Assignment 2 (A2): Language development in autistic and neurotypical children

Instructions

Ingrid Backman

```
data <- read_csv("data/data_clean.csv") %>%
  mutate(
    Gender = factor(Gender),
    Child.ID = factor(Child.ID),
    Diagnosis = factor(Diagnosis, levels = c("TD", "ASD"))
  )%>%
  rename(NonVerbal_IQ = MullenRaw, Verbal_IQ = ExpressiveLangRaw)
```

```
## 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.
```

```
sapply(data, class)
```

Intro

Autism Spectrum Disorder is often related to language impairment. However, this phenomenon has rarely been empirically traced in detail:

1. relying on actual naturalistic language production
2. over extended periods of time.

Around 30 kids with ASD and 30 typically developing kids were videotaped (matched by linguistic performance at visit 1) for ca. 30 minutes of naturalistic interactions with a parent. Data collection was repeated 6 times per kid, with 4 months between each visit. Following transcription of the data, the following quantities were computed:

1. the amount of words that each kid uses in each video. Same for the parent
2. the amount of unique words that each kid uses in each video. Same for the parent
3. .the amount of morphemes per utterance (Mean Length of Utterance) displayed by each child in each video. Same for the parent.

This data is in the file you prepared in the previous class, but you can also find it here (https://www.dropbox.com/s/d6eerv6cl6eksf3/data_clean.csv?dl=0)

Assignment structure

We will be spending a few weeks with this assignment. In particular, we will:

1. build our model, analyze our empirical data, and interpret the inferential results
2. use your model to predict the linguistic trajectory of new children and assess the performance of the model based on that.

As you work through these parts, you will have to produce a written document (separated from the code) answering the following questions:

1. Briefly describe the empirical data, your model(s) and their 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.

2. Given the model(s) from Q2, how well do they predict the data? Discuss both in terms of absolute error in training vs testing; and in terms of characterizing the new kids' language development as typical or in need of support.

Below you can find more detailed instructions for each part of the assignment.

In working through this part of the assignment, keep in mind the following workflow:

1. Formula definition
2. Prior definition
3. Prior predictive checking
4. Model fitting
5. Model quality checks
6. Model comparison

Analysis

Describe your sample (n, age, gender, clinical and cognitive features of the two groups) using plots and critically assess whether the groups (ASD and TD) are balanced.

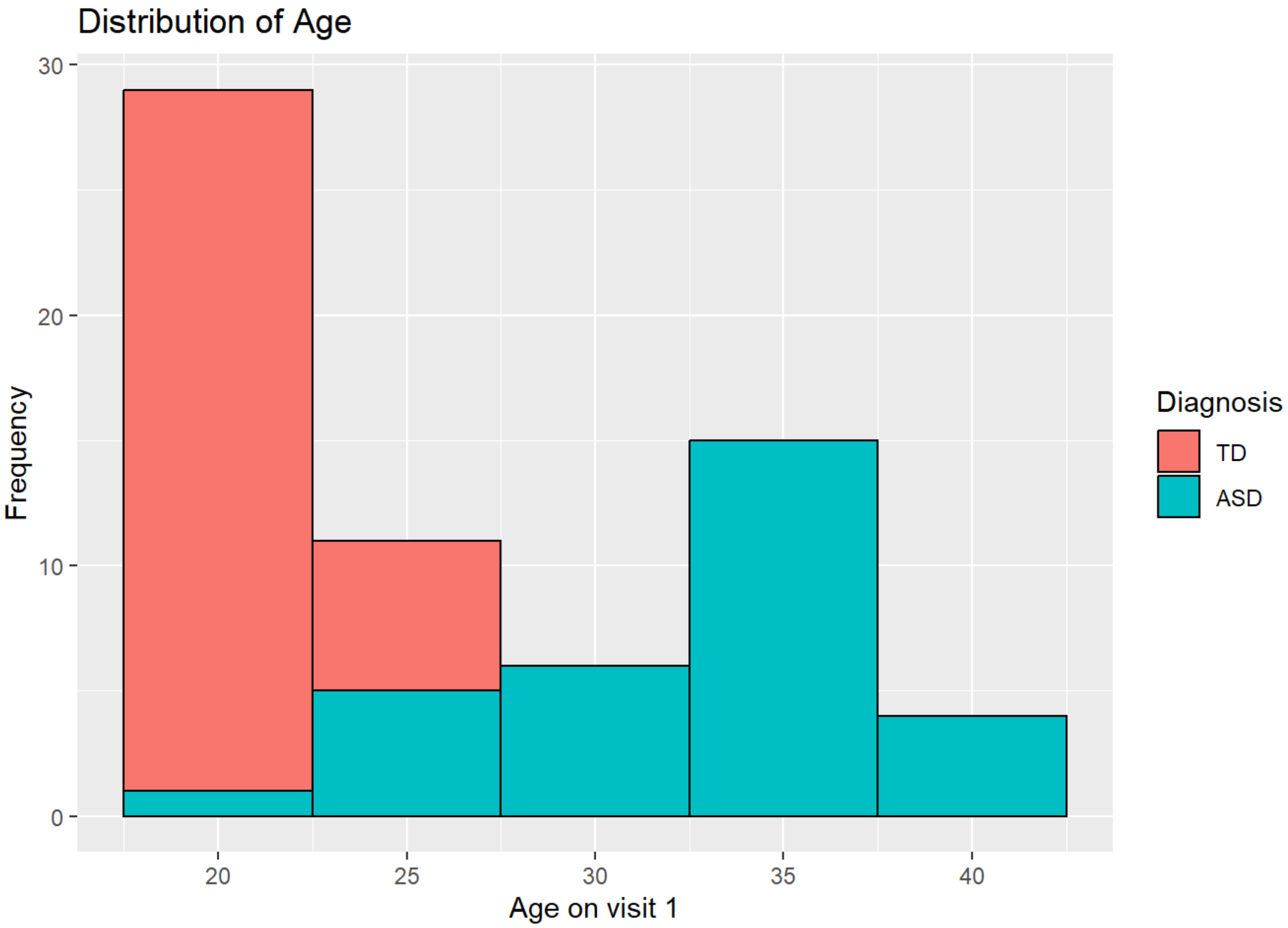
I am using the dataset provided (not the one cleaned in part 1). I looked at the 1st visit to ascertain whether the sample groups were balanced.

```
first_visit <- data %>%
  filter(Visit == 1)
```

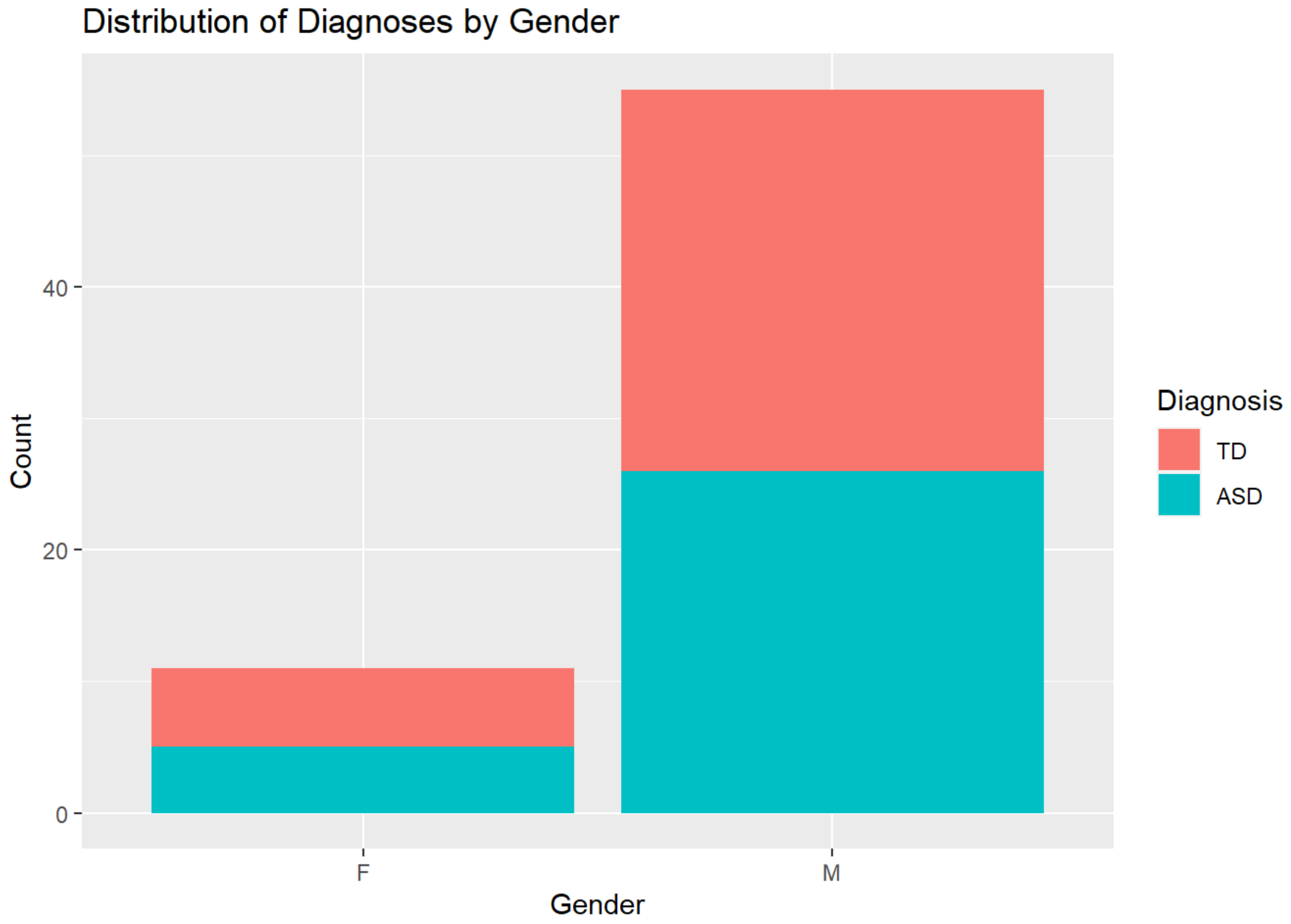
```
first_visit %>%
  group_by(Diagnosis) %>%
  summarize(Unique_Diagnoses = n_distinct(Child.ID))
```

```
## # A tibble: 2 × 2
##   Diagnosis Unique_Diagnoses
##   <fct>                <int>
## 1 TD                      35
## 2 ASD                     31
```

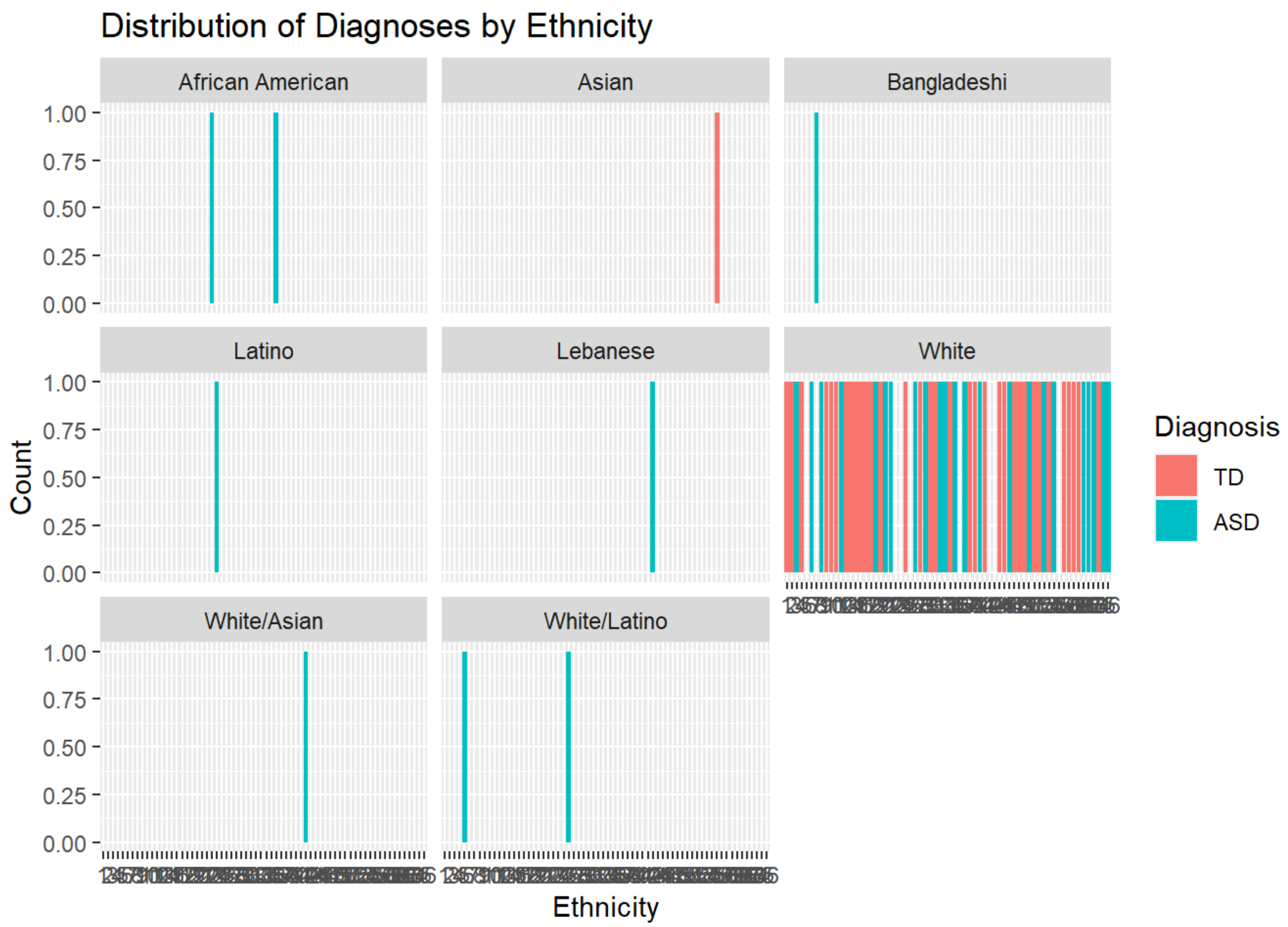
```
ggplot(first_visit, aes(x = Age, fill = Diagnosis)) +
  geom_histogram(binwidth = 5, color = "black") +
  labs(title = "Distribution of Age", x = "Age on visit 1", y = "Frequency") +
  guides(fill=guide_legend(title="Diagnosis"))
```



```
ggplot(first_visit, aes(x = Gender, fill = Diagnosis)) +
  geom_bar() +
  labs(title = "Distribution of Diagnoses by Gender", x = "Gender", y = "Count")
```



```
ggplot(first_visit, aes(x = Child.ID, fill = Diagnosis)) +
  geom_bar() +
  labs(title = "Distribution of Diagnoses by Ethnicity", x = "Ethnicity", y = "Count")+
  facet_wrap(~Ethnicity)
```



```
first_visit %>%
  group_by(Diagnosis, Ethnicity) %>%
  summarize(unique_ethnicity_count = n_distinct(Child.ID))
```

```
## `summarise()` has grouped output by 'Diagnosis'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 9 × 3
## # Groups:   Diagnosis [2]
##   Diagnosis Ethnicity        unique_ethnicity_count
##   <fct>     <chr>                             <int>
## 1 TD        Asian                                 1
## 2 TD        White                                34
## 3 ASD       African American                      2
## 4 ASD       Bangladeshi                           1
## 5 ASD       Latino                                1
## 6 ASD       Lebanese                              1
## 7 ASD       White                                23
## 8 ASD       White/Asian                           1
## 9 ASD       White/Latino                          2
```

```
first_visit %>%
  group_by(Ethnicity)%>%
  count(Ethnicity)
```

```
## # A tibble: 8 × 2
## # Groups:   Ethnicity [8]
##   Ethnicity            n
##   <chr>            <int>
## 1 African American     2
## 2 Asian                1
## 3 Bangladeshi          1
## 4 Latino               1
## 5 Lebanese             1
## 6 White               57
## 7 White/Asian          1
## 8 White/Latino         2
```

```
first_visit %>%
  group_by(Gender, Diagnosis) %>%
  summarize(Count = n()) %>%
  pivot_wider(names_from = Gender, values_from = Count) %>%
  mutate(Ratio_Women = F / sum(F), Ratio_Men = M / sum(M))
```

```
## `summarise()` has grouped output by 'Gender'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 2 × 5
##   Diagnosis     F     M Ratio_Women Ratio_Men
##   <fct>     <int> <int>       <dbl>     <dbl>
## 1 TD            6    29       0.545     0.527
## 2 ASD           5    26       0.455     0.473
```

n (Diagnosis):

In my dataset, there are 65 children: 34 TD,31 ASD. Their ratio appears to be roughly 50/50 and therefore balanced.

Age:

The age on visit 1 varies, with the youngest child being about ~20 months old, while the oldest was ~40 months old. The youngest were TD individuals, oldest ASD. Reason for this might be the fact that, developmentally, the children may be in a similar development/cognitive skill bracket, with ASD children simply being slower to develop. As such, it’s hard to say whether this is a balance issue.

Gender:

Though within genders the diagnoses appear to be balanced at roughly 50%, there are much fewer girls in the experiment (11 girls, 55 boys). This makes the experiment extremely unbalanced in terms of gender.

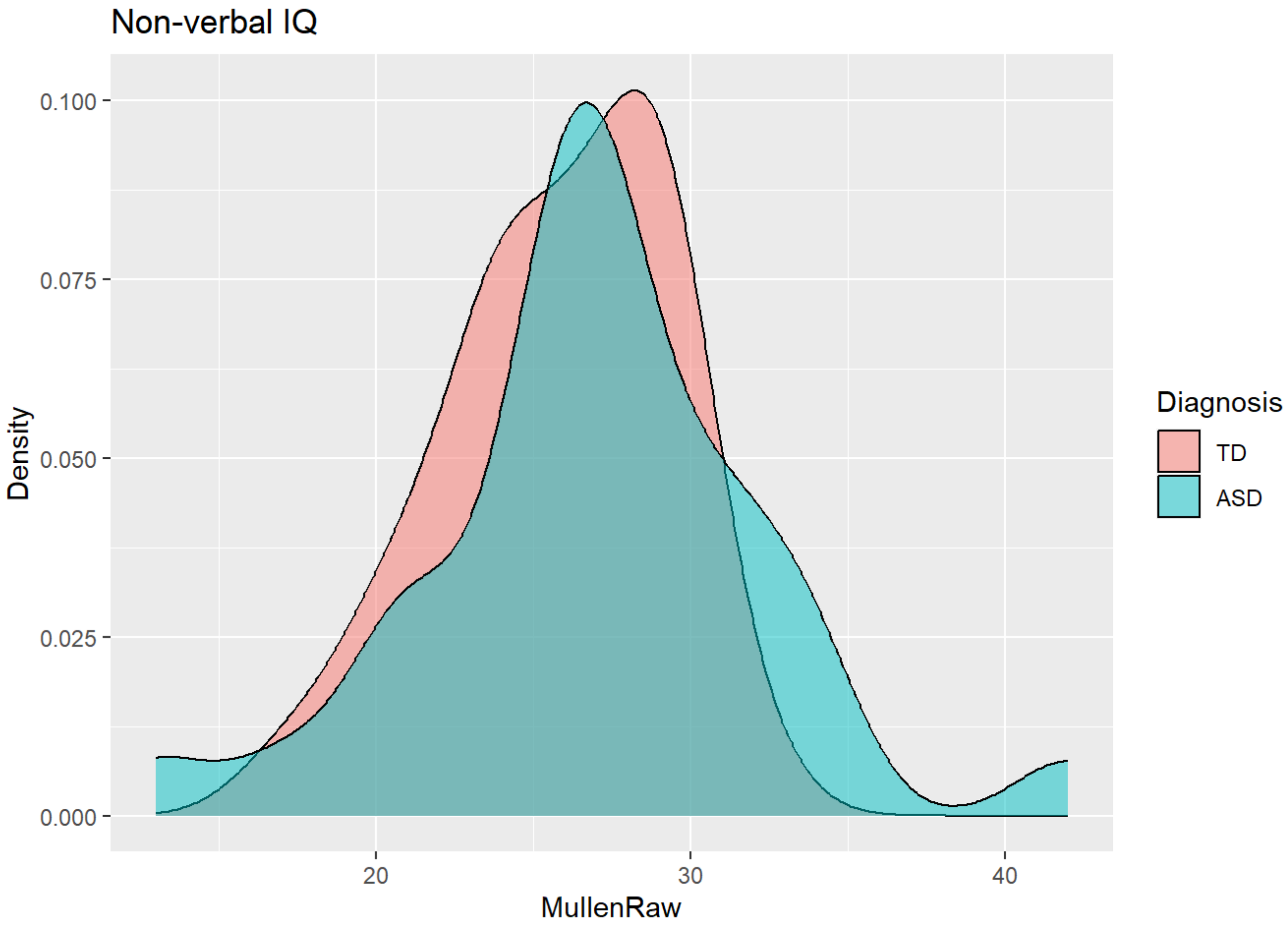
Ethnicity:

Of the 65 children 56 are fully “White” with there being only one Asian, for example. This makes the experiment very skewed in terms of ethnicity; though whether this matters or not is hard to say.

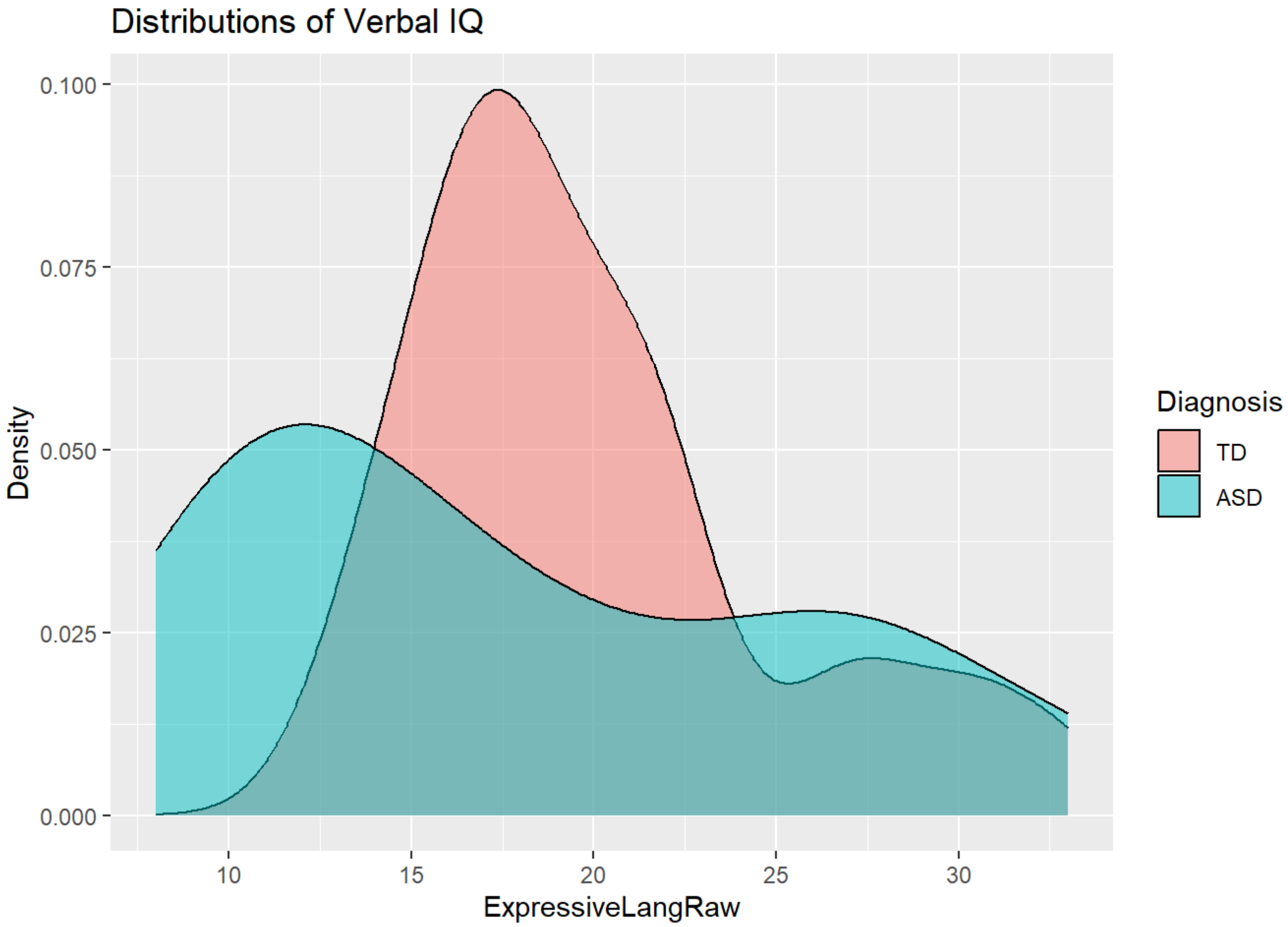
Balance:

When it comes to these metrics, the children are balanced in terms of Diagnosis, but not Gender. It’s hard to say whether Age is unbalanced, as it might be difficult to choose children, in particular ASD, who are developmentally similar. Ethnicity is very unbalanced, though it is possible that this was not a variable thought to be significant for this experiment. That said, this does mean that most of the children probably come from similar cultures with similar child-rearing methods.

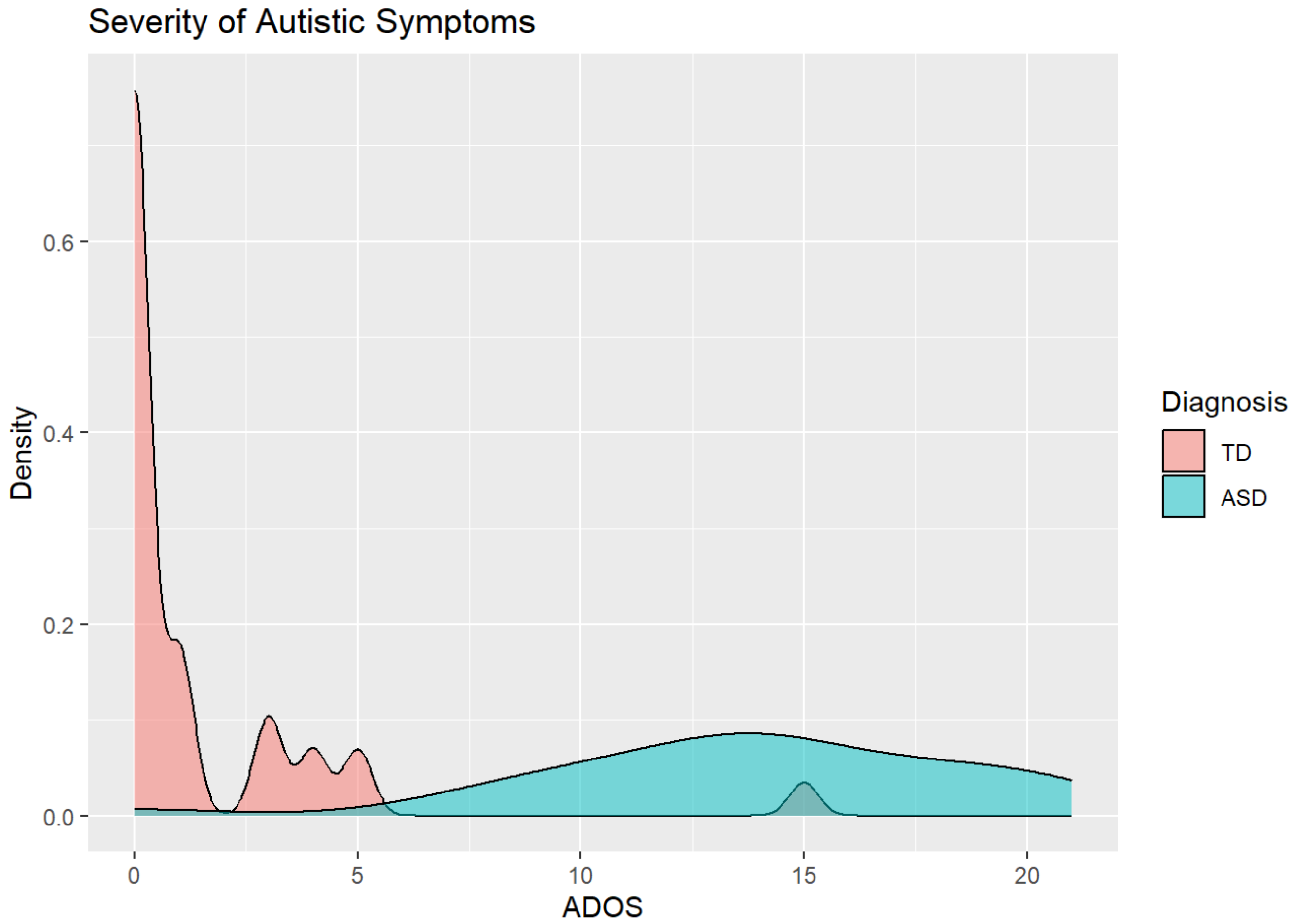
```
#Non-verbal IQ
ggplot(first_visit, aes(x = NonVerbal_IQ, fill = Diagnosis)) +
  geom_density(alpha = 0.5) +
  labs(title = "Non-verbal IQ", x = "MullenRaw", y = "Density")+
  guides(fill = guide_legend(title = "Diagnosis"))
```



```
#Verbal IQ
ggplot(first_visit, aes(x = Verbal_IQ, fill = Diagnosis)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distributions of Verbal IQ", x = "ExpressiveLangRaw", y = "Density")+
  guides(fill = guide_legend(title = "Diagnosis"))
```



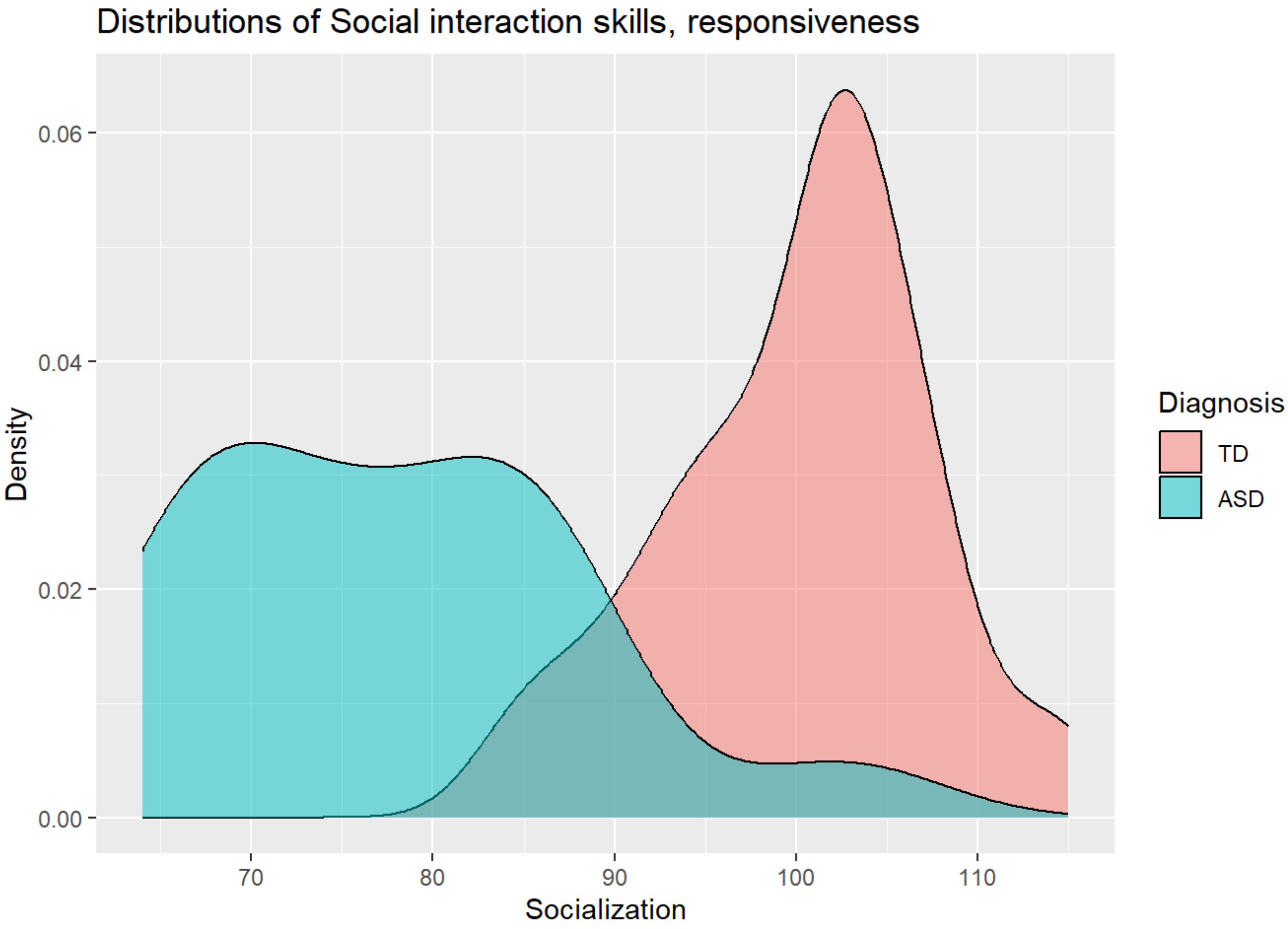
```
#ADOS
ggplot(first_visit, aes(x = ADOS, fill = Diagnosis)) +
  geom_density(alpha = 0.5) +
  labs(title = "Severity of Autistic Symptoms", x = "ADOS", y = "Density")+
  guides(fill = guide_legend(title = "Diagnosis"))
```



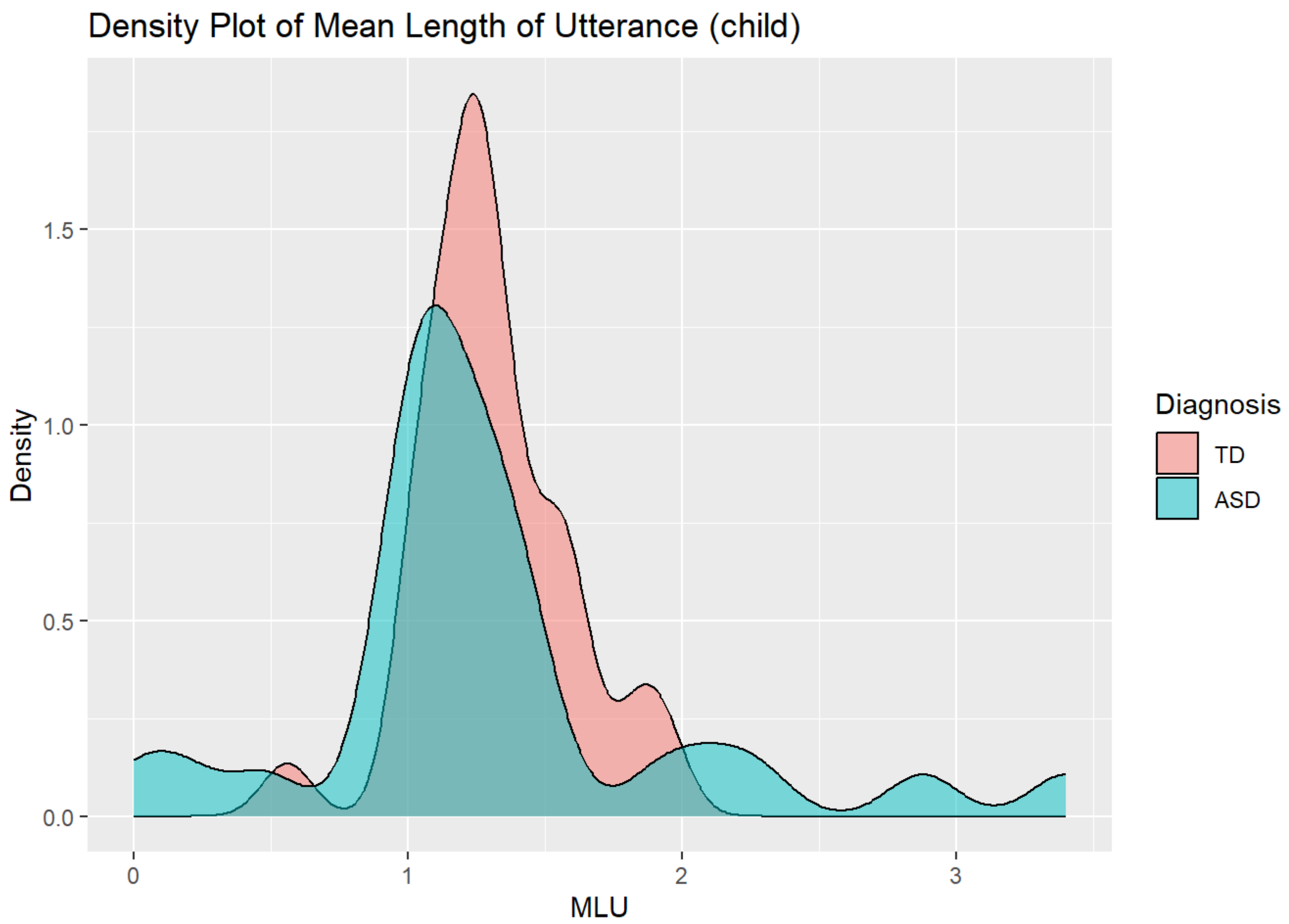
```
first_visit %>%
  group_by(Diagnosis)%>%
  summarise(mean_ADOS = mean(ADOS))
```

```
## # A tibble: 2 × 2
##   Diagnosis mean_ADOS
##   <fct>         <dbl>
## 1 TD             1.34
## 2 ASD           13.9
```

```
#Socialization
ggplot(first_visit, aes(x = Socialization, fill = Diagnosis)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distributions of Social interaction skills, responsiveness", x = "Socializ
         ation", y = "Density")+
  guides(fill = guide_legend(title = "Diagnosis"))
```



```
#CHI_MLU
ggplot(first_visit, aes(x = CHI_MLU, fill = Diagnosis)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot of Mean Length of Utterance (child)", x = "MLU", y = "Densit
         y") +
  guides(fill = guide_legend(title = "Diagnosis"))
```



### Cognitive features analysis (visually & first visit):

Non-verbal IQ:

TD and ASD children peak at roughly the same mean value, but ASD children's distribution has fatter tails, indicating that some ASD children have much lower Non-verbal IQ than the mean or much higher (max values at visit 1 are that of ASD children)

Verbal IQ:

TD children have a distribution with 2 peaks with most TD children having a similar level of Verbal IQ. That said, the second, smaller peak consists of TD children with higher than mean Verbal IQ, indicating that some are faster at developing their Verbal IQ than most.

For ASD children, the distribution is a lot less "clear" and evenly distributed, with some being on par with both the "average" and "gifted" TD children. That said, the highest "peak" for ASD children is near the bottom, indicating that, though some ASD children can keep up with TD children in terms of Verbal IQ, there are some who cannot.

ADOS:

Most TD children have 0 ADOS (severity of autistic symptoms), though there are a couple with more, going up to ~5. These children could be undiagnosed ASD children, but this is not necessarily the case. (mean = 0.9) Naturally, the ASD children are the ones with consistent values of ADOS over 5. (mean = 14)

Socialization:

TD children are again rather stable centered around a specific “distributional peak” with some being lower or higher in terms of their Socialization value; their values are predictable and even appear roughly gaussian.

ASD children have, for the most part, lower (and lowest) Socialization values (that is, social interaction skills and responsiveness) but this is not the case for all ASD children, as some (though few) can compete with TD values.

CHI_MLU:

TD children are fairly stable around their peak with lesser tails.

Most ASD children have comparative values to TD, but the fat tails indicate that there is much variability, with some even having higher mean length utterances than TD children.

Balance:

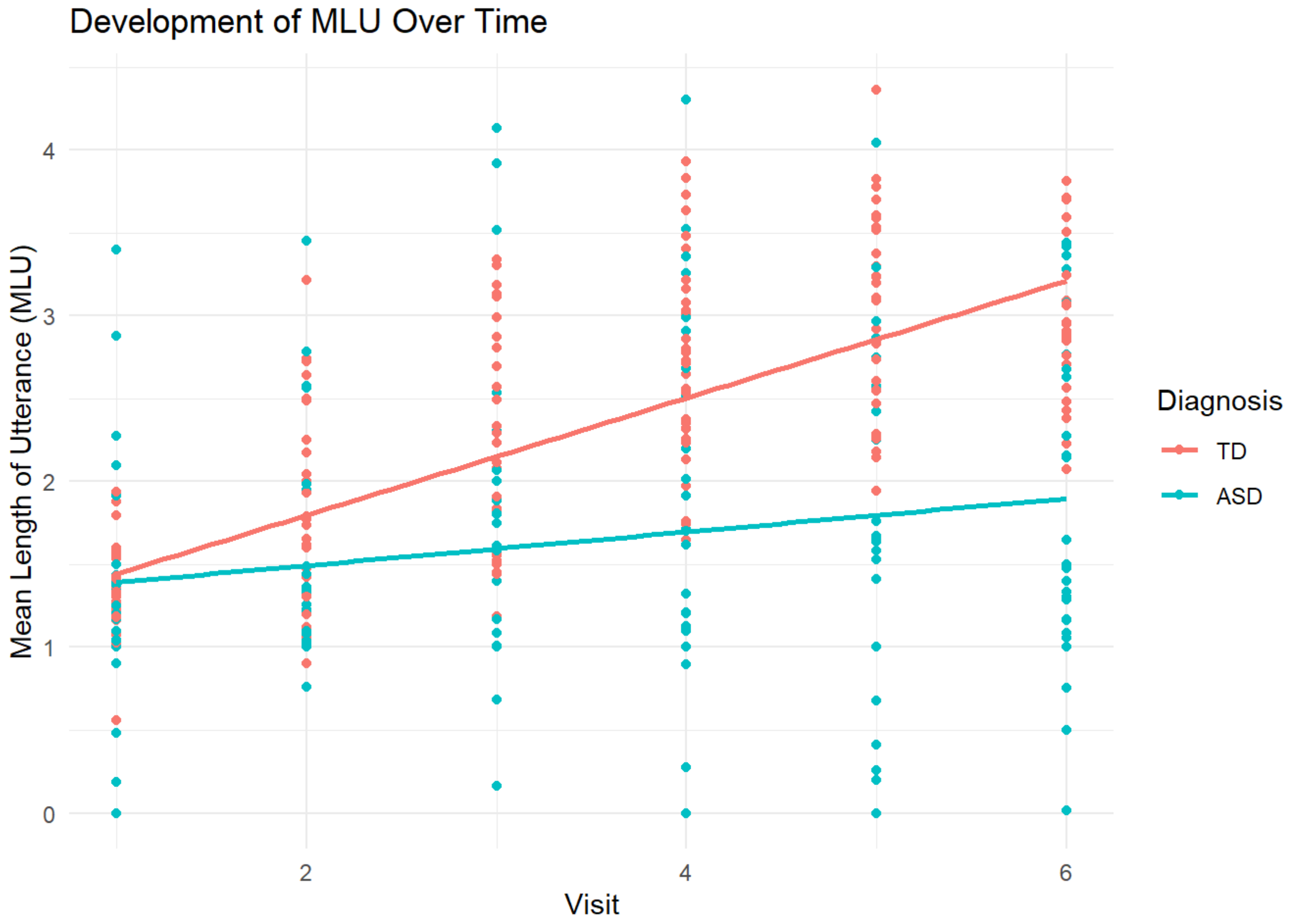
TD children tend to have rather stable density plots with few tails, with most TD children having “predictable” values around the mean, that is to say, they are developmentally comparable on many cognitive values with little spread. There are cases where certain TD children are more gifted than others, for example on Verbal IQ.

The density plots of ASD children are less Gaussian and often possess fat tails, indicating that their development on these cognitive features varies and is highly dependent on the individual; for example, some ASD individuals do not struggle with Verbal IQ while others are practically non-verbal.

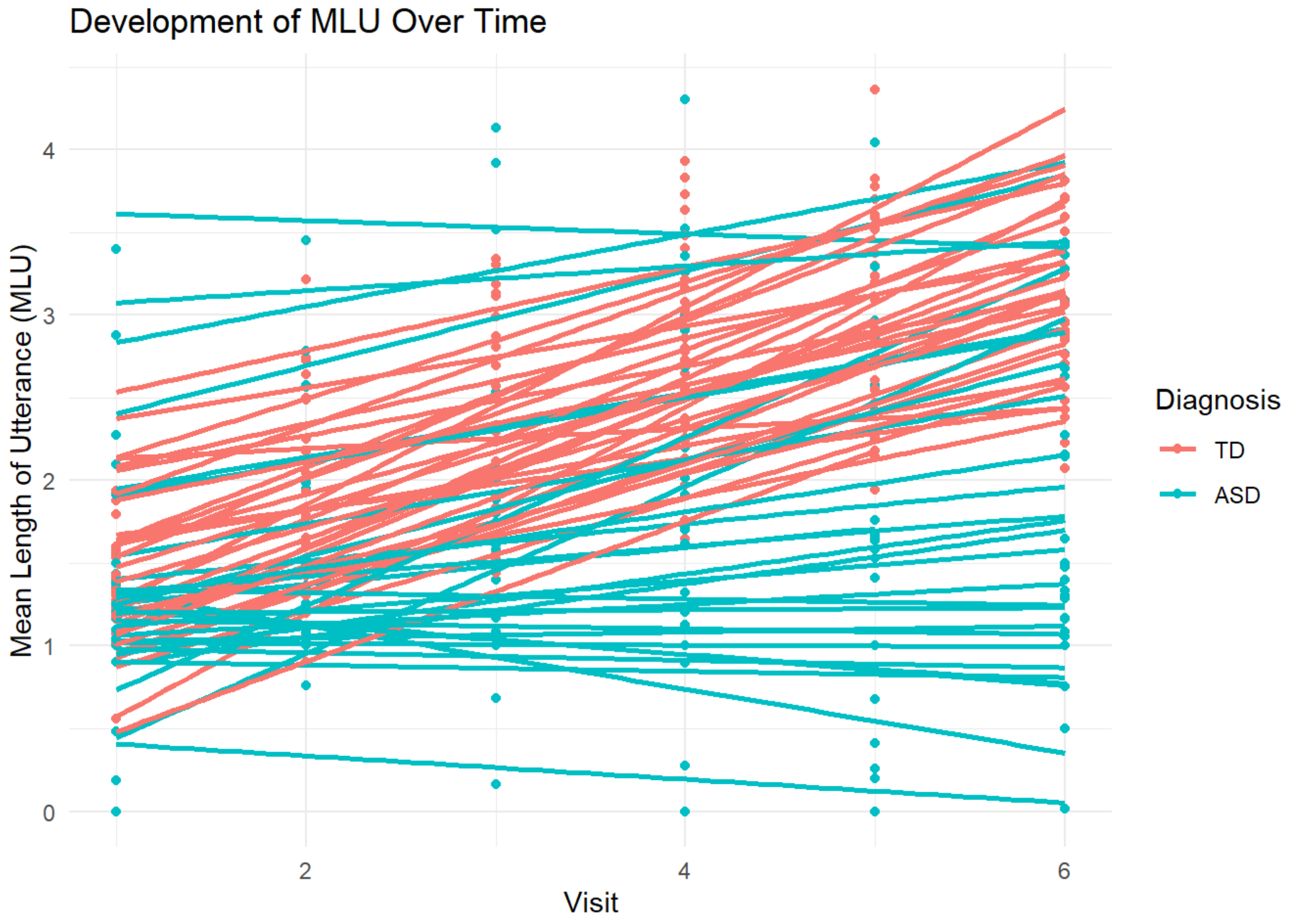
In terms of balance it’s hard to say, though perhaps the ASD individuals could’ve been somehow more balanced in the dataset; for example, the data could’ve been collected for children with a specific range of ADOS values, so as to avoid having too many “extreme” ASD children or too many “mild” ASD children? This depends on what the exact goal of the data is though, it might be beneficial to have a model that can diagnose milder cases than a model that is affected by the “obvious” ASD children with high ADOS.

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

```r
#CHI_MLU over time
ggplot(data, aes(x = Visit, y = CHI_MLU, color = Diagnosis)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, aes(group = Diagnosis)) +
  labs(title = "Development of MLU Over Time", x = "Visit", y = "Mean Length of Utterance
        (MLU)") +
  theme_minimal()
```



```
ggplot(data, aes(x = Visit, y = CHI_MLU, color = Diagnosis)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE, aes(group = Child.ID)) +
  labs(title = "Development of MLU Over Time", x = "Visit", y = "Mean Length of Utterance
         (MLU)") +
  theme_minimal()
```



```
data %>%
  group_by(Diagnosis, Visit) %>%
  filter(Visit == 1 | Visit == 6)%>%
  summarize(mean_MLU = mean(CHI_MLU, na.rm = TRUE), max_MLU = max(CHI_MLU, na.rm = TRUE),
         min_MLU = min(CHI_MLU, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'Diagnosis'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 × 5
## # Groups:   Diagnosis [2]
##   Diagnosis Visit mean_MLU max_MLU min_MLU
##   <fct>     <dbl>    <dbl>   <dbl>   <dbl>
## 1 TD            1     1.31    1.94  0.558
## 2 TD            6     2.93    3.81  2.07
## 3 ASD           1     1.30    3.4   0
## 4 ASD           6     1.89    3.44  0.0156
```

```r
data %>%
  filter(Visit %in% c(1, 6)) %>%  # Filter for only visits 1 and 6
  group_by(Child.ID, Visit, Diagnosis) %>%
  summarize(min_MLU = min(CHI_MLU)) %>%
  pivot_wider(names_from = Visit, values_from = min_MLU) %>%
  filter(`1` > `6`)
```

```
## `summarise()` has grouped output by 'Child.ID', 'Visit'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 6 × 4
## # Groups:   Child.ID [6]
##   Child.ID Diagnosis   `1`   `6`
##   <fct>    <fct>     <dbl> <dbl>
## 1 19       ASD        1.03 1
## 2 23       ASD        1    0.5
## 3 26       ASD        1.43 1.31
## 4 29       ASD        3.4  3.36
## 5 36       ASD        1.25 0.754
## 6 50       ASD        1.17 1.16
```

MLU over time:

On visit 1, both TD and ASD children have a mean MLU of about 1.3. By visit 6, however, TD have developed much faster, with a mean MLU of 2.93, while ASD children have only gotten to the mean of 1.89. This would indicate that TD children, as a whole, develop faster in terms of MLU.

Describe individual differences in linguistic development: do all kids follow the same path? Are all kids reflected by the general trend for their group?

Looking at individuals, TD children appear to have a rather predictable MLU development, with most following the mean and even visually appearing to have similar slopes. ASD children, however, vary widely individual-to-individual, with some even having a negative developmental slope. This would indicate that ASD children have a much more unpredictable development of MLU than TD children. In fact, 6 ASD children have a lower MLU value on visit 6 than they do on visit 1.

#Model making

```
#fixing NA by replacing the NA with a value from a visit before or after (by Child.ID), ki
        nd of sloppy but should work

noNa <- data %>%
  arrange(Child.ID, Visit) %>%
  group_by(Child.ID) %>%
  mutate(across(everything(), ~ ifelse(is.na(.), lag(., default = NA), .)))

for (i in 1:10) {
  noNa <- noNa %>%
    arrange(Child.ID, Visit) %>%
    group_by(Child.ID) %>%
    mutate(across(everything(), ~ ifelse(is.na(.), lag(., default = NA), .)))
}

#if certain values didnt get filled in due to missing in such a way where lag cannot help
        (being 1st or having no values before)
for (i in 1:10) {
noNa <- noNa %>%
  arrange(Child.ID, Visit) %>%
  group_by(Child.ID) %>%
  mutate(across(everything(), ~ ifelse(is.na(.), lead(., default = NA), .)))
}

#omitting children with bad data by hand since so few datapoints
noNa <- noNa %>%
  filter(Child.ID != 12 & Child.ID != 24 & Child.ID != 49 & Child.ID != 51 & Child.ID !=
        1) #12, 24, 49, 51, 1 have few visits with a lot of missing data, none of which c
        an be filled in
```

Formula Definition

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.

Based on my earlier (mostly visual) findings, I believe the following: non-verbal IQ is not particularly distinguishing (on visit 1) for the conditions and therefore can be omitted (both TD and ASD peak at similar values, though ASD have fatter tails) ADOS1 definitely distinguishes the conditions and ties into CHI_MLU as a result Verbal IQ is more distinguishing than non-verbal IQ and should be used in the model. Socialization definitely distinguishes the conditions (differing heavily between conditions)

```r
formula1 <-
  brms::bf(CHI_MLU ~ 1  + Visit + Diagnosis + MOT_MLU + Gender + Age + (1|Child.ID))

formula2 <-
  brms::bf(CHI_MLU ~ 1 + Visit + Diagnosis +  MOT_MLU + Gender + Age + (1 + Visit + ADOS1
         + Diagnosis ||Child.ID))

formula3 <- brms::bf(CHI_MLU ~ 1 + Visit + Diagnosis + MOT_MLU + Gender+ Age+ (1 + Visit +
         ADOS1 + Socialization1 + Diagnosis||Child.ID))


formula4 <-
  brms::bf(CHI_MLU ~ 1 + Visit + Diagnosis + MOT_MLU + Gender+ Age+(1 + Visit + ADOS1 + So
         cialization1 + verbalIQ1 + Diagnosis||Child.ID))
```

Prior Definition

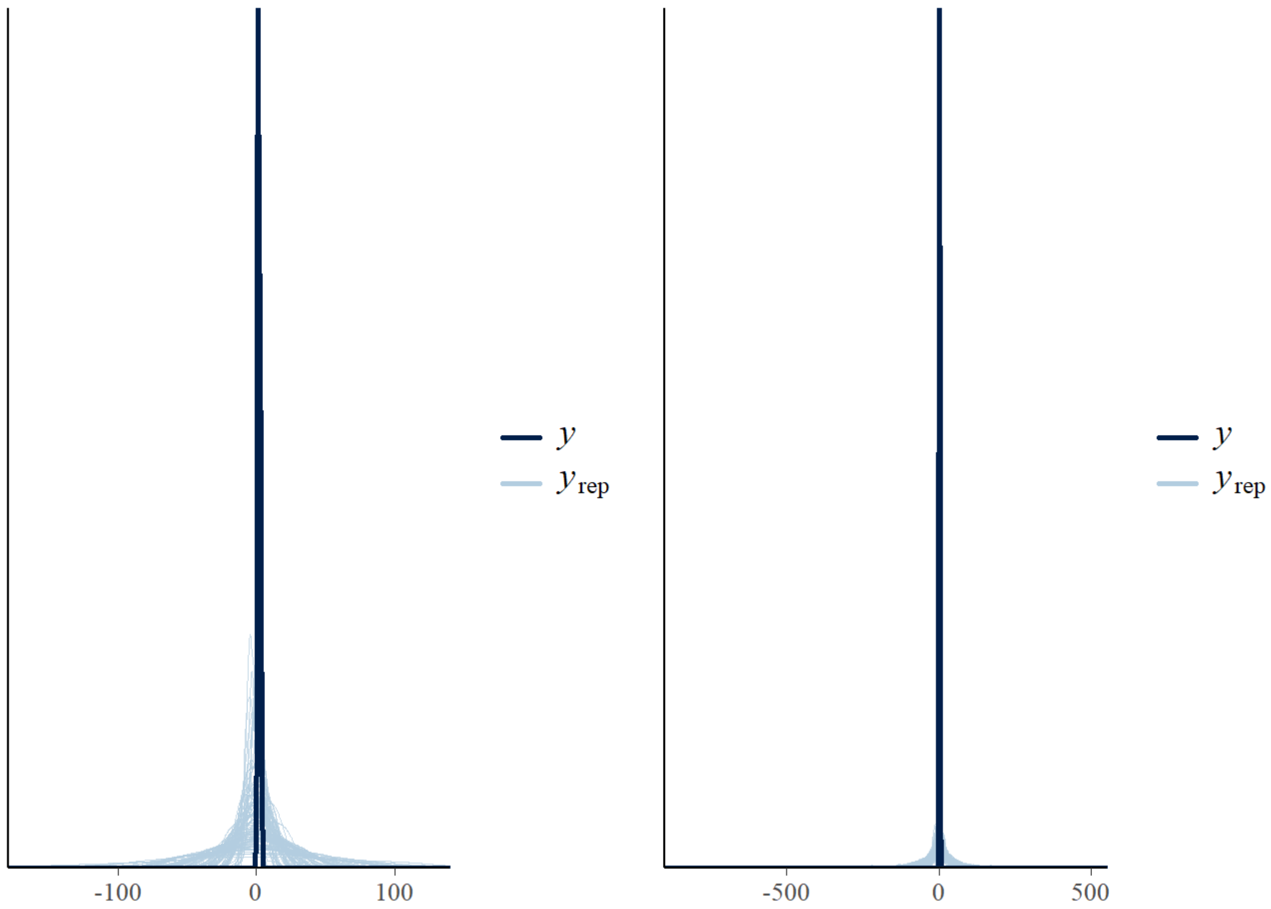
```
get_prior(formula1, data = data) %>% kable()
get_prior(formula2, data = data) %>% kable()
get_prior(formula3, data = data) %>% kable()
get_prior(formula4, data = data) %>% kable()

priors1 <- c(
  prior(normal(0, 2), class = b, coef = Age),
prior(normal(0, 2), class = b, coef = DiagnosisASD),
prior(normal(0, 2), class = b, coef = GenderM),
prior(normal(0, 2), class = b, coef = MOT_MLU),
 prior(normal(0, 2), class = b, coef = Visit),
 prior(normal(0, 2), class = Intercept)
)

priors2 <- c(
  prior(normal(0, 2), class = b, coef = Age),
prior(normal(0, 2), class = b, coef = DiagnosisASD),
prior(normal(0, 2), class = b, coef = GenderM),
prior(normal(0, 2), class = b, coef = MOT_MLU),
 prior(normal(0, 2), class = b, coef = Visit),
 prior(normal(0, 2), class = Intercept)

)

priors3 <- c(
prior(normal(0, 2), class = b, coef = Age),
prior(normal(0, 2), class = b, coef = DiagnosisASD),
prior(normal(0, 2), class = b, coef = GenderM),
prior(normal(0, 2), class = b, coef = MOT_MLU),
prior(normal(0, 2), class = b, coef = Visit),
 prior(normal(0, 2), class = Intercept)
)

priors4 <- c(
prior(normal(0, 2), class = b, coef = Age),
prior(normal(0, 2), class = b, coef = DiagnosisASD),
prior(normal(0, 2), class = b, coef = GenderM),
prior(normal(0, 2), class = b, coef = MOT_MLU),
prior(normal(0, 2), class = b, coef = Visit),
 prior(normal(0, 2), class = Intercept)
)

#I wound up setting the same priors for all since the random effects were the ones i kept
         changing; perhaps I should've done the fixed effects in tiers as well
```

Prior Predictive Checking

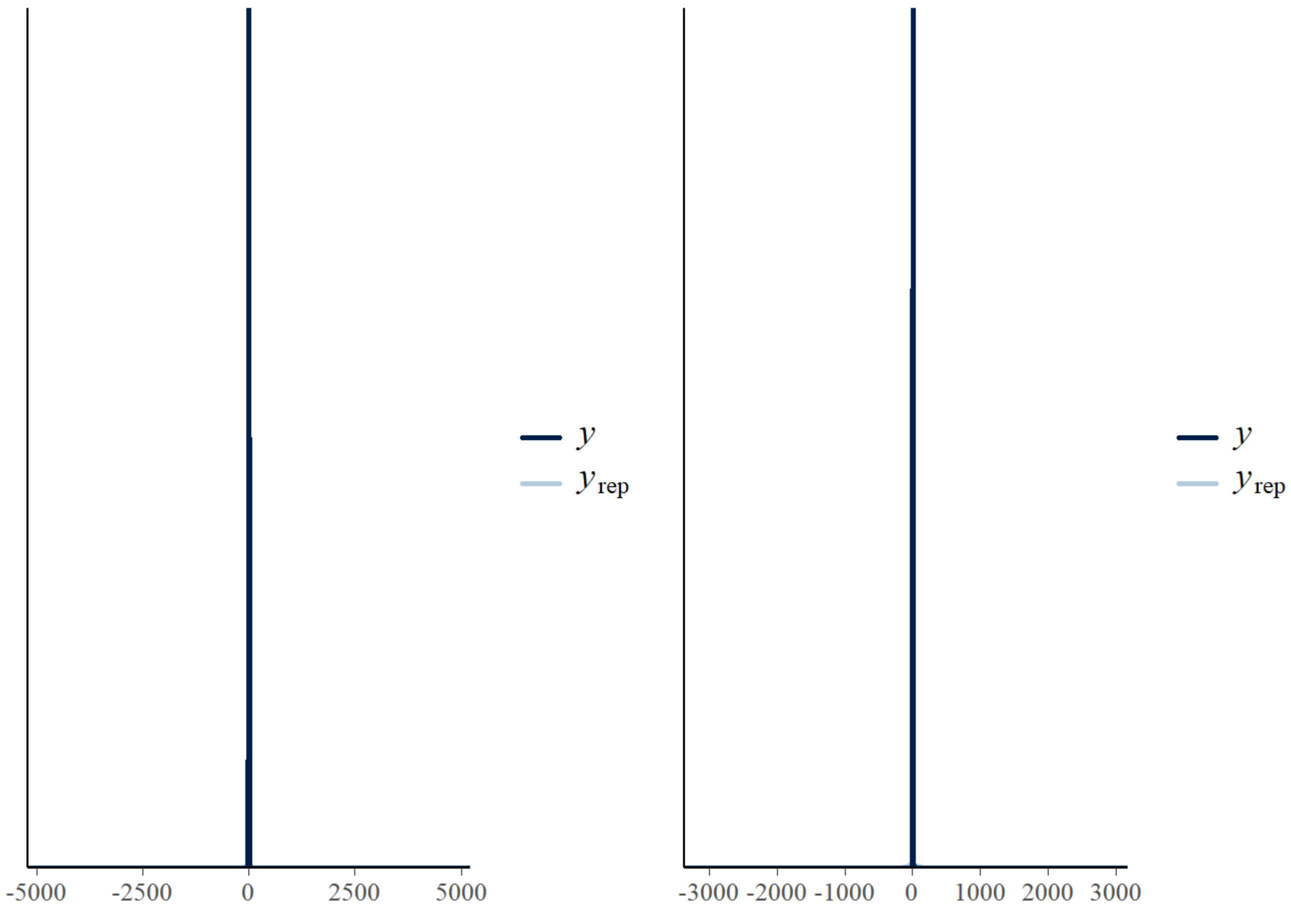
By setting *sample_prior* parameter is set to “only” in the **brm** function, draws are drawn solely from the priors, thus ignoring the likelihood. This allows among other things to generate draws from the prior predictive distribution.

```r
model1_prior <- brm(
  formula1,
  data,
  family = gaussian,
  prior = priors1,
  sample_prior = "only", #draws are drawn solely from the priors, ignoring the likelihood!
  file = 'data/model1_prior',
  backend = "cmdstanr",
  chains = 2,
  stan_model_args = list(stanc_options = list("O1"))
)
prior_predictive1 <- pp_check(model1_prior, ndraws = 100)

model2_prior <- brm(
  formula2,
  data,
  family = gaussian,
  prior = priors2,
  sample_prior = "only", #draws are drawn solely from the priors, ignoring the likelihood!
  file = 'data/model2_prior',
  backend = "cmdstanr",
  chains = 2,
  stan_model_args = list(stanc_options = list("O1"))
)
prior_predictive2 <- pp_check(model2_prior, ndraws = 100)


model3_prior <- brm(
  formula3,
  data,
  family = gaussian,
  prior = priors3,
  sample_prior = "only", #draws are drawn solely from the priors, ignoring the likelihood!
  file = 'data/model3_prior',
  backend = "cmdstanr",
  chains = 2,
  stan_model_args = list(stanc_options = list("O1"))
)
prior_predictive3 <- pp_check(model3_prior, ndraws = 100)


model4_prior <- brm(
  formula4, #is this stinkying it up?
  data,
  family = gaussian,
  prior = priors4,
  sample_prior = "only", #draws are drawn solely from the priors, ignoring the likelihood!
  file = 'data/model4_prior',
  backend = "cmdstanr",
  chains = 2,
  stan_model_args = list(stanc_options = list("O1"))
)
```

```
prior_predictive4 <- pp_check(model4_prior, ndraws = 100)

prior_predictive1 + prior_predictive2
```



```
prior_predictive3 + prior_predictive4
```

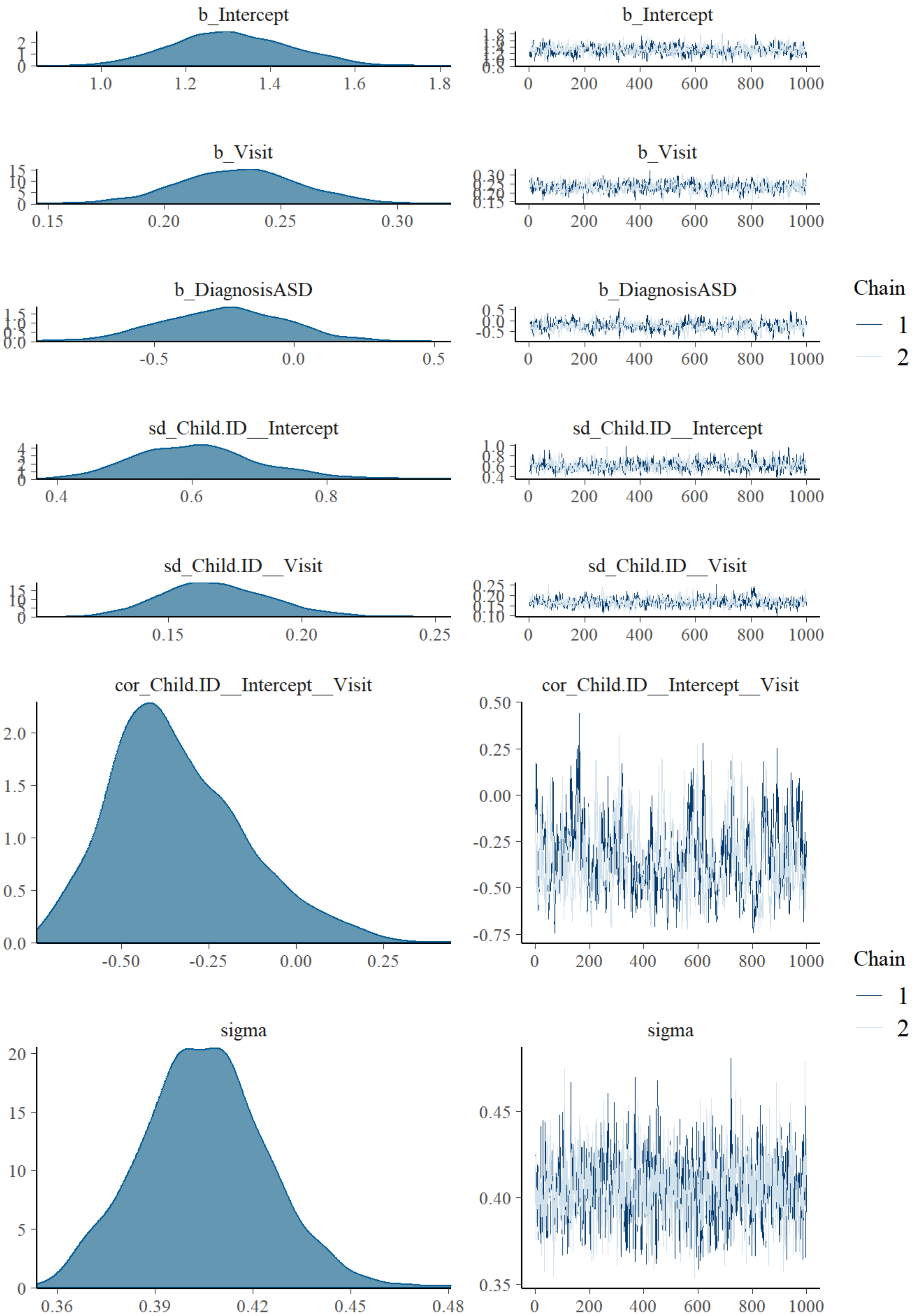

-5000
-2500
0
2500
5000
y
y_{rep}
-3000
-2000
-1000
0
1000
2000
3000
y
y_{rep}

Model Fitting

```
prior1model <- readRDS('data/model1_prior.rds')
prior2model <- readRDS('data/model2_prior.rds')
prior3model <- readRDS('data/model3_prior.rds')
prior4model <- readRDS('data/model4_prior.rds')

m1 <- brm(
  formula1,
  data,
  family = gaussian,
  prior = prior1model$prior,
  #save_pars = save_pars(all = TRUE),
  sample_prior = T,
  file = 'data/model1_fit',
  backend = "cmdstanr",
  chains = 2,
  cores = 2,
  threads = threading(2),
  control = list(
    adapt_delta = 0.9,
    max_treedepth = 20
  ),
  stan_model_args = list(stanc_options = list("O1"))
)

m2 <- brm(
  formula2,
  data,
  family = gaussian,
  prior = prior2model$prior,
  #save_pars = save_pars(all = TRUE),
  sample_prior = T,
  file = 'data/model2_fit',
  backend = "cmdstanr",
  chains = 2,
  cores = 2,
  threads = threading(2),
  control = list(
    adapt_delta = 0.90,
    max_treedepth = 20
  ),
  stan_model_args = list(stanc_options = list("O1"))
)

m3 <- brm(
  formula3,
  data,
  family = gaussian,
  prior = prior3model$prior,
  #save_pars = save_pars(all = TRUE),
  sample_prior = T,
  file = 'data/model3_fit',
  backend = "cmdstanr",
  chains = 2,
  cores = 2,
  threads = threading(2),
  control = list(
    adapt_delta = 0.9,
    max_treedepth = 20
  ),
  stan_model_args = list(stanc_options = list("O1"))
)

m4 <- brm(
  formula4,
  data,
  family = gaussian,
  prior = prior4model$prior,
  #save_pars = save_pars(all = TRUE),
  sample_prior = T,
  file = 'data/model4_fit',
  backend = "cmdstanr",
  chains = 2,
  cores = 2,
  threads = threading(2),
  control = list(
    adapt_delta = 0.90,
    max_treedepth = 20
  ),
  stan_model_args = list(stanc_options = list("O1"))
)
```

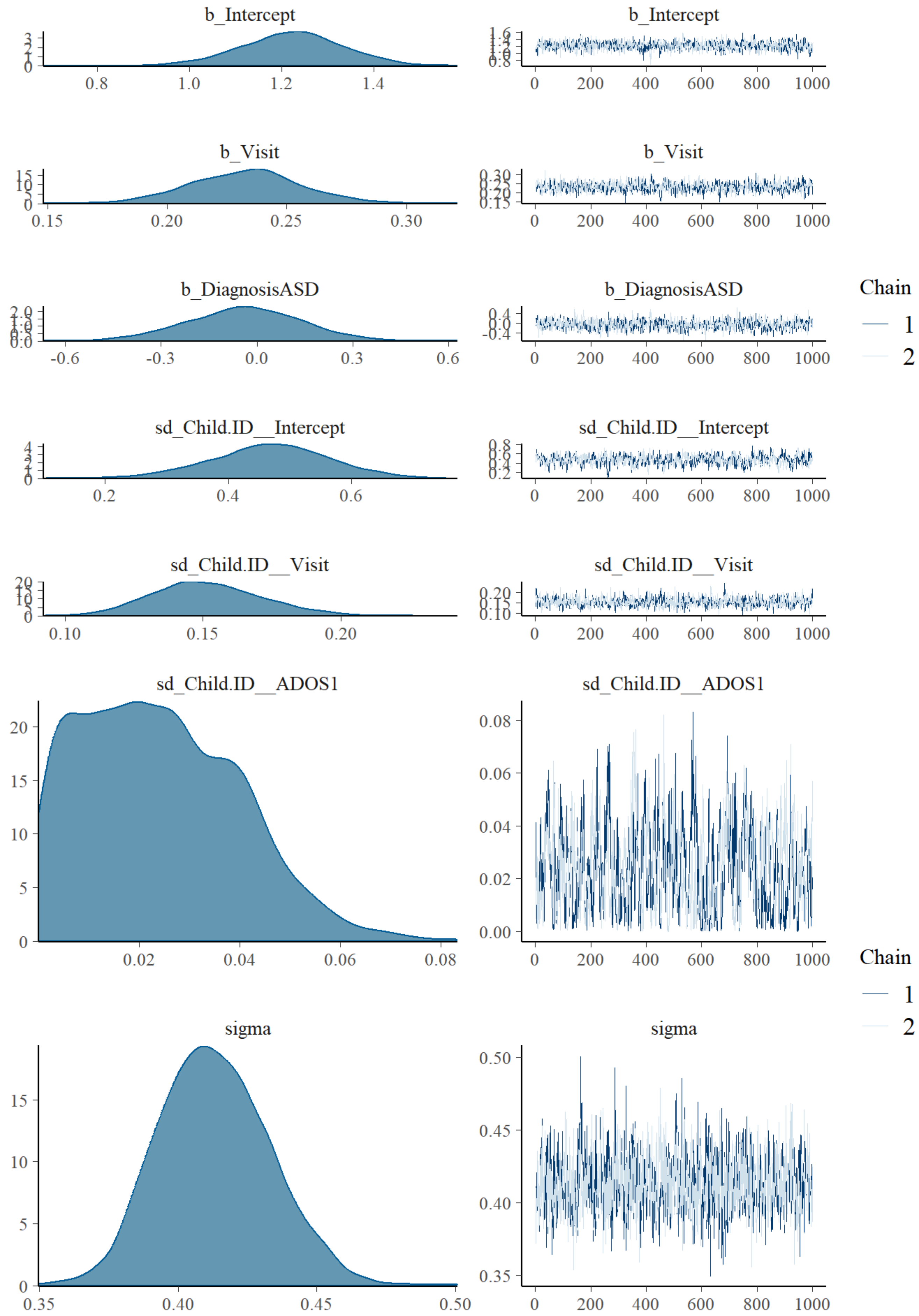
Model quality checks

```
plot(m1)
```

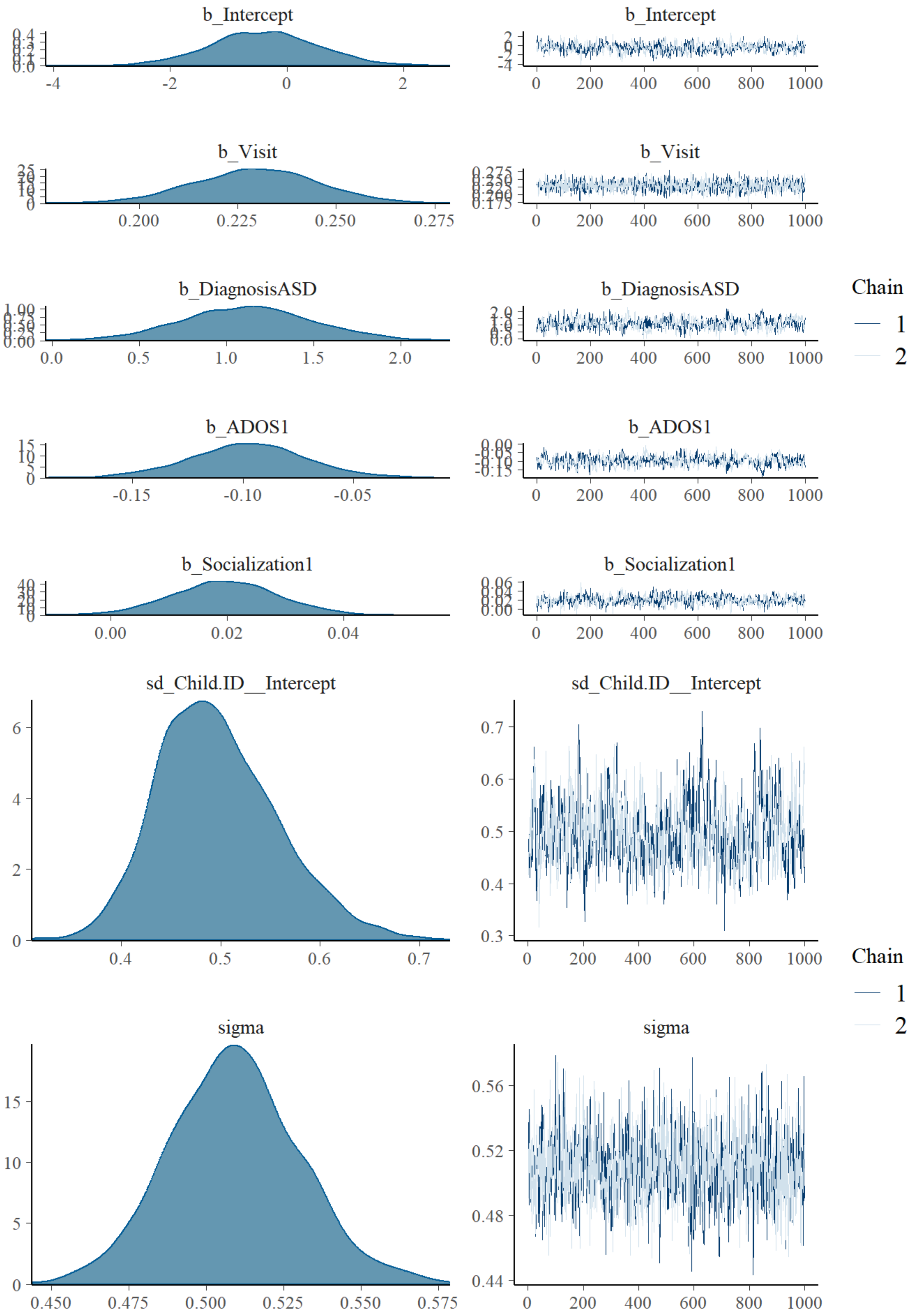


b_Intercept
0
1
2
1.0
1.2
1.4
1.6
1.8
b_Intercept
0.8
1.0
1.2
1.4
1.6
1.8
0
200
400
600
800
1000
b_Visit
0
5
10
15
0.15
0.20
0.25
0.30
b_Visit
0.15
0.20
0.25
0.30
0
200
400
600
800
1000
b_DiagnosisASD
0.0
0.5
1.0
1.5
-0.5
0.0
0.5
b_DiagnosisASD
-0.5
0.0
0.5
0
200
400
600
800
1000
Chain
1
2
sd_Child.ID__Intercept
0
1
2
3
4
0.4
0.6
0.8
sd_Child.ID__Intercept
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
sd_Child.ID__Visit
0
5
10
15
0.15
0.20
0.25
sd_Child.ID__Visit
0.10
0.15
0.20
0.25
0
200
400
600
800
1000
cor_Child.ID__Intercept__Visit
0.0
0.5
1.0
1.5
2.0
-0.50
-0.25
0.00
0.25
cor_Child.ID__Intercept__Visit
-0.75
-0.50
-0.25
0.00
0.25
0.50
0
200
400
600
800
1000
Chain
1
2
sigma
0
5
10
15
20
0.36
0.39
0.42
0.45
0.48
sigma
0.35
0.40
0.45
0
200
400
600
800
1000

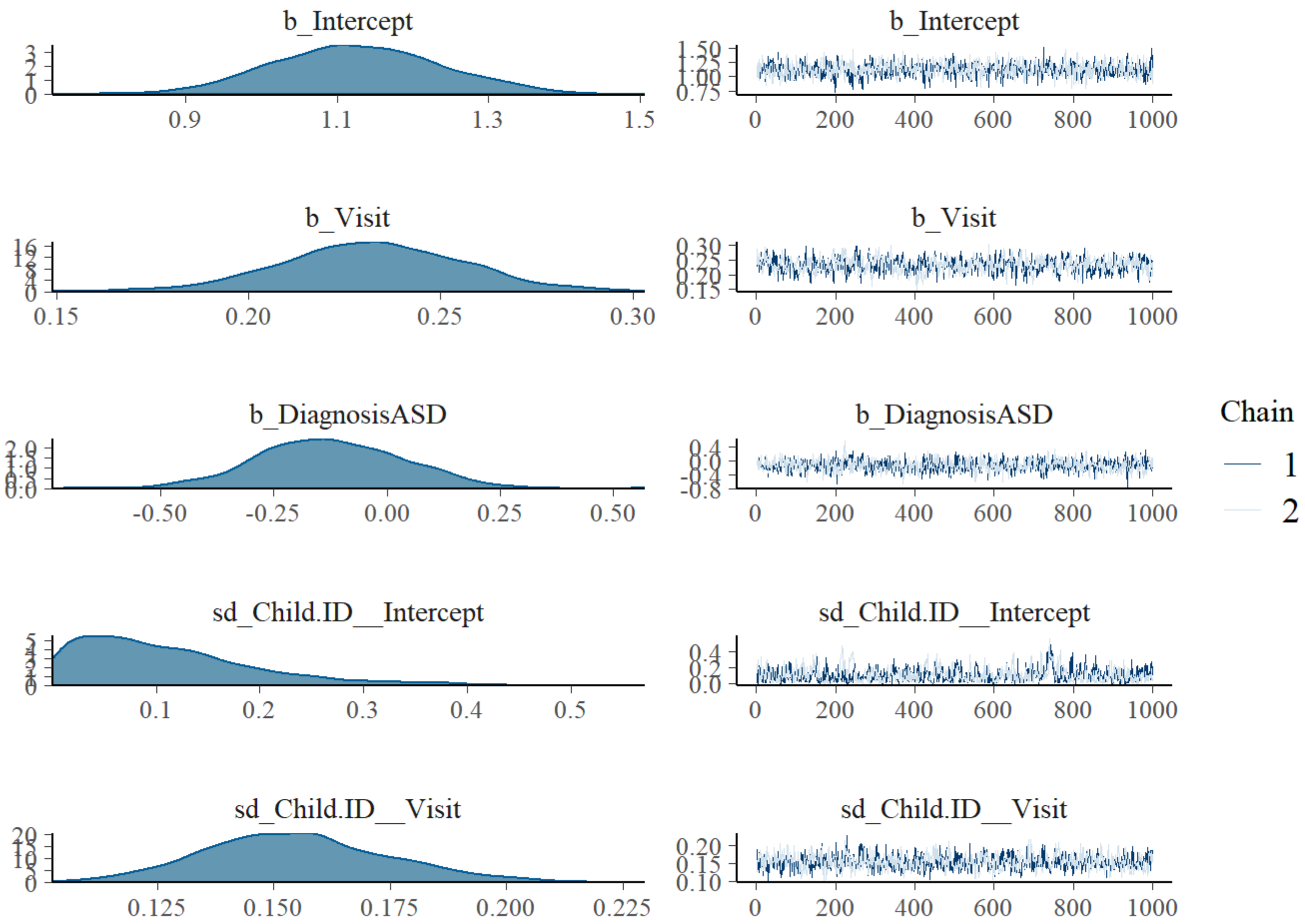
```r
plot(m2)
```

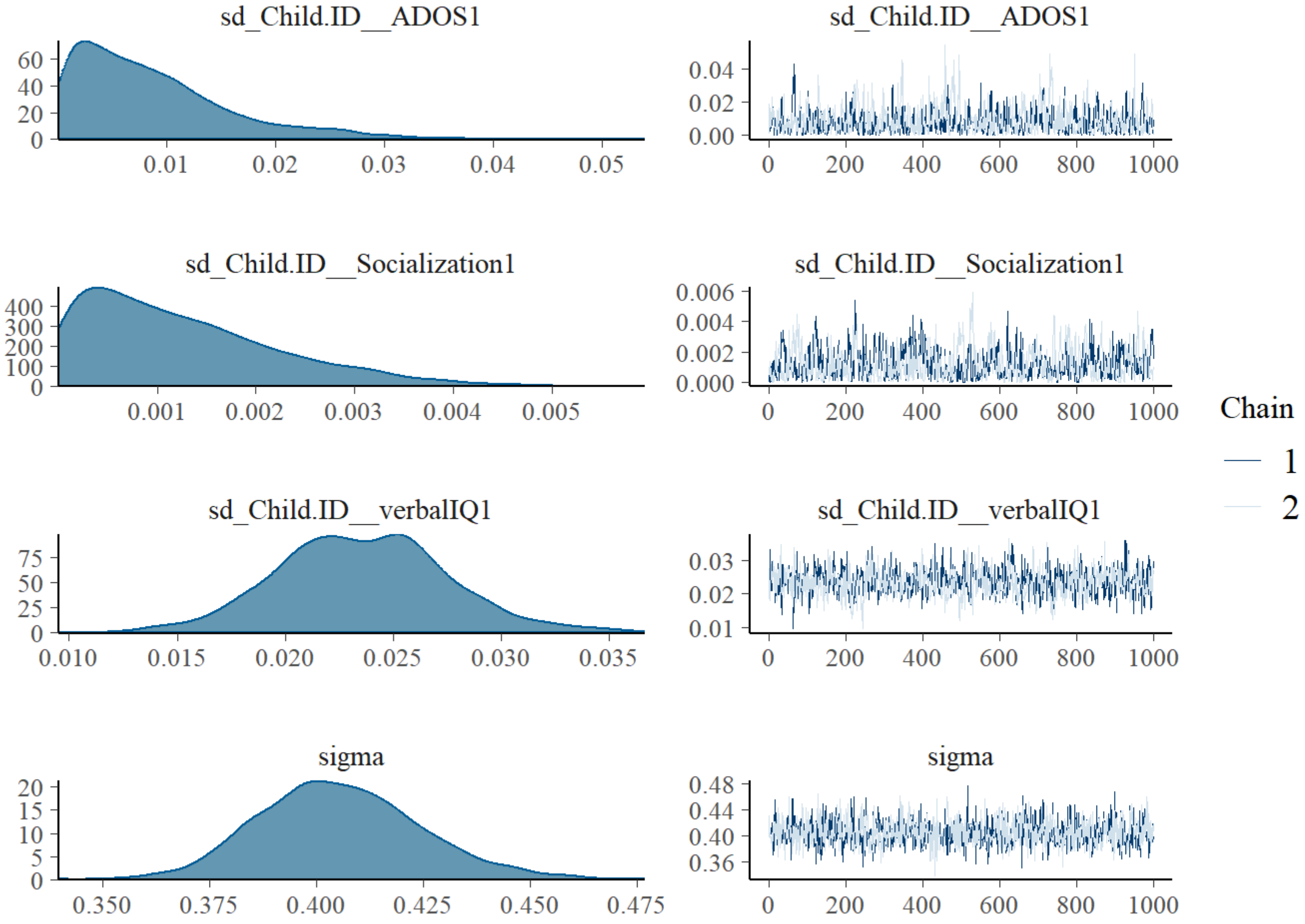

b_Intercept
0.8
1.0
1.2
1.4
0
1
2
3
1.6
1.4
1.0
0.8
0
200
400
600
800
1000
b_Visit
0.15
0.20
0.25
0.30
0
5
10
15
0.30
0.25
0.20
0.15
b_DiagnosisASD
-0.6
-0.3
0.0
0.3
0.6
0.0
0.5
1.0
2.0
0.4
0.0
-0.4
Chain
1
2
sd_Child.ID__Intercept
0.2
0.4
0.6
0
1
2
3
4
0.8
0.6
0.4
0.2
sd_Child.ID__Visit
0.10
0.15
0.20
0
5
10
15
20
0.20
0.15
0.10
sd_Child.ID__ADOS1
0
5
10
15
20
0.02
0.04
0.06
0.08
0.00
0.02
0.04
0.06
0.08
Chain
1
2
sigma
0
5
10
15
0.35
0.40
0.45
0.50
0.35
0.40
0.45
0.50

```
plot(m3)
```

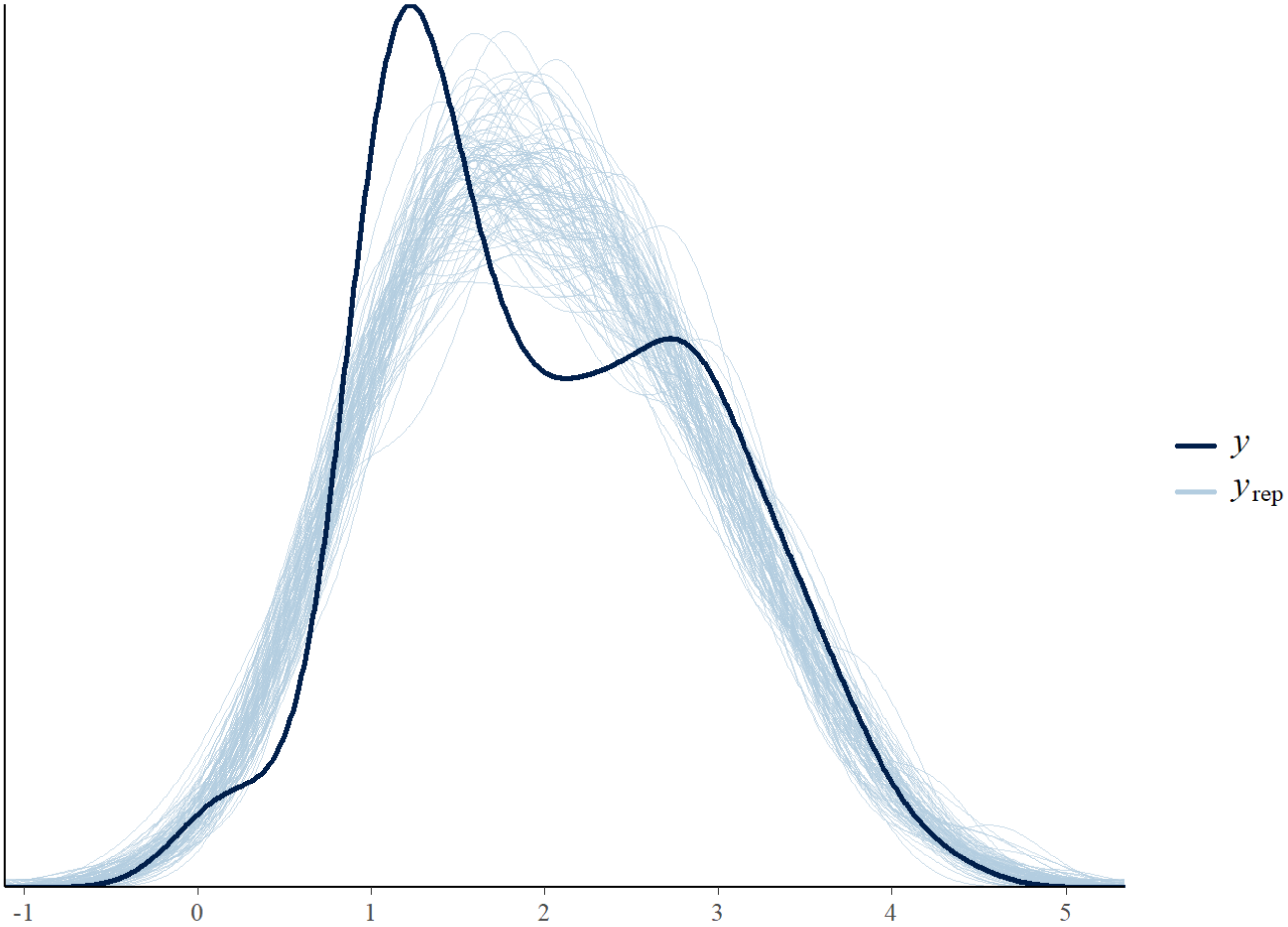

b_Intercept
0.4
0.0
-4
-2
0
2
2
0
-2
-4
0
200
400
600
800
1000
b_Visit
25
20
15
10
0
0.200
0.225
0.250
0.275
0.275
0.250
0.225
0.200
0.175
b_DiagnosisASD
1.00
0.75
0.50
0.25
0.00
0.0
0.5
1.0
1.5
2.0
2.0
1.5
1.0
0.5
0.0
Chain
1
2
b_ADOS1
15
10
5
0
-0.15
-0.10
-0.05
0.00
-0.05
-0.10
-0.15
b_Socialization1
40
30
20
10
0
0.00
0.02
0.04
0.06
0.04
0.02
0.00
sd_Child.ID__Intercept
6
4
2
0
0.4
0.5
0.6
0.7
0.7
0.6
0.5
0.4
0.3
Chain
1
2
sigma
15
10
5
0
0.450
0.475
0.500
0.525
0.550
0.575
0.56
0.52
0.48
0.44

```
plot(m4)
```

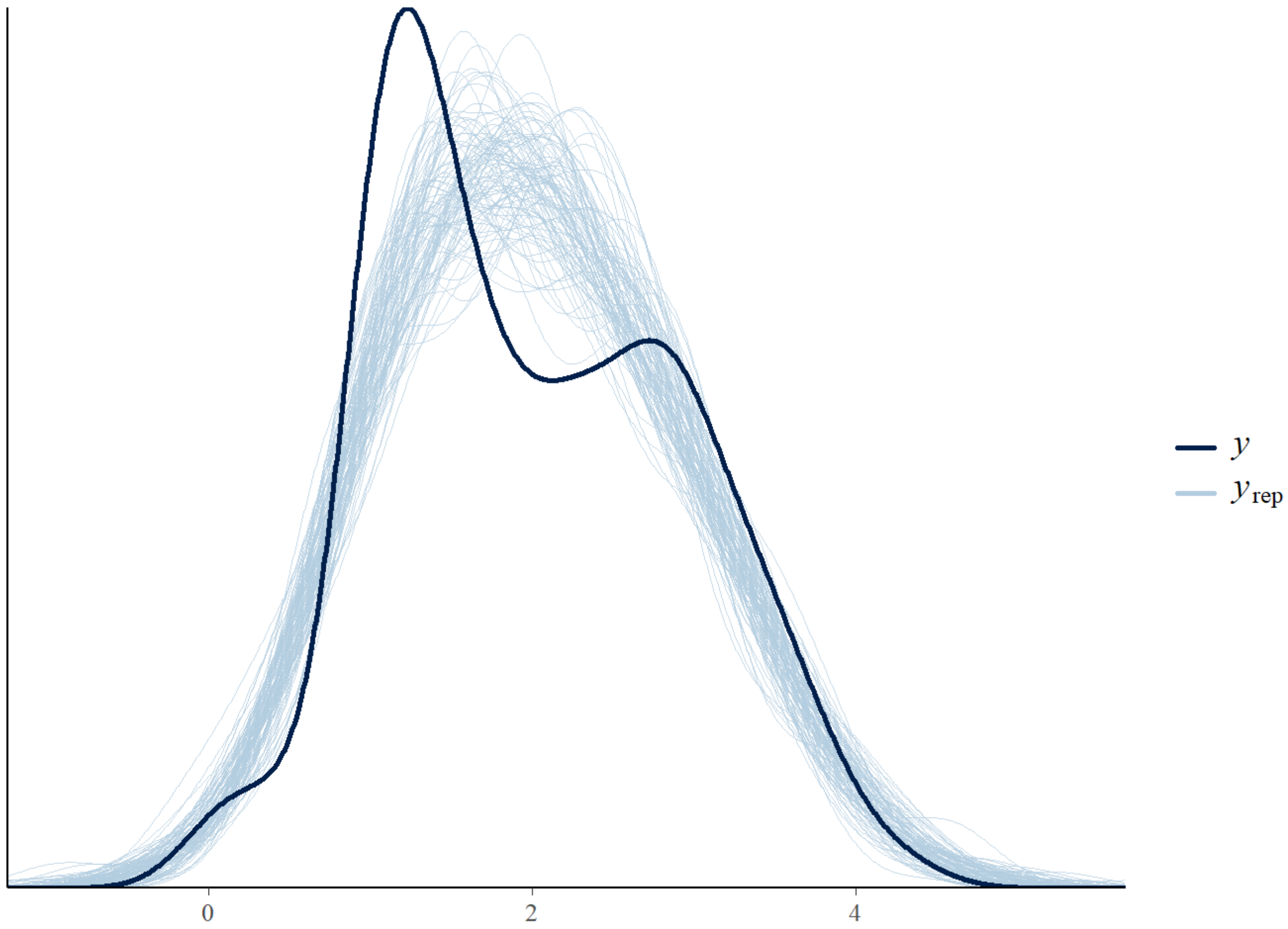

b_Intercept
0
1
2
3
0.9
1.1
1.3
1.5
b_Intercept
0.75
1.00
1.25
1.50
0
200
400
600
800
1000
b_Visit
0
4
8
12
16
0.15
0.20
0.25
0.30
b_Visit
0.15
0.20
0.25
0.30
0
200
400
600
800
1000
b_DiagnosisASD
0.0
0.5
1.0
1.5
2.0
-0.50
-0.25
0.00
0.25
0.50
b_DiagnosisASD
-0.8
-0.4
0.0
0.4
0
200
400
600
800
1000
Chain
1
2
sd_Child.ID__Intercept
0
1
2
3
4
5
0.1
0.2
0.3
0.4
0.5
sd_Child.ID__Intercept
0.0
0.2
0.4
0
200
400
600
800
1000
sd_Child.ID__Visit
0
5
10
15
20
0.125
0.150
0.175
0.200
0.225
sd_Child.ID__Visit
0.10
0.15
0.20
0
200
400
600
800
1000


sd_Child.ID__ADOS1
0
20
40
60
0.01
0.02
0.03
0.04
0.05
sd_Child.ID__ADOS1
0.00
0.02
0.04
0
200
400
600
800
1000
sd_Child.ID__Socialization1
0
100
200
300
400
0.001
0.002
0.003
0.004
0.005
sd_Child.ID__Socialization1
0.000
0.002
0.004
0.006
0
200
400
600
800
1000
Chain
1
2
sd_Child.ID__verbalIQ1
0
25
50
75
0.010
0.015
0.020
0.025
0.030
0.035
sd_Child.ID__verbalIQ1
0.01
0.02
0.03
0
200
400
600
800
1000
sigma
0
5
10
15
20
0.350
0.375
0.400
0.425
0.450
0.475
sigma
0.36
0.40
0.44
0.48
0
200
400
600
800
1000

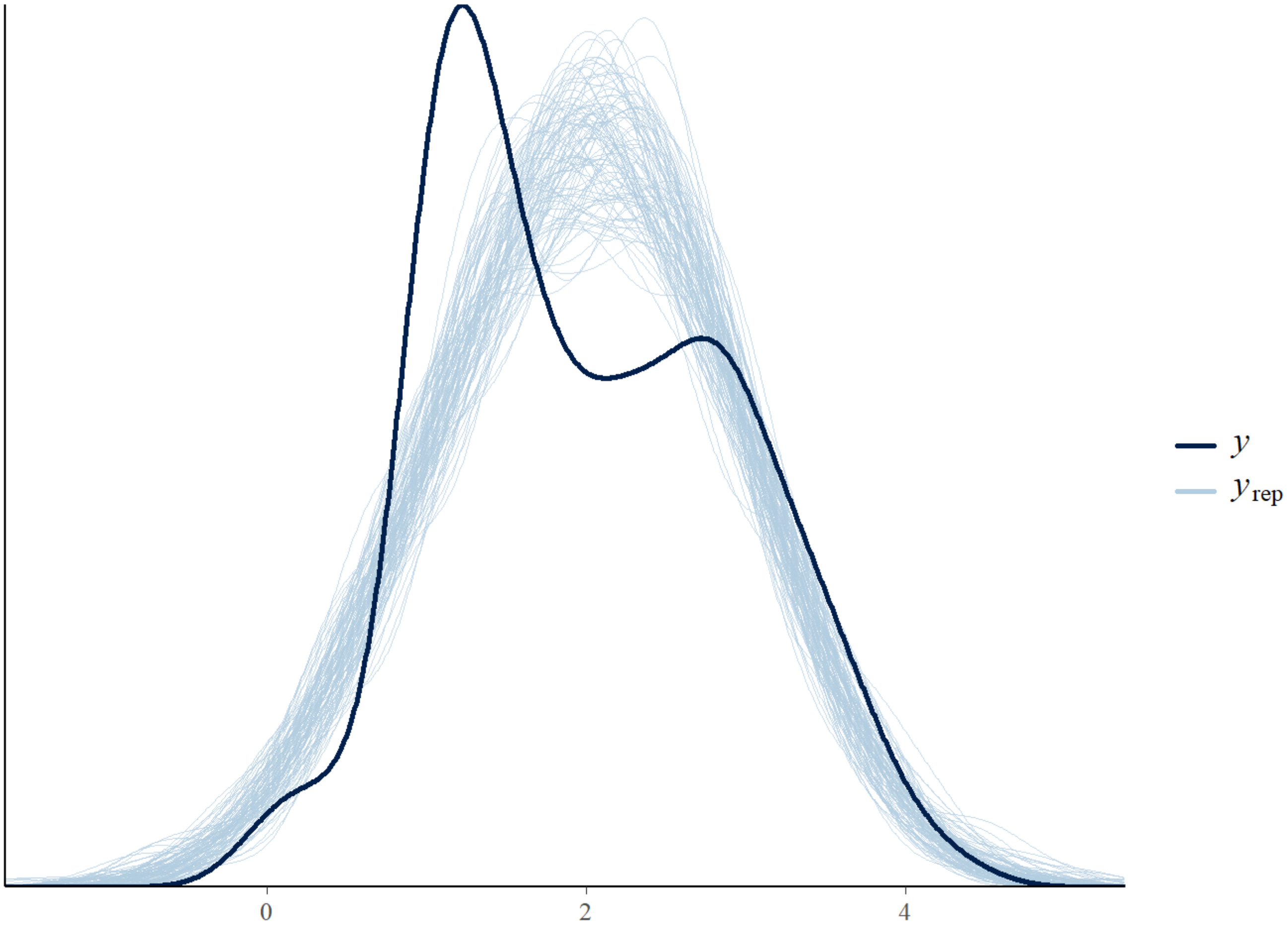
```
pp_check(m1, ndraws = 100)
```



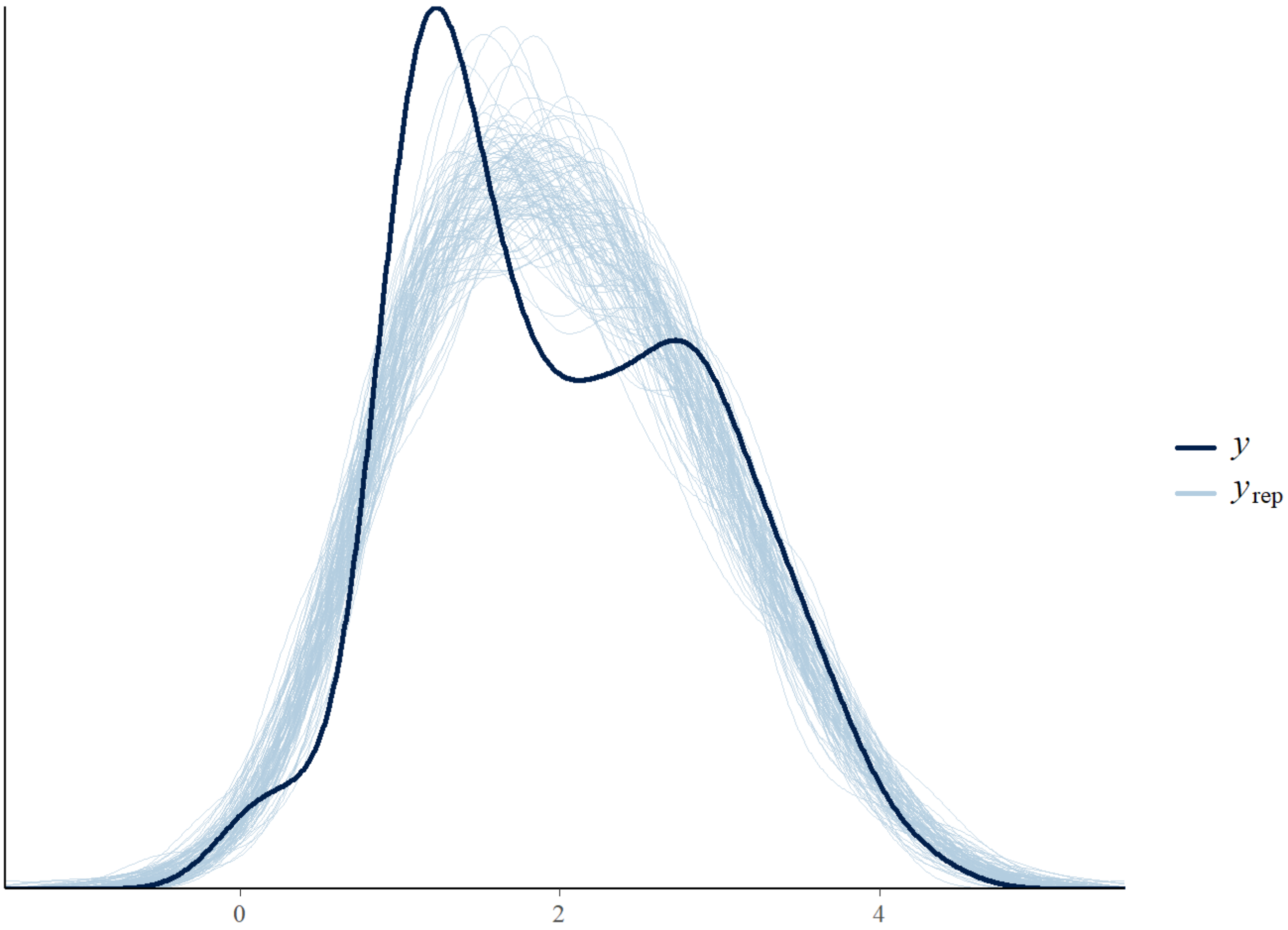
```
pp_check(m2, ndraws = 100)
```



```r
pp_check(m3, ndraws = 100)
```

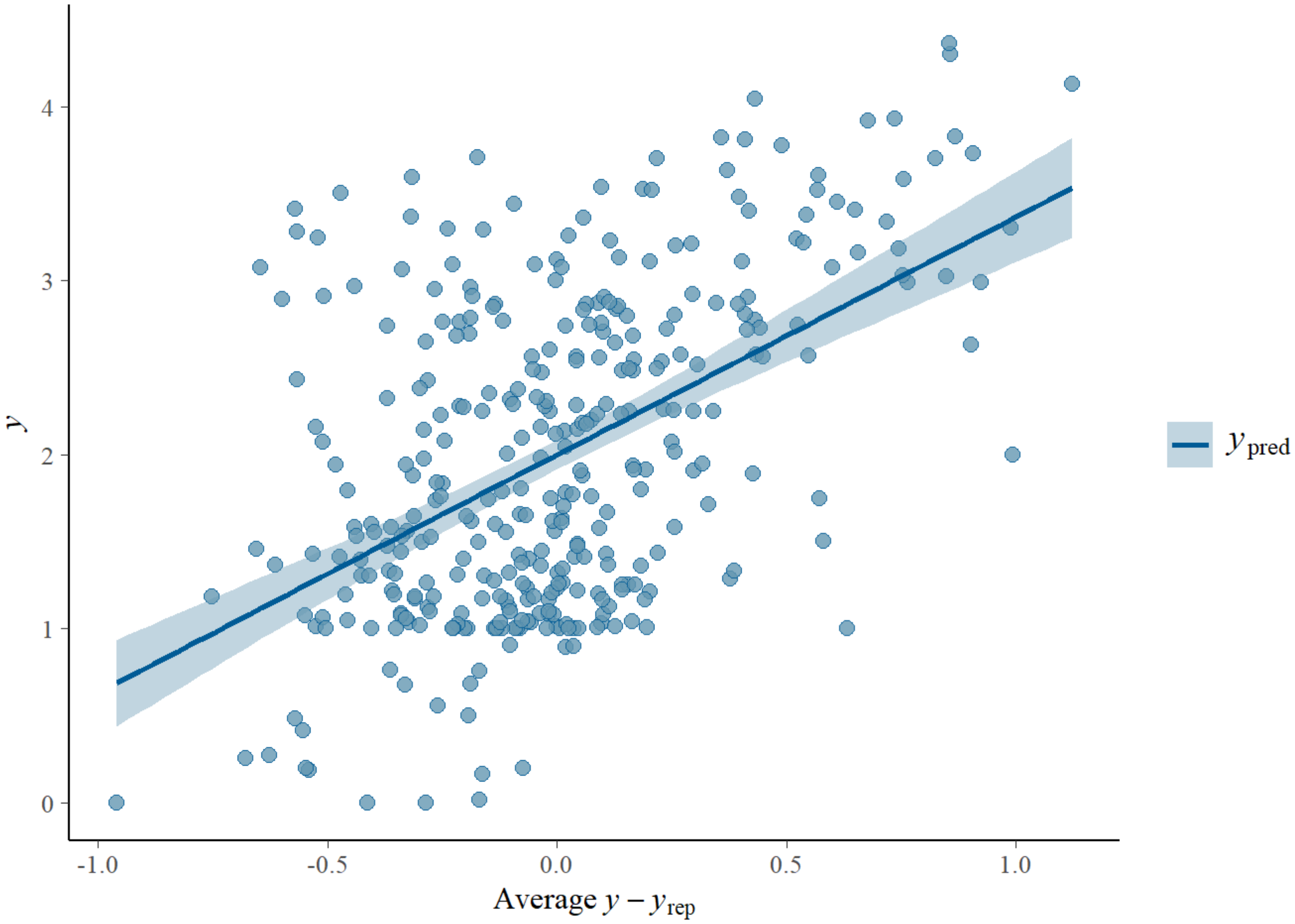


```
pp_check(m4, ndraws = 100)
```

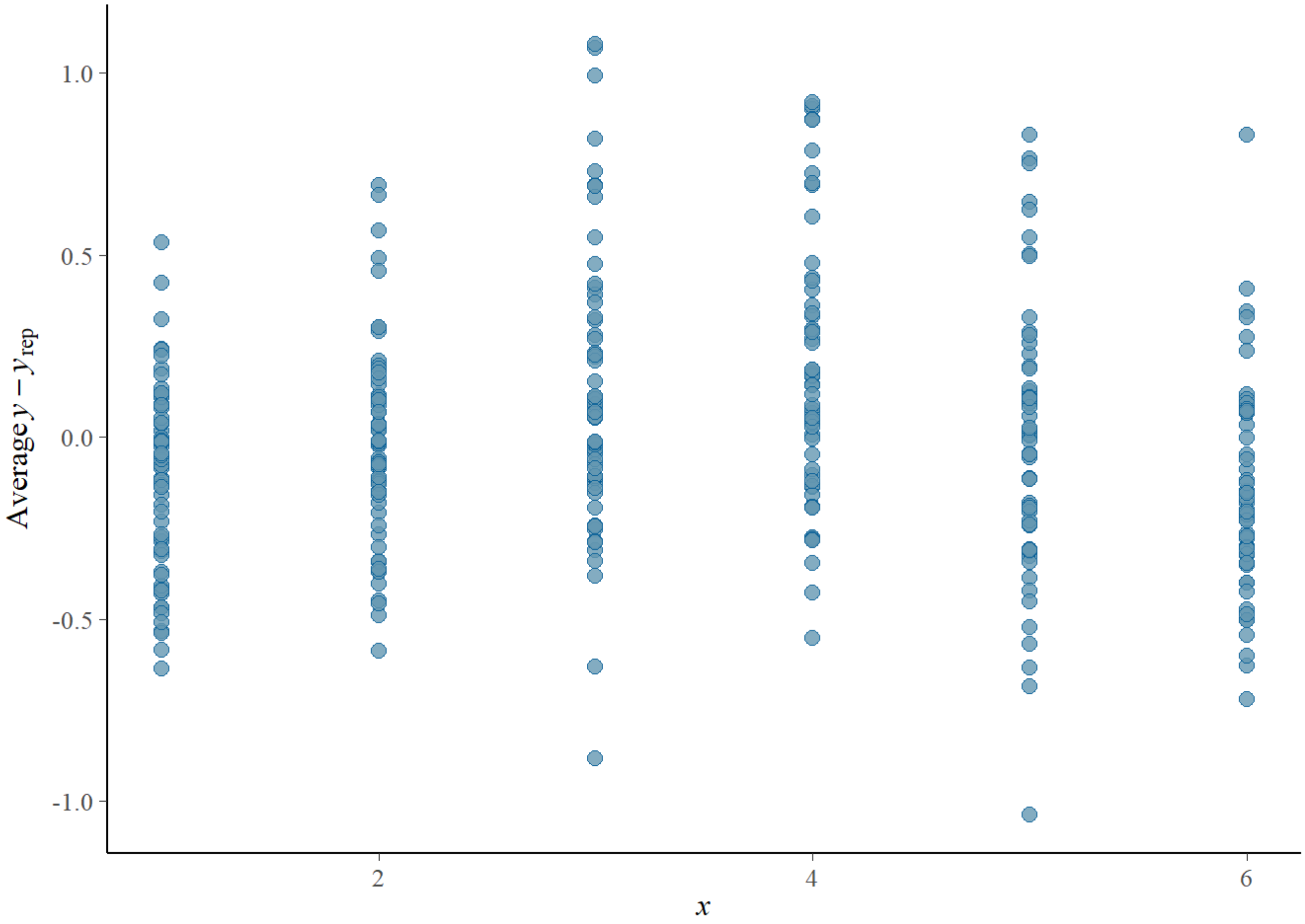


```
pp_check(m1, type = "error_scatter_avg", ndraws = 100) + geom_smooth(method = "lm")
```

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

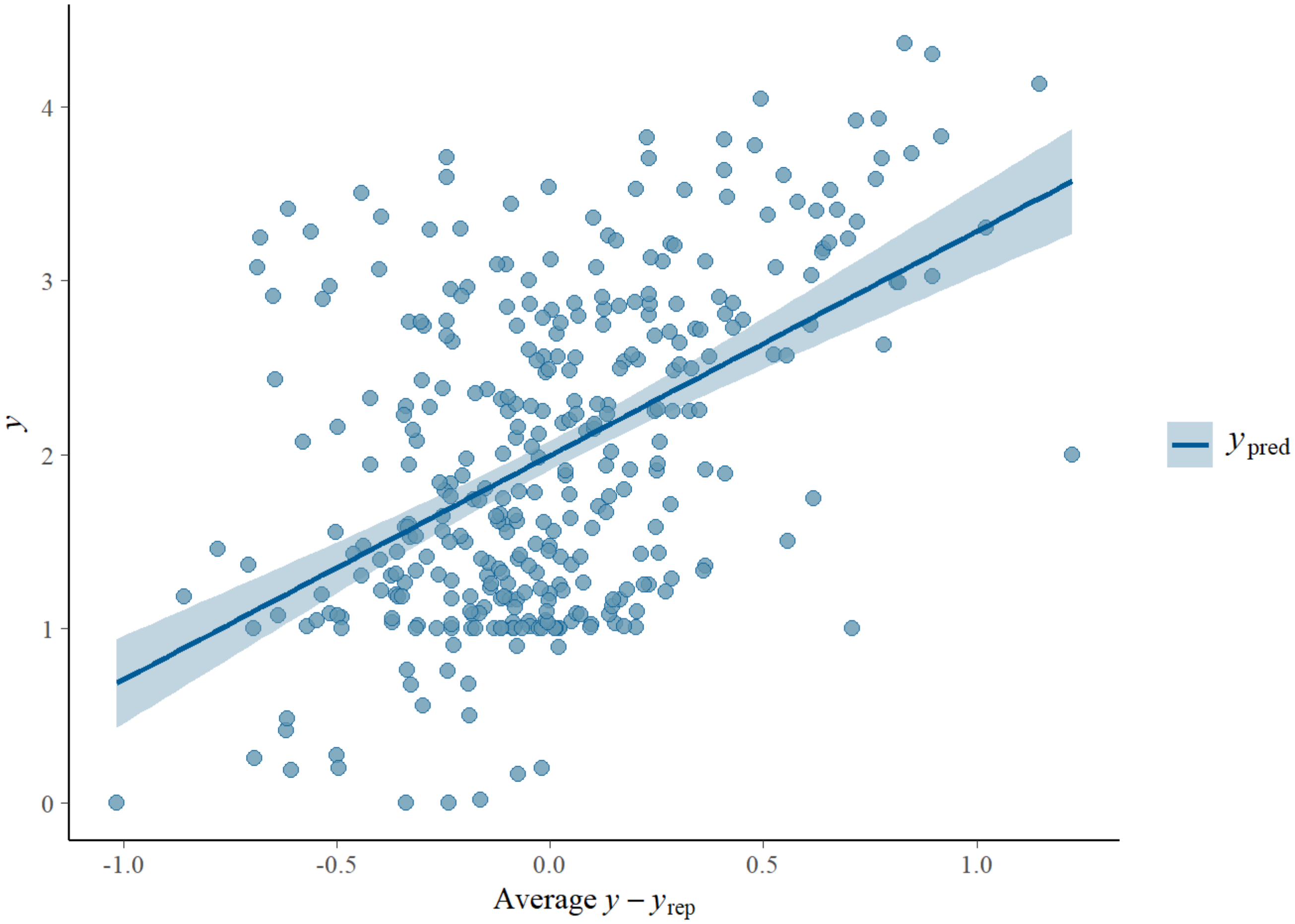


```
pp_check(m1, type = "error_scatter_avg_vs_x", x = "Visit", ndraws = 100)
```

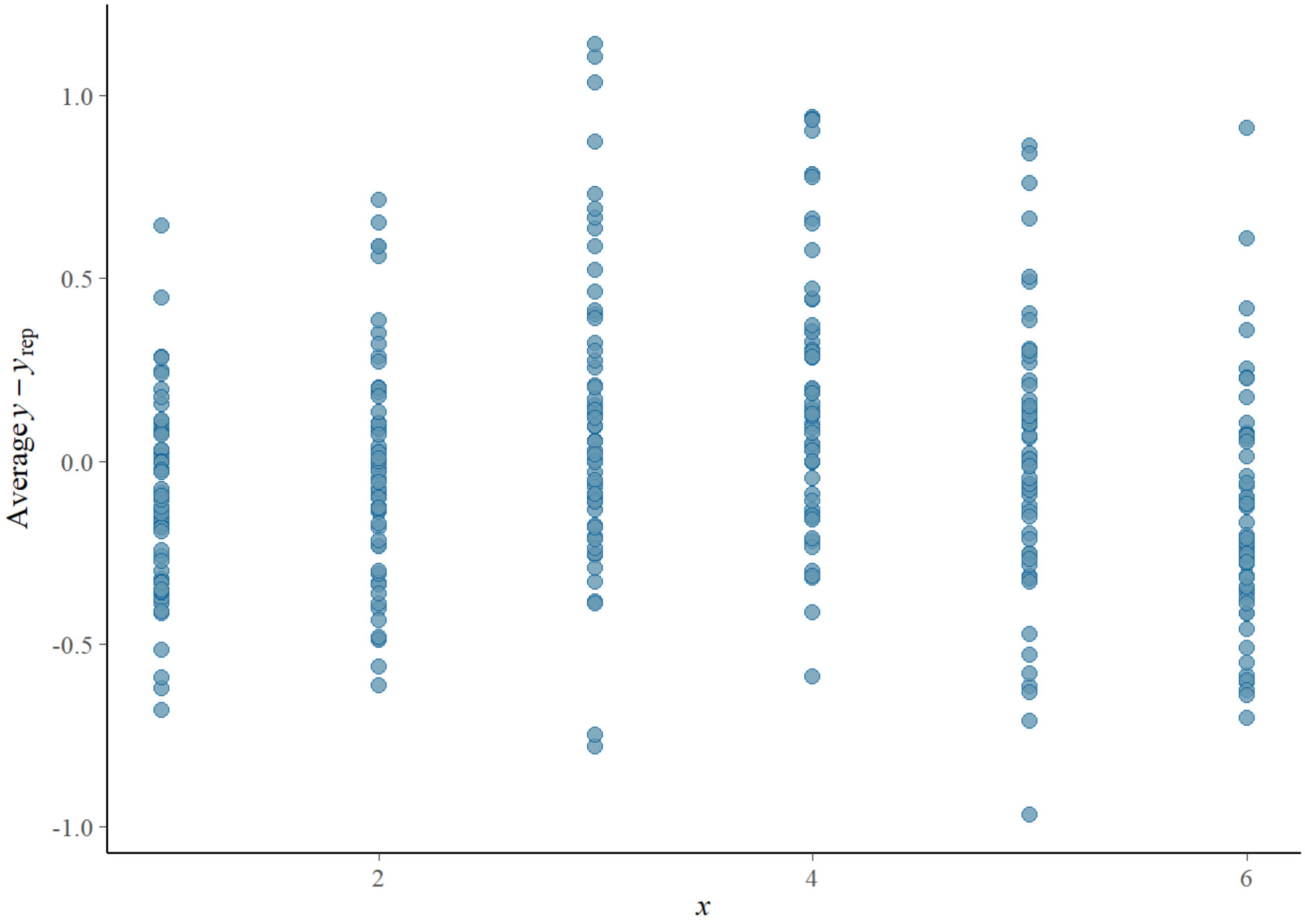


```
pp_check(m2, type = "error_scatter_avg", ndraws = 100) + geom_smooth(method = "lm")
```

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

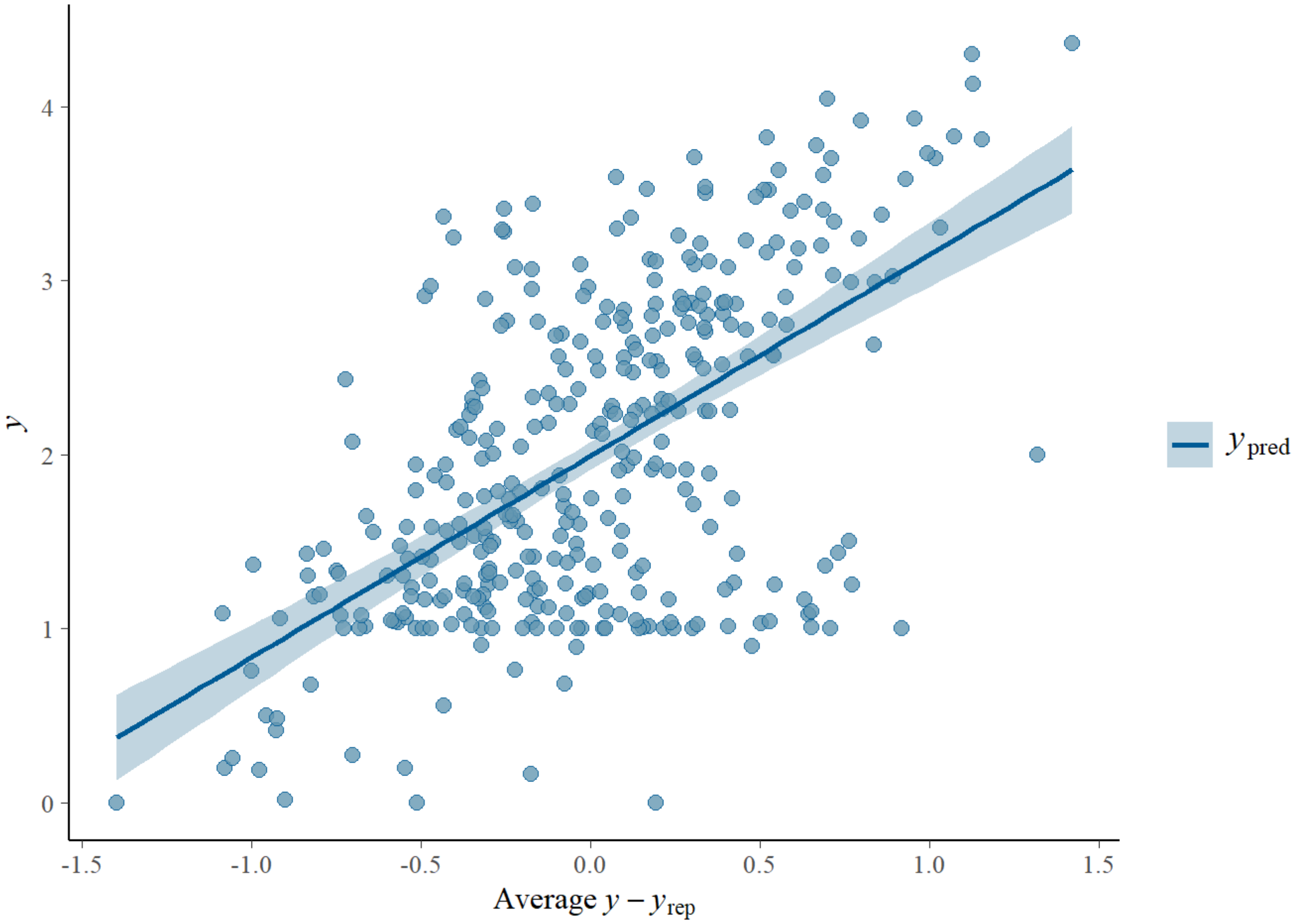


```
pp_check(m2, type = "error_scatter_avg_vs_x", x = "Visit", ndraws = 100)
```

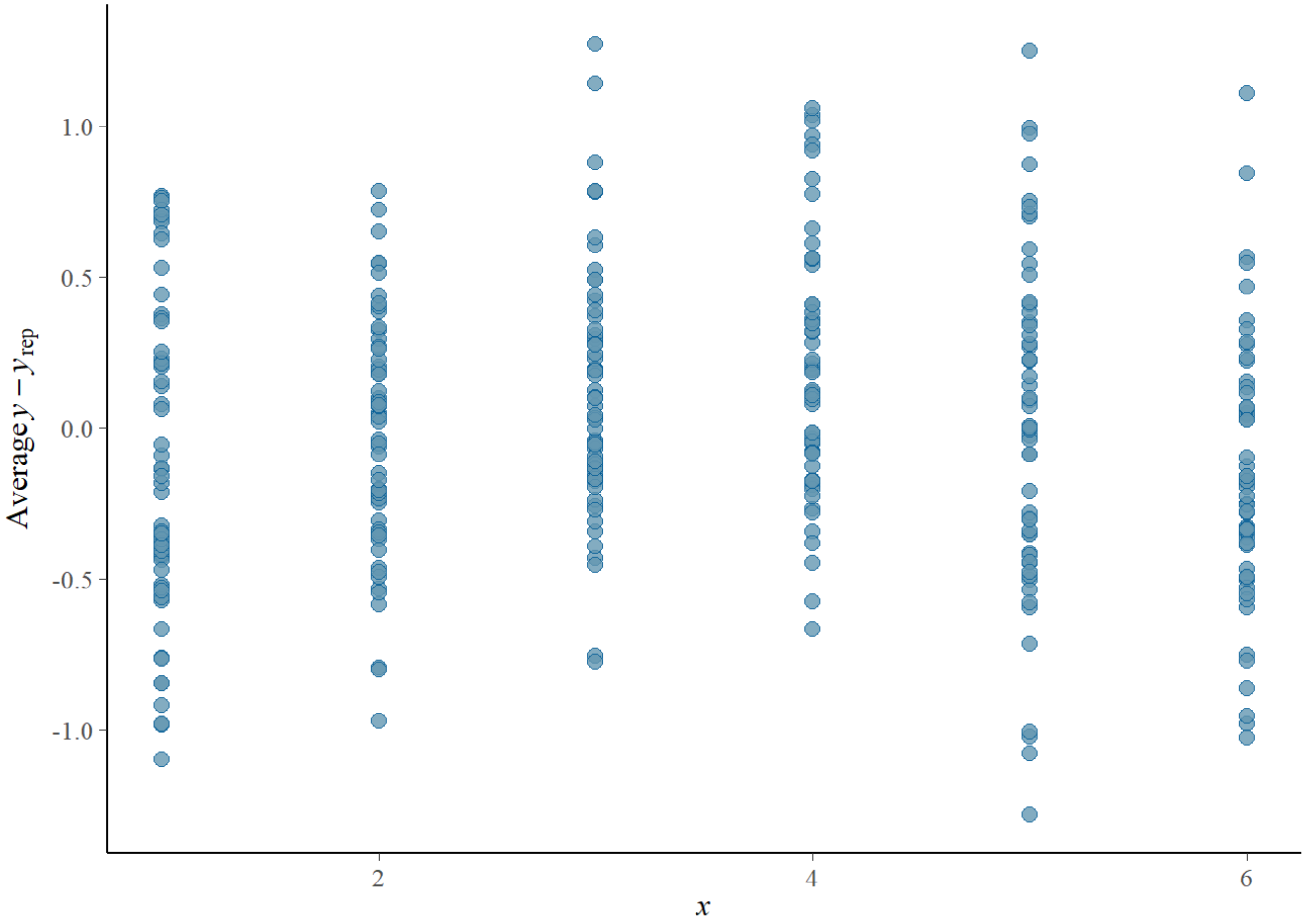


```
pp_check(m3, type = "error_scatter_avg", ndraws = 100) + geom_smooth(method = "lm")
```

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

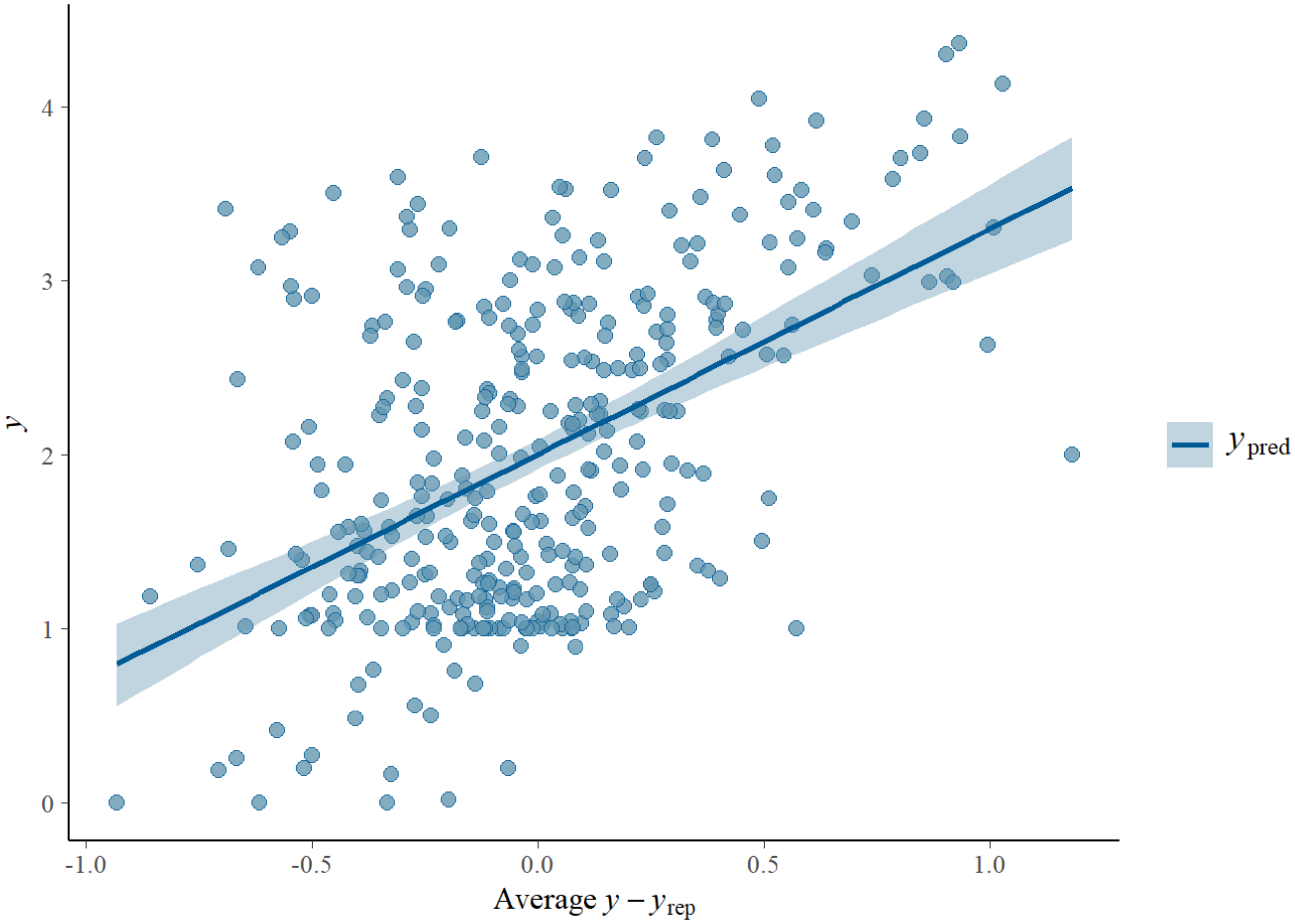


```
pp_check(m3, type = "error_scatter_avg_vs_x", x = "Visit", ndraws = 100)
```

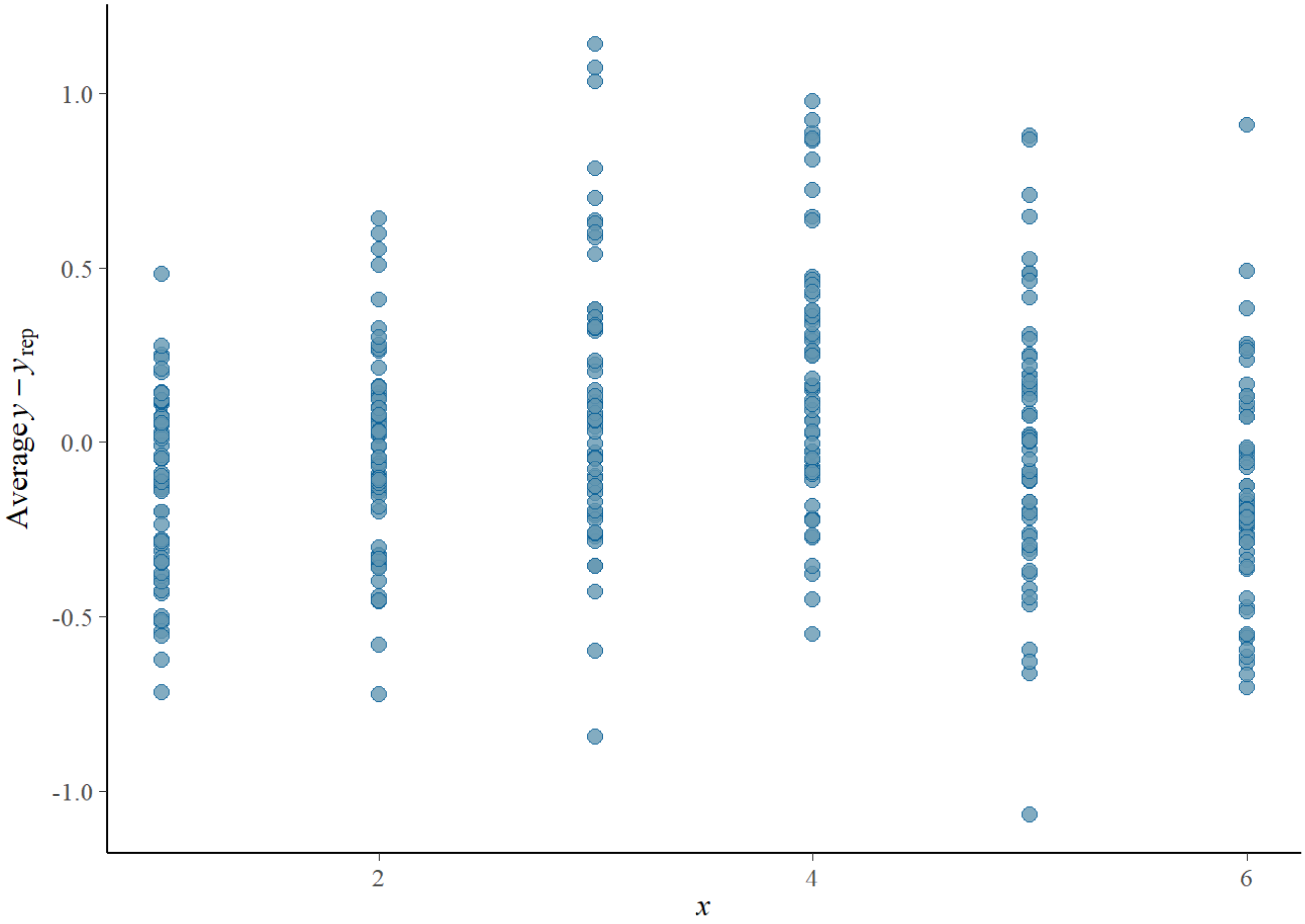


```
pp_check(m4, type = "error_scatter_avg", ndraws = 100) + geom_smooth(method = "lm")
```

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



```
pp_check(m4, type = "error_scatter_avg_vs_x", x = "Visit", ndraws = 100)
```



Model Comparison

```
m1 <- add_criterion(m1, criterion = "loo")
m2 <- add_criterion(m2, criterion = "loo")
m3 <- add_criterion(m3, criterion = "loo")
m4 <- add_criterion(m4, criterion = "loo")

# m1
# m2
# m3
m4
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: CHI_MLU ~ 1 + Visit + Diagnosis + (1 + Visit + ADOS1 + Socialization1 + verbal
IQ1 || Child.ID)
##    Data: data (Number of observations: 352)
##   Draws: 2 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 2000
##
## Group-Level Effects:
## ~Child.ID (Number of levels: 66)
##                      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)            0.11      0.09     0.00     0.33 1.01      276      378
## sd(Visit)                0.15      0.02     0.12     0.20 1.00      667      948
## sd(ADOS1)                0.01      0.01     0.00     0.03 1.01      602      452
## sd(Socialization1)       0.00      0.00     0.00     0.00 1.01      350      607
## sd(verbalIQ1)            0.02      0.00     0.02     0.03 1.00      633      726
##
## Population-Level Effects:
##                Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept          1.13      0.11     0.91     1.35 1.00     1085     1500
## Visit              0.23      0.02     0.19     0.28 1.00      605      920
## DiagnosisASD      -0.12      0.16    -0.43     0.18 1.00      876     1432
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.41      0.02     0.37     0.45 1.00     1246     1543
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
loo_compare( m1, m2, m3, m4)
```

```
##    elpd_diff se_diff
## m4   0.0       0.0
## m1  -2.9       2.0
## m2  -5.5       2.1
## m3 -55.0       9.4
```

```
loo_model_weights( m1, m2, m3, m4)
```

```
## Method: stacking
## ------
##    weight
## m1 0.000
## m2 0.000
## m3 0.008
## m4 0.992
```

```r
# bayes_R2(m1)
# bayes_R2(m2)
# bayes_R2(m3)
bayes_R2(m4)
```

```
##     Estimate  Est.Error      Q2.5     Q97.5
## R2 0.8132391 0.01323858 0.7842394 0.8352777
```

in loo_compare and loo_model_weights, Model 4 favored above all other models (0.992) Model 4 also has the highest Bayesian R2 (0.81) and lowest Est. error (0.013)

In the best model m4, CHI_MLU ~ 1 + Visit + Diagnosis + MOT_MLU + Gender+ Age+(1 + Visit + ADOS1 + Socialization1 + verbalIQ1 + Diagnosis||Child.ID) the intercept is estimated to be 1.13, falling between 0.91 and 1.35 95% of the time, meaning that the true population intercept for CHI_MLU lies between this value 95% of the time. Since the estimate falls within this range, it’s reasonable to conclude it is statistically plausible.

DiagnosisASD estimate is -0.12 with an interval of -0.43 to 0.18 (this includes zero, which might indicate that some ASD individuals have the same CHI_MLU as TD, meaning that they do not differ in a statistically significant manner). Compared to TD, this means that those with DiagnosisASD generally have a -0.12 lower value of CHI_MLU, but this is muddied up by the credible interval containing zero.

The standard deviation values grouped by Child.ID tell how the CHI_MLU values vary by Child.ID, for example the Intercept varies with 95% certainty between 0.0 and 0.33. Though, some of the sd values have fairly low ESS, meaning that they are not as reliable and may not have generated enough diverse samples. Either that, or its a convergence or model specification issue.

```
m1k <- kfold(
  m1,
  K = 10,
  folds = NULL,
  file = 'data/model1_kfold'
  )

m2k <- kfold(
  m2,
  K = 10,
  folds = NULL,
  file = 'data/model2_kfold'
)

m3k <- kfold(
  m3,
  K = 10,
  folds = NULL,
  file = 'data/model3_kfold'
)

m4k <- kfold(
  m4,
  K = 10,
  folds = NULL,
  file = 'data/model4_kfold'
)
```

```
m1k <- readRDS("data/model1_kfold.rds")
m2k <- readRDS("data/model2_kfold.rds")
m3k <- readRDS("data/model3_kfold.rds")
m4k <- readRDS("data/model4_kfold.rds")

m1k
m2k
m3k
m4k
```

Prediction

N.B. There are several data sets for this exercise, so pay attention to which one you are using!

1. The (training) data set from last time
2. The (test) data set on which you can test the models from last time:

- Demographic and clinical data (https://www.dropbox.com/s/ra99bdvm6fzay3g/demo_test.csv?dl=1)
- Utterance Length data (https://www.dropbox.com/s/uxtqqzl18nwxowq/LU_test.csv?dl=1)
- Word data (https://www.dropbox.com/s/1ces4hv8kh0stov/token_test.csv?dl=1)

Relying on the model(s) you trained in part 2 of the exercise, create predictions for the test set and assess how well they do compared to the actual data.

- Discuss the differences in performance of your model in training and testing data. Is the model any good?
- Let’s assume you are a speech therapy clinic. You want to assess whether the kids in your test sample will have a typical (like a TD) development, or they will have a worse one, in which case they should get speech therapy support. What do your predictions tell you about that? Which kids would you provide therapy for? Is the model any good?

```
## Rows: 36 Columns: 36
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): Child.ID, Ethnicity, Diagnosis, Birthdate
## dbl (29): Visit, ASD_check, ASD2, Gender, Total..Understands...Says., Total....
## num  (3): Age, Age2, CARS
##
## 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.
## Rows: 35 Columns: 12
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (2): SUBJ, VISIT
## dbl (10): MOT_MLU, MOT_LUstd, MOT_LU_q1, MOT_LU_q2, MOT_LU_q3, CHI_MLU, CHI_...
##
## 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.
## Rows: 35 Columns: 8
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (2): SUBJ, VISIT
## dbl (5): types_MOT, types_CHI, types_shared, tokens_MOT, tokens_CHI
## lgl (1): X
##
## 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.
```

Remove missing data to ease merging with predictions

```
demo %>%
  filter(Visit == 1) %>%
  select(-VinelandReceptive, -VinelandExpressive, -VinelandWritten)%>%
  summarise(Any_NA = any(is.na(.)))

demo <- demo %>%
  group_by(Child.ID) %>%
  mutate(ADOS = if_else(is.na(ADOS) & Visit != 1, first(ADOS), ADOS))%>% #kind of unnecessary since I only care about 1st value due to the model specification
  mutate(ADOS1 = first(ADOS))%>%
  mutate(Socialization1 = first(Socialization))%>%
  mutate(verbalIQ1 = first(ExpressiveLangRaw))%>%
  mutate(Diagnosis = if_else(Diagnosis == "A", "ASD", "TD"))%>%
  group_by(Child.ID) %>%
  mutate(Child.ID = cur_group_id()) #can do this since the order of the children is the same in each dataframe

uld <- uld %>%
  rename(Visit = VISIT) %>%
  mutate(Visit = as.integer(gsub("[^0-9]", "", Visit))) %>%
  rename(Child.ID = SUBJ) %>%
  group_by(Child.ID) %>%
  mutate(Child.ID = cur_group_id())

token <- token %>%
  rename(Child.ID = SUBJ) %>%
  group_by(Child.ID) %>% #can do this since the order of the children is the same in each dataframe
  mutate(Child.ID = cur_group_id())%>%
  rename(Visit = VISIT) %>%
  mutate(Visit = as.integer(gsub("[^0-9]", "", Visit)))%>%
  select(-X)

test_data <- inner_join(demo,uld)
```

```
## Joining with `by = join_by(Child.ID, Visit)`
```

```
test_data <- inner_join(test_data, token)
```

```
## Joining with `by = join_by(Child.ID, Visit)`
```

Here we should be using a model that'd have some more interesting predictors (to make sure we have something to predict)

Alternatively we could retrain the model to include visit 1 for all the test kids (and thus have the random effects)

```
model <- readRDS("data/model4_fit.rds")

training_data <- data %>%
  select(Child.ID, Visit, Diagnosis,MOT_MLU,Age, Socialization1, ADOS1, CHI_MLU, verbalIQ1)%>%
  na.omit()

test_data <- test_data %>%
  select(Child.ID, Visit, Diagnosis,MOT_MLU,Age, Socialization1, ADOS1, CHI_MLU, verbalIQ1)

#sloppily done with ChatGPT, probably is a way smarter way
datasets <- list(
  test = list(
    data = test_data,
    name = "Test Data"
  ),
  training = list(
    data = training_data,
    name = "Training Data"
  )
)

for (dataset in datasets) {
  posterior_samples <- posterior_predict(model, newdata = dataset$data, draws = 100, allow_new_levels = TRUE)
  predicted_values <- colMeans(posterior_samples)
  observed_values <- dataset$data$CHI_MLU

  model$residuals <- residuals(model)

  rmse_value <- sqrt(mean((observed_values - predicted_values)^2))

  cat("\n", dataset$name, ":\n")
  cat("RMSE:", rmse_value)
}
```

```
##
##  Test Data :
## RMSE: 1.107198
##  Training Data :
## RMSE: 0.3476537
```

Discuss the differences in performance of your model in training and testing data. Is the modelany good?

The higher RMSE on the test data indicates that the model is likely overfit on the training data and does not generalize particularly well to new data. This is somewhat to be expected, however my best model is likely not good enough given how big of an error 1.1 is.

The error is rather large considering how small the value range is:

```
data %>%
  group_by(Diagnosis)%>%
  summarize(
    mean_MLU = mean(CHI_MLU, na.rm = TRUE),
    max_MLU = max(CHI_MLU, na.rm = TRUE),
    min_MLU = min(CHI_MLU, na.rm = TRUE)
  )
```

```
## # A tibble: 2 × 4
##   Diagnosis mean_MLU max_MLU min_MLU
##   <fct>        <dbl>   <dbl>   <dbl>
## 1 TD            2.31    4.36   0.558
## 2 ASD           1.64    4.30   0
```

Show how ASD child fare in Child MLU compared to the average TD child at each visit

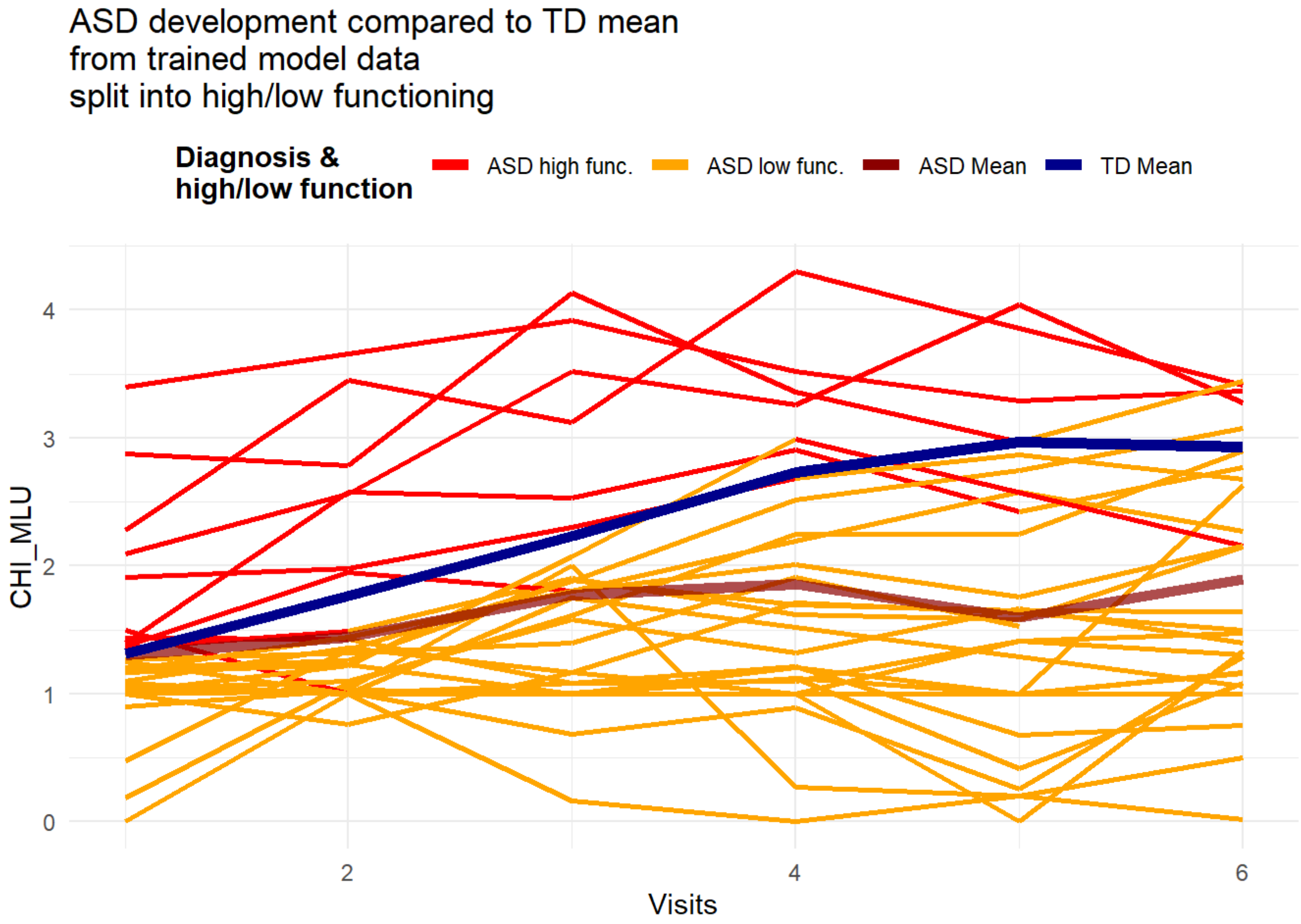
```
model <- readRDS("data/model4_fit.rds")
model_data <- model$data

#the mean is being based on the TD mean, "the average TD child at each visit"
td_mean_data <- model_data %>%
  subset(Diagnosis == "TD")%>%
  group_by(Visit) %>%
  summarize(Mean_CHI_MLU = mean(CHI_MLU))

asd_mean_data <- model_data %>%
  subset(Diagnosis == "ASD")%>%
  group_by(Visit) %>%
  summarize(Mean_CHI_MLU = mean(CHI_MLU))

#adding low and high functioning for the sake of the plot
model_data <- model_data %>%
  left_join(td_mean_data, by = "Visit") %>%
  mutate(functioning = ifelse(CHI_MLU >= Mean_CHI_MLU, "high", "low"))

asd_data <- subset(model_data, Diagnosis == "ASD")
td_data <- subset(model_data, Diagnosis == "TD")
```

```
#ratio between high and low functioning between diagnoses & visits (ignoring Child.id)
model_data %>%
  group_by(Diagnosis, Visit) %>%
  summarize(
    high_ratio = sum(functioning == "high") / n(),
    low_ratio = sum(functioning == "low") / n(),
    mean_CHI_MLU_diff = mean(CHI_MLU) - mean(Mean_CHI_MLU)
  ) %>%
  arrange(Visit)
```

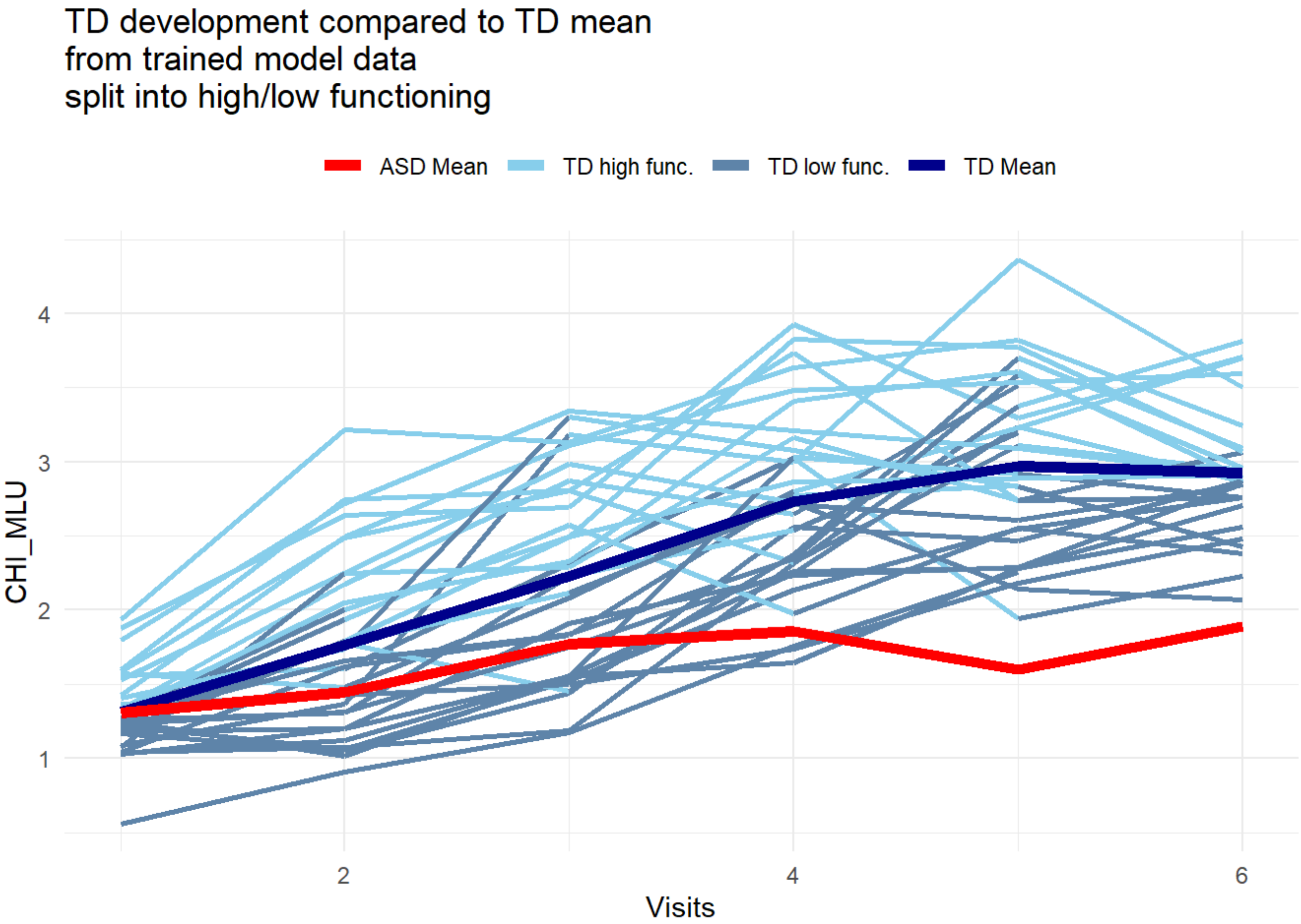
```
## `summarise()` has grouped output by 'Diagnosis'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 12 × 5
## # Groups:   Diagnosis [2]
##    Diagnosis Visit high_ratio low_ratio mean_CHI_MLU_diff
##    <fct>     <dbl>      <dbl>     <dbl>             <dbl>
##  1 TD            1     0.406     0.594           0
##  2 ASD           1     0.345     0.655          -0.00570
##  3 TD            2     0.469     0.531           0
##  4 ASD           2     0.214     0.786          -0.319
##  5 TD            3     0.516     0.484           0
##  6 ASD           3     0.214     0.786          -0.458
##  7 TD            4     0.469     0.531           0
##  8 ASD           4     0.222     0.778          -0.876
##  9 TD            5     0.484     0.516           0
## 10 ASD           5     0.0769    0.923          -1.37
## 11 TD            6     0.393     0.607           0
## 12 ASD           6     0.179     0.821          -1.04
```

```
ggplot() +
  geom_line(data = asd_data, aes(x = Visit, y = CHI_MLU, group = Child.ID, color = paste(D
       iagnosis, functioning, "func.")), size = 1, linetype = "solid") +
  geom_line(data = asd_mean_data, aes(x = Visit, y = Mean_CHI_MLU, color = "ASD Mean"), si
       ze = 2, linetype = "solid", alpha = 0.7) +
  geom_line(data = td_mean_data, aes(x = Visit, y = Mean_CHI_MLU, color = "TD Mean"), size
       = 2) +
  labs(x = "Visits", y = "CHI_MLU", color = "Diagnosis &\nhigh/low function") +
  scale_color_manual(values = c("ASD high func." = "red", "ASD low func." = "orange", "ASD
       Mean" = "darkred", "TD Mean" = "darkblue")) +
  theme_minimal() +
  theme(legend.position = "top", legend.title = element_text(face = "bold"))+
  ggtitle("ASD development compared to TD mean\nfrom trained model data\nsplit into high/l
       ow functioning")
```



```
ggplot() +
    geom_line(data = td_data, aes(x = Visit, y = CHI_MLU, group = Child.ID, color = paste
        (Diagnosis, functioning, "func.")), size = 1, linetype = "solid") +
  geom_line(data = td_mean_data, aes(x = Visit, y = Mean_CHI_MLU, color = "TD Mean"), size
        = 2) +
  labs(x = "Visits", y = "CHI_MLU", color = "") +
    geom_line(data = asd_mean_data, aes(x = Visit, y = Mean_CHI_MLU, color = "ASD Mean"),
        size = 2, linetype = "solid") +
  scale_color_manual(values = c("TD high func." = "skyblue", "TD low func." = "#5e84a9",
        "TD Mean" = "darkblue", "ASD Mean" = "red")) +
  theme_minimal() +
  theme(legend.position = "top", legend.title = element_text(face = "bold"))+
  ggtitle("TD development compared to TD mean\nfrom trained model data\nsplit into high/lo
        w functioning")
```



TD children have low within-group variance and tend to develop in a predictable manner. ASD children have high within-group variance, with there being both “high-functioning” and “low-functioning” ASD individuals with very different CHI_MLU development trajectories.

Generally, the first visit CHI_MLU is predictive of future ASD and TD CHI_MLU development, as those who begin below average generally stay below average, and those who begin above average stay that way as well.

Let’s assume you are a speech therapy clinic. You want to assess whether the kids in your test sample will have a typical (like a TD) development, or they will have a worse one, in which case they should get speech therapy support. What do your predictions tell you about that? Which kids would you provide therapy for? Is the model any good?

Since 1st visit CHI_MLU is rather predictive of the development of the child’s MLU, any ASD child with a value below the TD mean on their first visit may possibly benefit from speech therapy. Since there are also TD children who struggle with MLU, they may also benefit from speech therapy should their MLU be below average on the 1st and consequent visits. That said, they veer off from the mean expected value much less, meaning that ASD children with MLU issues are the ones that (generally) need the speech therapy the most.