Methods 3: Portfolio 1

Written by:
Katrine Munkholm, Mikkel Fischer, Thomas Steinthal, Rikke Neubert & Caroline Lemée
Study Group 9

Q1

We'll walk you through the code because we find this the most appropriate way of showing you our simulation process, its goals and what we learned. For the latter, I think we found it interesting to simulate this vast amount of data but we were primarily confirmed in our belief of which priors to include into the 'real' analysis.

Initial processing of input-values for data-simulation
Firstly, we identified the values we would expect to find. This would typically be done through a structured meta-analysis, but luckily we were provided with these values from Riccardo.
We log-transformed initiate means and standard deviations (to be exponentiated later) - this small transformation will normally decrease our data-quality, but it was done to avoid having negative predictions (because a child cannot have a negative MLU).
We also translated the expected slopes and uncertainties of these through experimenting with how much we expected the values to vary; we specified additional values of mu and sigma for both diagnoses. Mu was set to be higher for neurotypicals(.18) than for asd (.09). Sigma was set to be slightly smaller for neurotypicals(.025) than for asd(.035).

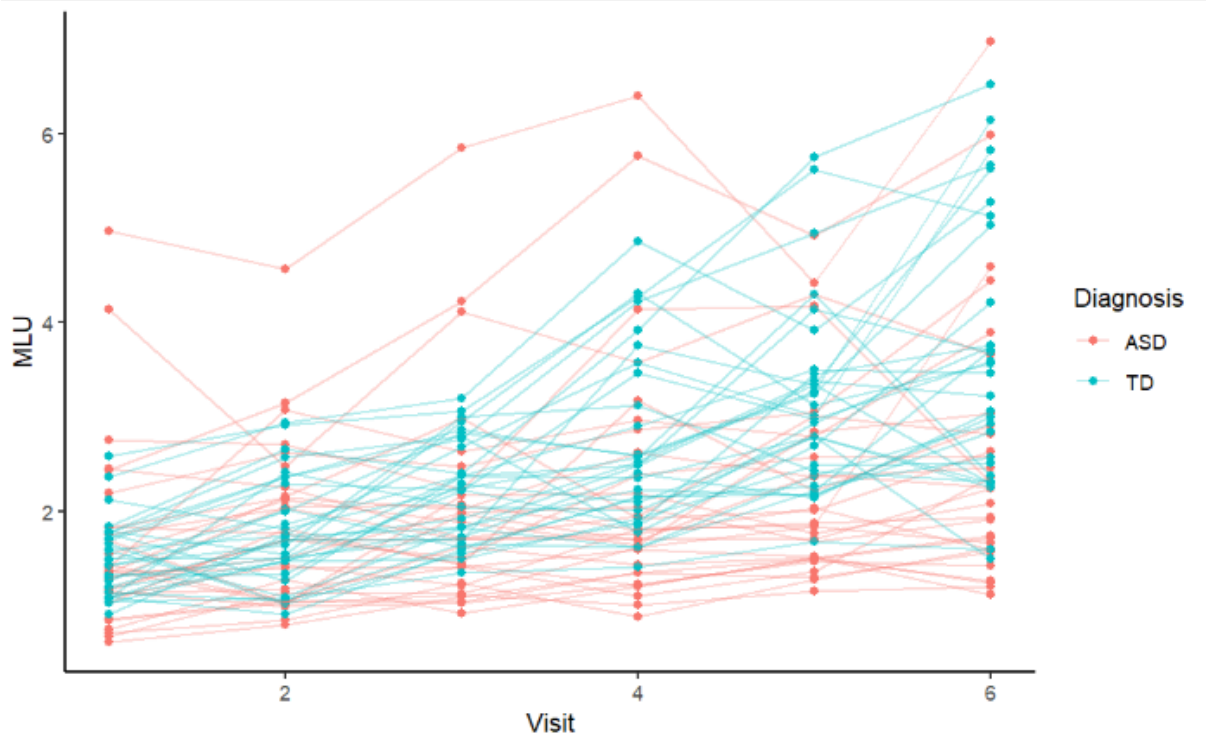
The error was set at .2

Simulating and plotting expected data
Initially, we created an ID-tibble for our expected data-collection setup: 60 participants (30 diagnosed with ASD, 30 being neurotypical) visiting 6 times each.
Then, we ran through the ID-frame, simulating individual intercepts and slopes for each participant based on the values from the meta-analysis and the diagnosis of the participant. These values made it possible for us to predict the actual data-points using rnorm(): The recorded MLU. Each of these data points were exponentiated to account for our earlier log-transformations.
We also wrote a function to make this data-simulation easier to understand, but it yielded slightly different data and we therefore decided to just showcase the function here:

```r
my_function <- function (n,v, li) {
#Initial ID-frame
d<-tibble(ID = rep(1:n, each = v),
          visit = rep(1:v, times = n),
          diagnosis = rep(0:1, each=n*3)
          )

#Simulating data: Individual intercept (with the same mean for all! )
Ind_b0<-c(rnorm(n/2, mean = li[1], sd = li[2]),
          rnorm(n/2, mean = li[1], sd = li[3]))

d<-d %>%
  mutate(Indi_b0 = rep(Ind_b0, each = v))

#Simulating data: Individual slope (SAME Through each visit!!!)
Ind_b1<-c(rnorm(n/2, mean = li[4], sd = li[6]),
          rnorm(n/2, mean = li[5], sd = li[7]))

d<-d %>%
  mutate(Indi_b1 = rep(Ind_b1, each = v))


#Simulating the final data - ASSUMING that the data fit a straight line with minor deviations (e=0.2)
d<-d %>%
  mutate(MLU=rnorm(n*v,mean=Indi_b0+Indi_b1*visit,sd=li[8])
         )

#d<-d %>%
# mutate(MLU=exp(MLU))

d<-d %>%
  mutate(diagnosis=as.factor(diagnosis))

return (d)
}
```

Note that the inputs of this function takes a sample size (n), a number of visits (v) and a list containing all the variables from the metaanalysis (li). This list is indexed to make the code more dense. We will use this function later for more concise analysis

When plotting the data, we generated the following plot. The plot corresponds with the values we were provided with initially (ASD-children vary more in both initial MLU and slope, BUT they overall tend to have a less change in MLU per visit).

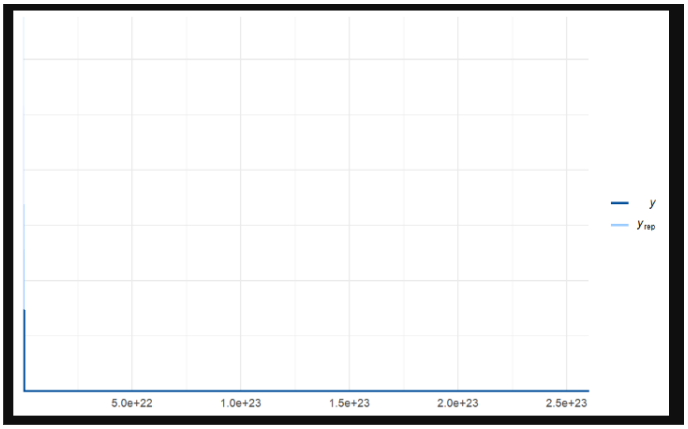


From the simulated data, we will therefore expect that some ASD-children will score quite high on MLU after 6 visits, but that they overall will score lower.

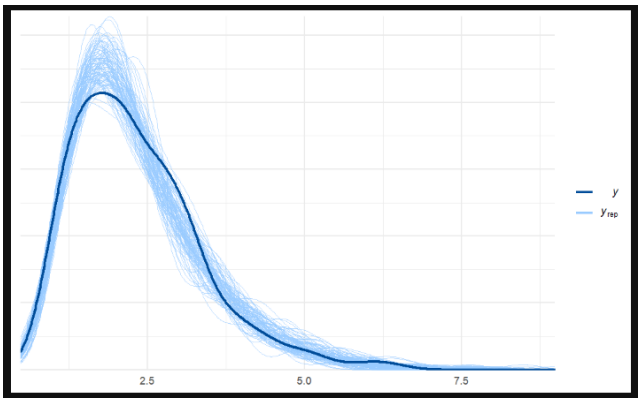
Analyzing simulated data

Here we've defined a formula, where we considered Diagnosis as well as the interaction between Diagnosis and Visit to be predictors. We included random slopes for Visit and ID because we expected an individual variability in the effects.
We then defined the priors by assessing values that yield a histogram of probable MLU values for both diagnoses.

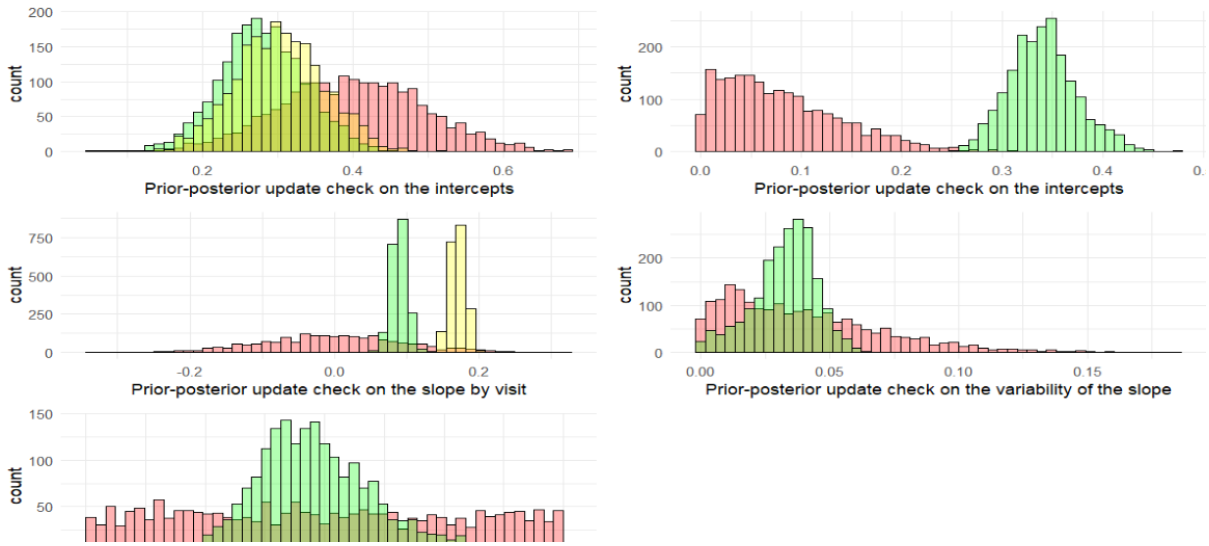
Using the brm_function, we fitted a model on our priors, in order to be able to do the pp checks. To start with, we sampled only from our priors and specified the family as lognormal. This yields extremely high values (see graph) during the pp_check, which we aren't really able to explain.



But we then created a different model with sample_prior=T, sampling from both simulated data and priors - our model_fit.
Using pp check, we assessed the model_fit and found our priors to fit much better:



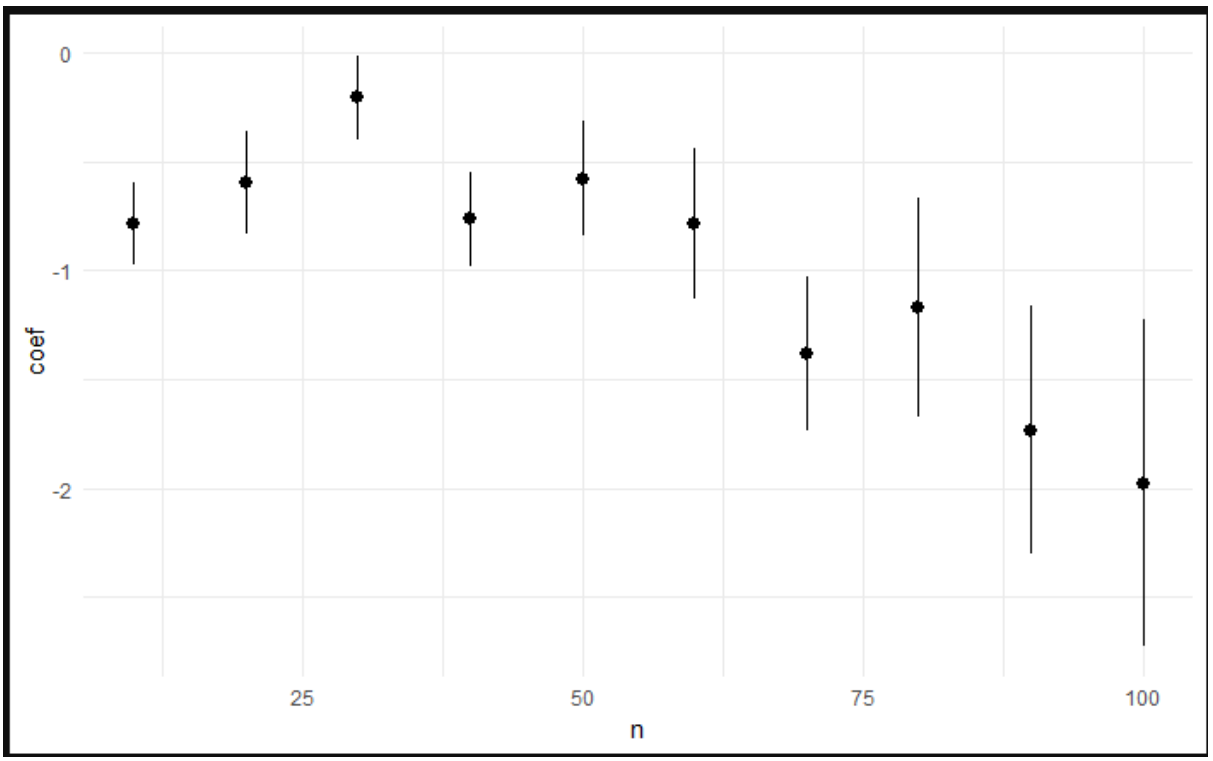
Thus, we injected this simulated data into the data-frame “posterior” from which we created different visualizations of our fit.



When it comes to precision analysis, we really struggled to get the Solomon Kurz-code up and running (often with errors concerned with the very R-installation when just simply copy-pasting his code).
To explore how it could work, we proposed a little simpler framework using the t-test for assessing sample size in the code written below:

```r
#First make model for a effect size and uncertainty, SE. Then

d_t_test <- function(d){
  t<-t.test(MLU ~ diagnosis, data = d, paired = T)
  d<-tibble(coef = t$estimate,
            CI_min = t$conf.int[1],
            CI_max = t$conf.int[2])

  return (d)
}

c<-c(10,20,30,40,50,60,70,80,90,100)

#power_analysis <- function (c,v,li) {
output<- tibble()

for (i in 1:length(c)) {
    a<-my_function(c[i],v,li)
    d<-d_t_test(a)

    output<- rbind(d,output)
}
output<-output %>%
  mutate(n = c)

ggplot(output, aes(n, coef))+
   geom_pointrange(aes(ymin = CI_min, ymax=CI_max))


```

In this we would first predict MLU based on diagnosis (and yes, here we would prospectively need to insert our model with far more predictors) before extracting the relevant values. By running several data-simulations (my_function - see above in the document) and thereby assessing the difference between the two groups, we yielded a plot relevant for plotting.
In this, we found the following:

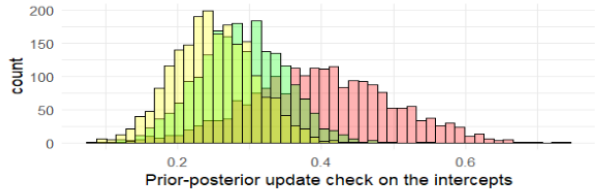


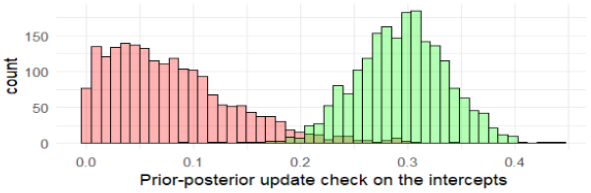
We are not sure that this is an alternative way of performing a power analysis using a non-bayesian framework, but we hope it at least proposes a try… At least it showed higher effect-sizes when moving up in the number of participants, which seems probable.

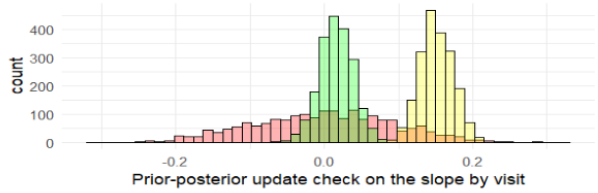
Q2

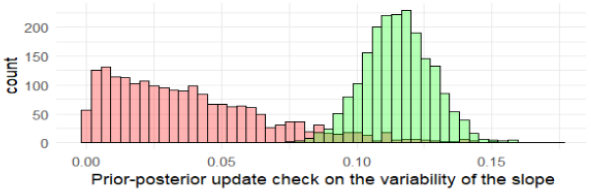
Part 2: Analyzing the real data
The second part of the analysis was pretty straightforward after having made the data-simulation. Of course, we needed to pluck out the relevant data-points, but then we created a model that would predict MLU from the dataset based on diagnosis, again with consideration to individual variance per datapoint.
From this we fitted a model, using the priors we decided on during the simulation and did similar posterior-checks to check the results.

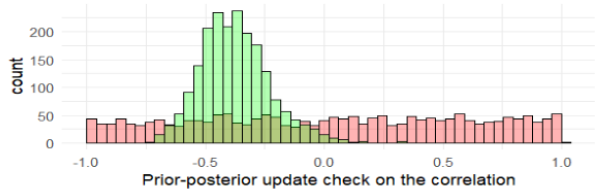
plots of the real data


count
200
150
100
50
0
0.2
0.4
0.6
Prior-posterior update check on the intercepts


count
150
100
50
0
0.0
0.1
0.2
0.3
0.4
Prior-posterior update check on the intercepts


count
400
300
200
100
0
-0.2
0.0
0.2
Prior-posterior update check on the slope by visit


count
200
150
100
50
0
0.00
0.05
0.10
0.15
Prior-posterior update check on the variability of the slope


count
200
150
100
50
0
-1.0
-0.5
0.0
0.5
1.0
Prior-posterior update check on the correlation