

## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;

## total post-warmup samples = 4000

##

## Group-Level Effects:

## ~id (Number of levels: 405)

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS

## sd(Intercept) 0.76 0.04 0.68 0.84 1.00 1279

## sd(occasion) 0.27 0.02 0.22 0.32 1.00 614

## cor(Intercept,occasion) 0.29 0.12 0.07 0.53 1.00 664

## Tail\_ESS

## sd(Intercept) 2218

## sd(occasion) 1082

## cor(Intercept,occasion) 1070

##

## Population-Level Effects:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## Intercept 2.70 0.05 2.60 2.79 1.00 1465 2297

## occasion 1.12 0.02 1.08 1.16 1.00 2963 3299

##

## Family Specific Parameters:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sigma 0.59 0.02 0.56 0.62 1.00 908 1758

##

## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample

## is a crude measure of effective sample size, and Rhat is the potential

## scale reduction factor on split chains (at convergence, Rhat = 1).

set.seed(25)

curran\_dat %>%

bind\_cols(as\_tibble(fitted(read4))) %>%

group\_by(id) %>%

nest() %>%

sample\_n(6) %>%

unnest() %>%

ggplot() +

geom\_point(aes(x = occasion, y = read), size = 4, alpha = .75, color = "dodgerblue2") +

geom\_point(aes(x = occasion, y = Estimate), shape = 1, size = 4, stroke = 1.5) +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Model 4: One Random Slope, One Covariate",

subtitle = "Blue points are observed values. Black circles are fitted values.") +

scale\_x\_continuous(expand = c(.075, .075), breaks = 0:3) +

facet\_wrap(~id, nrow = 1) +

theme\_bw(base\_size = 14) +

theme(plot.title = element\_text(hjust = .5))

I won’t belabour the output here. Just notice how the strength of the linear trend for the fitted values is different for each participant. We can see this across the whole sample with the following code:

curran\_dat %>%

bind\_cols(as\_tibble(fitted(read4))) %>%

ggplot() +

geom\_line(aes(x = occasion, y = Estimate, group = id), size = .75, alpha = .30) +

geom\_line(aes(x = occasion, y = read, group = id), size = .75, alpha = .15, color = "dodgerblue2") +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Model 4: One Random Slope, One Covariate",

subtitle = "Blue lines are observed values. Black lines are fitted.") +

theme\_minimal(base\_size = 16) +

theme(plot.title = element\_text(hjust = .5))

**Model 5: One Random Slope, Two Covariates**

Suppose we wanted to see whether there’s an effect of the amount of *cognitive stimulation* that children are exposed to at home on reading ability over time. Maybe cognitive stimulation impacts some children more strongly than others. Our last formula would thus be:

\[READ\_{ij} = \beta\_0+u\_{0i}+\beta\_1Time\_j+u\_{1ij}+\beta\_{2}Cog\_i+u\_{2ij}+\epsilon\_{ij}\]

Does the hypothesis hold up to a visual inspection? For illustrative purposes, we can split the data into equal bins of low, medium, and high cognitive stimulation. Of course, because stimulation is a continuous variable, it will stay continuous in our Bayesian model.

curran\_dat %>%

mutate(homecog\_grp = factor(ntile(homecog, 3))) %>%

ggplot(aes(x = occasion, y = read)) +

geom\_line(aes(group = id, color = homecog\_grp), size = .75, alpha = .50) +

geom\_smooth(method = 'lm', color = "black", size = 1.5) +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Cognitive Home Environment") +

facet\_wrap(~homecog\_grp, labeller = as\_labeller(c("1" = "Low",

"2" = "Medium",

"3" = "High"))) +

guides(color = FALSE) +

theme\_minimal(base\_size = 16) +

theme(plot.title = element\_text(hjust = .5, size = 14))

If there is an effect of cognitive stimulation, it’s a small one. So let’s find out:

read5 <- brm(data = curran\_dat,

family = gaussian,

read ~ 1 + occasion + homecog + (1 + occasion + homecog | id),

prior = c(prior(normal(0, 5), class = Intercept),

prior(normal(0, 1), class = b),

prior(cauchy(0, 1), class = sd),

prior(cauchy(0, 1), class = sigma),

prior(lkj\_corr\_cholesky(1.5), class = cor)),

iter = 2000, warmup = 1000, chains = 4, cores = 4,

control = list(adapt\_delta = .975, max\_treedepth = 20),

seed = 190831)

print(read5)

## Warning: The model has not converged (some Rhats are > 1.1). Do not analyse the results!

## We recommend running more iterations and/or setting stronger priors.

## Warning: There were 150 divergent transitions after warmup. Increasing adapt\_delta above 0.975 may help.

## See <http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup>

## Family: gaussian

## Links: mu = identity; sigma = identity

## Formula: read ~ 1 + occasion + homecog + (1 + occasion + homecog | id)

## Data: curran\_dat (Number of observations: 1325)

## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;

## total post-warmup samples = 4000

##

## Group-Level Effects:

## ~id (Number of levels: 405)

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS

## sd(Intercept) 0.72 0.11 0.50 0.99 1.02 456

## sd(occasion) 0.27 0.03 0.22 0.32 1.00 1010

## sd(homecog) 0.04 0.02 0.00 0.09 1.10 39

## cor(Intercept,occasion) 0.28 0.20 -0.14 0.70 1.02 178

## cor(Intercept,homecog) -0.07 0.39 -0.73 0.73 1.05 70

## cor(occasion,homecog) -0.02 0.38 -0.72 0.73 1.01 247

## Tail\_ESS

## sd(Intercept) 165

## sd(occasion) 1608

## sd(homecog) 54

## cor(Intercept,occasion) 319

## cor(Intercept,homecog) 369

## cor(occasion,homecog) 418

##

## Population-Level Effects:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## Intercept 2.15 0.16 1.84 2.46 1.00 2211 2188

## occasion 1.12 0.02 1.07 1.16 1.00 3363 3237

## homecog 0.06 0.02 0.03 0.10 1.00 2290 2399

##

## Family Specific Parameters:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sigma 0.59 0.02 0.56 0.63 1.00 1120 2369

##

## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample

## is a crude measure of effective sample size, and Rhat is the potential

## scale reduction factor on split chains (at convergence, Rhat = 1).

Running this model, we get a few warnings about large Rhats and divergent transitions. We would be better off running a few more iterations (maybe 4000) if we wanted more confidence in the results. Nevertheless, we see a pretty weak effect of cognitive stimulation. The credible interval for the population effect of cognitive stimulation doesn’t include zero, but the evidence is weak that it’s doing anything at all in this model. There’s some indication that the intercepts vary by cognitive stimulation, such that at higher stimulation children have a higher initial reading ability. But there’s no evidence that the slopes vary by cognitive stimulation.

curran\_dat %>%

bind\_cols(as\_tibble(fitted(read5))) %>%

mutate(homecog\_grp = factor(ntile(homecog, 3))) %>%

ggplot(aes(x = occasion, y = Estimate, group = id, color = homecog\_grp)) +

geom\_line(size = .75, alpha = .50) +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Cognitive Home Environment") +

facet\_wrap(~homecog\_grp, labeller = as\_labeller(c("1" = "Low",

"2" = "Medium",

"3" = "High"))) +

guides(color = FALSE) +

theme\_minimal(base\_size = 16) +

theme(plot.title = element\_text(hjust = .5, size = 14))

**Model 6: Dealing with Missing Data**

So far we’ve been dealing with complete data – where each assessment has no missing values. In doing so, we are effectively claiming that the data are missing completely at random (MCAR). This may not be a tenable assumption. Missingness on one variable may be correlated with another variable (Missing at Random; MAR) or, even worse, correlated with the variable itself (Missing Not At Random; MNAR). Without diving into the theoretical aspects of missing data. let’s end by running Bayesian imputation.

curran\_dat\_missing <- read\_sav("CurranLong.sav") %>%

select(id, occasion, read, homecog)

In *brms* we indicate missingness with *mi()*. Here we’re rerunning Model 5, but we’re also imputing missingness on reading ability and cognitive stimulation. We actually have two model formulae – one for each variable in the model with missing data. I’ll also note quickly that we have hidden missingness in this data as missing assessment points do not show up as rows in the dataset. For the purpose of this demonstration, it won’t be a problem, but in the real world, we’d want to impute these as well.

read6 <- brm(data = curran\_dat\_missing,

family = gaussian,

bf(read | mi() ~ 1 + occasion + mi(homecog) + (1 + occasion | id)) +

bf(homecog | mi() ~ 1) +

set\_rescor(FALSE),

prior = c(prior(normal(0, 5), class = Intercept),

prior(normal(0, 1), class = b),

prior(lkj\_corr\_cholesky(1.5), class = cor)),

iter = 2000, warmup = 1000, chains = 4, cores = 4,

control = list(adapt\_delta = .975, max\_treedepth = 20),

seed = 190831)

print(read6)

## Family: MV(gaussian, gaussian)

## Links: mu = identity; sigma = identity

## mu = identity; sigma = identity

## Formula: read | mi() ~ 1 + occasion + mi(homecog) + (1 + occasion | id)

## homecog | mi() ~ 1

## Data: curran\_dat\_missing (Number of observations: 1393)

## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;

## total post-warmup samples = 4000

##

## Group-Level Effects:

## ~id (Number of levels: 405)

## Estimate Est.Error l-95% CI u-95% CI

## sd(read\_Intercept) 0.75 0.04 0.68 0.83

## sd(read\_occasion) 0.27 0.02 0.22 0.32

## cor(read\_Intercept,read\_occasion) 0.25 0.12 0.03 0.51

## Rhat Bulk\_ESS Tail\_ESS

## sd(read\_Intercept) 1.00 1360 2987

## sd(read\_occasion) 1.01 770 1589

## cor(read\_Intercept,read\_occasion) 1.00 960 1492

##

## Population-Level Effects:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS

## read\_Intercept 2.15 0.16 1.84 2.47 1.00 2028

## homecog\_Intercept 8.91 0.07 8.77 9.05 1.00 9040

## read\_occasion 1.12 0.02 1.08 1.16 1.00 4599

## read\_mihomecog 0.06 0.02 0.03 0.09 1.00 2066

## Tail\_ESS

## read\_Intercept 2534

## homecog\_Intercept 2687

## read\_occasion 2956

## read\_mihomecog 2596

##

## Family Specific Parameters:

## Estimate Est.Error l-95% CI u-95% CI Rhat Bulk\_ESS Tail\_ESS

## sigma\_read 0.59 0.02 0.56 0.62 1.00 943 2256

## sigma\_homecog 2.57 0.05 2.47 2.66 1.00 12854 2668

##

## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample

## is a crude measure of effective sample size, and Rhat is the potential

## scale reduction factor on split chains (at convergence, Rhat = 1).

Output is all well and good, but to really see what’s going on we’ll return to our sampling technique. Here are the fitted values on 3 individuals from Model 5 (non-imputed) and the same 3 from Model 6 (imputed).

p1 <- curran\_dat %>%

bind\_cols(as\_tibble(fitted(read5))) %>%

filter(id %in% c(1048, 1050, 1498)) %>%

ggplot() +

geom\_point(aes(x = occasion, y = read), size = 5, alpha = .75, color = "dodgerblue2") +

geom\_point(aes(x = occasion, y = Estimate), shape = 1, size = 5, stroke = 1.5) +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Unimputed Data",

subtitle = "Blue points are observed values. Black circles are fitted values.") +

scale\_x\_continuous(expand = c(.1, .1), breaks = 0:3) +

scale\_y\_continuous(limits = c(2, 9)) +

facet\_wrap(~id, nrow = 1) +

theme\_bw(base\_size = 12) +

theme(plot.title = element\_text(hjust = .5))

p2 <- curran\_dat\_missing %>%

bind\_cols(as\_tibble(fitted(read6))) %>%

filter(id %in% c(1048, 1050, 1498)) %>%

ggplot() +

geom\_point(aes(x = occasion, y = read), size = 5, alpha = .75, color = "dodgerblue2") +

geom\_point(aes(x = occasion, y = Estimate.read), shape = 1, size = 5, stroke = 1.5) +

labs(x = "Assessment Period",

y = "Reading Ability",

title = "Bayesian Imputation",

subtitle = "Blue points are observed values. Black circles are fitted values.") +

scale\_x\_continuous(expand = c(.1, .1), breaks = 0:3) +

scale\_y\_continuous(limits = c(2, 9)) +

facet\_wrap(~id, nrow = 1) +

theme\_bw(base\_size = 12) +

theme(plot.title = element\_text(hjust = .5))

library(ggpubr)

ggarrange(p1, p2)

So we get reasonable estimates for the missing values. Our work here is done.

**Final Words**

This topic is so vast that I couldn’t possibly cover everything there is to say about Bayesian Linear Mixed Models. We only scratched the surface on interpreting Bayesian posteriors. For example, *brms* offers formal ways of comparing models and I highly recommend looking into these.