Q1 - Briefly describe your simulation process, its goals, and what you have learned from the simulation. Add at least a plot showcasing the results of the simulation. Make a special note on sample size considerations: how much data do you think you will need? what else could you do to increase the precision of your estimates?

The simulation process:

First, I transformed the mu and sigma^2 values given in the instructions with these functions:

The parameters μ and σ can be obtained, if the arithmetic mean and the arithmetic variance are known:

$$\mu = \ln\left(\frac{\mathrm{E}[X]^2}{\sqrt{\mathrm{E}[X^2]}}\right) = \ln\left(\frac{\mathrm{E}[X]^2}{\sqrt{\mathrm{Var}[X] + \mathrm{E}[X]^2}}\right),$$

$$\sigma^2 = \ln\left(\frac{\mathrm{E}[X^2]}{\mathrm{E}[X]^2}\right) = \ln\left(1 + \frac{\mathrm{Var}[X]}{\mathrm{E}[X]^2}\right).$$

(https://en.wikipedia.org/wiki/Log-normal_distribution#Arithmetic_moments)

This allowed me to plug them directly into a rlnorm() function, which I used to generate an individual intercept and an individual slopes for each participant. The true MLU value ('t_mlu' in the code) was then calculated to be either equal to the individual intercept in case of the first visit or calculated according to a formula: individual intercept + individual slope * Visit -1 for all visits after the first one.

As the last step, I calculated the measured MLU ('mlu' in code) – i.e. the true MLU with the measurement error. To model the measurement error I added a random draw from ~ Normal(0, 0.2) to each of the true MLU values.

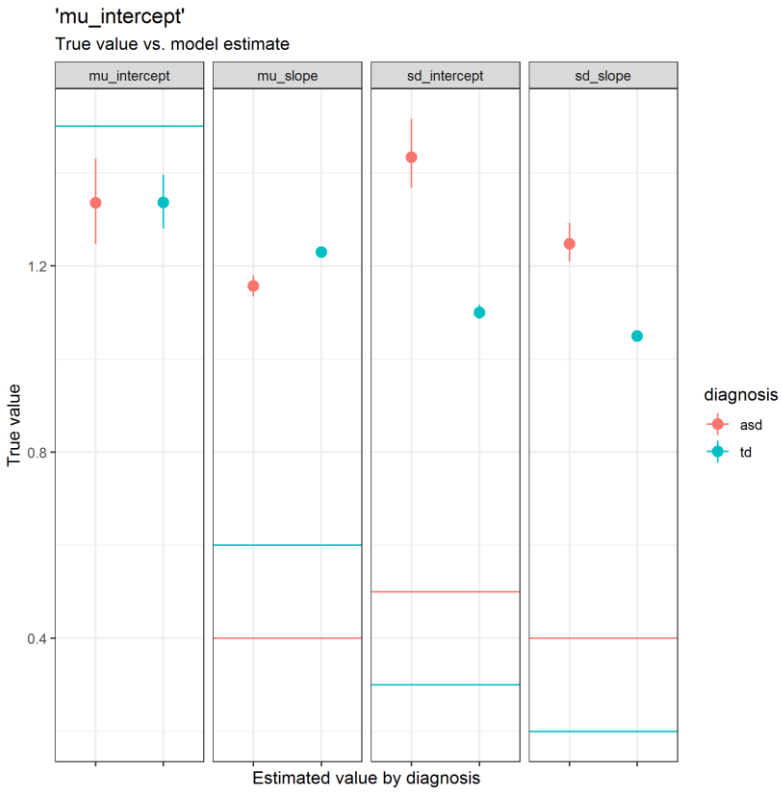
The goal of the simulation was to get data which true generative process would be known. This can help a lot in assessing the models we could fit to the real world data as well as understanding how they work or not work (how they break).

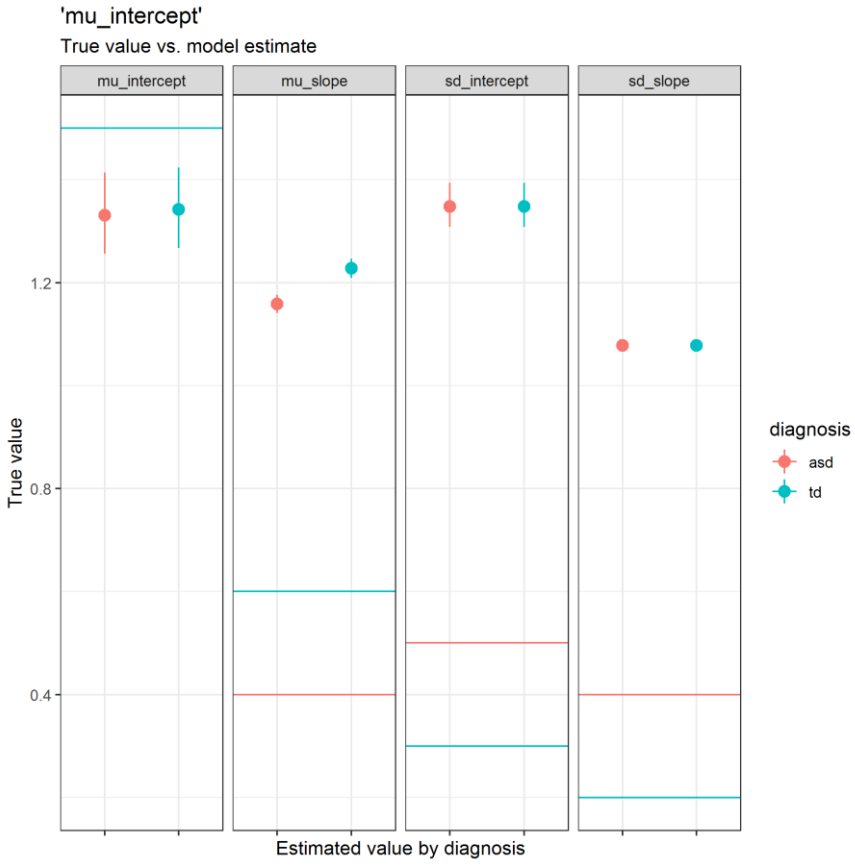
The simulation allowed me to compare two models of MLU.

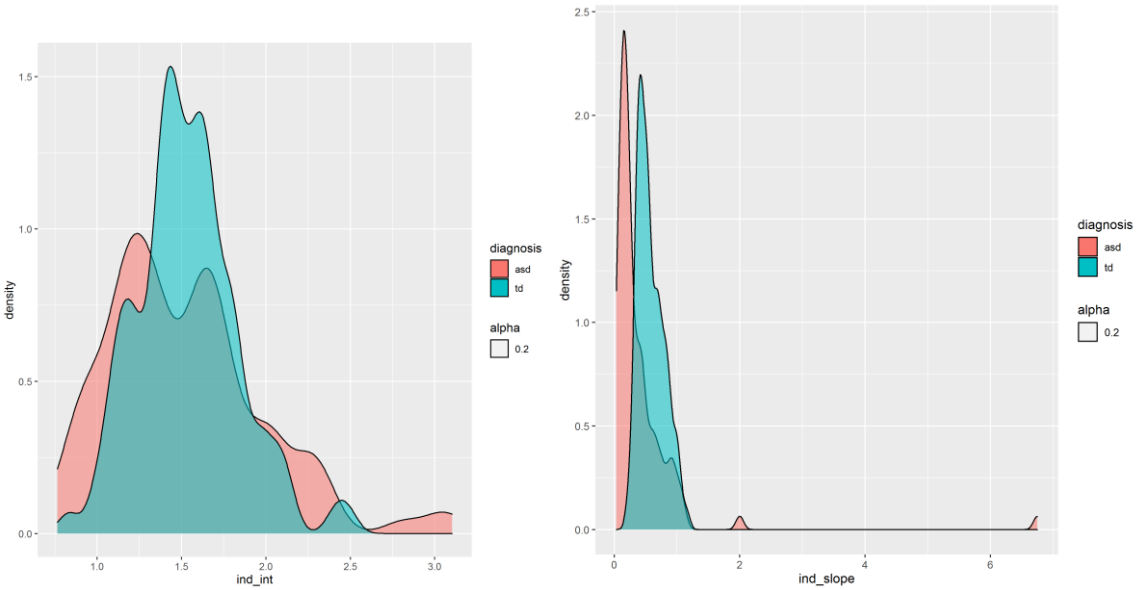
1. MLU ~ 0 + diagnosis + diagnosis:visit + (1 + visit|id)
2. MLU ~ 0 + diagnosis + diagnosis:visit + (1 + visit| gr(id, by = diagnosis))

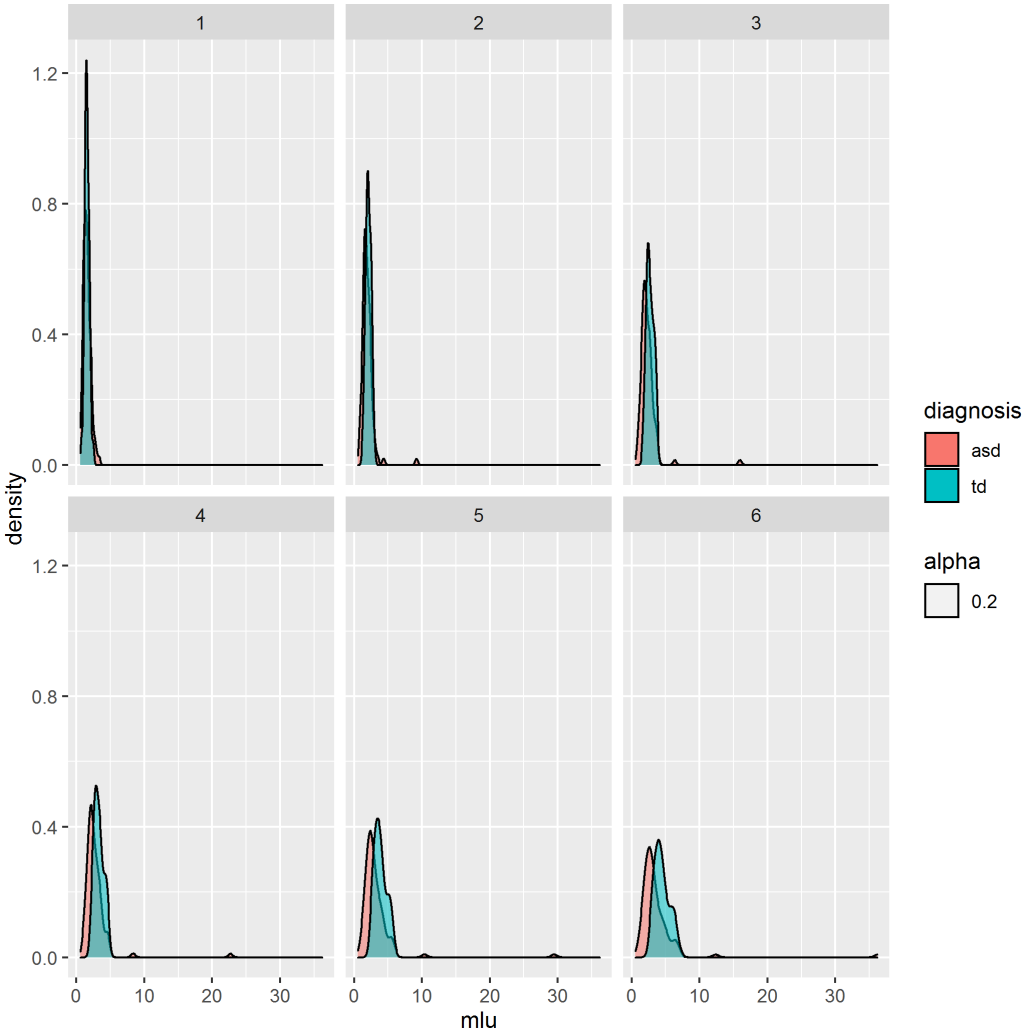
Both models allow varying slopes by participant, by one allows grouping of participants by diagnosis and the other does not. This results in partial pooling towards one aggregated means in model no. 1 and two different means for the different conditions in model no. 2.

Thanks to the simulation I learned how these models differ in their estimations of the simulated 'real' parameters.


'mu_intercept'
True value vs. model estimate
mu_intercept
mu_slope
sd_intercept
sd_slope
1.2
0.8
0.4
True value
Estimated value by diagnosis
diagnosis
asd
td

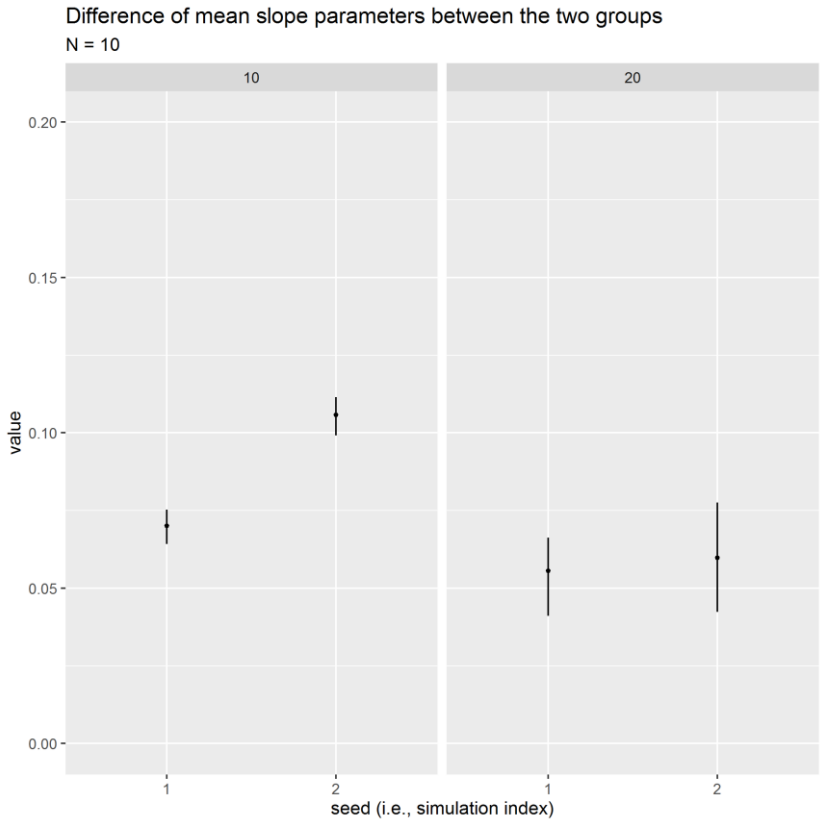

'mu_intercept'
True value vs. model estimate
mu_intercept
mu_slope
sd_intercept
sd_slope
1.2
0.8
0.4
True value
Estimated value by diagnosis
diagnosis
asd
td


density
ind_int
diagnosis
asd
td
alpha
0.2
density
ind_slope
diagnosis
asd
td
alpha
0.2


1
2
3
4
5
6
density
mlu
diagnosis
asd
td
alpha
0.2

Power / Precision analysis:

I unfortunately have to skip this part for now. I do have the code for the analysis in the .Rmd file, but I didn't manage to run it in the end. I have only checked whether the code works on just 2 N's and 2 seeds.



Increasing precision without increasing N:

Precision of the estimates could also be increased by making more informative priors. Wide priors assign a lot of the probability density / mass to the tails of the distribution, forcing the posterior to take a wider (less confident) distribution as well. This might however result in overfitting and the model not being generalisable.