Assignment 1 - Language development in autistic and neurotypical children

2022-08-15

Assignment 1 - Language development in autistic and neurotypical children

Quick recap

Autism Spectrum Disorder is often related to language impairment. However, this phenomenon has rarely been empirically traced in detail: i) relying on actual naturalistic language production, ii) over extended periods of time.

We therefore videotaped circa 30 kids with ASD and circa 30 comparison kids (matched by linguistic performance at visit 1) for ca. 30 minutes of naturalistic interactions with a parent. We repeated the data collection 6 times per kid, with 4 months between each visit. We transcribed the data and counted: i) the amount of words that each kid uses in each video. Same for the parent. ii) the amount of unique words that each kid uses in each video. Same for the parent. iii) the amount of morphemes per utterance (Mean Length of Utterance) displayed by each child in each video. Same for the parent.

The structure of the assignment

Produce a written document (separated from the code) answering the following questions:

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?

Q2 - Briefly describe the empirical data and how they compare to what you learned from the simulation (what can you learn from them?). Briefly describe your model(s) and model quality. Report the findings: how does development differ between autistic and neurotypical children (N.B. remember to report both population and individual level findings)? which additional factors should be included in the model? Add at least one plot showcasing your findings.

Part 1 - Simulating data

Before we even think of analyzing the data, we should make sure we understand the problem, and we plan the analysis. To do so, we need to simulate data and analyze the simulated data (where we know the ground truth).

In particular, 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. In other words, we need to define a few parameters: - average MLU for ASD (population mean) at Visit 1 and average individual deviation from that (population standard deviation) - average MLU for TD (population mean) at Visit 1 and average individual deviation from that (population standard deviation) - average change in MLU by visit for ASD (population mean) and average individual deviation from that (population standard deviation) - average change in MLU by visit for TD (population mean) and average individual deviation from that (population standard deviation) - an error term. Errors could be due to measurement, sampling, all sorts of noise.

Note that this makes a few assumptions: population means are exact values; change by visit is linear (the same between visit 1 and 2 as between visit 5 and 6). This is fine for the exercise. In real life research, you might want to vary the parameter values much more, relax those assumptions and assess how these things impact your inference.

We go through the literature and we settle for some values for these parameters: - average MLU for ASD and TD: 1.5 (remember the populations are matched for linguistic ability at first visit) - average individual variability in initial MLU for ASD 0.5; for TD 0.3 (remember ASD tends to be more heterogeneous) - average change in MLU for ASD: 0.4; for TD 0.6 (ASD is supposed to develop less) - average individual variability in change for ASD 0.4; for TD 0.2 (remember ASD tends to be more heterogeneous) - error is identified as 0.2

This would mean that on average the difference between ASD and TD participants is 0 at visit 1, 0.2 at visit 2, 0.4 at visit 3, 0.6 at visit 4, 0.8 at visit 5 and 1 at visit 6.

With these values in mind, simulate data, plot the data (to check everything is alright); and set up an analysis pipeline. 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

Once the pipeline is in place, loop through different sample sizes to assess how much data you would need to collect. N.B. for inspiration on how to set this up, check the tutorials by Kurz that are linked in the syllabus.

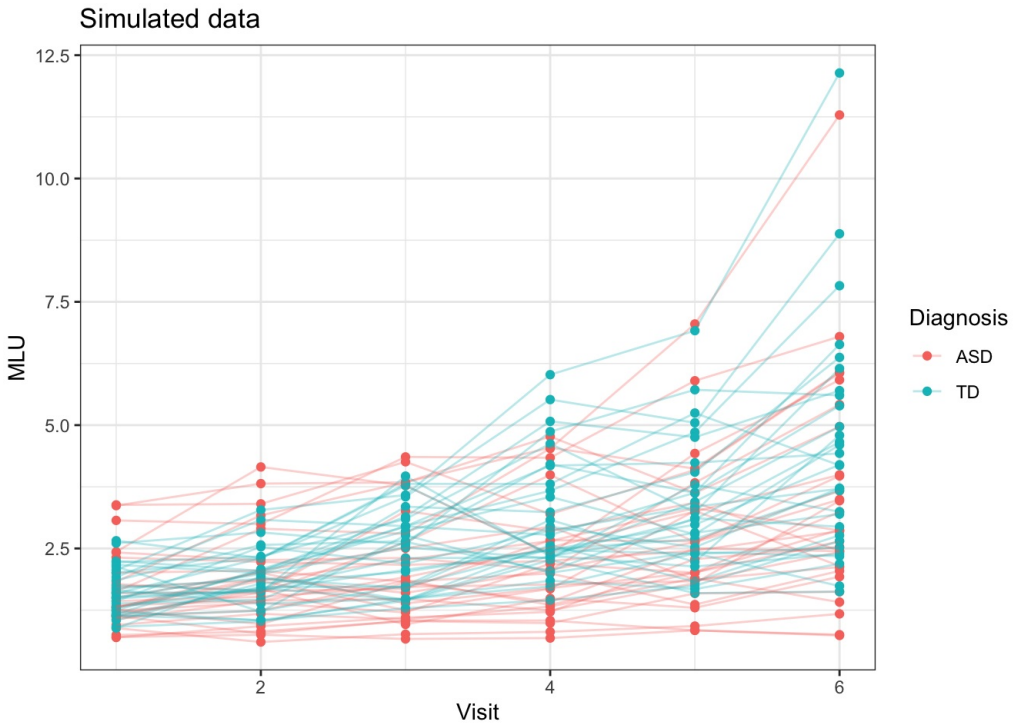
```
pacman::p_load(tidyverse, data.table, dplyr, tidybayes, ggplot2, ggridges, plyr, brms, cowplot, cmdstanr, purrr,
gridExtra)
```

Setting parameters for simulation

```
n <- 30
visit <- 6
error <- 0.2

#-----------------------------------
#ASD parameters

mean_MLU_asd <- log(1.5)
asd_sigma <- log(1.5) - log(1.5 - 0.5)        #More heterogeneous

mean_visit_asd <- 0.12                         #Develops less than TD, also making it relative to the average MLU
of ASD. #look at these values again, for TD and ASD.
sigma_visit_asd <- 0.1                         #More heterogeneity, making the sigma to be on the same scale as me
an_visit_asd.

#-------------------------------------
#TD parameters

mean_MLU_td <- log(1.5)
td_sigma <- log(1.5) - log(1.5 - 0.3)         #Less heterogeneous compared to ASD

mean_visit_td <- 0.2                           #Develops more than ASD, , also making it relative to the average M
LU of TD.
sigma_visit_td <- 0.06                         #Less heterogeneity, making the sigma to be on the same scale as me
an_visit_td.
```

Simulation of data

```
set.seed(2567) #Added random seed for simulation
#Making a function for simulating data
sim_f <- function(n, visit, mean_MLU_asd, mean_MLU_td, asd_sigma, td_sigma, error){
  s_df <- tibble(expand.grid(ID=seq(n),
                             Diagnosis= c("ASD", "TD"),
                             Visit = seq(visit))) %>%
    mutate(ID = ifelse(Diagnosis == "TD", ID + n, ID),
           Intercept = NA,
           Slope = NA,
           MLU = NA)

  for (i in seq(s_df$ID)) {
    #Assigning individual intercept
    s_df$Intercept[s_df$ID == i & s_df$Diagnosis == "ASD"] <- rnorm(1, mean_MLU_asd, asd_sigma)
    s_df$Intercept[s_df$ID == i & s_df$Diagnosis == "TD"] <- rnorm(1, mean_MLU_td, td_sigma)

    #Assigning individual slope
    s_df$Slope[s_df$ID == i & s_df$Diagnosis == "ASD"] <- rnorm(1, mean_visit_asd, sigma_visit_asd)
    s_df$Slope[s_df$ID == i & s_df$Diagnosis == "TD"] <- rnorm(1, mean_visit_td, sigma_visit_td)
  }

  for (i in seq(nrow(s_df))){
  s_df$MLU[i] <- exp(rnorm(1, (s_df$Intercept[i] + s_df$Slope[i] * (s_df$Visit[i]-1)), error))
  }
  return(s_df)
}

df <- sim_f(n, visit, mean_MLU_asd, mean_MLU_td, asd_sigma, td_sigma, error)

#Visualizing data
simulated_data <- ggplot(df, aes(Visit, MLU, color = Diagnosis, group = ID)) +
  theme_bw() +
  geom_point() +
  geom_line(alpha = 0.3) + ggtitle("Simulated data")
simulated_data
```



Building the model

```
model_f <- bf(MLU ~ 0 + Diagnosis + Diagnosis:Visit + (1 + Visit | ID))
get_prior(model_f, data = df, family = lognormal)
```

```
##                 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                        ID
##  student_t(3, 0, 2.5)    sd                        ID                  0
##  student_t(3, 0, 2.5)    sd                        ID                  0
##  student_t(3, 0, 2.5)    sd          Intercept     ID                  0
##  student_t(3, 0, 2.5)    sd              Visit     ID                  0
##  student_t(3, 0, 2.5) sigma                                            0
##        source
##       default
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##       default
##  (vectorized)
##       default
##  (vectorized)
##  (vectorized)
##  (vectorized)
##       default
```

Defining priors

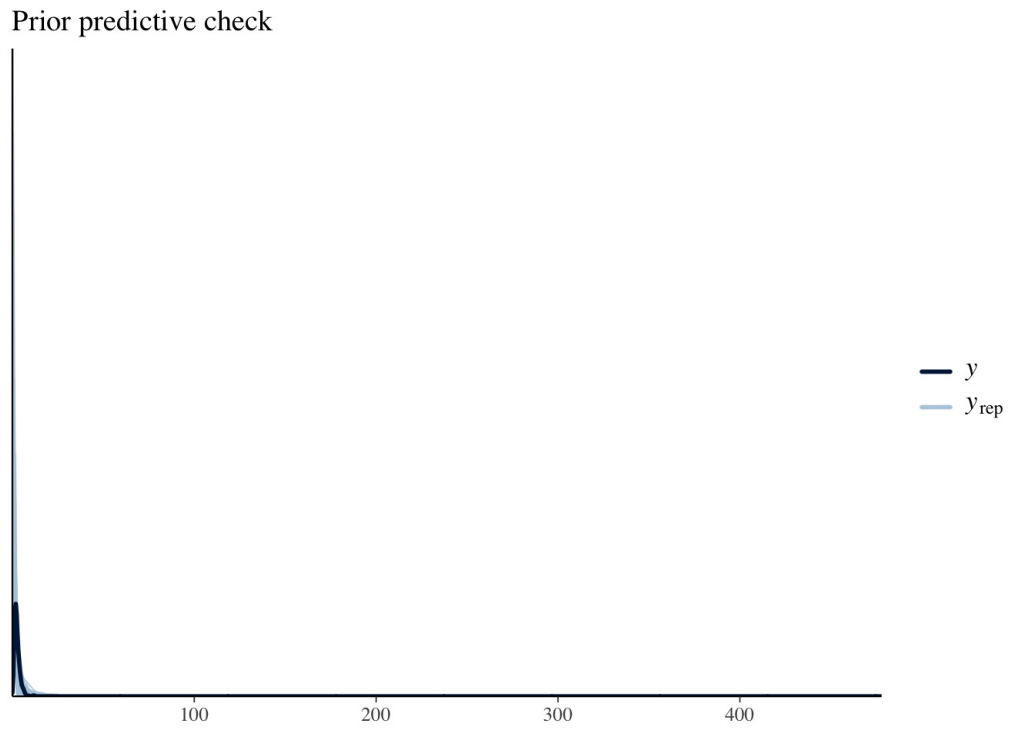
```
MLU_priors <- c(
  prior(normal(0.41, 0.4), class = b, coef = "DiagnosisASD"), #Just keep the same for both.
  prior(normal(0.41, 0.4), class = b, coef = "DiagnosisTD"),  #Keeping the mean and uncertainty same, based on th
e true values.
  prior(normal(0,0.2),class=b,coef="DiagnosisASD:Visit"), #Keeping the slopes the same
  prior(normal(0,0.2),class=b,coef="DiagnosisTD:Visit"),
  prior(normal(0, 0.3), class = sd, coef = Intercept, group = ID), #Took mean SD of both groups.
  prior(normal(0, 0.1), class = sd, coef = Visit, group= ID),
  prior(normal(0, 0.2), class = sigma),
  prior(lkj(2), class = cor) #Dampens extreme correlations.
)
```

Model to sample priors

```
MLU_mod_prior <- brm(
  model_f,
  data = df,
  family = lognormal,
  prior = MLU_priors,
  sample_prior = "only",
  backend = "cmdstanr",
  chains = 2,
  core = 2,
  control = list(adapt_delt = 0.99, max_treedepth = 20))
```

```
## Start sampling
```

```
pp_check(MLU_mod_prior, ndraws=100) + labs(title = "Prior predictive check") #Prior predictive check. Seems to be in the range of actual data, although looks odd when in log-scale.
```

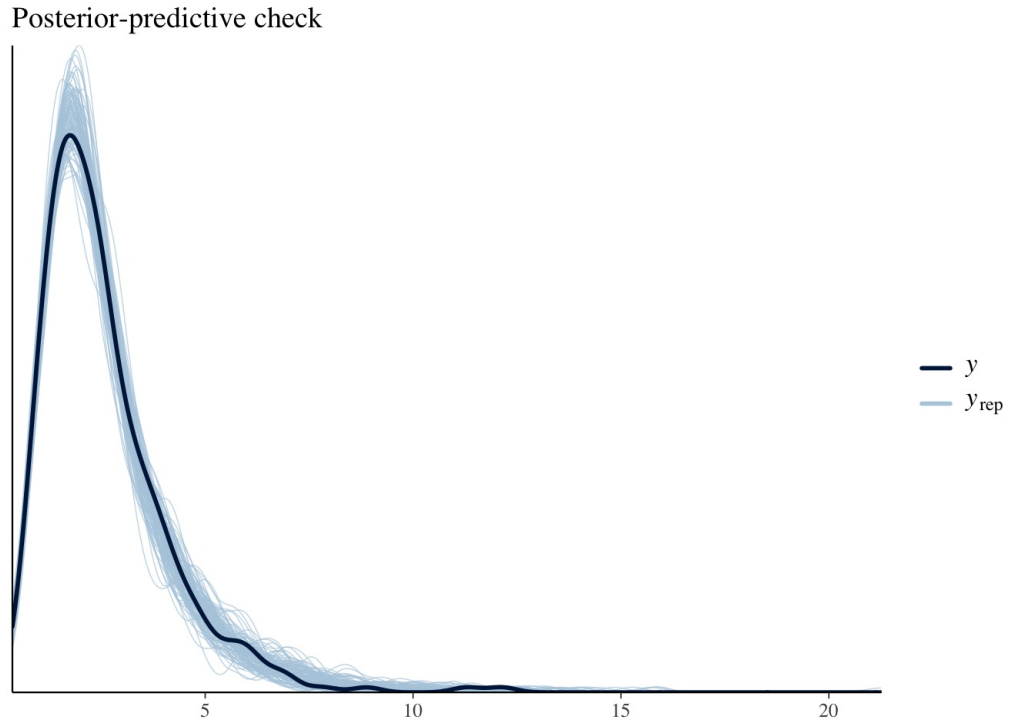


Model to compare prior model to posterior

```
MLU_model <- brm(
  model_f,
  data = df,
  family = lognormal,
  prior = MLU_priors,
  sample_prior = T,
  backend = "cmdstanr",
  chains = 2,
  core = 2,
  control = list(adapt_delt = 0.99, max_treedepth = 20))
```

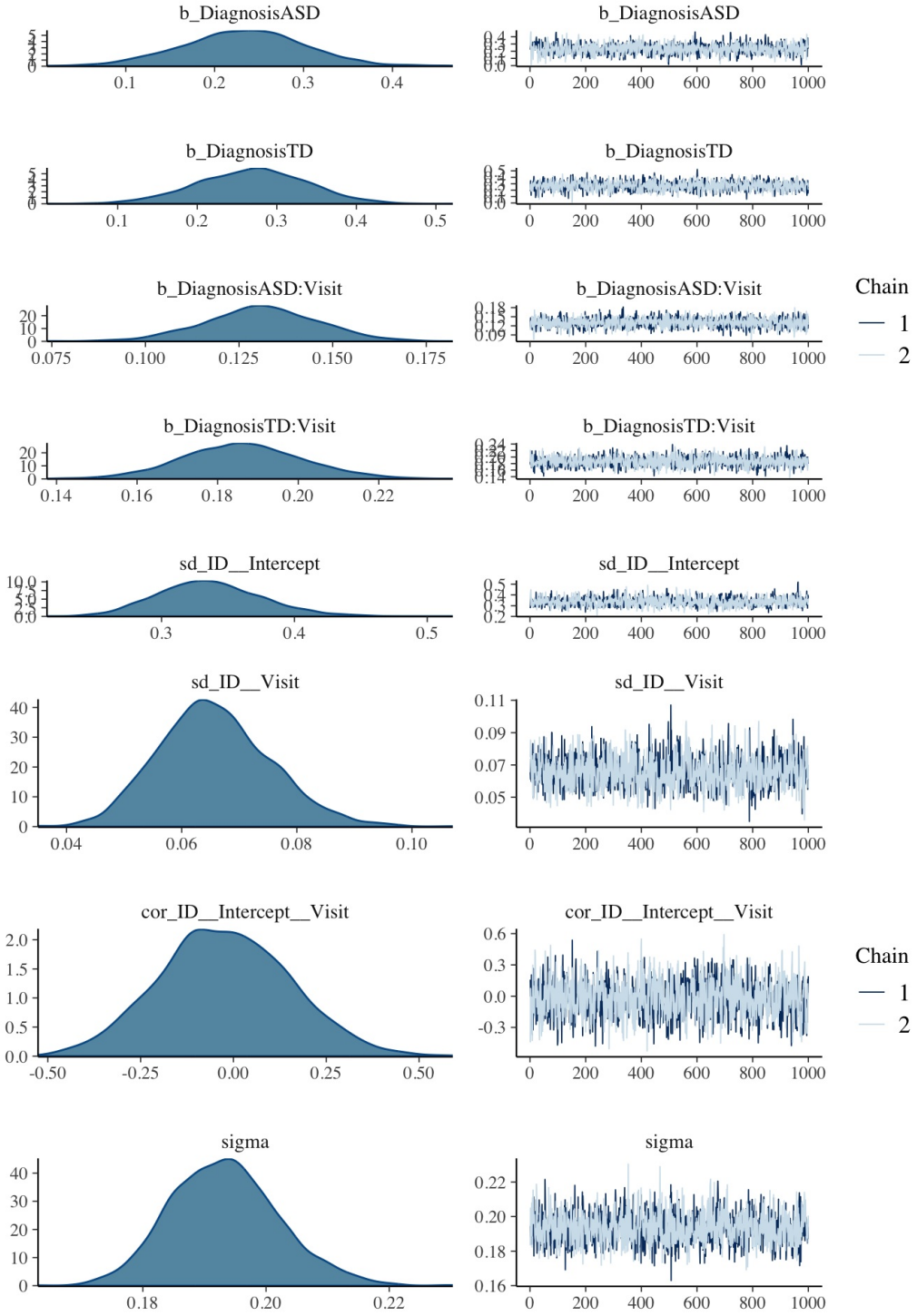
```
## Start sampling
```

```
pp_check(MLU_model, ndraws=100) + labs(title = "Posterior-predictive check")
```

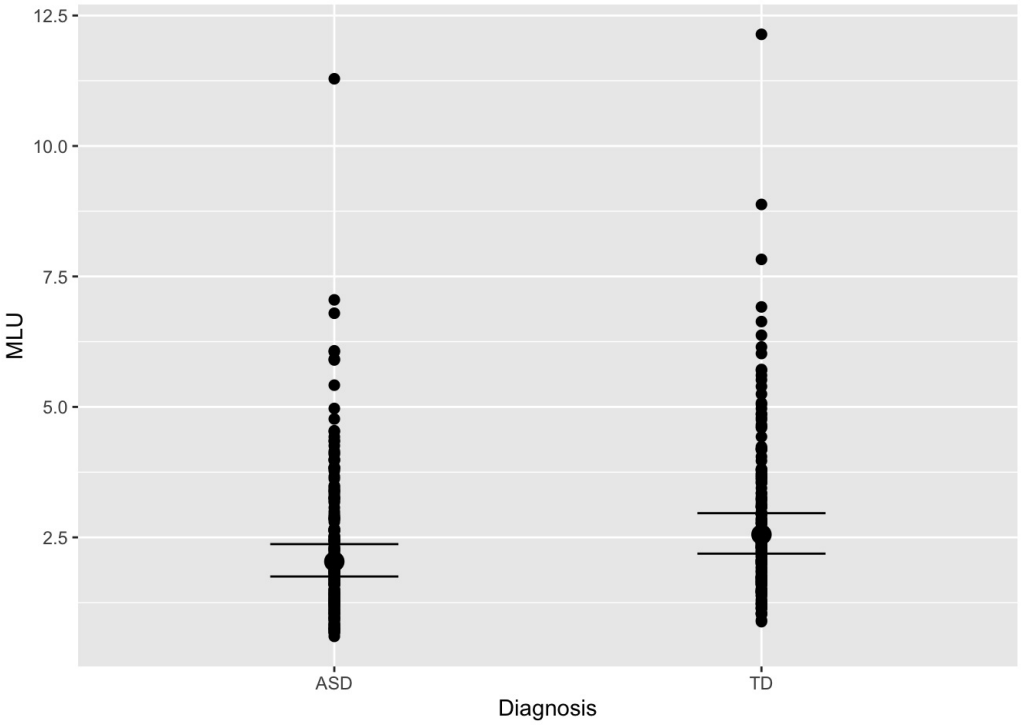


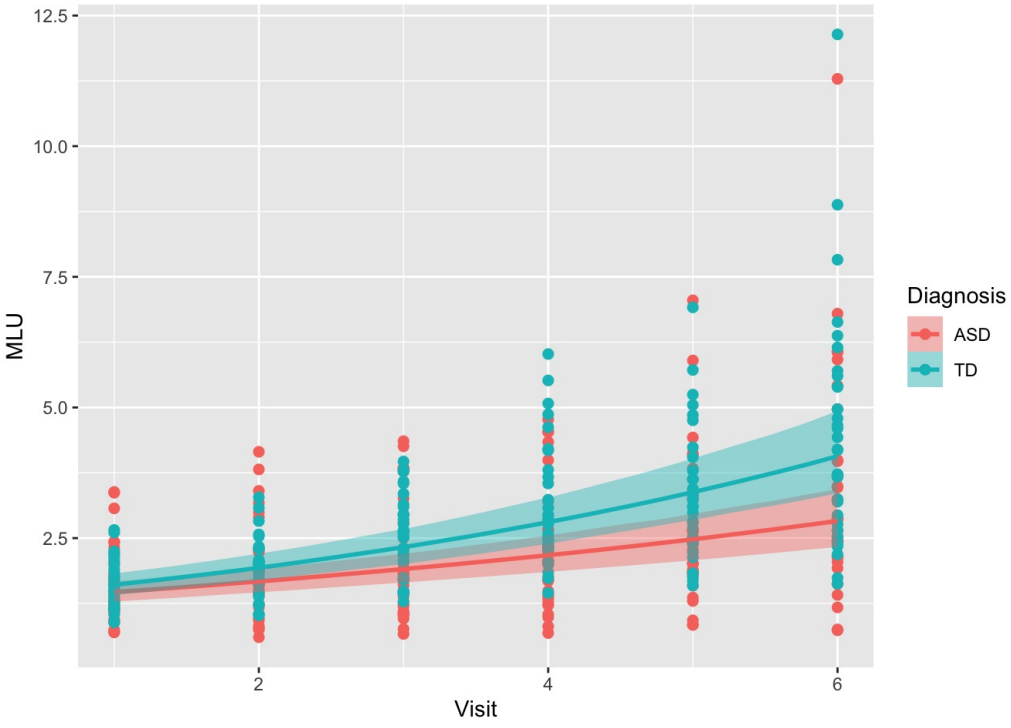
Plots

```
plot(MLU_model) #Trace plots.
```




```
# Model inference (population estimate) plus actual data. Another way of doing posterior predictive checks.
plot(conditional_effects(MLU_model), points = T)
```

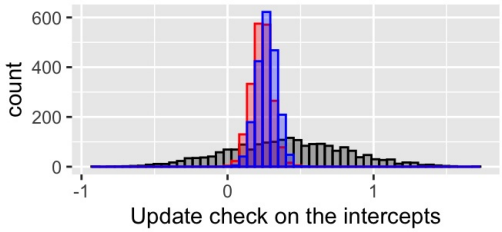


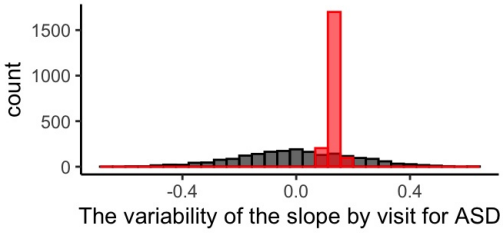


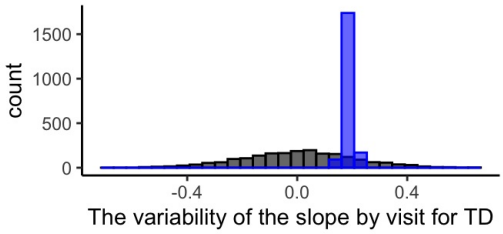
Comparisons:

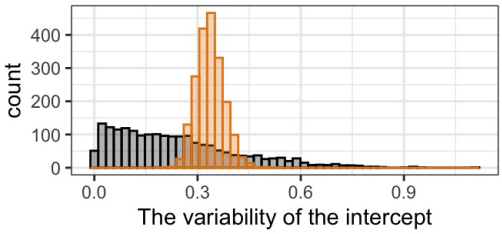
```
posterior <- as_draws_df(MLU_model)
#-------------------------------------------------------------------------------

plot_intercepts <- ggplot(posterior) +
  geom_histogram(aes(prior_b_DiagnosisASD),
                 fill = "black", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(b_DiagnosisASD),
                 fill = "red", color = "red", alpha = 0.3, bins = 50) +
    geom_histogram(aes(b_DiagnosisTD),
                 fill = "blue", color = "blue", alpha = 0.3, bins = 50)+
  xlab("Update check on the intercepts")

#-------------------------------------------------------------------------------

plot_b_visit_ASD <- ggplot(posterior) +
  geom_histogram(aes(`prior_b_DiagnosisASD:Visit`), fill="black", color="black",alpha=0.6,) +
  geom_histogram(aes(`b_DiagnosisASD:Visit`), fill="red", color="red",alpha=0.6) +
  theme_classic() +
   xlab("The variability of the slope by visit for ASD")

#-------------------------------------------------------------------------------

plot_b_visit_TD <- ggplot(posterior) +
  geom_histogram(aes(`prior_b_DiagnosisTD:Visit`), fill="black", color="black",alpha=0.6,) +
  geom_histogram(aes(`b_DiagnosisTD:Visit`), fill="blue", color="blue",alpha=0.6) +
  theme_classic() +
   xlab("The variability of the slope by visit for TD")

#-------------------------------------------------------------------------------

plot_sd_intercept <- ggplot(posterior) +
  geom_histogram(aes(prior_sd_ID__Intercept),
                 fill = "black", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(sd_ID__Intercept),
                 fill = "#E68613", color = "#E68613", alpha = 0.3, bins = 50) +
  theme_bw() +
  xlab("The variability of the intercept")

#-------------------------------------------------------------------------------

plot_sd_visit <- ggplot(posterior) +
  geom_histogram(aes(prior_sd_ID__Visit), fill="black", color="black",alpha=0.6,) +
  geom_histogram(aes(sd_ID__Visit), fill="#E68613", color="#E68613",alpha=0.6) +
  theme_classic() +
   xlab("The variability of the slope")

#-------------------------------------------------------------------------------

#Prior-posterior update checks:
grid.arrange(plot_intercepts, plot_b_visit_ASD, plot_b_visit_TD, plot_sd_intercept, plot_sd_visit)
```

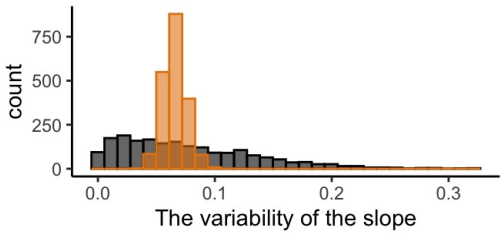
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```











Inspecting the parameters of our model

```
#summary(MLU_model)
```

Extracting and figuring out in which way the model is assessing the individual level estimate How wrong is the model when it tries to reconstruct the specific intercept and slope of all 60 children.

```
temp_re <- ranef(MLU_model)$ID
for (i in unique(df$ID)) {
  temp <- as.character(i)
  df$EstimatedIntercept[df$ID == i] <- temp_re[,,"Intercept"][temp,1]
  df$EstimatedIntercept_low[df$ID == i] <- temp_re[,,"Intercept"][temp,3]
  df$EstimatedIntercept_high[df$ID == i] <- temp_re[,,"Intercept"][temp,4]
  df$EstimatedSlope[df$ID == i] <- temp_re[,,"Visit"][temp,1]
  df$EstimatedSlope_low[df$ID == i] <- temp_re[,,"Visit"][temp,3]
  df$EstimatedSlope_high[df$ID == i] <- temp_re[,,"Visit"][temp,4]
}
```

```
## Warning: Unknown or uninitialised column: `EstimatedIntercept`.
```

```
## Warning: Unknown or uninitialised column: `EstimatedIntercept_low`.
```

```
## Warning: Unknown or uninitialised column: `EstimatedIntercept_high`.
```

```
## Warning: Unknown or uninitialised column: `EstimatedSlope`.
```

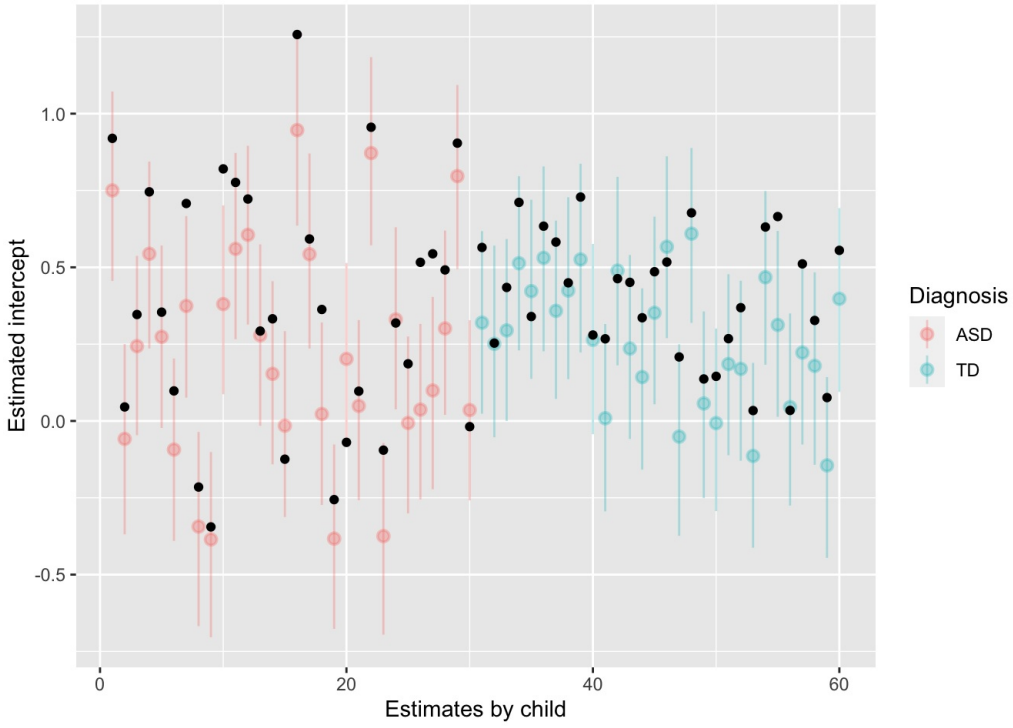
```
## Warning: Unknown or uninitialised column: `EstimatedSlope_low`.
```

```
## Warning: Unknown or uninitialised column: `EstimatedSlope_high`.
```

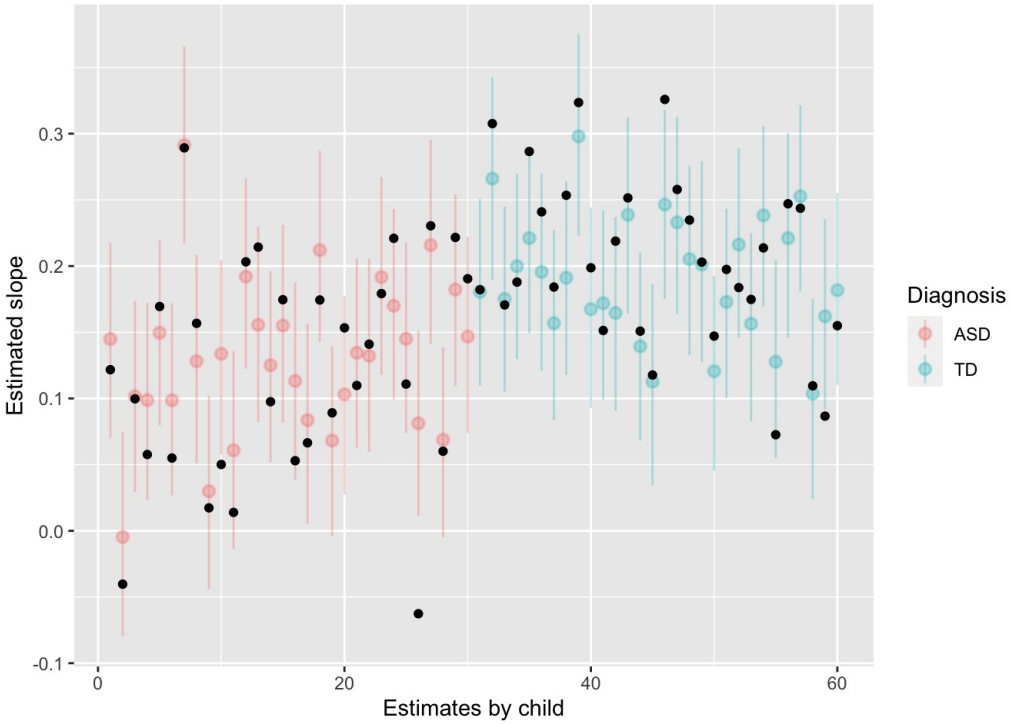
```
df_est1 <- df %>% subset(Visit==1) %>%
  mutate(
    EstimatedIntercept = ifelse(Diagnosis =="ASD",
                                EstimatedIntercept + 0.23, #Estimate for DiagnosisASD
                                EstimatedIntercept + 0.27), #Estimate for DiagnosisASD
    EstimatedIntercept_low = ifelse(Diagnosis=="ASD",
                                EstimatedIntercept_low + 0.23,
                                EstimatedIntercept_low + 0.27),
    EstimatedIntercept_high = ifelse(Diagnosis=="ASD",
                                EstimatedIntercept_high + 0.23,
                                EstimatedIntercept_high + 0.27),


    EstimatedSlope = ifelse(Diagnosis=="ASD",
                                EstimatedSlope + 0.13, #Estimate for DiagnosisASD:Visit
                                EstimatedSlope + 0.19),#Estimate for DiagnosisTD:Visit
    EstimatedSlope_low = ifelse(Diagnosis=="ASD",
                                EstimatedSlope_low + 0.13,
                                EstimatedSlope_low + 0.19),
    EstimatedSlope_high = ifelse(Diagnosis=="ASD",
                                EstimatedSlope_high + 0.13,
                                EstimatedSlope_high + 0.19)

  )
#head(temp_re)
```

Plotting estimates

```
Est_intercept <- ggplot(df_est1)+
  geom_pointrange(aes(x=as.numeric(as.factor(ID)),y=EstimatedIntercept,
                      ymin=EstimatedIntercept_low,ymax=EstimatedIntercept_high,
                      color = Diagnosis),alpha=0.3) +
  geom_point(aes(x=as.numeric(as.factor(ID)),y=Intercept))+
  xlab("Estimates by child")+
  ylab("Estimated intercept")


Est_slope <- ggplot(df_est1)+
  geom_pointrange(aes(x=as.numeric(as.factor(ID)),y=EstimatedSlope,
                      ymin=EstimatedSlope_low,ymax=EstimatedSlope_high,
                      color = Diagnosis),alpha=0.3) +
  geom_point(aes(x=as.numeric(as.factor(ID)),y=Slope))+
  xlab("Estimates by child")+
  ylab("Estimated slope")
Est_intercept
```




```
Est_slope
```




```
#grid.arrange(Est_intercept, Est_slope)
```

Creating update function for power analysis - not used.

```
#Set a new seed for the power analysis
set.seed(2)
#Simulate new data set based on the new seed.
df_new <- sim_f(n, visit, mean_MLU_asd, mean_MLU_td, asd_sigma, td_sigma, error)

updated_fit_df_power <-
  update(MLU_model,
         newdata = df_new,
         seed = 2)
```

```
## Start sampling
```

```
## Running MCMC with 2 sequential chains...
##
## Chain 1 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1 Iteration:  100 / 2000 [  5%]  (Warmup)
## Chain 1 Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1 Iteration:  300 / 2000 [ 15%]  (Warmup)
## Chain 1 Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1 Iteration:  500 / 2000 [ 25%]  (Warmup)
## Chain 1 Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1 Iteration:  700 / 2000 [ 35%]  (Warmup)
## Chain 1 Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1 Iteration:  900 / 2000 [ 45%]  (Warmup)
## Chain 1 Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1 Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 1 Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1 Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 1 Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1 Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 1 Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1 Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 1 Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1 Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 1 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1 finished in 13.2 seconds.
## Chain 2 Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2 Iteration:  100 / 2000 [  5%]  (Warmup)
## Chain 2 Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2 Iteration:  300 / 2000 [ 15%]  (Warmup)
## Chain 2 Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2 Iteration:  500 / 2000 [ 25%]  (Warmup)
## Chain 2 Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2 Iteration:  700 / 2000 [ 35%]  (Warmup)
## Chain 2 Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2 Iteration:  900 / 2000 [ 45%]  (Warmup)
## Chain 2 Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2 Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2 Iteration: 1100 / 2000 [ 55%]  (Sampling)
## Chain 2 Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2 Iteration: 1300 / 2000 [ 65%]  (Sampling)
## Chain 2 Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2 Iteration: 1500 / 2000 [ 75%]  (Sampling)
## Chain 2 Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2 Iteration: 1700 / 2000 [ 85%]  (Sampling)
## Chain 2 Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2 Iteration: 1900 / 2000 [ 95%]  (Sampling)
## Chain 2 Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2 finished in 12.8 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 13.0 seconds.
## Total execution time: 26.0 seconds.
```

Creating custom function that will simulate new data sets did not use it at the end.

```
sim_d <- function(seed, n) {
  visit <- 6
  mean_MLU_asd <- log(1.5)
  asd_sigma <- log(1.5) - log(1.5 - 0.5)
  mean_visit_asd <- 0.12
  sigma_visit_asd <- 0.1
  mean_MLU_td <- log(1.5)
  td_sigma <- log(1.5) - log(1.5 - 0.3)
  mean_visit_td <- 0.2
  sigma_visit_td <- 0.06
  set.seed(seed)

  s_df <- tibble(expand.grid(ID=seq(n),
                             Diagnosis= c("ASD", "TD"),
                             Visit = seq(visit))) %>%
    mutate(ID = ifelse(Diagnosis == "TD", ID + n, ID),
           Intercept = NA,
           Slope = NA,
           MLU = NA)

  for (i in seq(s_df$ID)) {
    #Assigning individual intercept
    s_df$Intercept[s_df$ID == i & s_df$Diagnosis == "ASD"] <- rnorm(1, mean_MLU_asd, asd_sigma)
    s_df$Intercept[s_df$ID == i & s_df$Diagnosis == "TD"] <- rnorm(1, mean_MLU_td, td_sigma)

    #Assigning individual slope
    s_df$Slope[s_df$ID == i & s_df$Diagnosis == "ASD"] <- rnorm(1, mean_visit_asd, sigma_visit_asd)
    s_df$Slope[s_df$ID == i & s_df$Diagnosis == "TD"] <- rnorm(1, mean_visit_td, sigma_visit_td)
  }

  for (i in seq(nrow(s_df))){
  s_df$MLU[i] <- exp(rnorm(1, (s_df$Intercept[i] + s_df$Slope[i] * (s_df$Visit[i]-1)), 0.2))
  }
  return(s_df)
}
```

Power analysis for n = 30.

```
sim_n <- 10 #To make it faster. 100 might be more accurate.

#Simulates 10 data sets with sample size of 30.
s_df_n30 <-
  tibble(seed = 1:sim_n) %>%
  mutate(data = map(seed, sim_d, n = 30))

#Tibble that stores all values that are needed for analysis.
estimates_df_n30 <- tibble(
  Model = seq(10),
  Mean_diff_b = NA,
  Upper = NA,
  Lower = NA
)

#Model_1_n30
m1_n30 <- update(MLU_model, newdata = s_df_n30$data[1])   #Creating a model
m1_n30_draws <- as_draws_df(m1_n30)                       #Taking the draws

m1_n30_diff <- m1_n30_draws$`b_DiagnosisTD:Visit` - m1_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[1,2] <- mean(m1_n30_diff)
estimates_df_n30[1,3] <- quantile(m1_n30_diff, 0.975)
estimates_df_n30[1,4] <- quantile(m1_n30_diff, 0.025)

#Model_2_n30
m2_n30 <- update(MLU_model, newdata = s_df_n30$data[2])
m2_n30_draws <- as_draws_df(m2_n30)

m2_n30_diff <- m2_n30_draws$`b_DiagnosisTD:Visit` - m2_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[2,2] <- mean(m2_n30_diff)
estimates_df_n30[2,3] <- quantile(m2_n30_diff, 0.975)
estimates_df_n30[2,4] <- quantile(m2_n30_diff, 0.025)

#Model_3_n30
m3_n30 <- update(MLU_model, newdata = s_df_n30$data[3])
m3_n30_draws <- as_draws_df(m3_n30)

m3_n30_diff <- m3_n30_draws$`b_DiagnosisTD:Visit` - m3_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[3,2] <- mean(m3_n30_diff)
estimates_df_n30[3,3] <- quantile(m3_n30_diff, 0.975)
```

```
estimates_df_n30[3,4] <- quantile(m3_n30_diff, 0.025)

#Model_4_n30
m4_n30 <- update(MLU_model, newdata = s_df_n30$data[4])
m4_n30_draws <- as_draws_df(m4_n30)

m4_n30_diff <- m4_n30_draws$`b_DiagnosisTD:Visit` - m4_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[4,2] <- mean(m4_n30_diff)
estimates_df_n30[4,3] <- quantile(m4_n30_diff, 0.975)
estimates_df_n30[4,4] <- quantile(m4_n30_diff, 0.025)

#Model_5_n30
m5_n30 <- update(MLU_model, newdata = s_df_n30$data[5])
m5_n30_draws <- as_draws_df(m5_n30)

m5_n30_diff <- m5_n30_draws$`b_DiagnosisTD:Visit` - m5_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[5,2] <- mean(m5_n30_diff)
estimates_df_n30[5,3] <- quantile(m5_n30_diff, 0.975)
estimates_df_n30[5,4] <- quantile(m5_n30_diff, 0.025)

#Model_6_n30
m6_n30 <- update(MLU_model, newdata = s_df_n30$data[6])
m6_n30_draws <- as_draws_df(m6_n30)

m6_n30_diff <- m6_n30_draws$`b_DiagnosisTD:Visit` - m6_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[6,2] <- mean(m6_n30_diff)
estimates_df_n30[6,3] <- quantile(m6_n30_diff, 0.975)
estimates_df_n30[6,4] <- quantile(m6_n30_diff, 0.025)

#Model_7_n30
m7_n30 <- update(MLU_model, newdata = s_df_n30$data[7])
m7_n30_draws <- as_draws_df(m7_n30)

m7_n30_diff <- m7_n30_draws$`b_DiagnosisTD:Visit` - m7_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[7,2] <- mean(m7_n30_diff)
estimates_df_n30[7,3] <- quantile(m7_n30_diff, 0.975)
estimates_df_n30[7,4] <- quantile(m7_n30_diff, 0.025)

#Model_8_n30
m8_n30 <- update(MLU_model, newdata = s_df_n30$data[8])
m8_n30_draws <- as_draws_df(m8_n30)

m8_n30_diff <- m8_n30_draws$`b_DiagnosisTD:Visit` - m8_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[8,2] <- mean(m8_n30_diff)
estimates_df_n30[8,3] <- quantile(m8_n30_diff, 0.975)
estimates_df_n30[8,4] <- quantile(m8_n30_diff, 0.025)

#Model_9_n30
m9_n30 <- update(MLU_model, newdata = s_df_n30$data[9])
m9_n30_draws <- as_draws_df(m9_n30)

m9_n30_diff <- m9_n30_draws$`b_DiagnosisTD:Visit` - m9_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[9,2] <- mean(m9_n30_diff)
estimates_df_n30[9,3] <- quantile(m9_n30_diff, 0.975)
estimates_df_n30[9,4] <- quantile(m9_n30_diff, 0.025)

#Model_10_n30
m10_n30 <- update(MLU_model, newdata = s_df_n30$data[10])
m10_n30_draws <- as_draws_df(m10_n30)

m10_n30_diff <- m10_n30_draws$`b_DiagnosisTD:Visit` - m10_n30_draws$`b_DiagnosisASD:Visit`
estimates_df_n30[10,2] <- mean(m10_n30_diff)
estimates_df_n30[10,3] <- quantile(m10_n30_diff, 0.975)
estimates_df_n30[10,4] <- quantile(m10_n30_diff, 0.025)

#Plot of difference

power_plot_n30 <- estimates_df_n30 %>%
  ggplot(aes(x = Model, y = Mean_diff_b, ymin = Lower, ymax = Upper)) +
  geom_pointrange(fatten = 1/2) +
  geom_hline(yintercept = 0.08, color = "red") + #True difference in mean: 0.2 (TD) - 0.12 (ASD)
  labs(x = "Seed",
       y = "Difference in slope") +
  ggtitle("N = 30")  +
  scale_x_continuous(breaks=seq(0,10,by=1)) +
  ylim(-0.04, 0.2)
power_plot_n30
```

Power analysis for n = 50.

```
s_df_n50 <-
  tibble(seed = 1:sim_n) %>%
  mutate(data = map(seed, sim_d, n = 50))

#Tibble that stores all values that are needed for analysis.
estimates_df_n50 <- tibble(
  Model = seq(10),
  Mean_diff_b = NA,
  Upper = NA,
  Lower = NA
)

#Model_1_n50
m1_n50 <- update(MLU_model, newdata = s_df_n50$data[1])
m1_n50_draws <- as_draws_df(m5_n30)

m1_n50_diff <- m1_n50_draws$`b_DiagnosisTD:Visit` - m1_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[1,2] <- mean(m1_n50_diff)
estimates_df_n50[1,3] <- quantile(m1_n50_diff, 0.975)
estimates_df_n50[1,4] <- quantile(m1_n50_diff, 0.025)

#Model_2_n50
m2_n50 <- update(MLU_model, newdata = s_df_n50$data[2])
m2_n50_draws <- as_draws_df(m5_n30)

m2_n50_diff <- m2_n50_draws$`b_DiagnosisTD:Visit` - m2_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[2,2] <- mean(m2_n50_diff)
estimates_df_n50[2,3] <- quantile(m2_n50_diff, 0.975)
estimates_df_n50[2,4] <- quantile(m2_n50_diff, 0.025)

#Model_3_n50
m3_n50 <- update(MLU_model, newdata = s_df_n50$data[3])
m3_n50_draws <- as_draws_df(m3_n50)

m3_n50_diff <- m3_n50_draws$`b_DiagnosisTD:Visit` - m3_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[3,2] <- mean(m3_n50_diff)
estimates_df_n50[3,3] <- quantile(m3_n50_diff, 0.975)
estimates_df_n50[3,4] <- quantile(m3_n50_diff, 0.025)

#Model_4_n50
m4_n50 <- update(MLU_model, newdata = s_df_n50$data[4])
m4_n50_draws <- as_draws_df(m4_n50)

m4_n50_diff <- m4_n50_draws$`b_DiagnosisTD:Visit` - m4_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[4,2] <- mean(m4_n50_diff)
estimates_df_n50[4,3] <- quantile(m4_n50_diff, 0.975)
estimates_df_n50[4,4] <- quantile(m4_n50_diff, 0.025)

#Model_5_n50
m5_n50 <- update(MLU_model, newdata = s_df_n50$data[5])
m5_n50_draws <- as_draws_df(m5_n50)

m5_n50_diff <- m5_n50_draws$`b_DiagnosisTD:Visit` - m5_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[5,2] <- mean(m5_n50_diff)
estimates_df_n50[5,3] <- quantile(m5_n50_diff, 0.975)
estimates_df_n50[5,4] <- quantile(m5_n50_diff, 0.025)

#Model_6_n50
m6_n50 <- update(MLU_model, newdata = s_df_n50$data[6])
m6_n50_draws <- as_draws_df(m6_n50)

m6_n50_diff <- m6_n50_draws$`b_DiagnosisTD:Visit` - m6_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[6,2] <- mean(m6_n50_diff)
estimates_df_n50[6,3] <- quantile(m6_n50_diff, 0.975)
estimates_df_n50[6,4] <- quantile(m6_n50_diff, 0.025)

#Model_7_n50
m7_n50 <- update(MLU_model, newdata = s_df_n50$data[7])
m7_n50_draws <- as_draws_df(m7_n50)

m7_n50_diff <- m7_n50_draws$`b_DiagnosisTD:Visit` - m7_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[7,2] <- mean(m7_n50_diff)
estimates_df_n50[7,3] <- quantile(m7_n50_diff, 0.975)
estimates_df_n50[7,4] <- quantile(m7_n50_diff, 0.025)

#Model_8_n50
```

```
m8_n50 <- update(MLU_model, newdata = s_df_n50$data[8])
m8_n50_draws <- as_draws_df(m8_n50)

m8_n50_diff <- m8_n50_draws$`b_DiagnosisTD:Visit` - m8_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[8,2] <- mean(m8_n50_diff)
estimates_df_n50[8,3] <- quantile(m8_n50_diff, 0.975)
estimates_df_n50[8,4] <- quantile(m8_n50_diff, 0.025)

#Model_9_n50
m9_n50 <- update(MLU_model, newdata = s_df_n50$data[9])
m9_n50_draws <- as_draws_df(m9_n50)

m9_n50_diff <- m9_n50_draws$`b_DiagnosisTD:Visit` - m9_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[9,2] <- mean(m9_n50_diff)
estimates_df_n50[9,3] <- quantile(m9_n50_diff, 0.975)
estimates_df_n50[9,4] <- quantile(m9_n50_diff, 0.025)

#Model_10_n50
m10_n50 <- update(MLU_model, newdata = s_df_n50$data[10])
m10_n50_draws <- as_draws_df(m10_n50)

m10_n50_diff <- m10_n50_draws$`b_DiagnosisTD:Visit` - m10_n50_draws$`b_DiagnosisASD:Visit`
estimates_df_n50[10,2] <- mean(m10_n50_diff)
estimates_df_n50[10,3] <- quantile(m10_n50_diff, 0.975)
estimates_df_n50[10,4] <- quantile(m10_n50_diff, 0.025)

#Plot of difference

power_plot_n50 <- estimates_df_n50 %>%
  ggplot(aes(x = Model, y = Mean_diff_b, ymin = Lower, ymax = Upper)) +
  geom_pointrange(fatten = 1/2) +
  geom_hline(yintercept = 0.08, color = "red") +
  labs(x = "Seed",
       y = "Difference in slope") +
  ggtitle("N = 50")  +
  scale_x_continuous(breaks=seq(0,10,by=1))  +
  ylim(-0.04, 0.2)
power_plot_n50
```

Power analysis for n = 70

```
s_df_n70 <-
  tibble(seed = 1:sim_n) %>%
  mutate(data = map(seed, sim_d, n = 70))

estimates_df_n70 <- tibble(
  Model = seq(10),
  Mean_diff_b = NA,
  Upper = NA,
  Lower = NA
)

#Model_1_n70
m1_n70 <- update(MLU_model, newdata = s_df_n70$data[1])
m1_n70_draws <- as_draws_df(m1_n70)

m1_n70_diff <- m1_n70_draws$`b_DiagnosisTD:Visit` - m1_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[1,2] <- mean(m1_n70_diff)
estimates_df_n70[1,3] <- quantile(m1_n70_diff, 0.975)
estimates_df_n70[1,4] <- quantile(m1_n70_diff, 0.025)

#Model_2_n70
m2_n70 <- update(MLU_model, newdata = s_df_n70$data[2])
m2_n70_draws <- as_draws_df(m2_n70)

m2_n70_diff <- m2_n70_draws$`b_DiagnosisTD:Visit` - m2_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[2,2] <- mean(m2_n70_diff)
estimates_df_n70[2,3] <- quantile(m2_n70_diff, 0.975)
estimates_df_n70[2,4] <- quantile(m2_n70_diff, 0.025)

#Model_3_n70
m3_n70 <- update(MLU_model, newdata = s_df_n70$data[3])
m3_n70_draws <- as_draws_df(m3_n70)

m3_n70_diff <- m3_n70_draws$`b_DiagnosisTD:Visit` - m3_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[3,2] <- mean(m3_n70_diff)
estimates_df_n70[3,3] <- quantile(m3_n70_diff, 0.975)
estimates_df_n70[3,4] <- quantile(m3_n70_diff, 0.025)
```

```
#Model_4_n70
m4_n70 <- update(MLU_model, newdata = s_df_n70$data[4])
m4_n70_draws <- as_draws_df(m4_n70)

m4_n70_diff <- m4_n70_draws$`b_DiagnosisTD:Visit` - m4_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[4,2] <- mean(m4_n70_diff)
estimates_df_n70[4,3] <- quantile(m4_n70_diff, 0.975)
estimates_df_n70[4,4] <- quantile(m4_n70_diff, 0.025)

#Model_5_n70
m5_n70 <- update(MLU_model, newdata = s_df_n70$data[5])
m5_n70_draws <- as_draws_df(m5_n70)

m5_n70_diff <- m5_n70_draws$`b_DiagnosisTD:Visit` - m5_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[5,2] <- mean(m5_n70_diff)
estimates_df_n70[5,3] <- quantile(m5_n70_diff, 0.975)
estimates_df_n70[5,4] <- quantile(m5_n70_diff, 0.025)

#Model_6_n70
m6_n70 <- update(MLU_model, newdata = s_df_n70$data[6])
m6_n70_draws <- as_draws_df(m6_n70)

m6_n70_diff <- m6_n70_draws$`b_DiagnosisTD:Visit` - m6_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[6,2] <- mean(m6_n70_diff)
estimates_df_n70[6,3] <- quantile(m6_n70_diff, 0.975)
estimates_df_n70[6,4] <- quantile(m6_n70_diff, 0.025)

#Model_7_n70
m7_n70 <- update(MLU_model, newdata = s_df_n70$data[7])
m7_n70_draws <- as_draws_df(m7_n70)

m7_n70_diff <- m7_n70_draws$`b_DiagnosisTD:Visit` - m7_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[7,2] <- mean(m7_n70_diff)
estimates_df_n70[7,3] <- quantile(m7_n70_diff, 0.975)
estimates_df_n70[7,4] <- quantile(m7_n70_diff, 0.025)

#Model_8_n70
m8_n70 <- update(MLU_model, newdata = s_df_n70$data[8])
m8_n70_draws <- as_draws_df(m8_n70)

m8_n70_diff <- m8_n70_draws$`b_DiagnosisTD:Visit` - m8_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[8,2] <- mean(m8_n70_diff)
estimates_df_n70[8,3] <- quantile(m8_n70_diff, 0.975)
estimates_df_n70[8,4] <- quantile(m8_n70_diff, 0.025)

#Model_9_n70
m9_n70 <- update(MLU_model, newdata = s_df_n70$data[9])
m9_n70_draws <- as_draws_df(m9_n70)

m9_n70_diff <- m9_n70_draws$`b_DiagnosisTD:Visit` - m9_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[9,2] <- mean(m9_n70_diff)
estimates_df_n70[9,3] <- quantile(m9_n70_diff, 0.975)
estimates_df_n70[9,4] <- quantile(m9_n70_diff, 0.025)

#Model_10_n70
m10_n70 <- update(MLU_model, newdata = s_df_n70$data[10])
m10_n70_draws <- as_draws_df(m10_n70)

m10_n70_diff <- m10_n70_draws$`b_DiagnosisTD:Visit` - m10_n70_draws$`b_DiagnosisASD:Visit`
estimates_df_n70[10,2] <- mean(m10_n70_diff)
estimates_df_n70[10,3] <- quantile(m10_n70_diff, 0.975)
estimates_df_n70[10,4] <- quantile(m10_n70_diff, 0.025)

#Plot of difference

power_plot_n70 <- estimates_df_n70 %>%
  ggplot(aes(x = Model, y = Mean_diff_b, ymin = Lower, ymax = Upper)) +
  geom_pointrange(fatten = 1/2) +
  geom_hline(yintercept = 0.08, color = "red") +
  labs(x = "Seed",
       y = "Difference in slope") +
  ggtitle("N = 70") +
  scale_x_continuous(breaks=seq(0,10,by=1))  +
  ylim(-0.04, 0.2)
power_plot_n70
```

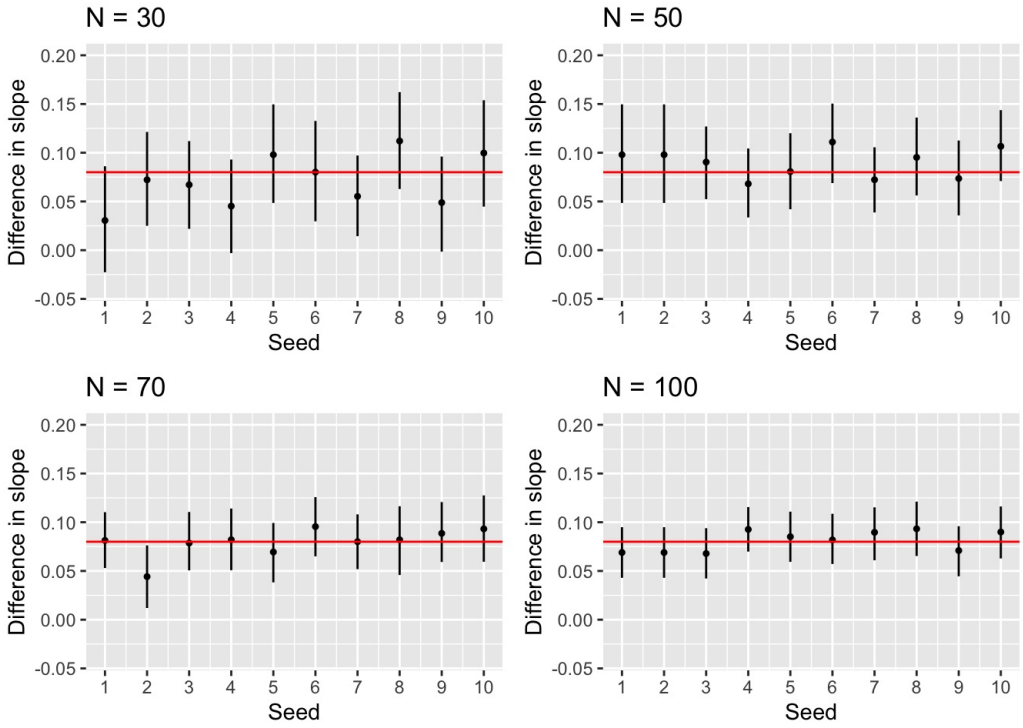
Power analysis for n = 100

```
s_df_n100 <-
  tibble(seed = 1:sim_n) %>%
  mutate(data = map(seed, sim_d, n = 100))

#Tibble that stores all values that are needed for analysis.
estimates_df_n100 <- tibble(
  Model = seq(10),
  Mean_diff_b = NA,
  Upper = NA,
  Lower = NA
)

#Model_1_n100
m1_n100 <- update(MLU_model, newdata = s_df_n100$data[1])
m1_n100_draws <- as_draws_df(m1_n100)

m1_n100_diff <- m1_n100_draws$`b_DiagnosisTD:Visit` - m1_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[1,2] <- mean(m1_n100_diff)
estimates_df_n100[1,3] <- quantile(m1_n100_diff, 0.975)
estimates_df_n100[1,4] <- quantile(m1_n100_diff, 0.025)

#Model_2_n100
m2_n100 <- update(MLU_model, newdata = s_df_n100$data[2])
m2_n100_draws <- as_draws_df(m2_n100)

m2_n100_diff <- m2_n100_draws$`b_DiagnosisTD:Visit` - m2_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[2,2] <- mean(m1_n100_diff)
estimates_df_n100[2,3] <- quantile(m1_n100_diff, 0.975)
estimates_df_n100[2,4] <- quantile(m1_n100_diff, 0.025)

#Model_3_n100
m3_n100 <- update(MLU_model, newdata = s_df_n100$data[3])
m3_n100_draws <- as_draws_df(m3_n100)

m3_n100_diff <- m3_n100_draws$`b_DiagnosisTD:Visit` - m3_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[3,2] <- mean(m3_n100_diff)
estimates_df_n100[3,3] <- quantile(m3_n100_diff, 0.975)
estimates_df_n100[3,4] <- quantile(m3_n100_diff, 0.025)

#Model_4_n100
m4_n100 <- update(MLU_model, newdata = s_df_n100$data[4])
m4_n100_draws <- as_draws_df(m4_n100)

m4_n100_diff <- m4_n100_draws$`b_DiagnosisTD:Visit` - m4_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[4,2] <- mean(m4_n100_diff)
estimates_df_n100[4,3] <- quantile(m4_n100_diff, 0.975)
estimates_df_n100[4,4] <- quantile(m4_n100_diff, 0.025)

#Model_5_n100
m5_n100 <- update(MLU_model, newdata = s_df_n100$data[5])
m5_n100_draws <- as_draws_df(m5_n100)

m5_n100_diff <- m5_n100_draws$`b_DiagnosisTD:Visit` - m5_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[5,2] <- mean(m5_n100_diff)
estimates_df_n100[5,3] <- quantile(m5_n100_diff, 0.975)
estimates_df_n100[5,4] <- quantile(m5_n100_diff, 0.025)

#Model_6_n100
m6_n100 <- update(MLU_model, newdata = s_df_n100$data[6])
m6_n100_draws <- as_draws_df(m6_n100)

m6_n100_diff <- m6_n100_draws$`b_DiagnosisTD:Visit` - m6_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[6,2] <- mean(m6_n100_diff)
estimates_df_n100[6,3] <- quantile(m6_n100_diff, 0.975)
estimates_df_n100[6,4] <- quantile(m6_n100_diff, 0.025)

#Model_7_n100
m7_n100 <- update(MLU_model, newdata = s_df_n100$data[7])
m7_n100_draws <- as_draws_df(m7_n100)

m7_n100_diff <- m7_n100_draws$`b_DiagnosisTD:Visit` - m7_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[7,2] <- mean(m7_n100_diff)
estimates_df_n100[7,3] <- quantile(m7_n100_diff, 0.975)
estimates_df_n100[7,4] <- quantile(m7_n100_diff, 0.025)

#Model_8_n100
m8_n100 <- update(MLU_model, newdata = s_df_n100$data[8])
m8_n100_draws <- as_draws_df(m8_n100)
```

```
m8_n100_diff <- m8_n100_draws$`b_DiagnosisTD:Visit` - m8_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[8,2] <- mean(m8_n100_diff)
estimates_df_n100[8,3] <- quantile(m8_n100_diff, 0.975)
estimates_df_n100[8,4] <- quantile(m8_n100_diff, 0.025)

#Model_9_n100
m9_n100 <- update(MLU_model, newdata = s_df_n100$data[9])
m9_n100_draws <- as_draws_df(m9_n100)

m9_n100_diff <- m9_n100_draws$`b_DiagnosisTD:Visit` - m9_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[9,2] <- mean(m9_n100_diff)
estimates_df_n100[9,3] <- quantile(m9_n100_diff, 0.975)
estimates_df_n100[9,4] <- quantile(m9_n100_diff, 0.025)

#Model_10_n100
m10_n100 <- update(MLU_model, newdata = s_df_n100$data[10])
m10_n100_draws <- as_draws_df(m10_n100)

m10_n100_diff <- m10_n100_draws$`b_DiagnosisTD:Visit` - m10_n100_draws$`b_DiagnosisASD:Visit`
estimates_df_n100[10,2] <- mean(m10_n100_diff)
estimates_df_n100[10,3] <- quantile(m10_n100_diff, 0.975)
estimates_df_n100[10,4] <- quantile(m10_n100_diff, 0.025)

#Plot of difference

power_plot_n100 <- estimates_df_n100 %>%
  ggplot(aes(x = Model, y = Mean_diff_b, ymin = Lower, ymax = Upper)) +
  geom_pointrange(fatten = 1/2) +
  geom_hline(yintercept = 0.08, color = "red") +
  labs(x = "Seed",
       y = "Difference in slope") +
  ggtitle("N = 100")  +
  scale_x_continuous(breaks=seq(0,10,by=1))  +
  ylim(-0.04, 0.2)
power_plot_n100
```

Plotting all at once:

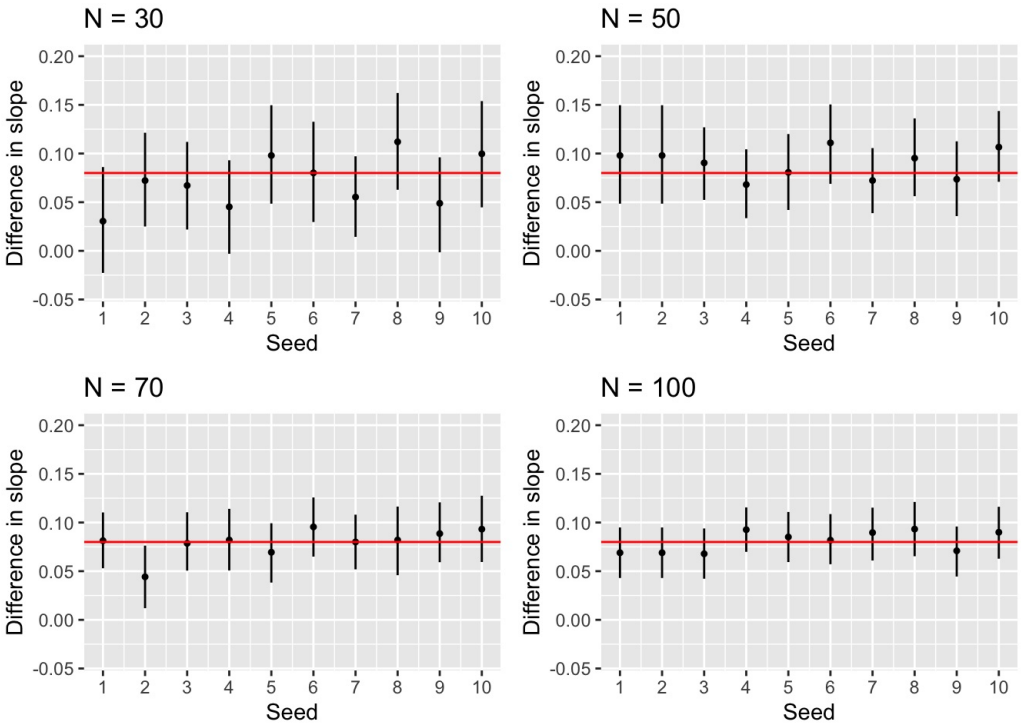
```
power_plots <- grid.arrange(power_plot_n30, power_plot_n50, power_plot_n70, power_plot_n100)
```




```
ggsave("Power_analysis.jpeg", plot = power_plots, path = "/Users/justina/Desktop/Desktop - Justina's MacBook Pro/
Aarhus_Uni/Semester_3/Methods 3/Assignment-1")
```

```
## Saving 7 x 5 in image
```

```
grid.arrange(power_plot_n30, power_plot_n50, power_plot_n70, power_plot_n100)
```



Part 2 - Strong in the Bayesian ken, you are now ready to analyse the actual data

Q1: Describe your sample (n, age, gender, clinical and cognitive features of the two groups) and critically assess whether the groups (ASD and TD) are balanced. Briefly discuss whether the data is enough given the simulations in part 1. Q2: Describe linguistic development (in terms of MLU over time) in TD and ASD children (as a function of group). Discuss the difference (if any) between the two groups. Q3: Describe individual differences in linguistic development: do all kids follow the same path? Are all kids reflected by the general trend for their group?

- Include additional predictors in your model of language development (N.B. not other indexes of child language: types and tokens, that'd be cheating). Identify the best model, by conceptual reasoning, model comparison or a mix. Report the model you choose (and name its competitors, if any) and discuss why it's the best model.

```
#Loading the data:
data_clean <- read_csv("data_clean.csv")
```

```
## Rows: 372 Columns: 22
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): Ethnicity, Diagnosis, Gender
## dbl (19): Child.ID, Visit, Age, ADOS, MullenRaw, ExpressiveLangRaw, Socializ...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
colnames(data_clean)[1] = "ID" #Renamed the first column.

#Filter out the data that has no MLU value
data_clean <- data_clean %>%
  filter(CHI_MLU != 0)
```

Describing the sample

```
#Sample size
data_clean$ID <- as.factor(data_clean$ID)
length(levels(data_clean$ID)) #n = 61
```

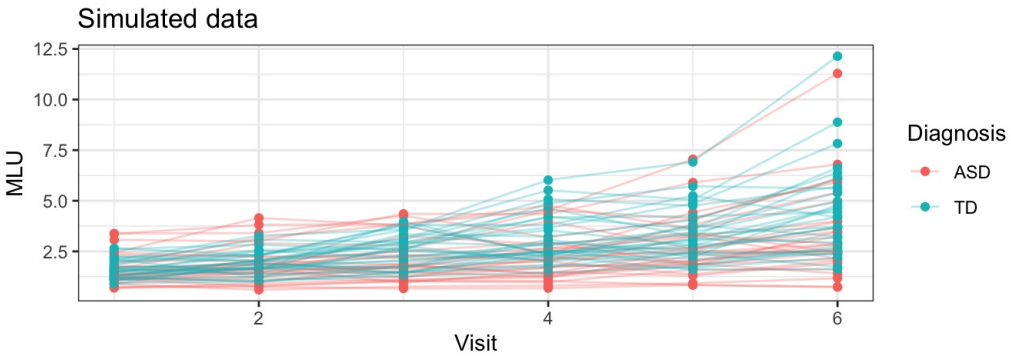
```
## [1] 61
```

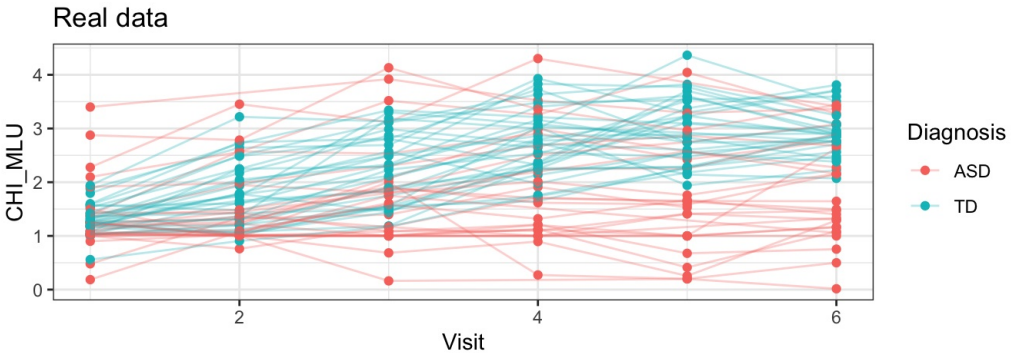
```
data_clean$Diagnosis <- as.factor(data_clean$Diagnosis) #Changed all variables that were needed (ID, Gender, Diag
nosis...)

#colnames(data_clean)

vis_2 <- subset(data_clean, Visit == 6) #Adjusted for each group/visit the same code.
asd_2 <- vis_2 %>%
  filter(Diagnosis == "TD")
summary(asd_2)
```

```
##        ID         Visit    Ethnicity         Diagnosis    Gender
##  2      : 1   Min.   :6   Length:28          ASD: 0    Length:28
##  4      : 1   1st Qu.:6   Class :character   TD :28    Class :character
##  9      : 1   Median :6   Mode  :character             Mode  :character
##  11     : 1   Mean   :6
##  13     : 1   3rd Qu.:6
##  14     : 1   Max.   :6
##  (Other):22
##       Age             ADOS       MullenRaw     ExpressiveLangRaw
##  Min.   :38.53   Min.   : NA   Min.   :33.00   Min.   :27
##  1st Qu.:39.80   1st Qu.: NA   1st Qu.:43.00   1st Qu.:36
##  Median :40.40   Median : NA   Median :45.00   Median :40
##  Mean   :41.18   Mean   :NaN   Mean   :44.11   Mean   :40
##  3rd Qu.:42.59   3rd Qu.: NA   3rd Qu.:46.00   3rd Qu.:45
##  Max.   :45.07   Max.   : NA   Max.   :50.00   Max.   :50
##                  NA's   :28    NA's   :1       NA's   :1
##  Socialization       MOT_MLU        MOT_LUstd        CHI_MLU
##  Min.   : 83.00   Min.   :3.392   Min.   :2.372   Min.   :2.072
##  1st Qu.: 97.00   1st Qu.:4.168   1st Qu.:2.541   1st Qu.:2.744
##  Median :101.00   Median :4.318   Median :2.590   Median :2.881
##  Mean   : 99.96   Mean   :4.395   Mean   :2.633   Mean   :2.928
##  3rd Qu.:103.00   3rd Qu.:4.664   3rd Qu.:2.733   3rd Qu.:3.077
##  Max.   :116.00   Max.   :5.587   Max.   :3.014   Max.   :3.811
##  NA's   :1
##    CHI_LUstd       types_MOT       types_CHI       tokens_MOT
##  Min.   :1.785   Min.   :249.0   Min.   : 34.0   Min.   :1024
##  1st Qu.:2.021   1st Qu.:336.0   1st Qu.:156.0   1st Qu.:1650
##  Median :2.187   Median :386.5   Median :175.5   Median :1878
##  Mean   :2.199   Mean   :401.7   Mean   :174.6   Mean   :1953
##  3rd Qu.:2.292   3rd Qu.:475.8   3rd Qu.:214.0   3rd Qu.:2351
##  Max.   :2.795   Max.   :595.0   Max.   :260.0   Max.   :2895
##
##    tokens_CHI         ADOS1          verbalIQ1      nonVerbalIQ1
##  Min.   :  61.0   Min.   :0.0000   Min.   :13.00   Min.   :19.00
##  1st Qu.: 431.5   1st Qu.:0.0000   1st Qu.:16.75   1st Qu.:23.75
##  Median : 625.5   Median :0.0000   Median :18.50   Median :26.50
##  Mean   : 606.6   Mean   :0.6786   Mean   :20.04   Mean   :25.89
##  3rd Qu.: 712.2   3rd Qu.:1.0000   3rd Qu.:22.00   3rd Qu.:29.00
##  Max.   :1294.0   Max.   :5.0000   Max.   :33.00   Max.   :32.00
##
##  Socialization1
##  Min.   : 86.0
##  1st Qu.: 96.0
##  Median :102.0
##  Mean   :100.5
##  3rd Qu.:104.5
##  Max.   :115.0
##
```

Simulated vs Real data

```
real_data <- ggplot(data_clean, aes(Visit, CHI_MLU, color = Diagnosis, group = ID)) +
  theme_bw() +
  geom_point() +
  geom_line(alpha = 0.3) +
  ggtitle("Real data")
grid.arrange(simulated_data, real_data)
```





Defining the formula and setting priors (the same as in the simulation above, just with broader priors)

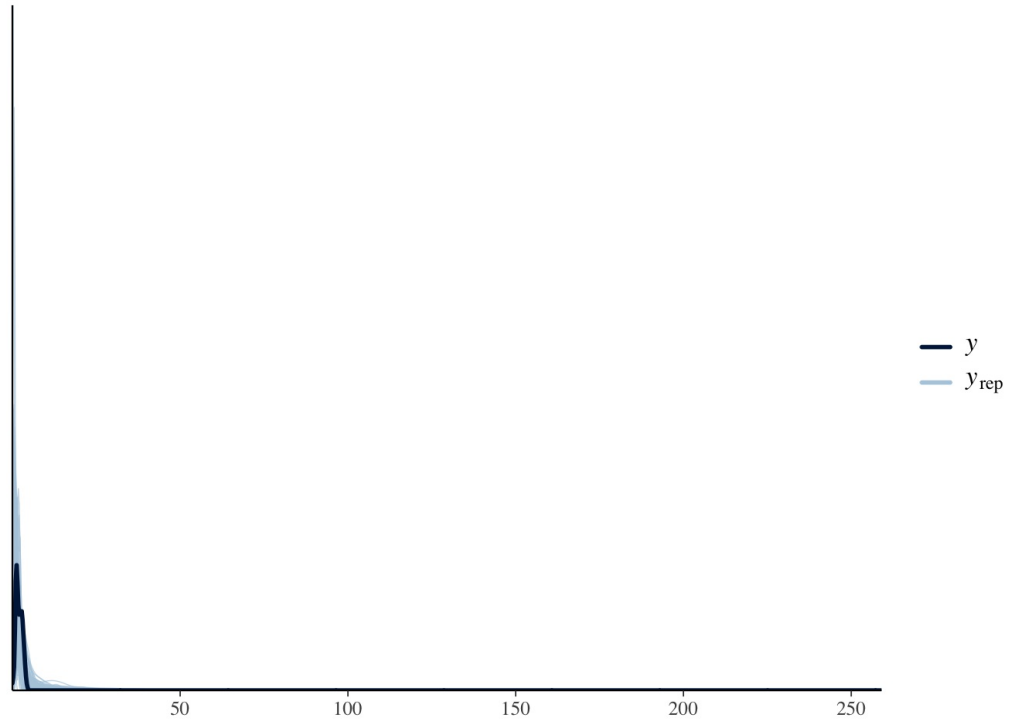
```
unWords_f <- bf(CHI_MLU ~ 0 + Diagnosis + Diagnosis:Visit + (1 + Visit|ID))

CHI_MLU_priors <- c(
  prior(normal(0.41, 0.5), class = b, coef = "DiagnosisASD"), #Just keep the same for both.
  prior(normal(0.41, 0.5), class = b, coef = "DiagnosisTD"),  #Keeping the mean and uncertainty same, based on th
e true values.
  prior(normal(0,0.2),class=b,coef="DiagnosisASD:Visit"), #Keeping the slopes the same
  prior(normal(0,0.2),class=b,coef="DiagnosisTD:Visit"),
  prior(normal(0, 0.5), class = sd, coef = Intercept, group = ID), #Took mean SD of both groups.
  prior(normal(0, 0.1), class = sd, coef = Visit, group= ID),
  prior(normal(0, 0.2), class = sigma),
  prior(lkj(2), class = cor) #Dampens extreme correlations.
)
```

Fitting the models

```
MLU_f_prior_s <-
  brm(
    unWords_f,
    data = data_clean,
    family = lognormal,
    prior = CHI_MLU_priors,
    sample_prior = "only",
    backend = "cmdstanr",
    cores = 2,
    chains = 2,
    control = list(adapt_delta = 0.99, max_treedepth = 20))
```

```
## Start sampling
```

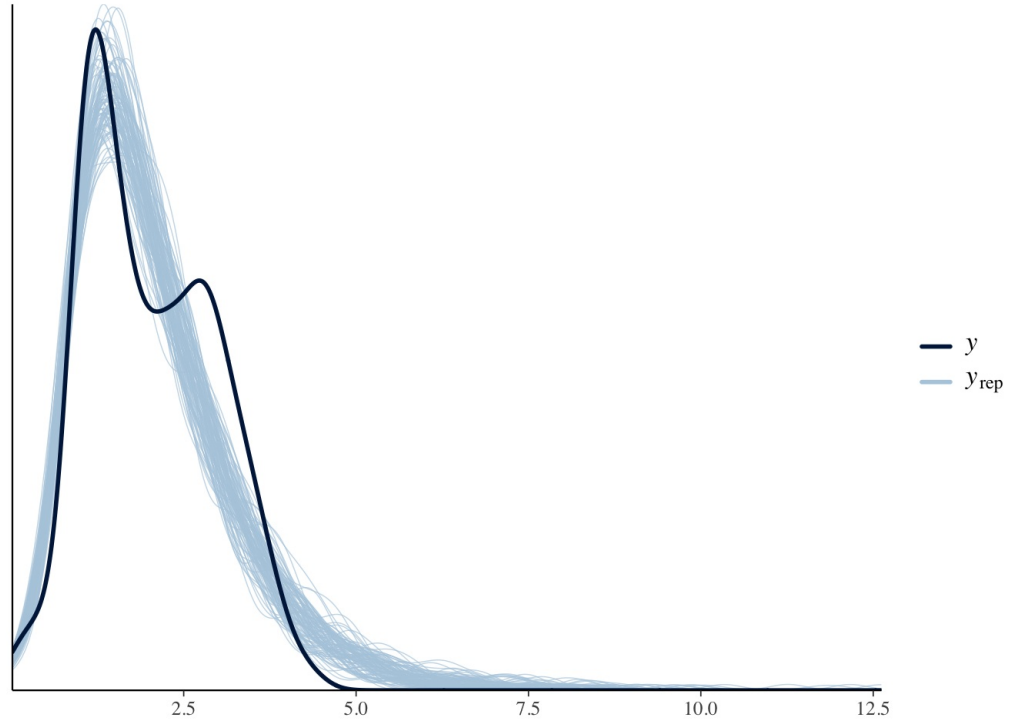
```
pp_check(MLU_f_prior_s, ndraws = 100)
```




```
MLU_f_posterior <-
  brm(
    unWords_f,
    data = data_clean,
    family = lognormal,
    prior = CHI_MLU_priors,
    sample_prior = T,
    backend = "cmdstanr",
    cores = 2,
    chains = 2,
    control = list(adapt_delta = 0.99, max_treedepth = 20))
```

```
## Start sampling
```

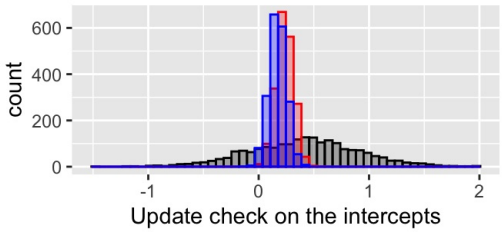
```
pp_check(MLU_f_posterior, ndraws = 100)
```

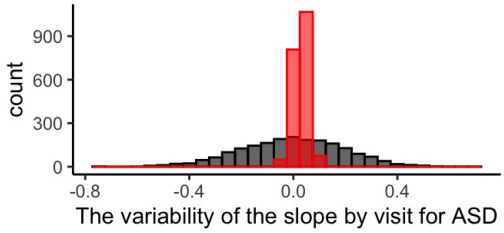


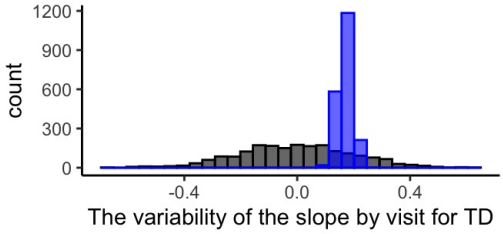
Prior-posterior update checks

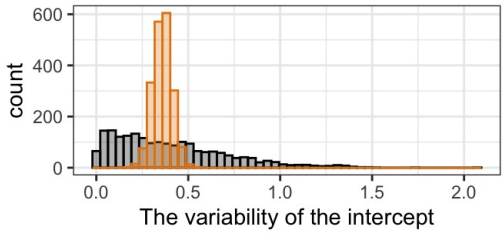
```
#variables(MLU_f_posterior)

posterior_real <- as_draws_df(MLU_f_posterior)
#---------------------------------------------------------------------------

plot_intercepts_real <- ggplot(posterior_real) +
  geom_histogram(aes(prior_b_DiagnosisASD),
                 fill = "black", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(b_DiagnosisASD),
                 fill = "red", color = "red", alpha = 0.3, bins = 50) +
    geom_histogram(aes(b_DiagnosisTD),
                 fill = "blue", color = "blue", alpha = 0.3, bins = 50)+
  xlab("Update check on the intercepts")

#---------------------------------------------------------------------------

plot_b_visit_ASD_real <- ggplot(posterior_real) +
  geom_histogram(aes(`prior_b_DiagnosisASD:Visit`), fill="black", color="black",alpha=0.6,) +
  geom_histogram(aes(`b_DiagnosisASD:Visit`), fill="red", color="red",alpha=0.6) +
  theme_classic() +
   xlab("The variability of the slope by visit for ASD")

#---------------------------------------------------------------------------

plot_b_visit_TD_real <- ggplot(posterior_real) +
  geom_histogram(aes(`prior_b_DiagnosisTD:Visit`), fill="black", color="black",alpha=0.6,) +
  geom_histogram(aes(`b_DiagnosisTD:Visit`), fill="blue", color="blue",alpha=0.6) +
  theme_classic() +
   xlab("The variability of the slope by visit for TD")

#---------------------------------------------------------------------------

plot_sd_intercept_real <- ggplot(posterior_real) +
  geom_histogram(aes(prior_sd_ID__Intercept),
                 fill = "black", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(sd_ID__Intercept),
                 fill = "#E68613", color = "#E68613", alpha = 0.3, bins = 50) +
  theme_bw() +
  xlab("The variability of the intercept")

#---------------------------------------------------------------------------

plot_sd_visit_real <- ggplot(posterior_real) +
  geom_histogram(aes(prior_sd_ID__Visit), fill="black", color="black",alpha=0.6,) +
  geom_histogram(aes(sd_ID__Visit), fill="#E68613", color="#E68613",alpha=0.6) +
  theme_classic() +
   xlab("The variability of the slope")

#---------------------------------------------------------------------------

#Prior-posterior update checks:
grid.arrange(plot_intercepts_real, plot_b_visit_ASD_real, plot_b_visit_TD_real, plot_sd_intercept_real, plot_sd_v
isit_real)
```

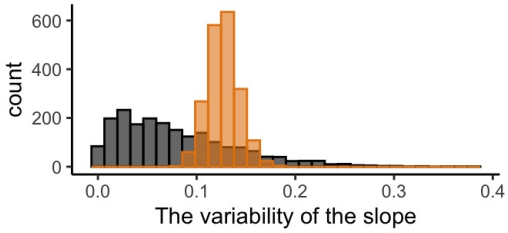
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```











Inspecting the parameters

```
#summary(MLU_f_posterior)
```

Hypothesis testing: does the development rate for TD compared to ASD is significant?

```
hypothesis(MLU_f_posterior, "DiagnosisTD:Visit > DiagnosisASD:Visit") #taking the distributions of the slope for TD and ASD
```

```
## Hypothesis Tests for class b:
##                 Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio
## 1 (DiagnosisTD:Visi... > 0     0.14      0.04     0.09     0.21        Inf
##   Post.Prob Star
## 1         1    *
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

The average difference is 0.14, with standard error of 0.04. Looking at the CI.Lower and CI.Upper it is clear that the distribution is not normal, as it does not include 0. Also, it indicates that the probability of TD developing faster than the ASD group is 100%. (Which might seem a bit too confident?)

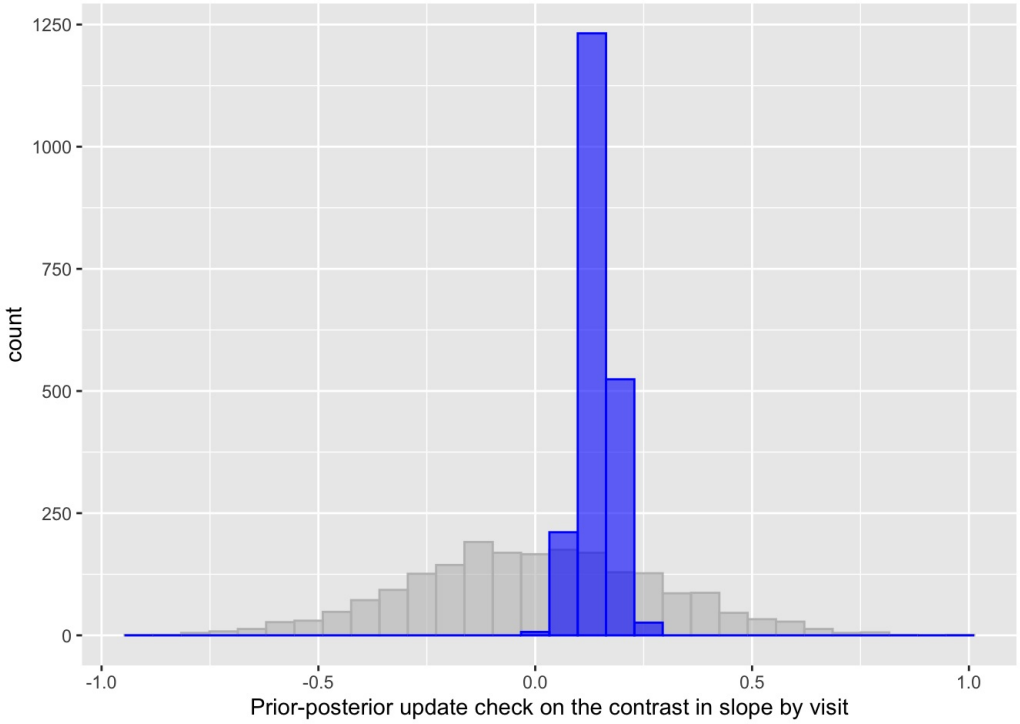
Plotting the prior distribution of differences:

```
#Prior samples:

#Sampling from the prior of ASD
samples_asd <- prior_draws(MLU_f_posterior, "b_DiagnosisASD:Visit")
samples_asd <- as.tibble(samples_asd) #data frame that contains prior draws
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## ℹ Please use `as_tibble()` instead.
## ℹ The signature and semantics have changed, see `?as_tibble`.
```

```
#Sampling from the prior of TD
samples_td <- prior_draws(MLU_f_posterior, "b_DiagnosisTD:Visit")
samples_td <- tibble(samples_td) #data frame that contains prior draws

#Calculating the difference of prior slopes between both groups and saving it as a tibble.
prior_dis_of_diff <- samples_td$`b_DiagnosisTD:Visit` - samples_asd$`b_DiagnosisASD:Visit`
prior_dis_of_diff <- tibble(Difference = prior_dis_of_diff)

#--------------------------------------------------------------------------------

posterior_real <- as_draws_df(MLU_f_posterior)

#Plotting the slope values
prior_dis_of_diff_plot <- ggplot() +
  geom_histogram(data = prior_dis_of_diff, aes(x = Difference), color = "grey", fill = "grey", alpha = 0.6) +
  geom_histogram(data = posterior_real, aes(x = `b_DiagnosisTD:Visit` - `b_DiagnosisASD:Visit`), color = "blue",
fill = "blue", alpha = 0.6) +
  labs(x =  "Prior-posterior update check on the contrast in slope by visit")

prior_dis_of_diff_plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#Posterior distribution of the intercept for TD, same for ASD, posterior distribution per each visit of the slope
.
#variables(MLU_f_posterior)
```

Plotting individual level estimates

```
temp_re_real <- ranef(MLU_f_posterior)$ID
for (i in unique(data_clean$ID)) {
  temp <- as.character(i)
  data_clean$EstimatedIntercept[data_clean$ID == i] <- temp_re_real[,,"Intercept"][temp,1] #Estimate
  data_clean$EstimatedIntercept_low[data_clean$ID == i] <- temp_re_real[,,"Intercept"][temp,3] #Q2.5
  data_clean$EstimatedIntercept_high[data_clean$ID == i] <- temp_re_real[,,"Intercept"][temp,4] #Q97.5
  data_clean$EstimatedSlope[data_clean$ID == i] <- temp_re_real[,,"Visit"][temp,1]
  data_clean$EstimatedSlope_low[data_clean$ID == i] <- temp_re_real[,,"Visit"][temp,3]
  data_clean$EstimatedSlope_high[data_clean$ID == i] <- temp_re_real[,,"Visit"][temp,4]
}
```

```
## Warning: Unknown or uninitialised column: `EstimatedIntercept`.
```

```
## Warning: Unknown or uninitialised column: `EstimatedIntercept_low`.
```

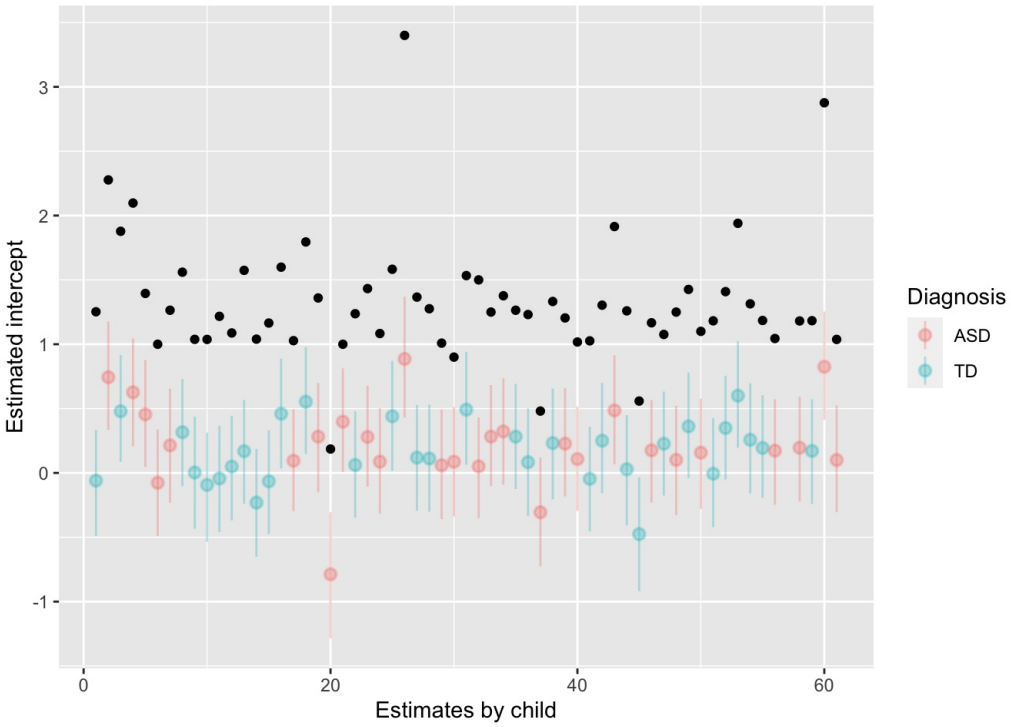
```
## Warning: Unknown or uninitialised column: `EstimatedIntercept_high`.
```

```
## Warning: Unknown or uninitialised column: `EstimatedSlope`.
```

```
## Warning: Unknown or uninitialised column: `EstimatedSlope_low`.
```

```
## Warning: Unknown or uninitialised column: `EstimatedSlope_high`.
```

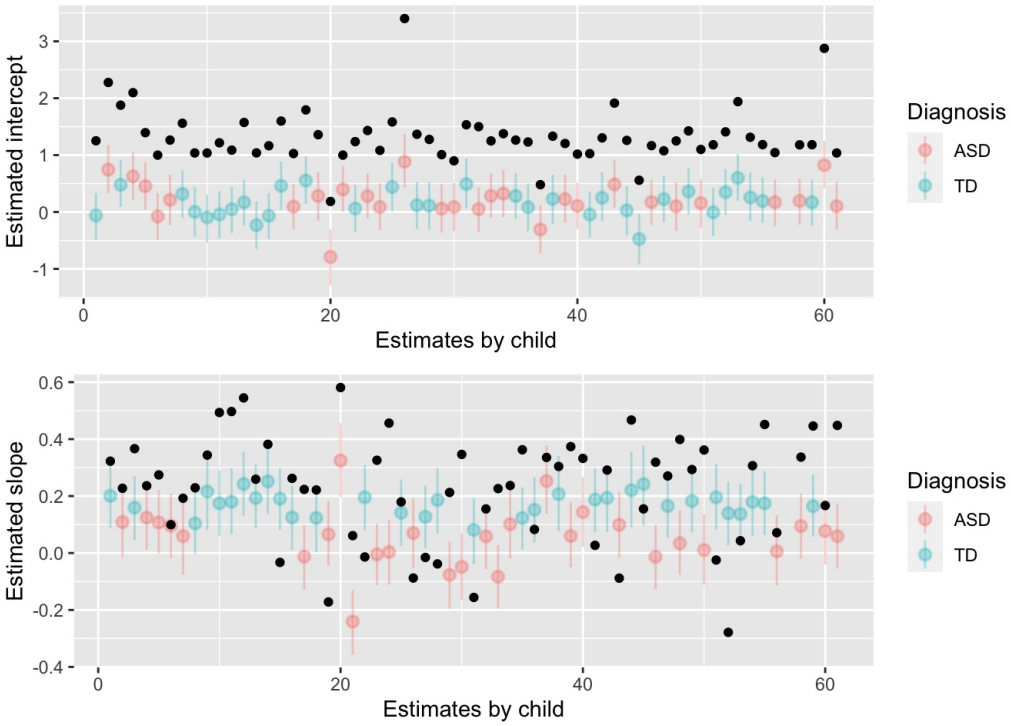
```
df_est1_real <- data_clean %>% subset(Visit==1) %>%
  mutate(
    EstimatedIntercept = ifelse(Diagnosis =="ASD",
                                EstimatedIntercept + 0.24, #Estimate for DiagnosisASD
                                EstimatedIntercept + 0.17), #Estimate for DiagnosisTD
    EstimatedIntercept_low = ifelse(Diagnosis=="ASD",
                                EstimatedIntercept_low + 0.24,
                                EstimatedIntercept_low + 0.17),
    EstimatedIntercept_high = ifelse(Diagnosis=="ASD",
                                EstimatedIntercept_high + 0.24,
                                EstimatedIntercept_high + 0.17),


    EstimatedSlope = ifelse(Diagnosis=="ASD",
                                EstimatedSlope + 0.03, #Estimate for DiagnosisASD:Visit
                                EstimatedSlope + 0.17),#Estimate for DiagnosisTD:Visit
    EstimatedSlope_low = ifelse(Diagnosis=="ASD",
                                EstimatedSlope_low + 0.03,
                                EstimatedSlope_low + 0.17),
    EstimatedSlope_high = ifelse(Diagnosis=="ASD",
                                EstimatedSlope_high + 0.03,
                                EstimatedSlope_high + 0.17)

  )
#--------------------------------------------------------------------------------
#Extracting the MLU values of the child at visit 1:
Chi_MLU_v1 <- data_clean%>%
  filter(Visit == 1
         )
Chi_MLU_values <- Chi_MLU_v1$CHI_MLU

#Plotting estimates intercepts vs MLU values at visit 1
Est_intercept_real <- ggplot(df_est1_real)+
  geom_pointrange(aes(x=as.numeric(as.factor(ID)),y=EstimatedIntercept,
                      ymin=EstimatedIntercept_low,ymax=EstimatedIntercept_high,
                      color = Diagnosis),alpha=0.3) +
  geom_point(aes(x=as.numeric(as.factor(ID)), y= Chi_MLU_values))+
  xlab("Estimates by child")+
  ylab("Estimated intercept")
Est_intercept_real
```



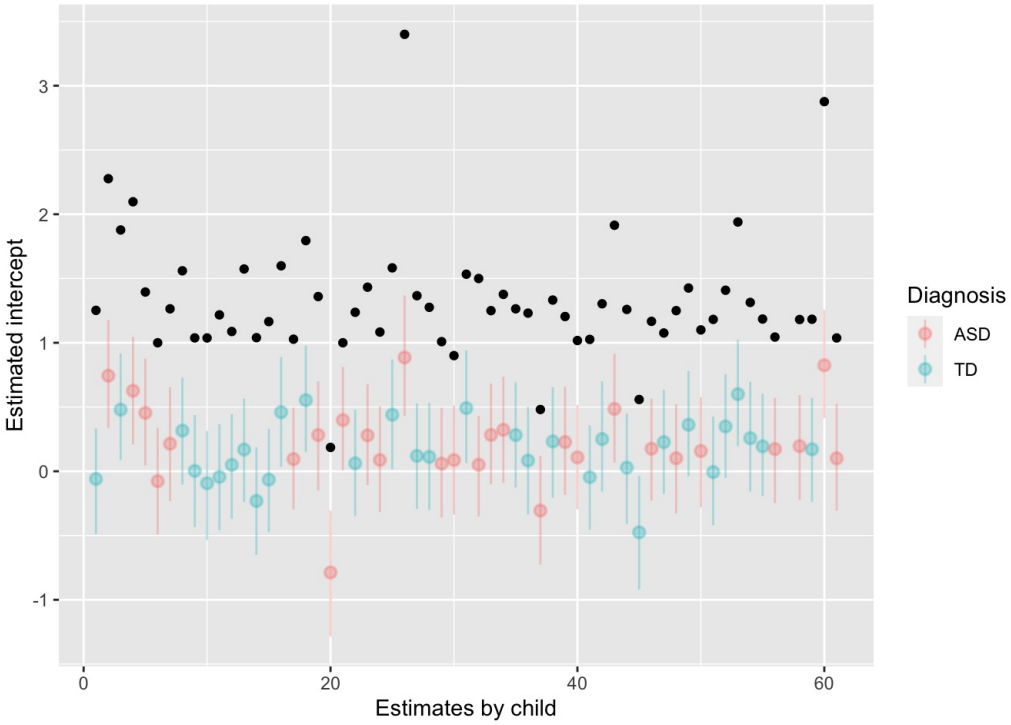
```
#-------------------------------------------------------------------------------
#Estimating the slope for the children from real data:
mlu_v6 <- data_clean%>%
  filter(Visit == 6)
Chi_MLU_values_v6 <- mlu_v6$CHI_MLU
diff_slope_real <- (Chi_MLU_values_v6-Chi_MLU_values)/5
```

```
## Warning in Chi_MLU_values_v6 - Chi_MLU_values: longer object length is not a
## multiple of shorter object length
```

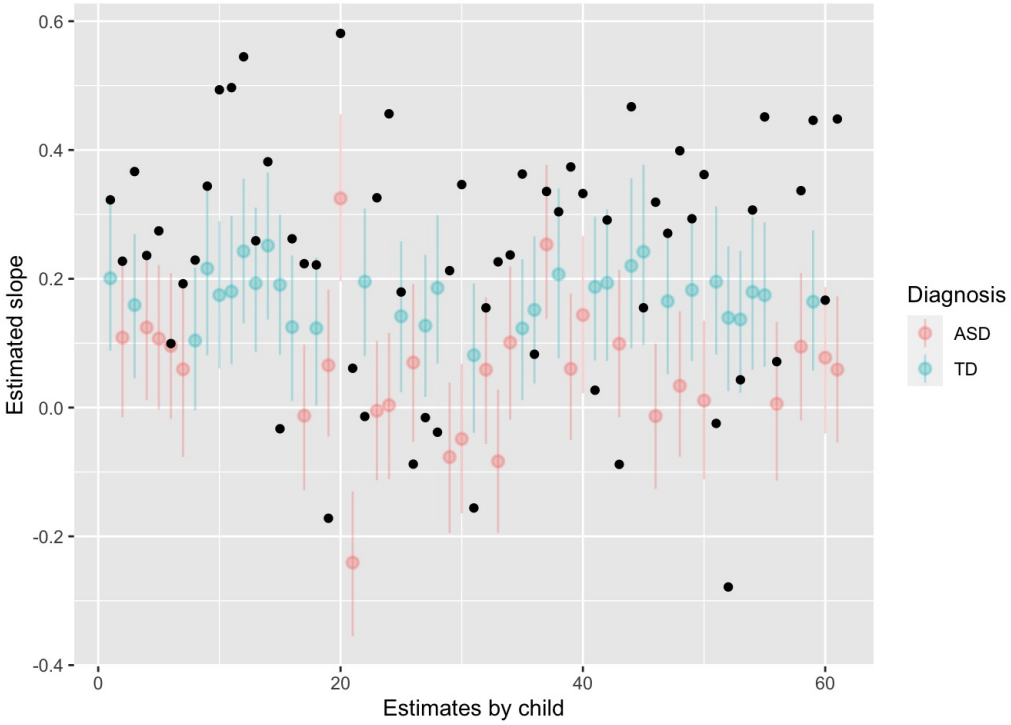
```
#Plotting the estimated slopes vs the average difference at any 2 visits (of MLU)
Est_slope_real <- ggplot(df_est1_real)+
  geom_pointrange(aes(x=as.numeric(as.factor(ID)),y=EstimatedSlope,
                      ymin=EstimatedSlope_low,ymax=EstimatedSlope_high,
                      color = Diagnosis),alpha=0.3) +
  geom_point(aes(x=as.numeric(as.factor(ID)),y = diff_slope_real))+
  xlab("Estimates by child")+
  ylab("Estimated slope")

grid.arrange(Est_intercept_real, Est_slope_real)
```




```
Est_intercept_real
```




```
Est_slope_real
```



MOT_MLU as an additional factor:

```
addition_f1 <- bf(CHI_MLU ~ 0 + Diagnosis + Diagnosis:Visit + Diagnosis:MOT_MLU + (1 + Visit|ID))
get_prior(addition_f1, data_clean, lognormal)

#Kept the same priors, + new ones.
MLU_add_1_p <- c(
  prior(normal(0.41, 0.5), class = b, coef = "DiagnosisASD"),
  prior(normal(0.41, 0.5), class = b, coef = "DiagnosisTD"),
  prior(normal(0,0.2),class=b,coef="DiagnosisASD:Visit"),
  prior(normal(0,0.2),class=b,coef="DiagnosisTD:Visit"),
  prior(normal(0, 0.3), class = b, coef = "DiagnosisASD:MOT_MLU"),
  prior(normal(0, 0.3), class = b, coef = "DiagnosisTD:MOT_MLU"),
  prior(normal(0, 0.5), class = sd, coef = Intercept, group = ID),
  prior(normal(0, 0.1), class = sd, coef = Visit, group= ID),
  prior(normal(0, 0.2), class = sigma),
  prior(lkj(2), class = cor)
)

m_1_add <-
  brm(
    addition_f1,
    data = data_clean,
    family = lognormal,
    prior = MLU_add_1_p,
    sample_prior = T,
    backend = "cmdstanr",
    cores = 2,
    chains = 2,
    control = list(adapt_delta = 0.99, max_treedepth = 20))
#update(m_1_add)

#Information criteria
m_1_add <- add_criterion(m_1_add, criterion = "loo")

data_clean$looic <- m_1_add$criteria$loo$pointwise[,"looic"]
ggplot(data_clean, aes(x = ID , y = looic, color = Diagnosis)) + geom_point() + theme_bw()

MLU_f_posterior <- add_criterion(MLU_f_posterior, criterion = "loo")
data_clean$looic_MLU_post <- MLU_f_posterior$criteria$loo$pointwise[,"looic_MLU_post"]
ggplot(data_clean, aes(x = ID , y = looic_MLU_post, color = Diagnosis)) + geom_point() + theme_bw()

loo_compare(MLU_f_posterior, m_1_add)
loo_model_weights(MLU_f_posterior, m_1_add)
```