Methods 3, Week 5:

Multilevel simulation

Exercice with simulated data

Let's imagine we have n autistic and n neurotypical children. We are simulating their average utterance length (Mean Length of Utterance or MLU) in terms of words, starting at Visit 1 and all the way to Visit 6.

- Assumptions
  - Population means are exact values
  - Change by visit is linear (same between each visit)

Remember the usual bayesian workflow: - define the formula - define the prior - prior predictive checks - fit the model - model quality checks: traceplots, divergences, rhat, effective samples - model quality checks: posterior predictive checks, prior-posterior update checks - model comparison

```r
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_knit$set(root.dir = rprojroot::find_rstudio_root_file())

if(!'cmdstanr' %in% installed.packages()){
  remotes::install_github("stan-dev/cmdstanr")
  cmdstanr::install_cmdstan()}


pacman::p_load(
  tidyverse,
  brms,
  patchwork
)
```

```r
# Define the sample size per group
n <- 50
visits <- 6

# Define the dataset (1000 td, 1000 asd, 6 visits)
sim_data <- tibble(expand.grid(child_id = seq(n), diagnosis = c("asd", "td"), visit = seq(visits)), int
# Make sure ASD and TD do not share a ID
sim_data$child_id[sim_data$diagnosis == "td"] <-
  sim_data$child_id[sim_data$diagnosis == "td"] + n

sim_data
```

```
## # A tibble: 600 x 5
##    child_id diagnosis visit intercept slope
##       <dbl> <fct>     <int> <lgl>     <lgl>
##  1        1 asd           1 NA        NA
##  2        2 asd           1 NA        NA
##  3        3 asd           1 NA        NA
##  4        4 asd           1 NA        NA
##  5        5 asd           1 NA        NA
##   6         6 asd             1 NA        NA
##   7         7 asd             1 NA        NA
##   8         8 asd             1 NA        NA
##   9         9 asd             1 NA        NA
##  10        10 asd             1 NA        NA
## # i 590 more rows
```

```
# Define the parameters, based on literature and assumptions
params <- list(
  mu_asd = 2,
  sigma_asd = 0.4,
  mu_td = 2,
  sigma_td = 0.3,
  mu_visit_asd = 0.4,
  sigma_visit_asd = 0.3,
  mu_visit_td = 0.6,
  sigma_visit_td = 0.2,
  error = 0.2)
```

```
#define the intercept and slope

for (i in seq(unique(sim_data$child_id))) {
  sim_data$intercept[sim_data$child_id == i &
                     sim_data$diagnosis == "asd"] <-
    rnorm(1, params$mu_asd, params$sigma_asd)
  sim_data$intercept[sim_data$child_id == i &
                     sim_data$diagnosis == "td"] <-
    rnorm(1, params$mu_td, params$sigma_td)
  sim_data$slope[sim_data$child_id == i &
                 sim_data$diagnosis == "asd"] <-
    rnorm(1, params$mu_visit_asd, params$sigma_visit_asd)
  sim_data$slope[sim_data$child_id == i &
                 sim_data$diagnosis == "td"] <-
    rnorm(1, params$mu_visit_td, params$sigma_visit_td)
}

# Calculate mlu per each data point
sim_data <- sim_data %>%
  mutate(
    mlu = intercept +
      (slope * (visit - 1)) + rnorm(1, 0, params$error))
```

Explore Simulated Data

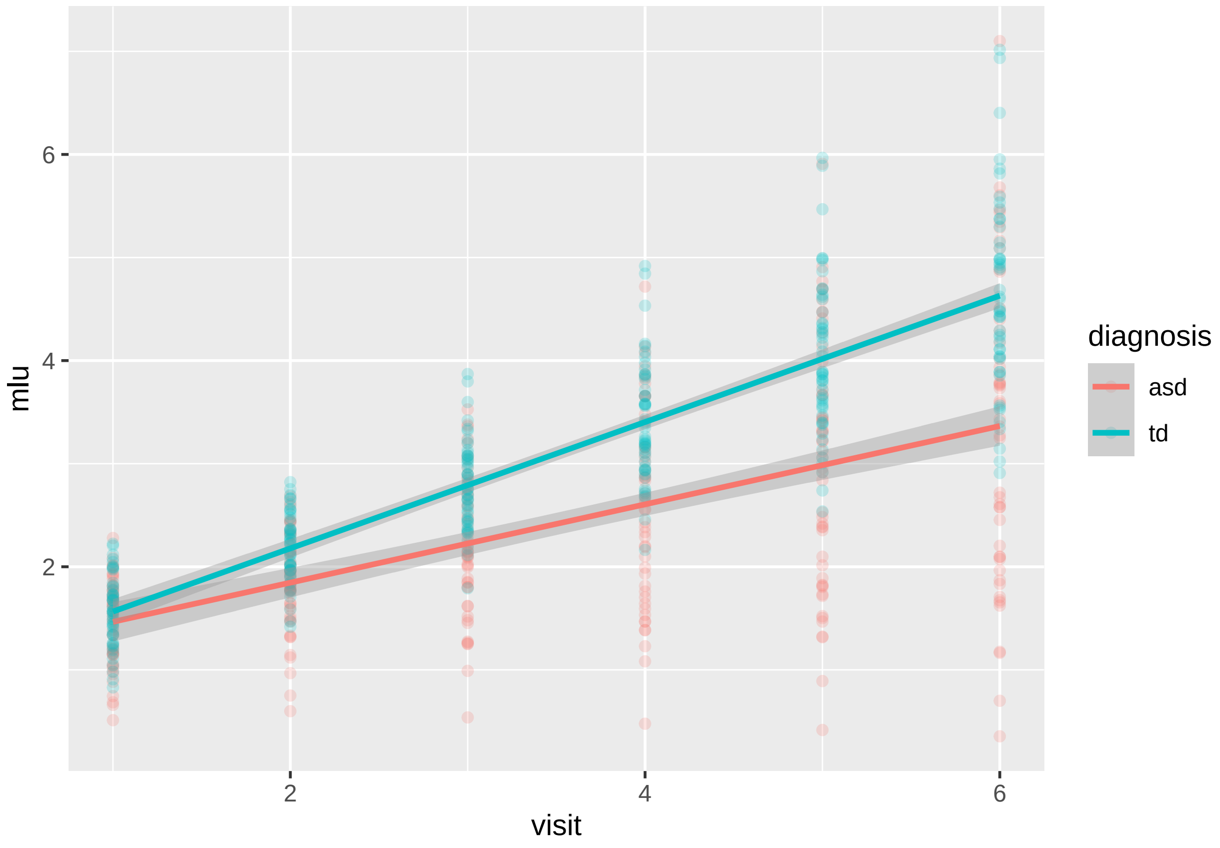
```
# Check the data - looking at the first few rows
sim_data
```

```
## # A tibble: 600 x 6
##    child_id diagnosis visit intercept   slope   mlu
##       <dbl> <fct>     <int>     <dbl>   <dbl> <dbl>
##   1        1 asd             1      1.65  0.186  1.15
##   2        2 asd             1      2.39 -0.0422 1.89
##   3        3 asd             1      1.66  0.161  1.16
##   4        4 asd             1      1.54  0.913  1.04
##   5         5 asd            1      2.24  0.426  1.74
##   6         6 asd            1      2.05  0.377  1.55
##   7         7 asd            1      2.49  0.695  1.99
##   8         8 asd            1      2.51  0.351  2.01
##   9         9 asd            1      2.02  0.448  1.52
##  10        10 asd            1      1.25  0.731  0.745
## # i 590 more rows
```

Plot Simulated Data

```
# Check the data - plotting the data
ggplot(sim_data, aes(visit, mlu, color = diagnosis)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Formulas

```
# The left-hand side defines the outcome (what is predicted)
# The right-hand side defines the predictors
# Constant effects are added first, followed by varying ones
# Varying effects are specified between brackets. Again the value 1 is to indicate the intercept and th
# All correlations between each pair of random effects are estimated
# Use || between two random effects instead of | to prevent this.
```

```
formula1 <-
  brms::bf(mlu ~ visit + (1|visit) + (1 + visit|child_id))

# Intercepts are assumed by default in R.
# Hence, the above formula is equivalent to the following one:
# mlu ~ 1 + visit + (1 + visit |child_id)
# To fit a model without a population-level intercept, replace the 1 in the previous formula by a 0
# Cross level interactions between two predictors can be created by the : sign or by multiplying them i

formula2 <-
  brms::bf(mlu ~ 0 +
             diagnosis*visit +
             (1 + visit | gr(child_id, by = diagnosis)))


formula3 <- brms::bf(mlu ~ 0 +
                  diagnosis +
                  diagnosis:visit +
                  (1 + visit | gr(child_id, by = diagnosis)),
                sigma ~ 0 +
                  diagnosis +
                  (1 | gr(child_id, by = diagnosis)))
```

Priors

```
# Get priors
get_prior(formula1, data = sim_data)
```

```
##                    prior     class      coef    group resp dpar nlpar lb ub
##                   (flat)         b
##                   (flat)         b     visit
##                   lkj(1)       cor
##                   lkj(1)       cor           child_id
##  student_t(3, 2.5, 2.5) Intercept
##    student_t(3, 0, 2.5)        sd                                    0
##    student_t(3, 0, 2.5)        sd           child_id                 0
##    student_t(3, 0, 2.5)        sd Intercept child_id                 0
##    student_t(3, 0, 2.5)        sd     visit child_id                 0
##    student_t(3, 0, 2.5)        sd              visit                 0
##    student_t(3, 0, 2.5)        sd Intercept    visit                 0
##    student_t(3, 0, 2.5)     sigma                                    0
##        source
##       default
##  (vectorized)
##       default
##  (vectorized)
##       default
##       default
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##       default
```

```r
priors1 <- c(
  prior(normal(1, 1), class = Intercept),
  prior(normal(0, .5), class = b),
  prior(normal(0, 1), class = sd),
  prior(normal(0, 1), class = sigma),
  prior(lkj(3), class = cor)
)

get_prior(formula2, data = sim_data)
```

```
##                 prior class              coef    group resp dpar nlpar lb ub
##                (flat)     b
##                (flat)     b      diagnosisasd
##                (flat)     b       diagnosistd
##                (flat)     b diagnosistd:visit
##                (flat)     b             visit
##                lkj(1)   cor
##                lkj(1)   cor                   child_id
##  student_t(3, 0, 2.5)    sd                                             0
##  student_t(3, 0, 2.5)    sd                   child_id                  0
##  student_t(3, 0, 2.5)    sd         Intercept child_id                  0
##  student_t(3, 0, 2.5)    sd             visit child_id                  0
##  student_t(3, 0, 2.5) sigma                                             0
##        source
##       default
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##       default
##  (vectorized)
##       default
##  (vectorized)
##  (vectorized)
##  (vectorized)
##       default
```

```r
priors2 <- c(
  prior(normal(0, .5), class = b),
  prior(normal(1, 1), class = b, coef = diagnosisasd),
  prior(normal(1, 1), class = b, coef = diagnosistd),
  prior(normal(0, 1), class = sd),
  prior(normal(0, 1), class = sigma),
  prior(lkj(3), class = cor)
)

get_prior(formula3, data = sim_data)
```

```
##                 prior class               coef    group resp  dpar nlpar lb ub
##                (flat)     b
##                (flat)     b       diagnosisasd
##                (flat)     b diagnosisasd:visit
##                 (flat)     b        diagnosistd
##                 (flat)     b  diagnosistd:visit
##                 lkj(1)   cor
##                 lkj(1)   cor                     child_id
##   student_t(3, 0, 2.5)    sd                                                   0
##   student_t(3, 0, 2.5)    sd                     child_id                      0
##   student_t(3, 0, 2.5)    sd          Intercept child_id                       0
##   student_t(3, 0, 2.5)    sd              visit child_id                       0
##                 (flat)     b                                   sigma
##                 (flat)     b       diagnosisasd                sigma
##                 (flat)     b        diagnosistd                sigma
##   student_t(3, 0, 2.5)    sd                                   sigma        0
##   student_t(3, 0, 2.5)    sd                     child_id      sigma        0
##   student_t(3, 0, 2.5)    sd          Intercept child_id      sigma        0
##         source
##        default
##   (vectorized)
##   (vectorized)
##   (vectorized)
##   (vectorized)
##        default
##   (vectorized)
##        default
##   (vectorized)
##   (vectorized)
##   (vectorized)
##        default
##   (vectorized)
##   (vectorized)
##        default
##   (vectorized)
##   (vectorized)
```

```
priors3 <- c(
  prior(normal(1, 1), class = b),
  prior(normal(1, 1), class = b, coef = diagnosisasd),
  prior(normal(1, 1), class = b, coef = diagnosistd),
  prior(normal(0, 1), class = sd),
  #prior(normal(0, 1), class = b, dpar = sigma),
  prior(normal(0, .5), class = b, dpar = sigma),
  #prior(normal(0, 1), class = sd, dpar = sigma),
  prior(normal(0, .1), class = sd, dpar = sigma),
  prior(lkj(3), class = cor)
)
```

```
# Generate prior predictive models

model1_prior <- brm(
  formula1,
  sim_data,
  family = gaussian,
  prior = prior1,
  sample_prior = "only",
  file = 'data/bhm_simulation/model1_prior',
  backend = "cmdstanr",
```

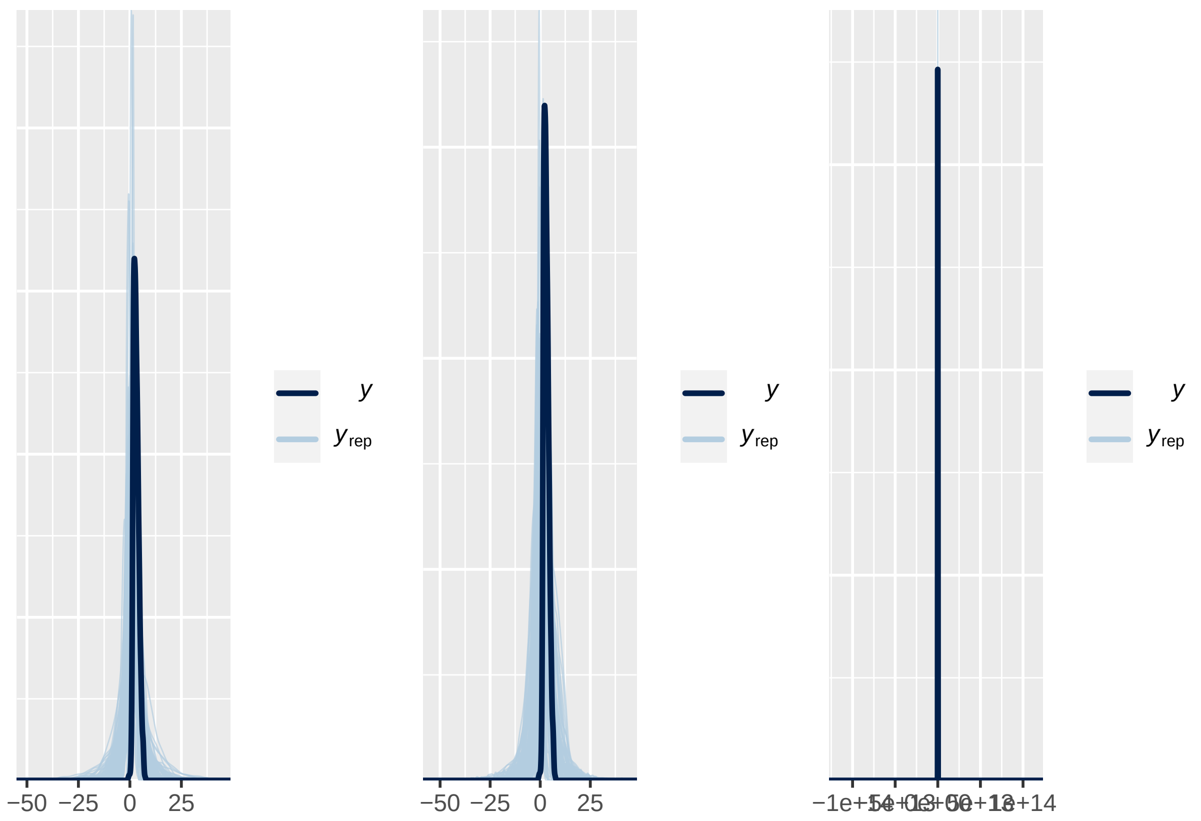
```r
  chains = 2,
  stan_model_args = list(stanc_options = list("O1"))
)

model2_prior <- brm(
  formula2,
  sim_data,
  family = gaussian,
  prior = prior2,
  sample_prior = "only",
  file = 'data/bhm_simulation/model2_prior',
  backend = "cmdstanr",
  chains = 2,
  stan_model_args = list(stanc_options = list("O1"))
)

model3_prior <- brm(
  formula3,
  sim_data,
  family = gaussian,
  prior = prior3,
  sample_prior = "only",
  file = 'data/bhm_simulation/model3_prior',
  backend = "cmdstanr",
  chains = 2,
  stan_model_args = list(stanc_options = list("O1"))
)

# Check prior predictive checks
prior_predictive1 <- pp_check(model1_prior, ndraws = 100)
prior_predictive2 <- pp_check(model2_prior, ndraws = 100)
prior_predictive3 <- pp_check(model3_prior, ndraws = 100)

prior_predictive1 + prior_predictive2 + prior_predictive3
```


−50
−25
0
25
y
y_rep
−50
−25
0
25
y
y_rep
−1e+14
−5e+13
0e+00
5e+13
1e+14
y
y_rep