Model Specification

Tristan Mahr

This document describes the model specification and the code used to fit the model.

Data overview

Let's take note of the data.

Here are the first 10 rows of the data.

```
library(tidyverse)
data
#> # A tibble: 513 x 4
#>    sid     age slpg    intel
#>    <chr> <dbl> <fct>   <dbl>
#>  1 c65      53 NSMI    0.890
#>  2 c65      60 NSMI    0.892
#>  3 c65      65 NSMI    0.951
#>  4 c65      70 NSMI    0.975
#>  5 c65      76 NSMI    0.977
#>  6 c65      81 NSMI    0.974
#>  7 c65      87 NSMI    0.990
#>  8 c65      96 NSMI    0.988
#>  9 c45      45 SMI-LCT 0.417
#> 10 c45      51 SMI-LCT 0.706
#> # ... with 503 more rows
```

Where:

- `sid` uniquely identifies each child
- age is a child's age in months
- `slpg` is the child's speech-language profile group
- `intel` is the child's intelligibility measurement

Children are nested in speech-language profile groups and visits are nested in children.

```
data %>%
  group_by(slpg, sid) %>%
  summarise(
    n_visits_by_child = n()
  )
#> # A tibble: 65 x 3
#> # Groups:   slpg [3]
#>    slpg  sid   n_visits_by_child
#>    <fct> <chr>             <int>
#>  1 NSMI  c12                  10
#>  2 NSMI  c13                  10
#>  3 NSMI  c15                   7
#>  4 NSMI  c23                   4
#>  5 NSMI  c26                  10
#>   6 NSMI  c28                  11
#>   7 NSMI  c32                  10
#>   8 NSMI  c37                  11
#>   9 NSMI  c4                    8
#> 10 NSMI  c40                    8
#> # ... with 55 more rows

data %>%
  group_by(slpg) %>%
  summarise(
    n_visits_in_group = n(),
    n_children_in_group = n_distinct(sid)
  )
#> # A tibble: 3 x 3
#>   slpg    n_visits_in_group n_children_in_group
#>   <fct>               <int>               <int>
#> 1 NSMI                  191                  22
#> 2 SMI-LCT               248                  31
#> 3 SMI-LCI                74                  12
```

Intelligibility is a proportion between 0 and 1. Age ranges from 24 months to 96 months.

```
summary(data$intel, digits = 2)
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>  0.0058  0.3000  0.6000  0.5500  0.8300  0.9900

summary(data$age, digits = 3)
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>    24.0    51.0    64.0    64.3    78.0    96.0
```

Building the BRMS model

To fit our model, we need to specify a model family, the model formula and parameters, and the priors for the parameters.

brms has a rich formula-based language for building up regression models. It resembles the usual syntax for making linear mixed models in R with lme4, but it has elaborates on this syntax in a number of ways.

The `bf()` function (short for *brms formula*) is used to build up model specifications. An ordinary linear model regressing some y onto x is set up as follows.

```
library(brms)

bf(
  y ~ x,
  family = gaussian()
)
```

Response family

Because the outcome is a proportion between 0 and 1, we use a beta regression model which is parameterized using with mean parameter and a precision parameter (phi). The precision is strictly positive so it is modeled with a log-link function.

We fit a ‘submodel’ for both of these parameters: A nonlinear model for the mean and linear model for the precision. Here is how the model formula looks at this point, using placeholders for the formulas.

```
bf(
  intel ~ `some nonlinear formula for the mean`,
  phi ~ `some linear formula for the precision`,
  nl = TRUE,
  family = Beta(link = "identity", link_phi = "log")
)
```

`nl = TRUE` signals that there is a nonlinear model inside of the formula.

Nonlinear formula for the mean

Let’s start with overall model of the mean. We are fitting a logistic curve. These usually have the form:

```
f(y) = asymptote / (1 + exp((mid - x) * scale))
```

where `asymptote`, `mid` and `scale` are parameters that estimated. The model assumes that children start at 0 intelligibility, show accelerating and then decelerating growth, and eventually plateau at some mature level of performance (`asymptote`). The point where growth is the steepest is the midpoint (`mid`), and the `scale` feature controls how steep the growth curve is.

Later on, each of these features will get a linear model. Thus, our model formula expands to:

```
bf(
  intel ~ asymptote / (1 + exp((mid - age) * scale)),
  asymptote ~ `some linear formula`,
  mid ~ `some linear formula`,
  scale ~ `some linear formula`,
  phi ~ `some linear formula for the precision`,
  nl = TRUE,
  family = Beta(link = "identity", link_phi = "log")
)
```

We make two alterations to the logistic function. First, we exponentiate the `scale` term, so the slope is always positive. This operation rules out degenerate growth curves with negative slopes.

Second, we apply the inverse-logit function to the asymptote. Intelligibility is on the proportion scale, and so the `asymptote` needs to be on the proportion scale too. But we are later going to incorporate group and child variation in the asymptote feature by means of a linear model, so we need to constrain the asymptote to the proportion scale. We handle this by estimating the asymptote on the real-valued logit (log-odds) scale and converting it into to a proportion using the inverse logit function.

The updated formula is as follows:

```
inv_logit <- function(x) 1 / (1 + exp(-x))

bf(
  intel ~ inv_logit(asymlogit) / (1 + exp((mid - age) * exp(scale))),
  asymlogit ~ `some linear formula`,
  mid ~ `some linear formula`,
  scale ~ `some linear formula`,
  phi ~ `some linear formula for the precision`,
  nl = TRUE,
  family = Beta(link = "identity", link_phi = "log")
)
```

Linear formulas for the curve parameters

Now that we have the nonlinear model for the mean sketched out, we have to specify the linear model for each of the curve parameters. The same linear model is used for each one:

```
asymlogit ~ 1 + slpg + (0 + slpg | ID | sid)
mid       ~ 1 + slpg + (0 + slpg | ID | sid)
scale     ~ 1 + slpg + (0 + slpg | ID | sid)
```

Let's work through the parts of the formula step by step.

The first two terms are the population average in each group:

- `1` - fit an intercept (the average of the NSMI group)
- `+ slpg` - fit group (`slpg`) differences from intercept (NSMI vs SMI-LCT and NSMI vs SMI-LCT)

The remaining terms are the population variation in each group:

- `+ ( ... | sid)` - include random effects. Observations are nested within the `sid` variable.
- `+ ( ... | ID | ...)` - include correlation between random effects between formulas. The `ID` works as an identifier saying which formulas should have correlations between them. In this case, all of them are correlated.
- `(0 + slpg | ...)` - estimate a separate random intercept variance term for each group. `0` means suppress the intercept.

This model estimates the correlation among all random effect terms. For example, in the NSMI group, the model estimates the correlations among the midpoints, asymptotes, and scale features, and it estimates analogous correlations in the SMI-LCT and in the SMI-LCI groups. It *also* estimates the correlations among the midpoints, asymptotes, and scale features between groups; for example, the correlation between the NSMI asymptotes and SMI-LCT midpoints is estimated. Removing these cross-group correlations would provide a more parsimonious model, but the model without the correlations does not converge (due to divergent iterations). Therefore, we allow the cross-group correlations.

```
inv_logit <- function(x) 1 / (1 + exp(-x))

bf(
  intel ~ inv_logit(asymlogit) / (1 + exp((mid - age) * exp(scale))),
  asymlogit ~ 1 + slpg + (0 + slpg | ID | sid),
  mid ~ 1 + slpg + (0 + slpg | ID | sid),
  scale ~ 1 + slpg + (0 + slpg | ID | sid),
  phi ~ `some linear formula for the precision`,
  nl = TRUE,
  family = Beta(link = "identity", link_phi = "log")
)
```

Linear formula for the precision

We allow the precision to change linearly with age and allow the average precision to change by group:

```
phi ~ 1 + age + slpg
```

The full formula

Here is the full model formula.

```
inv_logit <- function(x) 1 / (1 + exp(-x))

full_formula <- bf(
  intel ~ inv_logit(asymlogit) / (1 + exp((mid - age) * exp(scale))),
  asymlogit ~ 1 + slpg + (0 + slpg | ID | sid),
  mid       ~ 1 + slpg + (0 + slpg | ID | sid),
  scale     ~ 1 + slpg + (0 + slpg | ID | sid),
  phi ~ 1 + age + slpg,
  nl = TRUE,
  family = Beta(link = "identity", link_phi = "log")
)
```

Model priors

We provide three sets of priors: Priors of the population averages in each group (fixed effects), priors for the population variation (random effects), and priors for the precision.

In the population average priors, we specify by prior distributions for the NSMI group (`coef = "Intercept"`) and for the group differences (`class = "b"`, *b* as in a *beta* in a regression equation.)

```
prior_fixef <- c(
  prior(normal(1.25, .5), nlpar = "asymlogit", coef = "Intercept"),
  prior(normal(-.5, .5), nlpar = "asymlogit", class = "b"),
  prior(normal(50, 6), nlpar = "mid", coef = "Intercept"),
  prior(normal(0, 12), nlpar = "mid", class = "b"),
  prior(normal(-2, 1), nlpar = "scale", coef = "Intercept"),
  prior(normal(0, .5), nlpar = "scale", class = "b")
)
```

Let's work through one example about what these priors are saying.

For the midpoint intercept parameter (`nlpar = "mid"`), this prior information says the following: Before seeing any data, we think that a plausible set of values for the average age of steepest growth in the NSMI group is a normal distribution with a mean of 50 months and standard deviation of 6 months, and as a result, the 99% most plausible NSMI-average midpoints will fall between 34.5 and 66.5 months. For the group differences, we tell the model that group differences on the order of 12 months are plausible. This prior is less informative than the one for the reference group: It is centered at 0, meaning that both earlier-than-NSMI and later-than-NSMI midpoints are plausible group averages. Group differences up to 24 months are plausible.

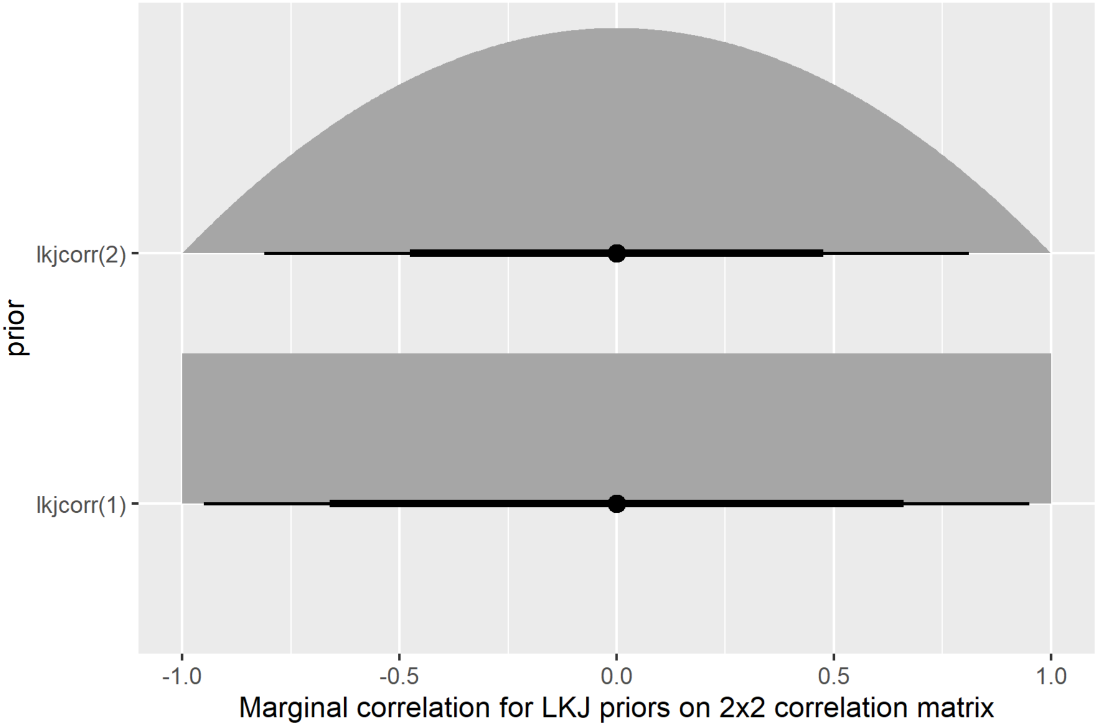
These priors are computational devices: They supply some information not available in the data that will help the model sample the space of parameter values by ruling out a priori implausible parameter values.

The priors for the population variation are given in terms of standard deviations and an LKJ prior for the correlation matrix.

```
prior_ranef <- c(
  prior(normal(0, 1.25), class = "sd", nlpar = "asymlogit"),
  prior(normal(10, 2.5), class = "sd", nlpar = "mid"),
  prior(normal(0, .5), class = "sd", nlpar = "scale"),
  prior(lkj_corr_cholesky(2), class = "L")
)
```

Here, for the midpoints, our prior says the between-child variability (as a standard deviation) between 5 and 15 months is plausible in each group.

The LKJ prior specifies what correlations are plausible. For example, in a 2x2 matrix, LKJ(1) puts a uniform distribution over the correlations whereas LKJ(2) rules out correlations of -1 or 1:



We use the LKJ(2) prior because it provides a weakly informative prior: Enough information to rule out degenerate correlations.

Finally, for the precision parameter, we use weakly informative priors:

```
prior_phi <- c(
  prior(normal(2, 1), dpar = "phi", class = "Intercept"),
  prior(normal(0, 1), dpar = "phi", class = "b")
)
```

For prior selection, we used combined subject matter knowledge and evaluation of the prior predictive distribution. That is, for things like the midpoint feature, we have a good sense of when children's speech develops, so we selected priors that encompassed that age range. For more computational features, in particular the `scale` parameter, we had the model simulate fake data. If the fake data were implausible, like implying 100% intelligibility at 18 months or changing from 20% intelligible to 80% intelligible in a month, we tuned the priors so that these the fake data became more plausible.

Full model fitting code

For completeness, here is the exact code used to fit the model in the manuscript, include sampling settings. (There are some slight variables name changes and syntax changes compared to the above exposition).

```r
fit_model <- function(data, chains = 4, cores = 4, sample_prior = "no") {
  inv_logit <- function(x) 1 / (1 + exp(-x))

  formula_beta <- bf(
    multiword_intel2 ~
      inv_logit(asymlogit) * inv(1 + exp((mid - age) * exp(scale))),
    asymlogit ~ 1 + slpg + (0 + slpg | ID | sid),
    mid ~ 1 + slpg + (0 + slpg | ID | sid),
    scale ~ 1 + slpg + (0 + slpg | ID | sid),
    phi ~ 1 + age + slpg,
    nl = TRUE
  )

  prior_fixef <- c(
    prior(normal(1.25, .5), nlpar = "asymlogit", coef = "Intercept"),
    prior(normal(-.5, .5), nlpar = "asymlogit", class = "b"),
    prior(normal(50, 6), nlpar = "mid", coef = "Intercept"),
    prior(normal(0, 12), nlpar = "mid", class = "b"),
    prior(normal(-2, 1), nlpar = "scale", coef = "Intercept"),
    prior(normal(0, .5), nlpar = "scale", class = "b")
  )

  prior_phi <- c(
    prior(normal(2, 1), dpar = "phi", class = "Intercept"),
    prior(normal(0, 1), dpar = "phi", class = "b")
  )

  prior_ranef <- c(
    prior(normal(0, 1.25), class = "sd", nlpar = "asymlogit"),
    prior(normal(10, 2.5), class = "sd", nlpar = "mid"),
    prior(normal(0, .5), class = "sd", nlpar = "scale"),
    prior(lkj_corr_cholesky(2), class = "L")
  )

  fit_beta <- brm(
    formula_beta,
    data = data,
    prior = c(prior_fixef, prior_phi, prior_ranef),
    family = Beta(link = identity, link_phi = "log"),
    iter = 2000,
    chains = chains,
    refresh = 25,
    sample_prior = sample_prior,
    cores = cores,
    control = list(adapt_delta = 0.92, max_treedepth = 15)
  )

  fit_beta
}

fit <- fit_model(data, cores = 4, chains = 4)
```

Model summary

Here is model output (posterior median, SD, 95% intervals):

```
#>  Family: beta 
#>   Links: mu = identity; phi = log 
#> Formula: multiword_intel2 ~ inv_logit(asymlogit) * inv(1 + exp((mid - age) * exp(scale))) 
#>          asymlogit ~ 1 + slpg + (0 + slpg | ID | sid)
#>          mid ~ 1 + slpg + (0 + slpg | ID | sid)
#>          scale ~ 1 + slpg + (0 + slpg | ID | sid)
#>          phi ~ 1 + age + slpg
#>    Data: data (Number of observations: 513) 
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>          total post-warmup samples = 4000
#> 
#> Group-Level Effects: 
#> ~sid (Number of levels: 65) 
#>                                                  Estimate Est.Error l-95% CI u-95% CI
#> sd(asymlogit_slpgNSMI)                               0.71      0.25     0.26     1.29
#> sd(asymlogit_slpgSMIMLCT)                            2.44      0.41     1.76     3.37
#> sd(asymlogit_slpgSMIMLCI)                            2.23      0.52     1.33     3.35
#> sd(mid_slpgNSMI)                                     6.80      1.45     4.30     9.96
#> sd(mid_slpgSMIMLCT)                                 10.63      1.59     7.68    13.88
#> sd(mid_slpgSMIMLCI)                                 11.01      2.60     5.86    16.00
#> sd(scale_slpgNSMI)                                   0.19      0.12     0.01     0.43
#> sd(scale_slpgSMIMLCT)                                0.23      0.12     0.02     0.47
#> sd(scale_slpgSMIMLCI)                                0.46      0.31     0.03     1.15
#> cor(asymlogit_slpgNSMI,asymlogit_slpgSMIMLCT)       -0.00      0.29    -0.54     0.54
#> cor(asymlogit_slpgNSMI,asymlogit_slpgSMIMLCI)       -0.00      0.29    -0.56     0.54
#> cor(asymlogit_slpgSMIMLCT,asymlogit_slpgSMIMLCI)    -0.01      0.29    -0.57     0.56
#> cor(asymlogit_slpgNSMI,mid_slpgNSMI)                -0.06      0.24    -0.51     0.42
#> cor(asymlogit_slpgSMIMLCT,mid_slpgNSMI)             -0.00      0.29    -0.57     0.55
#> cor(asymlogit_slpgSMIMLCI,mid_slpgNSMI)             -0.01      0.29    -0.55     0.54
#> cor(asymlogit_slpgNSMI,mid_slpgSMIMLCT)              0.01      0.29    -0.55     0.55
#> cor(asymlogit_slpgSMIMLCT,mid_slpgSMIMLCT)           0.05      0.23    -0.42     0.48
#> cor(asymlogit_slpgSMIMLCI,mid_slpgSMIMLCT)          -0.01      0.29    -0.56     0.52
#> cor(mid_slpgNSMI,mid_slpgSMIMLCT)                    0.00      0.29    -0.55     0.57
#> cor(asymlogit_slpgNSMI,mid_slpgSMIMLCI)              0.01      0.28    -0.54     0.55
#> cor(asymlogit_slpgSMIMLCT,mid_slpgSMIMLCI)           0.00      0.29    -0.55     0.55
#> cor(asymlogit_slpgSMIMLCI,mid_slpgSMIMLCI)          -0.06      0.26    -0.57     0.45
#> cor(mid_slpgNSMI,mid_slpgSMIMLCI)                   -0.00      0.29    -0.55     0.57
#> cor(mid_slpgSMIMLCT,mid_slpgSMIMLCI)                 0.00      0.29    -0.54     0.55
#> cor(asymlogit_slpgNSMI,scale_slpgNSMI)               0.14      0.27    -0.41     0.63
#> cor(asymlogit_slpgSMIMLCT,scale_slpgNSMI)           -0.00      0.29    -0.55     0.56
#> cor(asymlogit_slpgSMIMLCI,scale_slpgNSMI)            0.00      0.29    -0.56     0.57
#> cor(mid_slpgNSMI,scale_slpgNSMI)                     0.09      0.27    -0.46     0.57
#> cor(mid_slpgSMIMLCT,scale_slpgNSMI)                  0.01      0.28    -0.55     0.55
#> cor(mid_slpgSMIMLCI,scale_slpgNSMI)                  0.00      0.29    -0.56     0.55
#> cor(asymlogit_slpgNSMI,scale_slpgSMIMLCT)            0.01      0.29    -0.56     0.56
#> cor(asymlogit_slpgSMIMLCT,scale_slpgSMIMLCT)        -0.03      0.27    -0.56     0.50
#> cor(asymlogit_slpgSMIMLCI,scale_slpgSMIMLCT)         0.01      0.28    -0.54     0.56
#> cor(mid_slpgNSMI,scale_slpgSMIMLCT)                 -0.01      0.29    -0.56     0.56
#> cor(mid_slpgSMIMLCT,scale_slpgSMIMLCT)               0.02      0.25    -0.46     0.50
#> cor(mid_slpgSMIMLCI,scale_slpgSMIMLCT)              -0.00      0.29    -0.56     0.56
#> cor(scale_slpgNSMI,scale_slpgSMIMLCT)               -0.00      0.29    -0.55     0.55
#> cor(asymlogit_slpgNSMI,scale_slpgSMIMLCI)           -0.01      0.29    -0.56     0.55
#> cor(asymlogit_slpgSMIMLCT,scale_slpgSMIMLCI)         0.00      0.29    -0.54     0.55
#> cor(asymlogit_slpgSMIMLCI,scale_slpgSMIMLCI)         0.02      0.28    -0.53     0.56
#> cor(mid_slpgNSMI,scale_slpgSMIMLCI)                  0.00      0.29    -0.54     0.55
#> cor(mid_slpgSMIMLCT,scale_slpgSMIMLCI)              -0.01      0.29    -0.56     0.54
#> cor(mid_slpgSMIMLCI,scale_slpgSMIMLCI)              -0.00      0.28    -0.54     0.54
#> cor(scale_slpgNSMI,scale_slpgSMIMLCI)                0.00      0.29    -0.57     0.57
#> cor(scale_slpgSMIMLCT,scale_slpgSMIMLCI)             0.01      0.29    -0.55     0.56
#> 
#> Population-Level Effects: 
#>                       Estimate Est.Error l-95% CI u-95% CI
#> phi_Intercept             2.44      0.31     1.83     3.05
#> asymlogit_Intercept       2.53      0.24     2.05     2.99
#> asymlogit_slpgSMIMLCT    -0.69      0.40    -1.45     0.09
#> asymlogit_slpgSMIMLCI    -1.48      0.51    -2.46    -0.44
#> mid_Intercept            39.06      1.70    35.75    42.49
#> mid_slpgSMIMLCT          15.36      2.97     9.64    21.25
#> mid_slpgSMIMLCI          12.76      9.56    -4.94    32.77
#> scale_Intercept          -2.38      0.10    -2.58    -2.19
#> scale_slpgSMIMLCT        -0.23      0.13    -0.47     0.04
#> scale_slpgSMIMLCI        -0.71      0.35    -1.38     0.02
#> phi_age                   0.02      0.00     0.01     0.03
#> phi_slpgSMIMLCT          -0.35      0.16    -0.67    -0.04
#> phi_slpgSMIMLCI          -0.72      0.22    -1.17    -0.30
#> 
#> Samples were drawn 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).
```

Here is the diagnostic output. Rhat should be < 1.05. Effective sample size should be > 400.

```
#>  Family: beta 
#>   Links: mu = identity; phi = log 
#> Formula: multiword_intel2 ~ inv_logit(asymlogit) * inv(1 + exp((mid - age) * exp(s
cale))) 
#>          asymlogit ~ 1 + slpg + (0 + slpg | ID | sid)
#>          mid ~ 1 + slpg + (0 + slpg | ID | sid)
#>          scale ~ 1 + slpg + (0 + slpg | ID | sid)
#>          phi ~ 1 + age + slpg
#>    Data: data (Number of observations: 513) 
#> Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
#>          total post-warmup samples = 4000
#> 
#> Group-Level Effects: 
#> ~sid (Number of levels: 65) 
#>                                                  Rhat Bulk_ESS Tail_ESS
#> sd(asymlogit_slpgNSMI)                           1.01     1065     1133
#> sd(asymlogit_slpgSMIMLCT)                        1.00     2382     2888
#> sd(asymlogit_slpgSMIMLCI)                        1.00     2361     2354
#> sd(mid_slpgNSMI)                                 1.00     2821     3467
#> sd(mid_slpgSMIMLCT)                              1.00     3461     2928
#> sd(mid_slpgSMIMLCI)                              1.00     3563     2215
#> sd(scale_slpgNSMI)                               1.01      819     1793
#> sd(scale_slpgSMIMLCT)                            1.00      711     1421
#> sd(scale_slpgSMIMLCI)                            1.00     2471     2559
#> cor(asymlogit_slpgNSMI,asymlogit_slpgSMIMLCT)    1.01      531     1256
#> cor(asymlogit_slpgNSMI,asymlogit_slpgSMIMLCI)    1.00     1289     2426
#> cor(asymlogit_slpgSMIMLCT,asymlogit_slpgSMIMLCI) 1.00     1483     2456
#> cor(asymlogit_slpgNSMI,mid_slpgNSMI)             1.00     2291     2801
#> cor(asymlogit_slpgSMIMLCT,mid_slpgNSMI)          1.00     1422     2305
#> cor(asymlogit_slpgSMIMLCI,mid_slpgNSMI)          1.00     1205     2459
#> cor(asymlogit_slpgNSMI,mid_slpgSMIMLCT)          1.00      807     1685
#> cor(asymlogit_slpgSMIMLCT,mid_slpgSMIMLCT)       1.00     2315     2767
#> cor(asymlogit_slpgSMIMLCI,mid_slpgSMIMLCT)       1.00      990     1565
#> cor(mid_slpgNSMI,mid_slpgSMIMLCT)                1.01     1013     2102
#> cor(asymlogit_slpgNSMI,mid_slpgSMIMLCI)          1.00     5526     3134
#> cor(asymlogit_slpgSMIMLCT,mid_slpgSMIMLCI)       1.00     5714     3001
#> cor(asymlogit_slpgSMIMLCI,mid_slpgSMIMLCI)       1.00     6011     3265
#> cor(mid_slpgNSMI,mid_slpgSMIMLCI)                1.00     4993     2953
#> cor(mid_slpgSMIMLCT,mid_slpgSMIMLCI)             1.00     5288     3442
#> cor(asymlogit_slpgNSMI,scale_slpgNSMI)           1.00     5699     3479
#> cor(asymlogit_slpgSMIMLCT,scale_slpgNSMI)        1.00     3546     3424
#> cor(asymlogit_slpgSMIMLCI,scale_slpgNSMI)        1.00     3386     3296
#> cor(mid_slpgNSMI,scale_slpgNSMI)                 1.00     4435     3313
#> cor(mid_slpgSMIMLCT,scale_slpgNSMI)              1.00     3615     3411
#> cor(mid_slpgSMIMLCI,scale_slpgNSMI)              1.00     2886     3688
#> cor(asymlogit_slpgNSMI,scale_slpgSMIMLCT)        1.00     2975     3159
#> cor(asymlogit_slpgSMIMLCT,scale_slpgSMIMLCT)     1.00     4081     3102
#> cor(asymlogit_slpgSMIMLCI,scale_slpgSMIMLCT)     1.00     2787     2935
#> cor(mid_slpgNSMI,scale_slpgSMIMLCT)              1.00     3010     3387
#> cor(mid_slpgSMIMLCT,scale_slpgSMIMLCT)           1.00     3610     3096
#> cor(mid_slpgSMIMLCI,scale_slpgSMIMLCT)           1.00     2942     3543
#> cor(scale_slpgNSMI,scale_slpgSMIMLCT)            1.00     3011     3519
#> cor(asymlogit_slpgNSMI,scale_slpgSMIMLCI)        1.00     6844     3092
#> cor(asymlogit_slpgSMIMLCT,scale_slpgSMIMLCI)     1.00     6062     3319
#> cor(asymlogit_slpgSMIMLCI,scale_slpgSMIMLCI)     1.00     7379     2966
#> cor(mid_slpgNSMI,scale_slpgSMIMLCI)              1.00     5616     3328
#> cor(mid_slpgSMIMLCT,scale_slpgSMIMLCI)           1.00     4729     3314
#> cor(mid_slpgSMIMLCI,scale_slpgSMIMLCI)           1.00     3467     3394
#> cor(scale_slpgNSMI,scale_slpgSMIMLCI)            1.00     3156     3755
#> cor(scale_slpgSMIMLCT,scale_slpgSMIMLCI)         1.00     3877     3563
#> 
#> Population-Level Effects: 
#>                       Rhat Bulk_ESS Tail_ESS
#> phi_Intercept         1.00     4462     3025
#> asymlogit_Intercept   1.00     1064     1577
#> asymlogit_slpgSMIMLCT 1.00     3681     2909
#> asymlogit_slpgSMIMLCI 1.00     4619     3164
#> mid_Intercept         1.00     2555     2794
#> mid_slpgSMIMLCT       1.00     2745     2814
#> mid_slpgSMIMLCI       1.00     1519     1828
#> scale_Intercept       1.00     2419     2549
#> scale_slpgSMIMLCT     1.00     2893     2855
#> scale_slpgSMIMLCI     1.00     2979     2831
#> phi_age               1.00     4518     3435
#> phi_slpgSMIMLCT       1.00     2845     3405
#> phi_slpgSMIMLCI       1.00     4498     3681
#> 
#> Samples were drawn 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).
```

Here is Stan's internal diagonistic check.

```
rstan::check_hmc_diagnostics(fit$fit)
#> 
#> Divergences:
#> 0 of 4000 iterations ended with a divergence.
#> 
#> Tree depth:
#> 0 of 4000 iterations saturated the maximum tree depth of 15.
#> 
#> Energy:
#> E-BFMI indicated no pathological behavior.
```