A_2.final

Study group 3

2022-11-04

Loading required packages

```
pacman::p_load(glue,
       tidyr,
       data.table,
       moments,
       tidybayes,
       tibble,
       cowplot,
       viridis,
       brms,
       stringr,
       rstan,
       cmdstanr,
       magrittr,
       gridExtra,
       grid,
       lattice,
       ggplot2,
       ggridges,
       ellipse,
       Rmisc,
       dplyr,
       "rmarkdown",
       knitr)
pacman::p_load(tidyverse)
```

```
## Error in completeSubclasses(classDef2, class1, obj, where) :
##   trying to get slot "subclasses" from an object of a basic class ("NULL") with
no slots
##
## The downloaded binary packages are in
##  /var/folders/hf/2cjx77xd24b01rtsfd8v4mmh0000gn/T//RtmpGE2FYm/downloaded_packag
es
## Error in completeSubclasses(classDef2, class1, obj, where) :
##   trying to get slot "subclasses" from an object of a basic class ("NULL") with
no slots
```

```
pacman::p_load(purrr)
pacman::p_load(MCMCglmm)
pacman::p_load(readxl)
pacman::p_load(metafor)
```

Assignment 2: meta-analysis

Questions to be answered

1. Simulate data to setup the analysis and gain insight on the structure of the problem. Simulate one dataset of 100 studies (n of participants should follow a normal distribution with mean of 20, sd of 10, but no fewer than 10 participants), with a mean effect size of 0.4, average deviation by study of .4 and measurement error of .8. The data you get should have one row per study, with an effect size mean and standard error. Build a proper bayesian model to analyze the simulated data. Then simulate publication bias (only some of the studies you simulate are likely to be published, which?), the effect of publication bias on your estimates (re-run the model on published studies, assess the difference), and use at least one technique to assess publication bias. remember to use at least one plot to visualize your results. BONUS question: do a power/precision analysis.

2. What is the current evidence for distinctive vocal patterns in schizophrenia? Use the data from Parola et al (2020) - https://www.dropbox.com/s/0l9ur0gaabr80a8/Matrix_MetaAnalysis_Diagnosis_updated290719.xlsx?dl=0 (https://www.dropbox.com/s/0l9ur0gaabr80a8/Matrix_MetaAnalysis_Diagnosis_updated290719.xlsx?dl=0) - focusing on pitch variability (PITCH_F0SD). Describe the data available (studies, participants). Using the model from question 1 analyze the data, visualize and report the findings: population level effect size; how well studies reflect it; influential studies, publication bias. BONUS question: add the findings from https://www.medrxiv.org/content/10.1101/2022.04.03.22273354v2 (https://www.medrxiv.org/content/10.1101/2022.04.03.22273354v2). BONUS question: assess the effect of task on the estimates (model comparison with baseline model)

Sara

Question 1

Simulation

Outlining prior parameter provided by the assignment description

```
mean_effect <- 0.4
effect_sd <- 0.4
meas_error <- 0.8
par_mean <- 20
par_sd <- 10
n <- 100
```

A simulation of participant data of multiple visits using the provided data

```
set.seed(954)

sim_studies <-
  tibble(
    study_ID = seq(1:n),
    n_participants =
        round(rtnorm(n, mean=par_mean, sd=par_sd, lower=10))
    )


for (i in seq(nrow(sim_studies))){
    sim_studies$study_effect[i] <-
      rnorm(1,mean_effect,effect_sd)
    temp <-
      rnorm(sim_studies$n_participants[i],sim_studies$study_effect[i], meas_error)
    sim_studies$mean_effect_size[i] <-
      mean(temp)
    sim_studies$sd_effect[i] <-
      sd(temp)
    sim_studies$standard_error[i] <-
      sim_studies$sd_effect[i]/sqrt(sim_studies$n_participants[i])
  }
```

Bayesian model

A Bayesian model illustrating potential effect sizes on individual participants

```
model_study <- bf(mean_effect_size|se(standard_error) ~1 + (1|study_ID))
```

Priors

Generating prior data simulations to model, using parameters provided in class

```
get_prior(data = sim_studies, family = gaussian, model_study)
```

```
##                    prior     class      coef    group resp dpar nlpar lb ub
##  student_t(3, 0.3, 2.5) Intercept
##    student_t(3, 0, 2.5)        sd                                     0
##    student_t(3, 0, 2.5)        sd           study_ID                  0
##    student_t(3, 0, 2.5)        sd Intercept study_ID                  0
##        source
##       default
##       default
##  (vectorized)
##  (vectorized)
```

```
priors <- c(
  prior(normal(0, 0.3), class=Intercept),
  prior(normal(0, 0.3), class=sd))
```

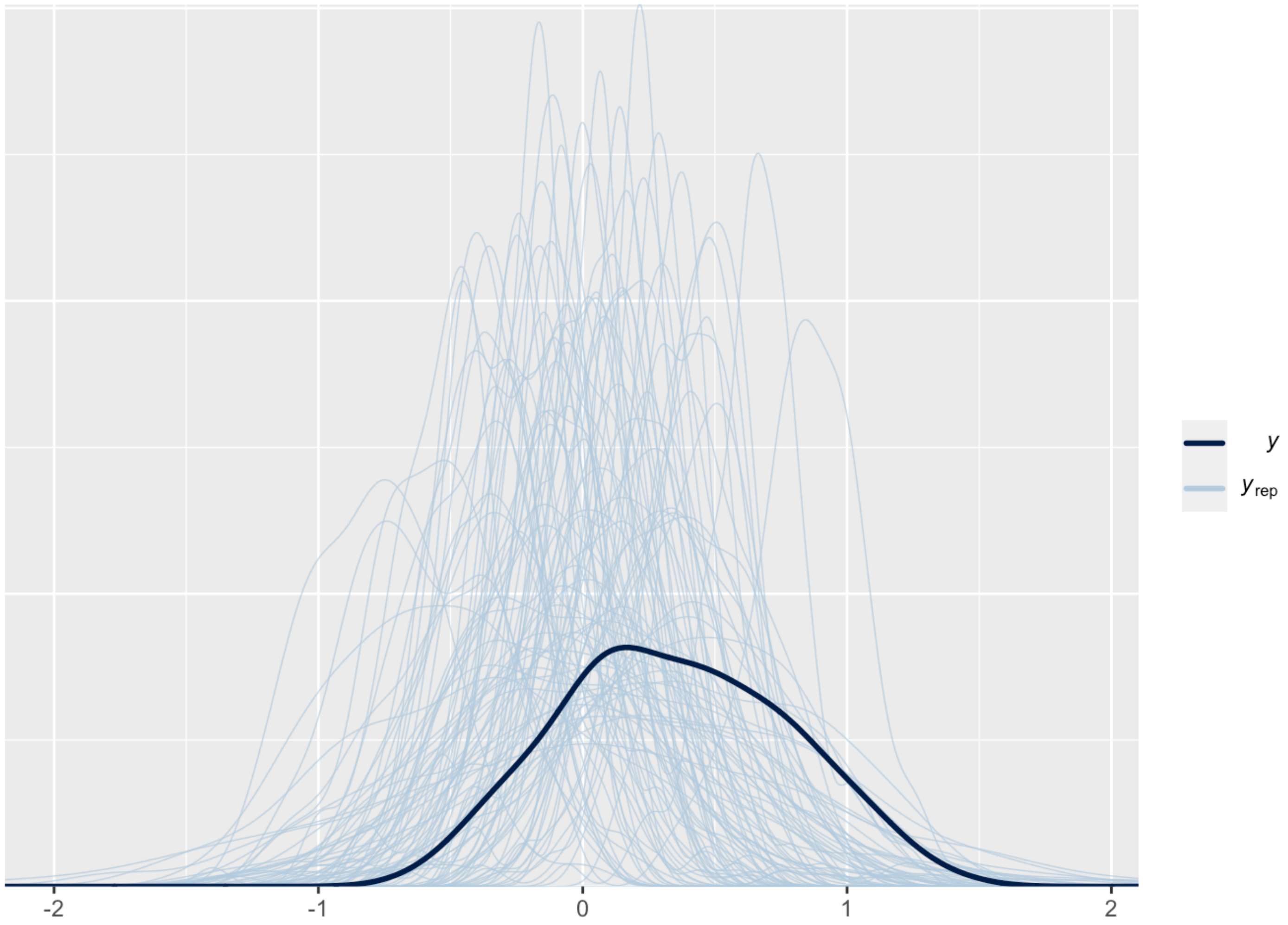
Model

Modeling using sample_prior = ‘only’

```
model_prior <- brm(
  model_study,
  data = sim_studies,
  prior = priors,
  family = gaussian,
  refresh=0,
  sample_prior = 'only',
  iter=10000,
  warmup = 1000,
  backend = "cmdstanr",
  threads = threading(2),
  chains = 2,
  cores = 2,
  control = list(
    adapt_delta = 0.99,
    max_treedepth = 20
)
)
```

```
## Running MCMC with 2 parallel chains, with 2 thread(s) per chain...
##
## Chain 1 finished in 1.6 seconds.
## Chain 2 finished in 1.6 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 1.6 seconds.
## Total execution time: 1.6 seconds.
```

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



## Manuela

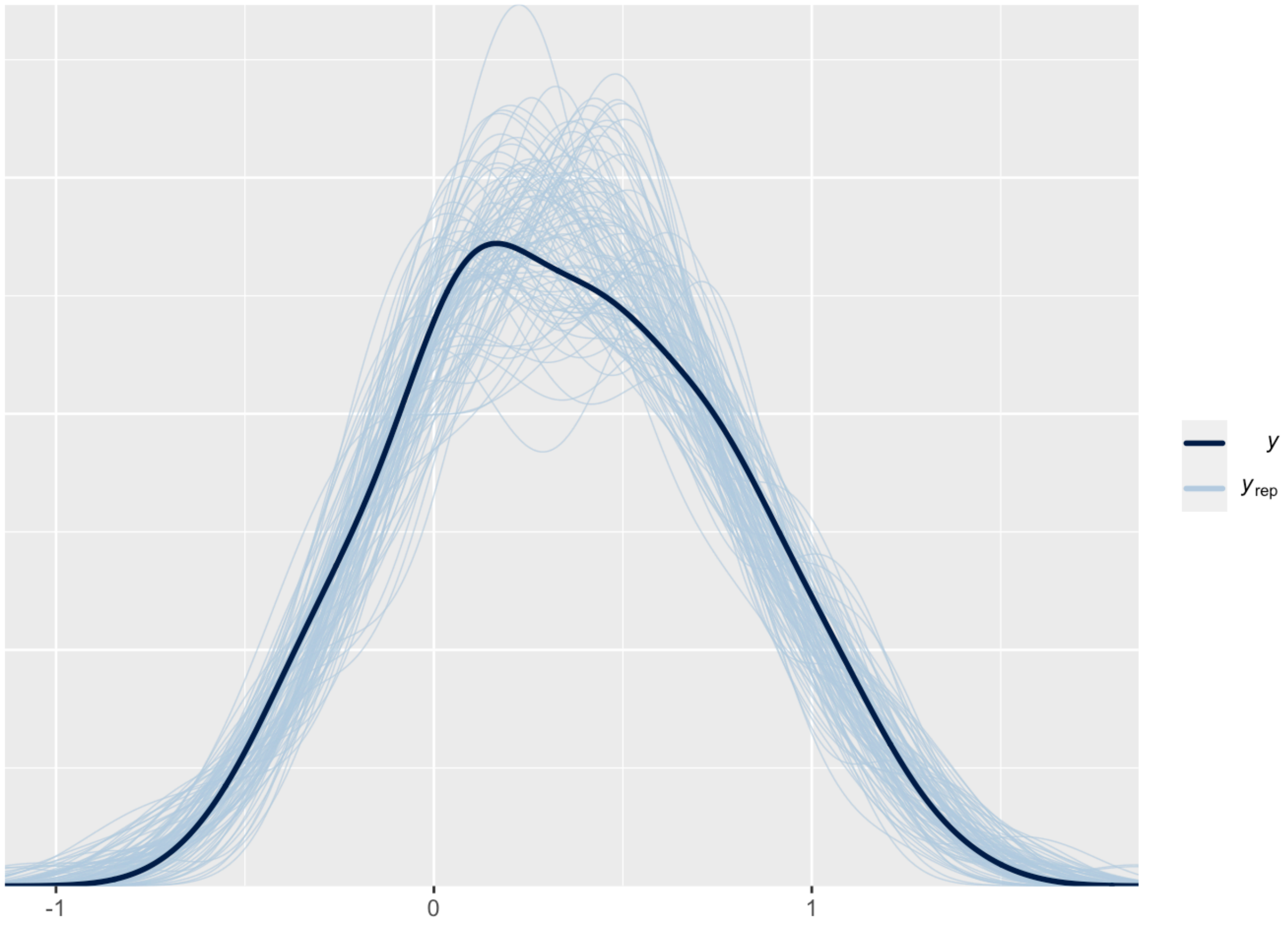
Fitting model

Modeling the sampled priors along with the simulation

```
model_prior_fit <- brm(
  model_study,
  data = sim_studies,
  prior = priors,
  family = gaussian,
  refresh=0,
  sample_prior = TRUE,
  iter=10000,
  warmup = 1000,
  backend = "cmdstanr",
  threads = threading(2),
  chains = 2,
  cores = 2,
  control = list(
    adapt_delta = 0.99,
    max_treedepth = 20
)
)
```

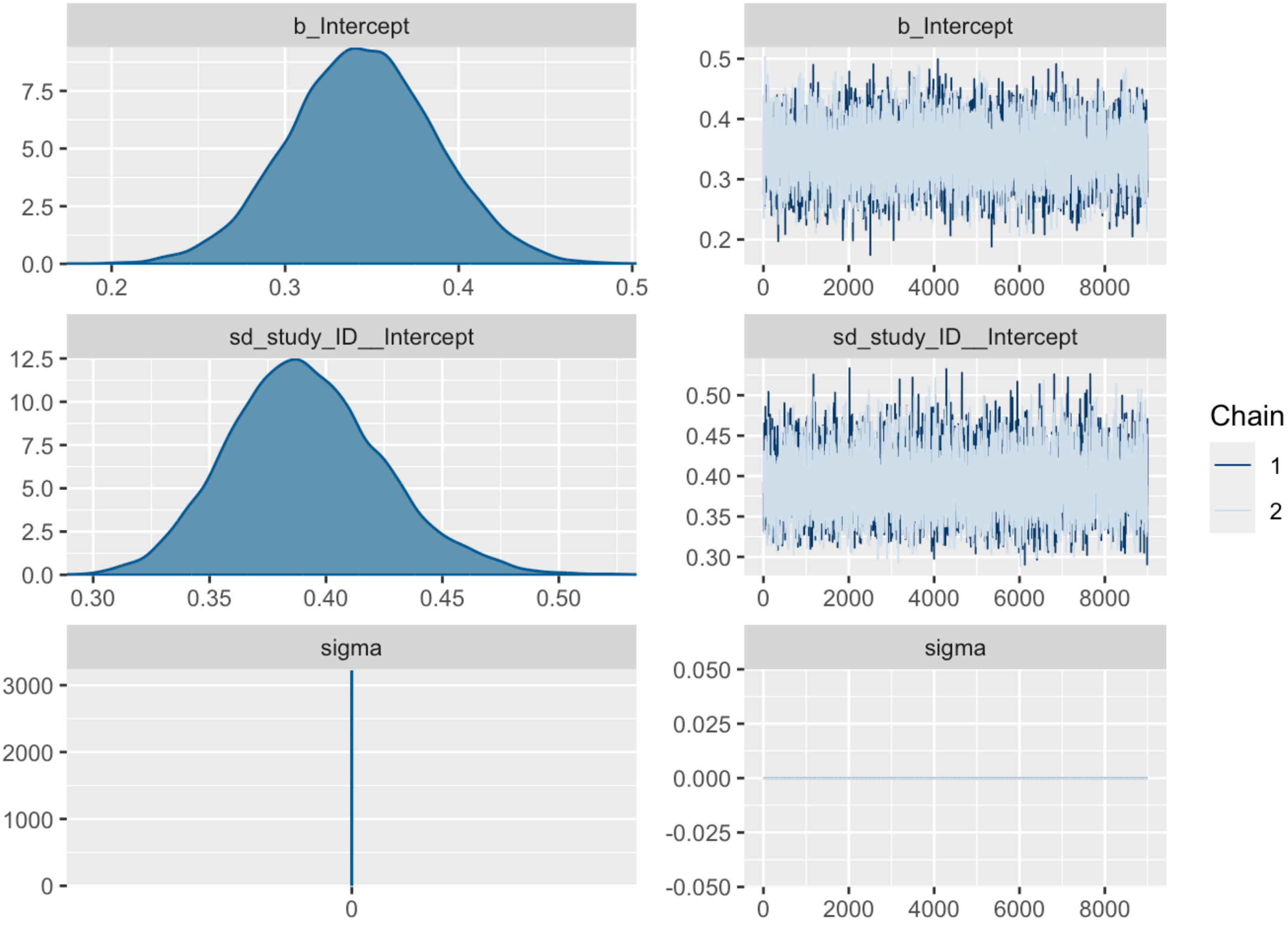
```
## Running MCMC with 2 parallel chains, with 2 thread(s) per chain...
##
## Chain 1 finished in 3.0 seconds.
## Chain 2 finished in 4.7 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 3.9 seconds.
## Total execution time: 4.7 seconds.
```

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



#### Plotting and visualizing

```
plot(model_prior_fit)
```



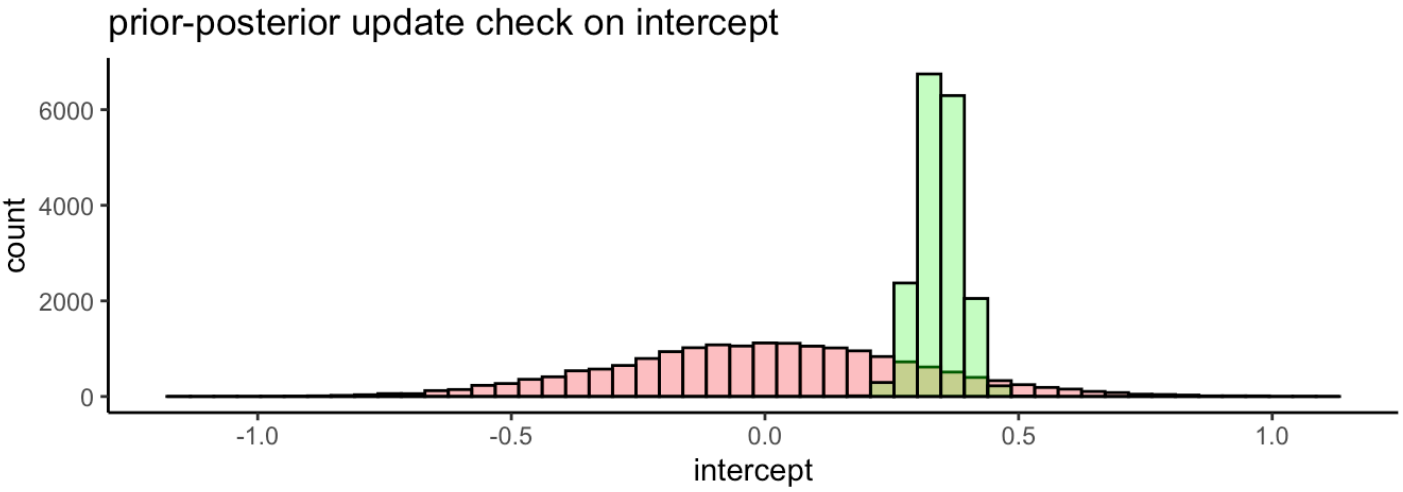
```
summary(model_prior_fit)
```

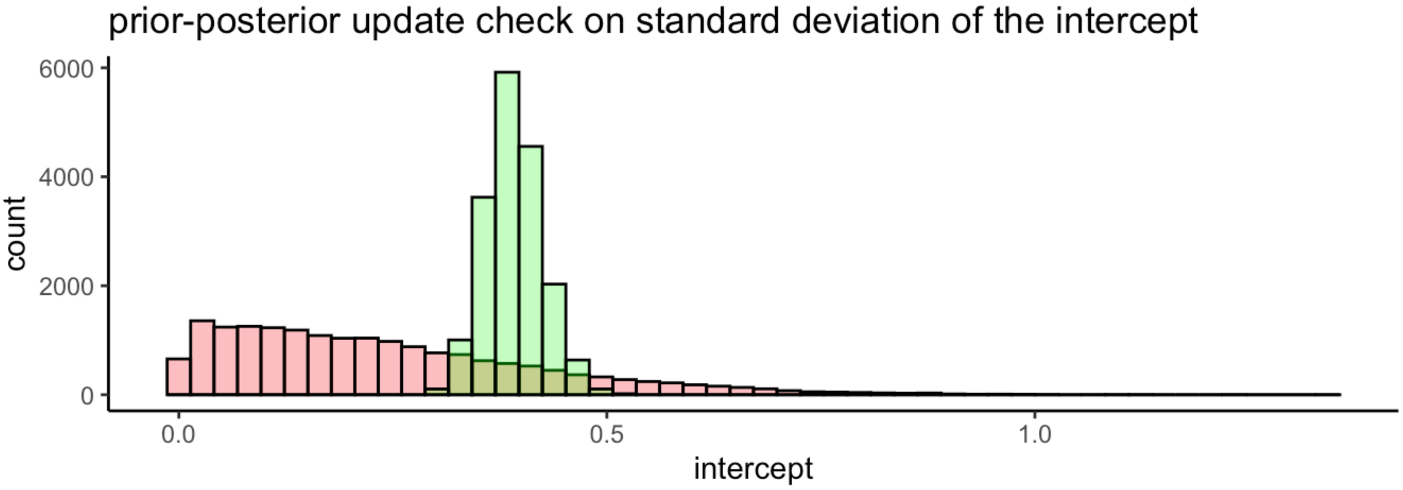
```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: mean_effect_size | se(standard_error) ~ 1 + (1 | study_ID)
##    Data: sim_studies (Number of observations: 100)
##   Draws: 2 chains, each with iter = 10000; warmup = 1000; thin = 1;
##          total post-warmup draws = 18000
##
## Group-Level Effects:
## ~study_ID (Number of levels: 100)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.39      0.03     0.33     0.46 1.00     5268     8430
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.34      0.04     0.26     0.43 1.00     4104     7005
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.00      0.00     0.00     0.00   NA       NA       NA
##
## 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).
```

Prior posterior update check

Plotting “prior-posterior update check on intercept” and “prior-posterior update check on standard deviation of the intercept”

```
model_posterior <- as_draws_df(model_prior_fit)

plot1 <- ggplot(model_posterior)+
  geom_histogram(aes(prior_Intercept), fill='red', color='black', alpha=0.3, bins=
50)+
  geom_histogram(aes(Intercept), fill='green', color='black', alpha=0.3, bins=50)+
  theme_classic()+
  ggtitle('prior-posterior update check on intercept')+
  xlab('intercept')
plot2 <- ggplot(model_posterior)+
  geom_histogram(aes(prior_sd_study_ID), fill='red', color='black', alpha=0.3, bin
s=50)+
  geom_histogram(aes(sd_study_ID__Intercept), fill='green', color='black', alpha=0
.3, bins=50)+
  theme_classic()+
  ggtitle('prior-posterior update check on standard deviation of the intercept')+
  xlab('intercept')
grid.arrange(plot1, plot2)
```





Simulation of publication bias, the effect of publication bias on our estimate and asses the publication bias (remember to visualize our results)

Simulating the effect size of the publication factors and filtering data for only published studies

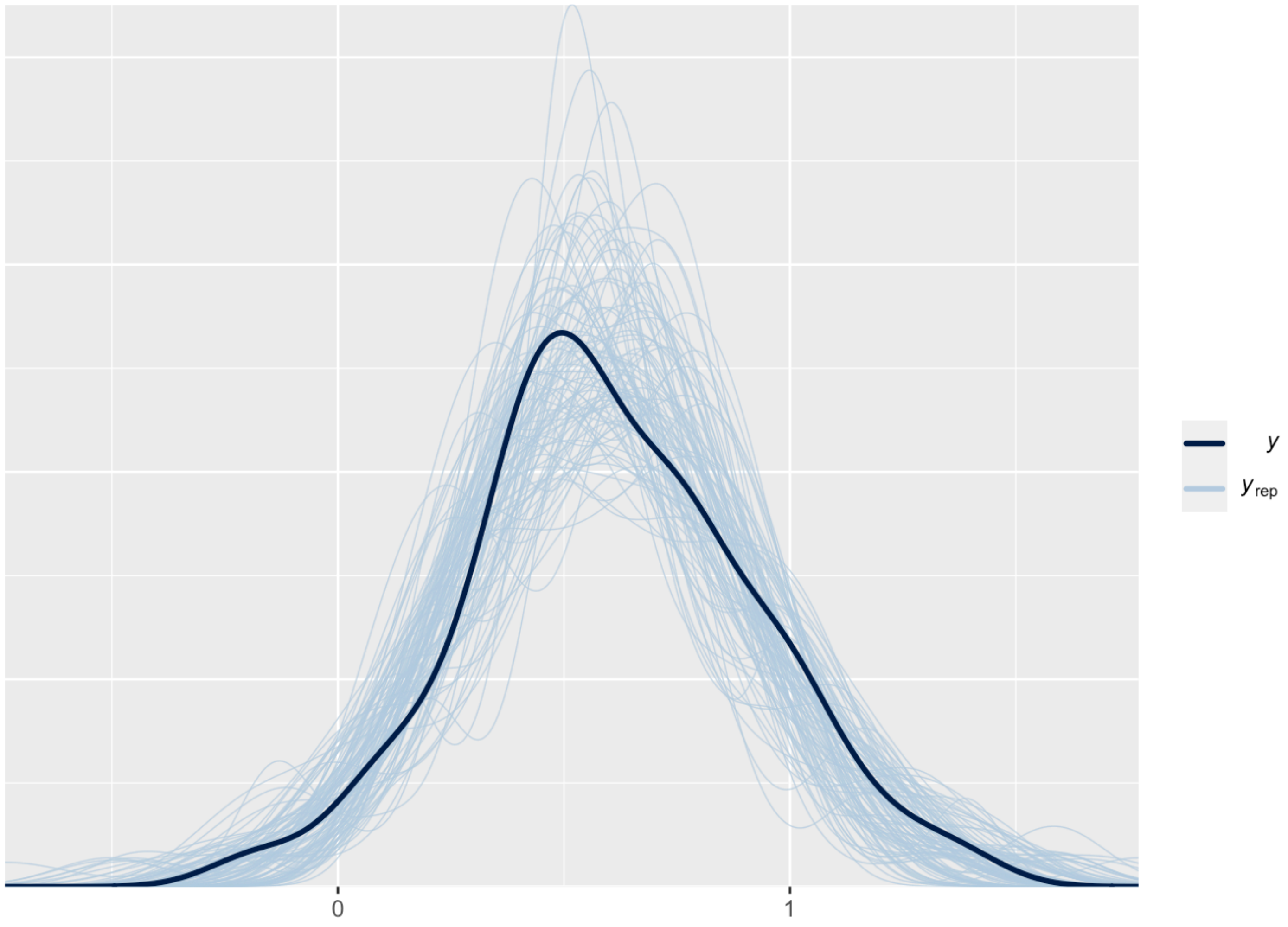
```
set.seed(843)

for (i in seq(nrow(sim_studies))){
  sim_studies$published[i] <-
    ifelse(abs(
      sim_studies$mean_effect_size[i])-(2*sim_studies$standard_error[i])>0
      & sim_studies$mean_effect_size[i]>0,
      rbinom(1,1,0.9), rbinom(1,1,0.1))}

sim_studies <- sim_studies %>%
  mutate(published=as.factor(published))

pub_sim_studies <- dplyr::filter(sim_studies, published==1)
```

Modeling using sample_prior = ‘only’

```
pub_model_prior_fit <- brm(
  model_study,
  data = pub_sim_studies,
  prior = priors,
  family = gaussian,
  refresh=0,
  sample_prior = TRUE,
  iter=10000,
  warmup = 1000,
  backend = "cmdstanr",
  threads = threading(2),
  chains = 2,
  cores = 2,
  control = list(
    adapt_delta = 0.99,
    max_treedepth = 20
)
)
```

```
## Running MCMC with 2 parallel chains, with 2 thread(s) per chain...
##
## Chain 2 finished in 2.1 seconds.
## Chain 1 finished in 2.1 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 2.1 seconds.
## Total execution time: 2.2 seconds.
```

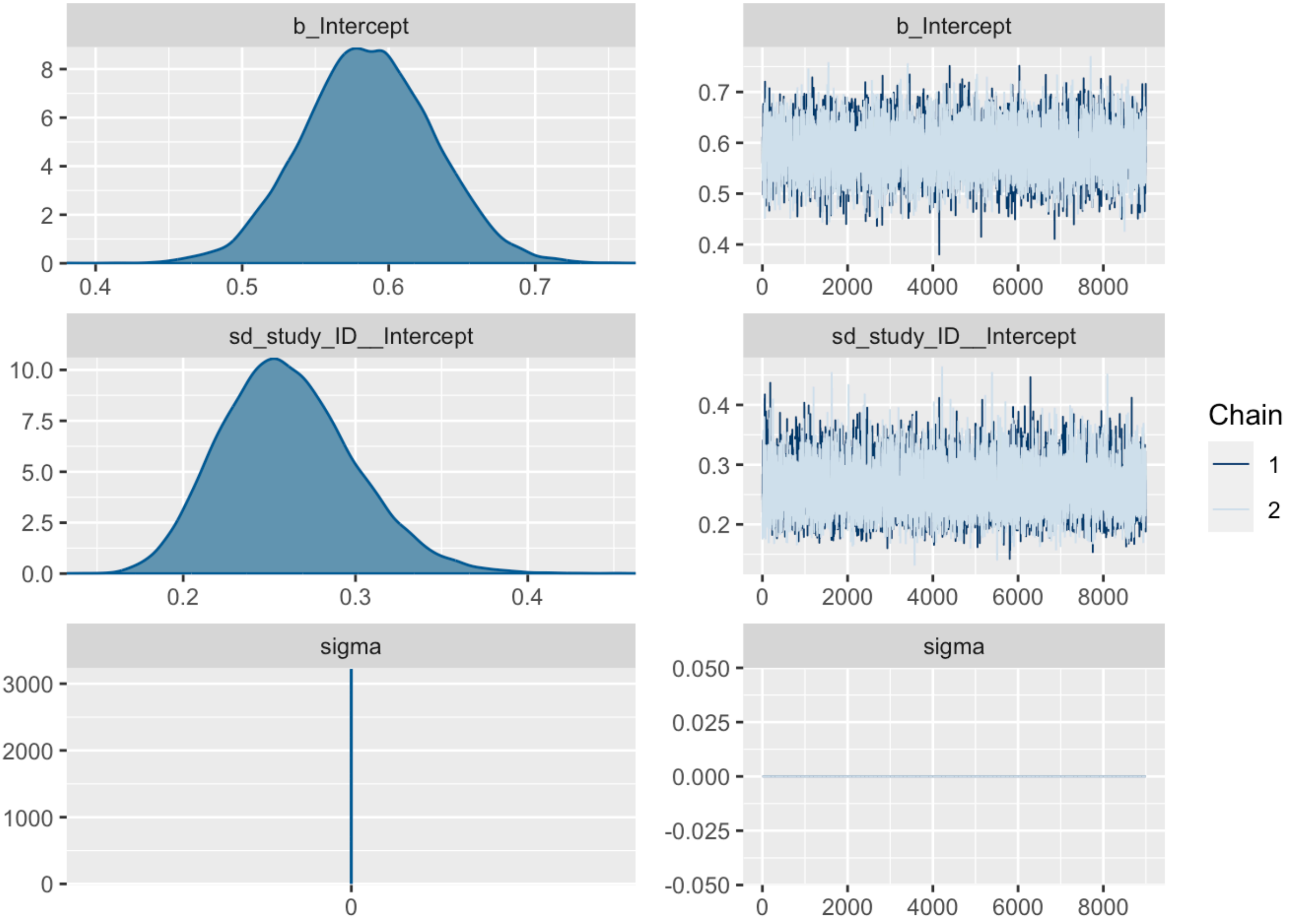
```
pp_check(pub_model_prior_fit, ndraws=100)
```



Ditlev

Potting and assesing and transforming the brmsfit to draws

```
plot(pub_model_prior_fit)
```



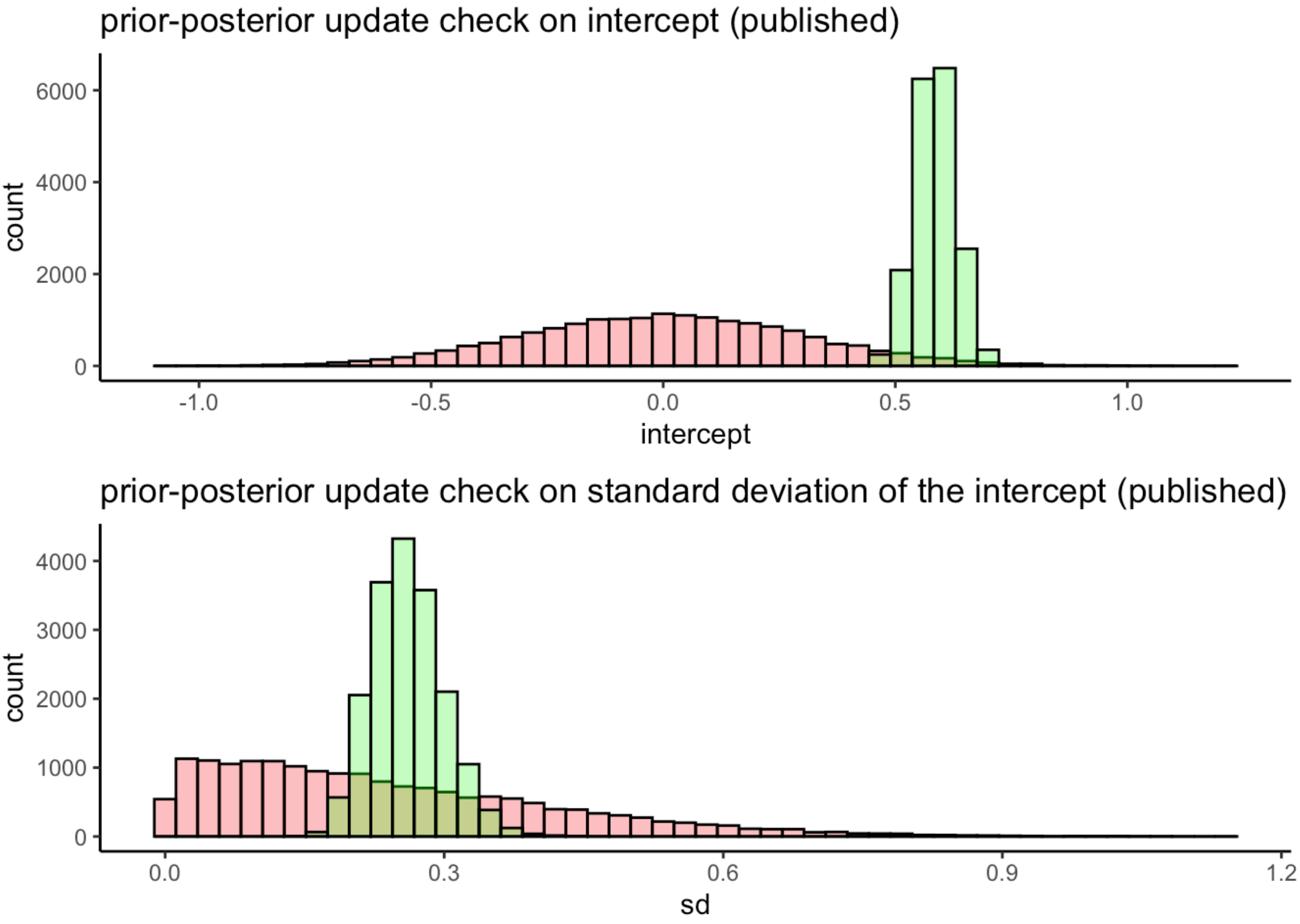
```
summary(pub_model_prior_fit)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: mean_effect_size | se(standard_error) ~ 1 + (1 | study_ID)
##    Data: pub_sim_studies (Number of observations: 48)
##   Draws: 2 chains, each with iter = 10000; warmup = 1000; thin = 1;
##          total post-warmup draws = 18000
##
## Group-Level Effects:
## ~study_ID (Number of levels: 48)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.26      0.04     0.19     0.34 1.00     5900     9863
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.59      0.04     0.50     0.67 1.00     4538     8095
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.00      0.00     0.00     0.00   NA       NA       NA
##
## 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).
```

```
pub_posterior <- as_draws_df(pub_model_prior_fit)
```

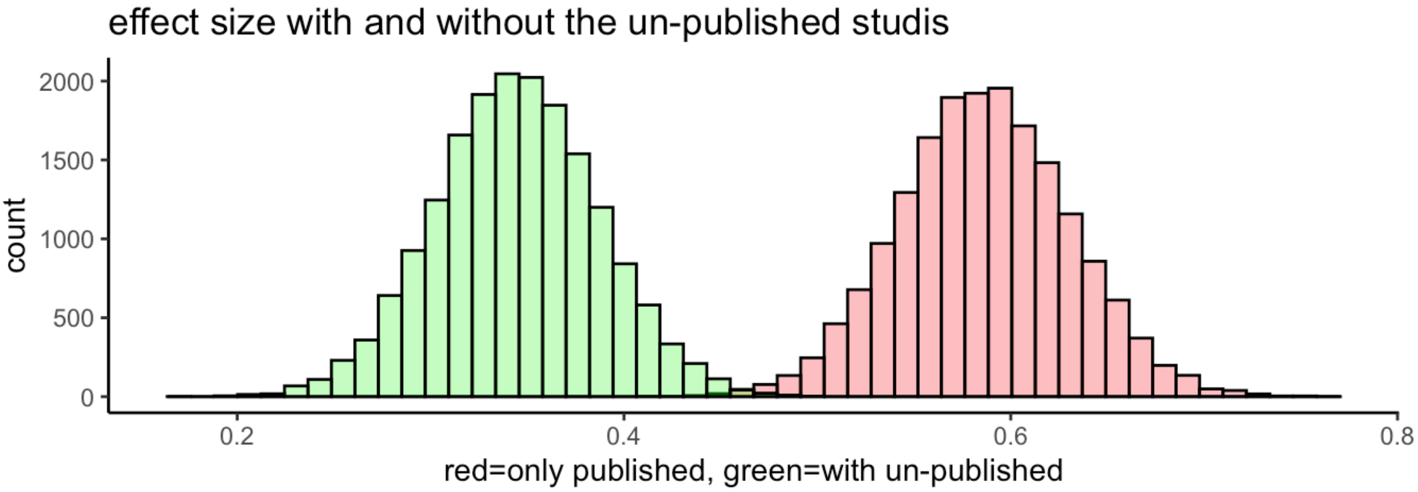
Plotting “prior-posterior update check on intercept” and “prior-posterior update check on standard deviation of the intercept”

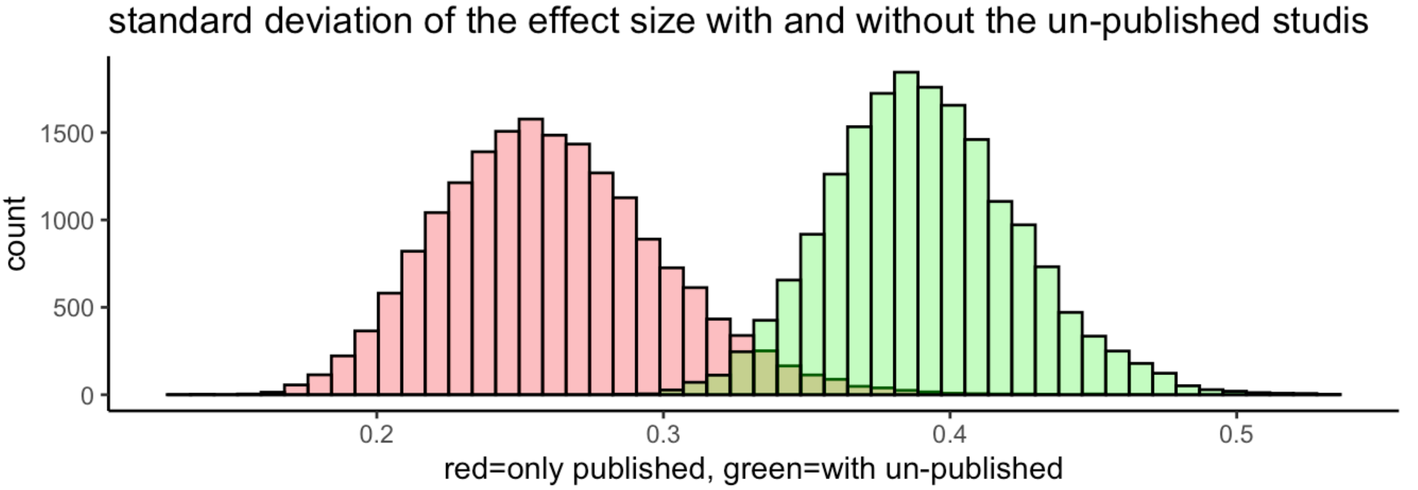
```
pub_plot1 <- ggplot(pub_posterior)+
  geom_histogram(aes(prior_Intercept), fill='red', color='black', alpha=0.3, bins=
50)+
  geom_histogram(aes(Intercept), fill='green', color='black', alpha=0.3, bins=50)+
  theme_classic()+
  ggtitle('prior-posterior update check on intercept (published)')+
  xlab('intercept')
pub_plot2 <- ggplot(pub_posterior)+
  geom_histogram(aes(prior_sd_study_ID), fill='red', color='black', alpha=0.3, bin
s=50)+
  geom_histogram(aes(sd_study_ID__Intercept), fill='green', color='black', alpha=0
.3, bins=50)+
  theme_classic()+
  ggtitle('prior-posterior update check on standard deviation of the intercept (pu
blished)')+
  xlab('sd')
grid.arrange(pub_plot1, pub_plot2)
```



Plotting “effect size with and without the un-published studis” and “tandard deviation of the effect size with and without the un-published studis”

```
plot3 <- ggplot()+
  geom_histogram(aes(pub_posterior$Intercept), fill='red', color='black', alpha=0.3, bins=50)+
  geom_histogram(aes(model_posterior$Intercept), fill='green', color='black', alpha=0.3, bins=50)+
  theme_classic()+
  ggtitle('effect size with and without the un-published studis')+
  xlab('red=only published, green=with un-published')
plot4 <- ggplot()+
  geom_histogram(aes(pub_posterior$sd_study_ID__Intercept), fill='red', color='black', alpha=0.3, bins=50)+
  geom_histogram(aes(model_posterior$sd_study_ID__Intercept), fill='green', color='black', alpha=0.3, bins=50)+
  theme_classic()+
  ggtitle('standard deviation of the effect size with and without the un-published studis')+
  xlab('red=only published, green=with un-published')
grid.arrange(plot3, plot4)
```





Question 2

Loading the data

```
matrix_ma <- read_excel("/Users/patrikmolnar/Desktop/Cognitive Science/Semester 3/
Methods3/Matrix_MetaAnalysis.xlsx")
```

Describing the data

Filtering out NA & NR for SZ

```
matrix_ma_filter_for_analysis <- matrix_ma %>%
  dplyr::filter(AGE_M_SZ!="NR") %>%
  dplyr::filter(AGE_M_SZ!="NA")
matrix_ma_filter_for_analysis <- matrix_ma_filter_for_analysis %>%
  dplyr::filter(AGE_SD_SZ!="NR") %>%
  dplyr::filter(AGE_SD_SZ!="NA")
matrix_ma_filter_for_analysis <- matrix_ma_filter_for_analysis %>%
  dplyr::filter(MALE_SZ!="NR") %>%
  dplyr::filter(MALE_SZ!="NA")
matrix_ma_filter_for_analysis <- matrix_ma_filter_for_analysis %>%
  dplyr::filter(FEMALE_SZ!="NR") %>%
  dplyr::filter(FEMALE_SZ!="NA")
```

Filtering out NA & NR for HC

```
matrix_ma_filter_for_analysis <- matrix_ma_filter_for_analysis %>%
  dplyr::filter(AGE_M_HC!="NR") %>%
  dplyr::filter(AGE_M_HC!="NA")
matrix_ma_filter_for_analysis <- matrix_ma_filter_for_analysis %>%
  dplyr::filter(AGE_SD_HC!="NR") %>%
  dplyr::filter(AGE_SD_HC!="NA")
matrix_ma_filter_for_analysis <- matrix_ma_filter_for_analysis %>%
  dplyr::filter(MALE_HC!="NR") %>%
  dplyr::filter(MALE_HC!="NA")
matrix_ma_filter_for_analysis <- matrix_ma_filter_for_analysis %>%
  dplyr::filter(FEMALE_HC!="NR") %>%
  dplyr::filter(FEMALE_HC!="NA")
```

Making the variables numeric for SZ

```
matrix_ma_filter_for_analysis$AGE_M_SZ <- as.numeric(matrix_ma_filter_for_analysis$AGE_M_SZ)
matrix_ma_filter_for_analysis$AGE_SD_SZ <- as.numeric(matrix_ma_filter_for_analysis$AGE_SD_SZ)
matrix_ma_filter_for_analysis$MALE_SZ <- as.numeric(matrix_ma_filter_for_analysis$MALE_SZ)
matrix_ma_filter_for_analysis$FEMALE_SZ <- as.numeric(matrix_ma_filter_for_analysis$FEMALE_SZ)
```

Making the variables numeric for HC

```
matrix_ma_filter_for_analysis$AGE_M_HC <- as.numeric(matrix_ma_filter_for_analysis
$AGE_M_HC)
matrix_ma_filter_for_analysis$AGE_SD_HC <- as.numeric(matrix_ma_filter_for_analysi
s$AGE_SD_HC)
matrix_ma_filter_for_analysis$MALE_HC <- as.numeric(matrix_ma_filter_for_analysis$
MALE_HC)
matrix_ma_filter_for_analysis$FEMALE_HC <- as.numeric(matrix_ma_filter_for_analysi
s$FEMALE_HC)
```

Patrik

Making both tibbles in order to combine them and make them easier for the eye

```
a <- tibble(diagnosis = "SZ",
         mean_sample_size=mean(matrix_ma_filter_for_analysis$SAMPLE_SIZE_SZ),
         mean_numer_of_males=mean(matrix_ma_filter_for_analysis$MALE_SZ),
         mean_number_of_females=mean(matrix_ma_filter_for_analysis$FEMALE_SZ),
         mean_age=mean(matrix_ma_filter_for_analysis$AGE_M_SZ),
         mean_sd_age=mean(matrix_ma_filter_for_analysis$AGE_SD_SZ)
         )

b <- tibble(diagnosis = "HC",
         mean_sample_size=mean(matrix_ma_filter_for_analysis$SAMPLE_SIZE_HC),
         mean_numer_of_males=mean(matrix_ma_filter_for_analysis$MALE_HC),
         mean_number_of_females=mean(matrix_ma_filter_for_analysis$FEMALE_HC),
         mean_age=mean(matrix_ma_filter_for_analysis$AGE_M_HC),
         mean_sd_age=mean(matrix_ma_filter_for_analysis$AGE_SD_HC)
         )
```

Binding the rows together

```
Demographic_overview <- bind_rows(a,b)
```

Showing the tibble

```
Demographic_overview
```

```
## # A tibble: 2 × 6
##   diagnosis mean_sample_size mean_numer_of_males mean_number_o…¹ mean_…² mean_…
³
##   <chr>                <dbl>               <dbl>           <dbl>   <dbl>   <dbl
>
## 1 SZ                    40.5                27.3            14.3    35.9    8.3
9
## 2 HC                    31.2                17.7            14.7    34.9    8.9
2
## # … with abbreviated variable names ¹mean_number_of_females, ²mean_age,
## #   ³mean_sd_age
```

Selceting the relevant variables

```
matrix_pitch <- matrix_ma %>%
  select('StudyID','Article','SAMPLE_SIZE_SZ','SAMPLE_SIZE_HC', 'PITCH_F0SD_HC_M',
'PITCH_F0SD_HC_SD','PITCH_F0SD_SZ_M','PITCH_F0SD_SZ_SD')
```

Filtering out the NA

```
matrix_pitch <- matrix_pitch %>%
  na.omit()
```

Merging diagnosis into one variable

```
matrix_pitch <- matrix_pitch %>%
  mutate(sample_size=(SAMPLE_SIZE_SZ+SAMPLE_SIZE_HC))
```

Creating IDs for the studies

```
matrix_pitch <- matrix_pitch %>%
  mutate(StudyID=as.factor(StudyID))
matrix_pitch <- matrix_pitch %>%
  mutate(StudyID=as.numeric(StudyID))
matrix_pitch <- matrix_pitch %>%
  mutate(StudyID=as.factor(StudyID))
```

Getting normalized results

```
matrix_pitch <- escalc('SMD',
n1i=SAMPLE_SIZE_HC,
n2i=SAMPLE_SIZE_SZ,
m1i = PITCH_F0SD_HC_M,
m2i=PITCH_F0SD_SZ_M,
sd1i = PITCH_F0SD_HC_SD,
sd2i = PITCH_F0SD_SZ_SD,
data = matrix_pitch)
matrix_pitch <- matrix_pitch %>%
  rename(effect_size=yi)
```

Creating a loop to calculate sd effect size and se

```
for (i in seq(nrow(matrix_pitch))){
  matrix_pitch$sd_effect[i] <- sqrt((sum((matrix_pitch$effect_size[i] - mean(matri
x_pitch$effect_size))^2))/length(matrix_pitch))
  matrix_pitch$standard_error[i] <- matrix_pitch$sd_effect[i]/sqrt(matrix_pitch$sa
mple_size)
}
```

Setting model

```
model_matrix <- bf(effect_size|se(standard_error) ~1 + (1|StudyID))
```

Getting priors

```
get_prior(data = matrix_pitch, family = gaussian, model_matrix)
```

```
##                   prior     class      coef   group resp dpar nlpar lb ub
##  student_t(3, 0.3, 2.5) Intercept
##    student_t(3, 0, 2.5)        sd                                    0
##    student_t(3, 0, 2.5)        sd            StudyID                  0
##    student_t(3, 0, 2.5)        sd Intercept StudyID                  0
##        source
##       default
##       default
##  (vectorized)
##  (vectorized)
```

Setting priors

```
matrix_priors <- c(
  prior(normal( .3, 2.5), class=Intercept),
  prior(normal( 0, 2.5), class=sd))
```

Priors

```
matrix_prior_fit <- brm(
  model_matrix,
  data = matrix_pitch,
  prior = matrix_priors,
  family = gaussian,
  refresh=0,
  sample_prior = 'only',
  iter=10000,
  warmup = 1000,
  backend = "cmdstanr",
  threads = threading(2),
  chains = 2,
  cores = 2,
  control = list(
    adapt_delta = 0.99,
    max_treedepth = 20
)
)
```

```
## Running MCMC with 2 parallel chains, with 2 thread(s) per chain...
##
## Chain 2 finished in 0.4 seconds.
## Chain 1 finished in 0.5 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 0.4 seconds.
## Total execution time: 0.5 seconds.
```

Bryan

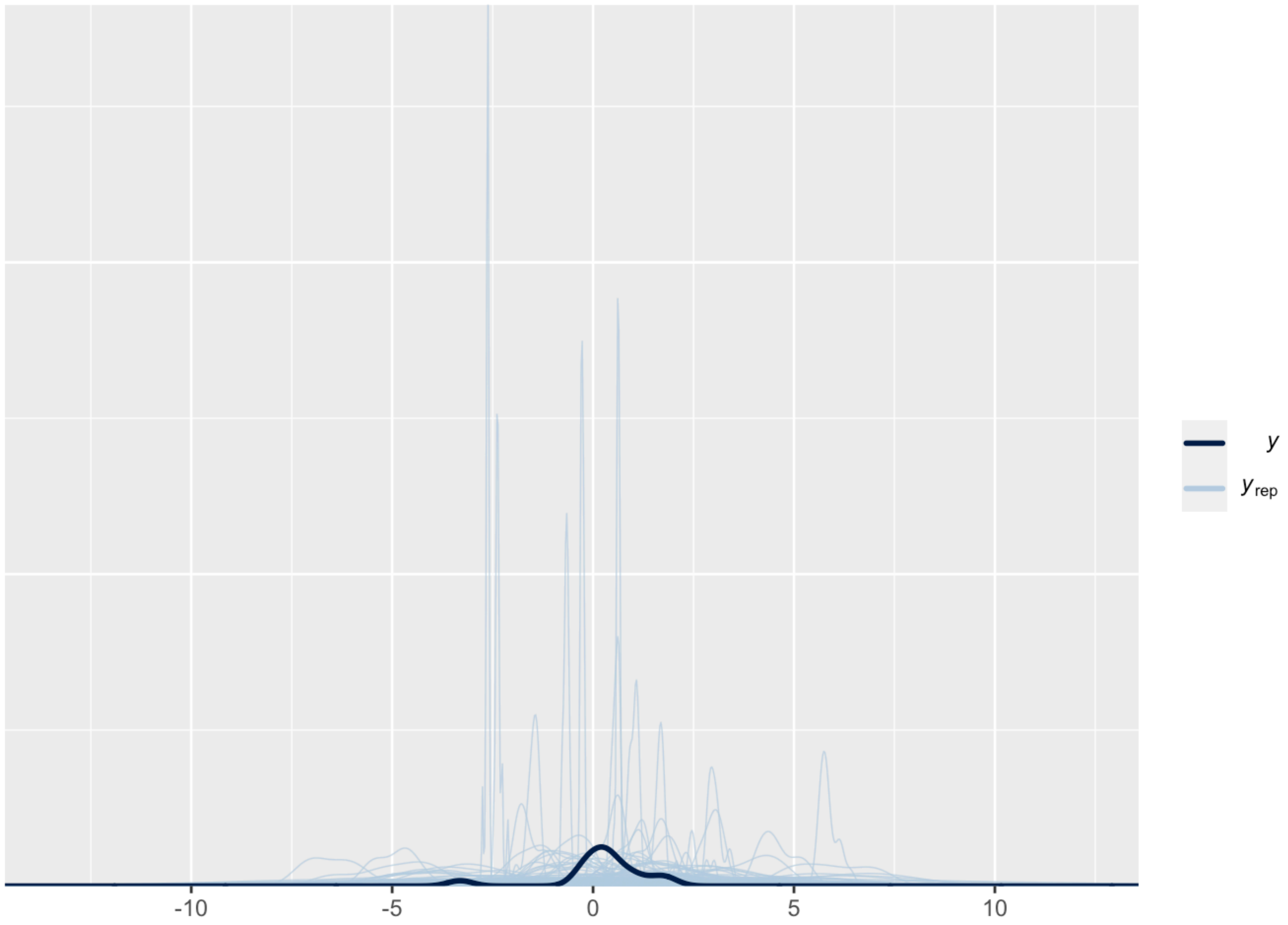
Assesing results

```
matrix_prior_fit
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: effect_size | se(standard_error) ~ 1 + (1 | StudyID)
##    Data: matrix_pitch (Number of observations: 15)
##   Draws: 2 chains, each with iter = 10000; warmup = 1000; thin = 1;
##          total post-warmup draws = 18000
##
## Group-Level Effects:
## ~StudyID (Number of levels: 12)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.98      1.52     0.08     5.62 1.00    14754     8411
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.29      2.51    -4.62     5.09 1.00    26582    13577
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.00      0.00     0.00     0.00   NA       NA       NA
##
## 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).
```

PP check

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



Both data and priors

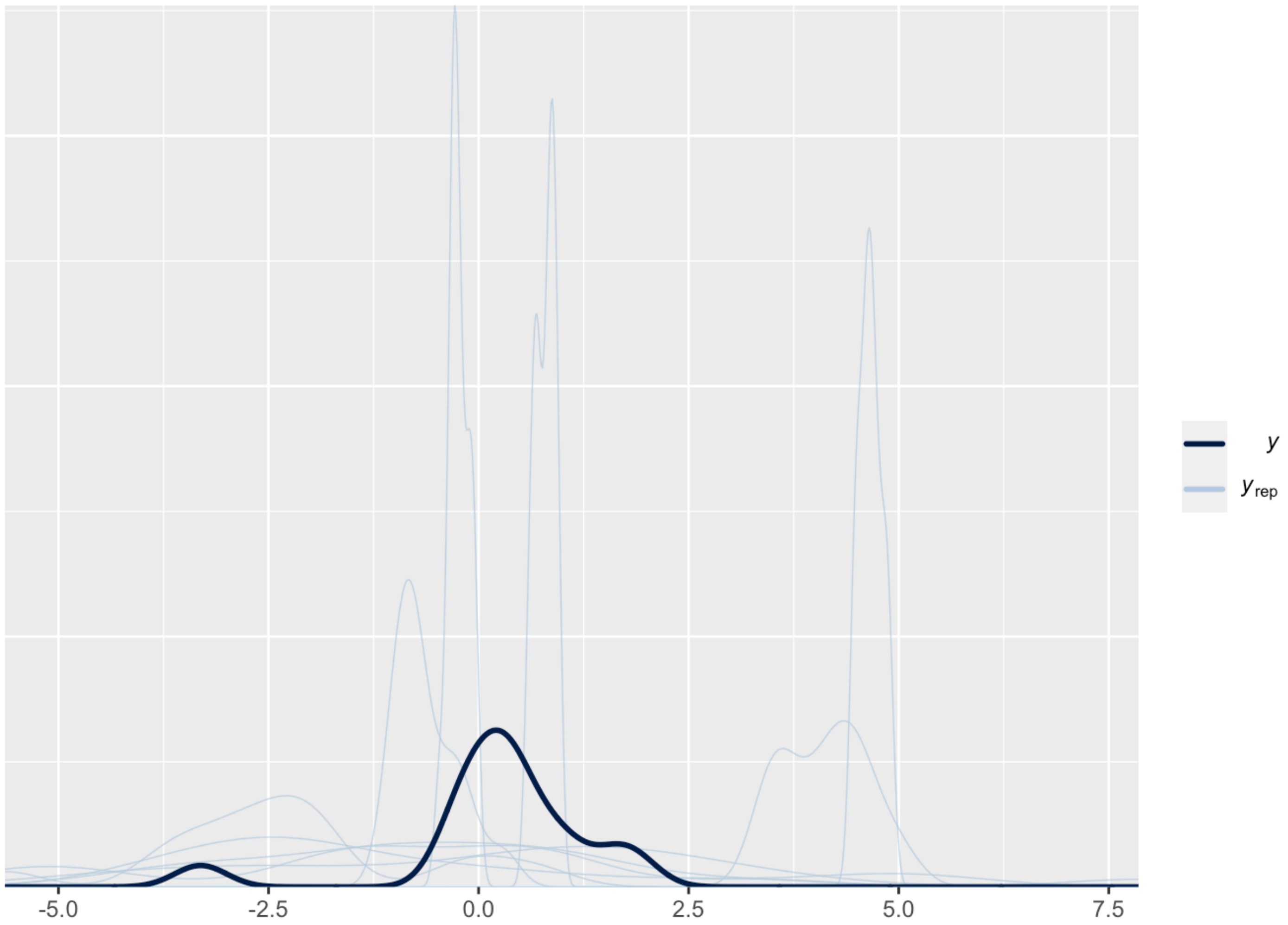
Including both data and priors

```
matrix_fit <- brm(
  model_matrix,
  data = matrix_pitch,
  prior = matrix_priors,
  family = gaussian,
  refresh=0,
  sample_prior = 'only',
  iter=10000,
  warmup = 1000,
  backend = "cmdstanr",
  threads = threading(2),
  chains = 2,
  cores = 2,
  control = list(
    adapt_delta = 0.99,
    max_treedepth = 20
)
)
```

```
## Running MCMC with 2 parallel chains, with 2 thread(s) per chain...
##
## Chain 1 finished in 0.4 seconds.
## Chain 2 finished in 0.6 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 0.5 seconds.
## Total execution time: 0.7 seconds.
```

PP check

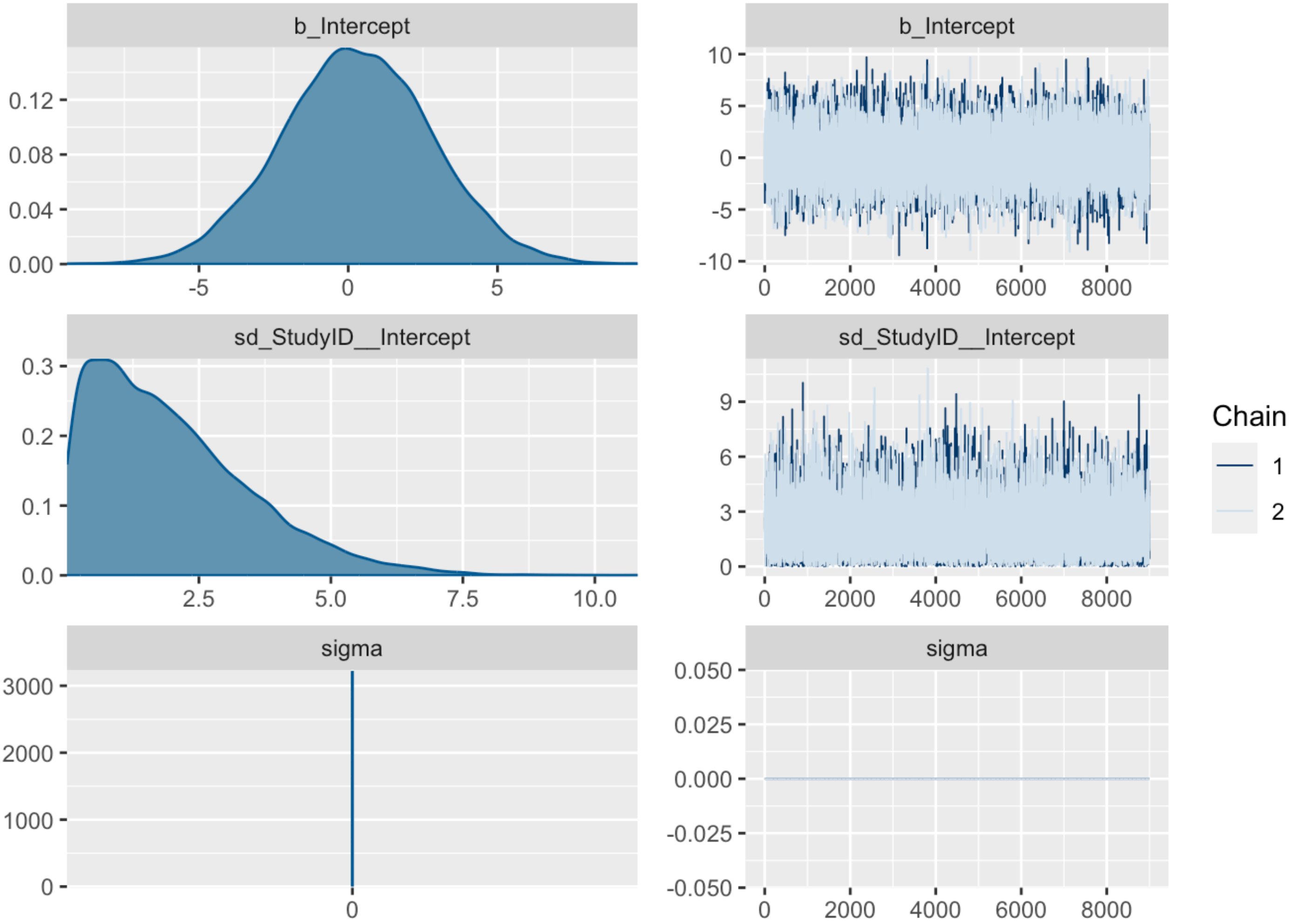
```
pp_check(matrix_fit)
```



Visualize and report

Plotting and assesing the results

```
plot(matrix_fit)
```



```
summary(matrix_fit)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: effect_size | se(standard_error) ~ 1 + (1 | StudyID)
##    Data: matrix_pitch (Number of observations: 15)
##   Draws: 2 chains, each with iter = 10000; warmup = 1000; thin = 1;
##          total post-warmup draws = 18000
##
## Group-Level Effects:
## ~StudyID (Number of levels: 12)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.99      1.51     0.08     5.65 1.00    13897     7399
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.29      2.52    -4.62     5.20 1.00    24126    12836
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.00      0.00     0.00     0.00   NA       NA       NA
##
## 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).
```

Visualizing and assesing intercepts

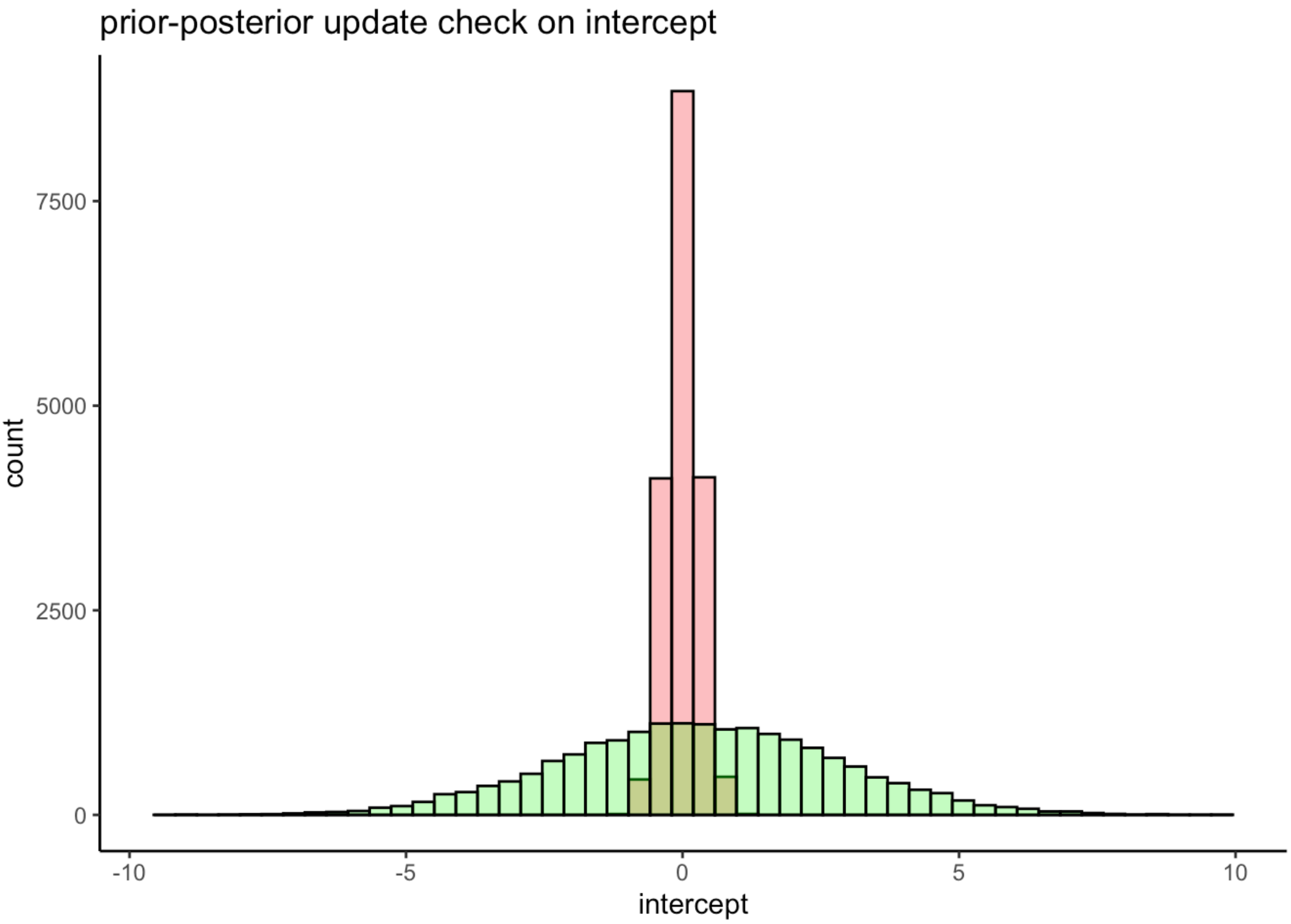
```
matrix_posterior <- as_draws_df(matrix_fit)
plot1 <- ggplot(matrix_posterior)+
  geom_histogram(aes(model_posterior$prior_Intercept), fill='red', color='black',
alpha=0.3, bins=50)+
  geom_histogram(aes(Intercept), fill='green', color='black', alpha=0.3, bins=50)+
  theme_classic()+
  ggtitle('prior-posterior update check on intercept')+
  xlab('intercept')
```

Visualizing standard deviation

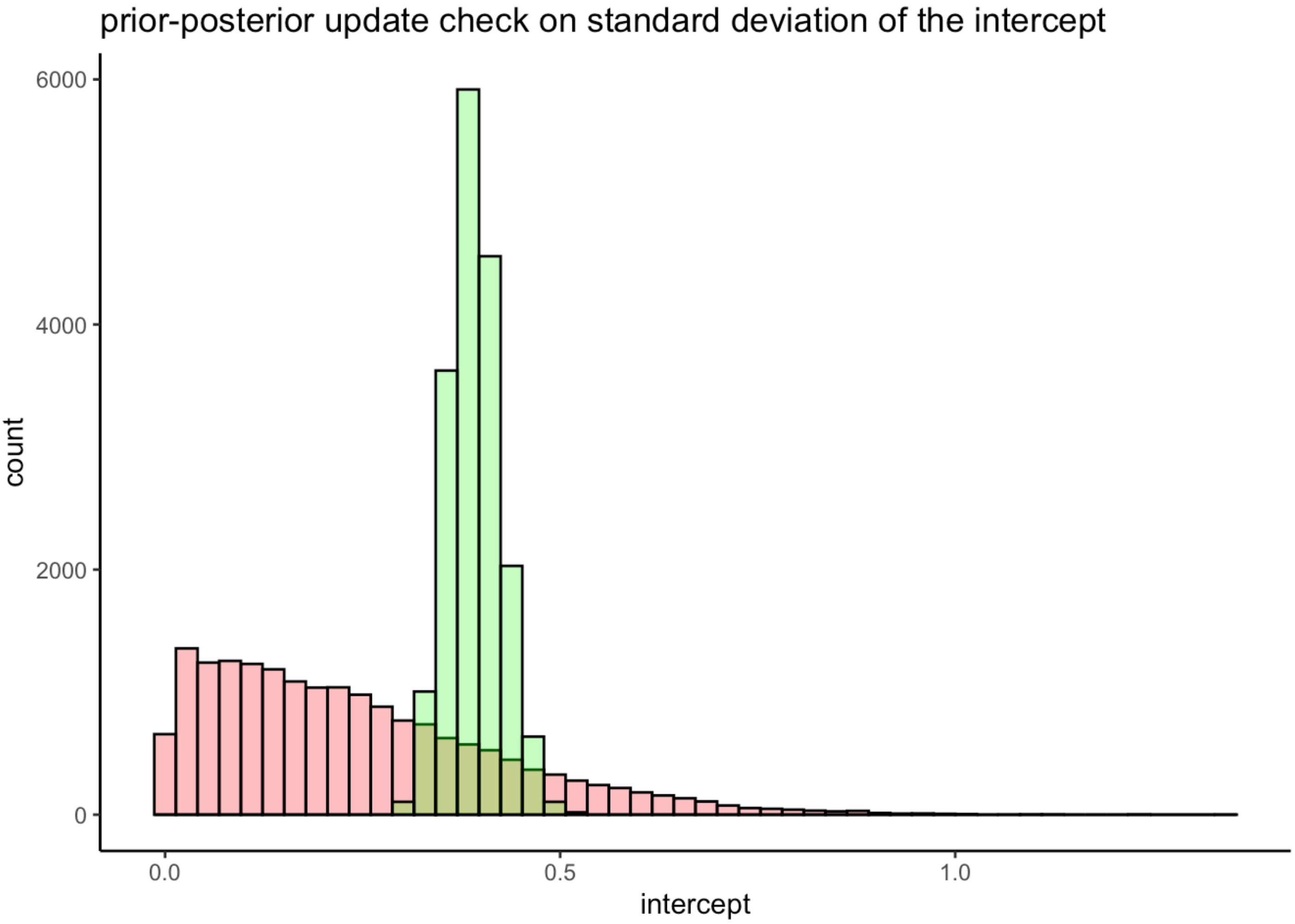
```
plot2 <- ggplot(matrix_posterior)+
  geom_histogram(aes(model_posterior$prior_sd_study_ID), fill='red', color='black'
, alpha=0.3, bins=50)+
  geom_histogram(aes(model_posterior$sd_study_ID__Intercept), fill='green', color=
'black', alpha=0.3, bins=50)+
  theme_classic()+
  ggtitle('prior-posterior update check on standard deviation of the intercept')+
  xlab('intercept')
```

Printing plots

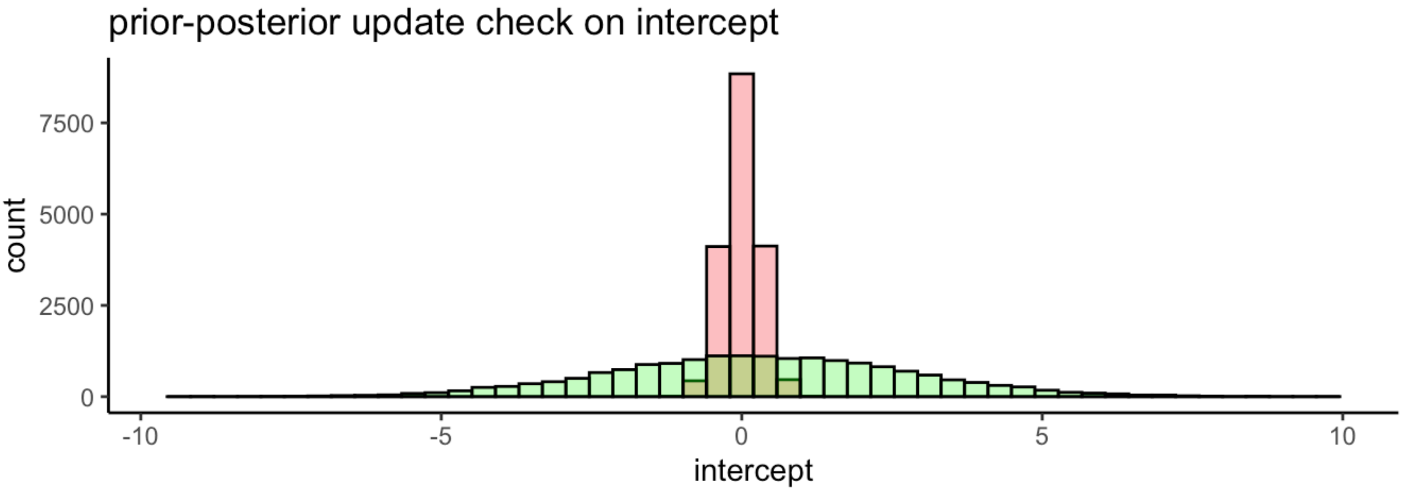
```
plot1
```

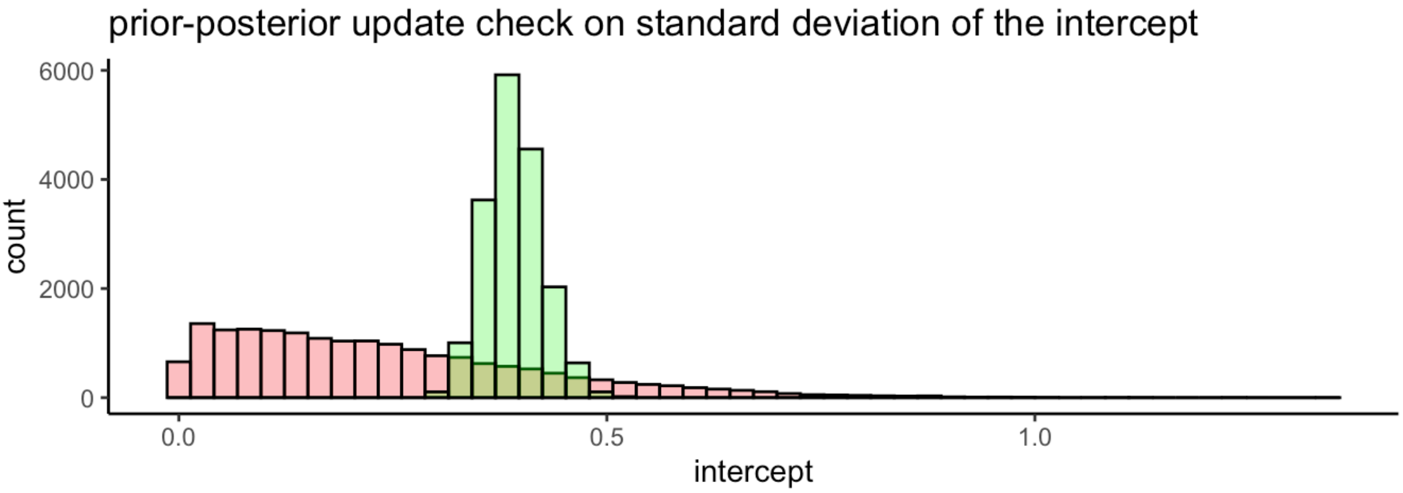

prior-posterior update check on intercept
count
7500
5000
2500
0
-10
-5
0
5
10
intercept

```
plot2
```



```
grid.arrange(plot1, plot2)
```


prior-posterior update check on intercept


prior-posterior update check on standard deviation of the intercept

Influencial studies

Excluding the Cohen et al. (2014) by indexing

```
excluded_matrix <- matrix_pitch %>%
  dplyr::filter(StudyID!=6)
```

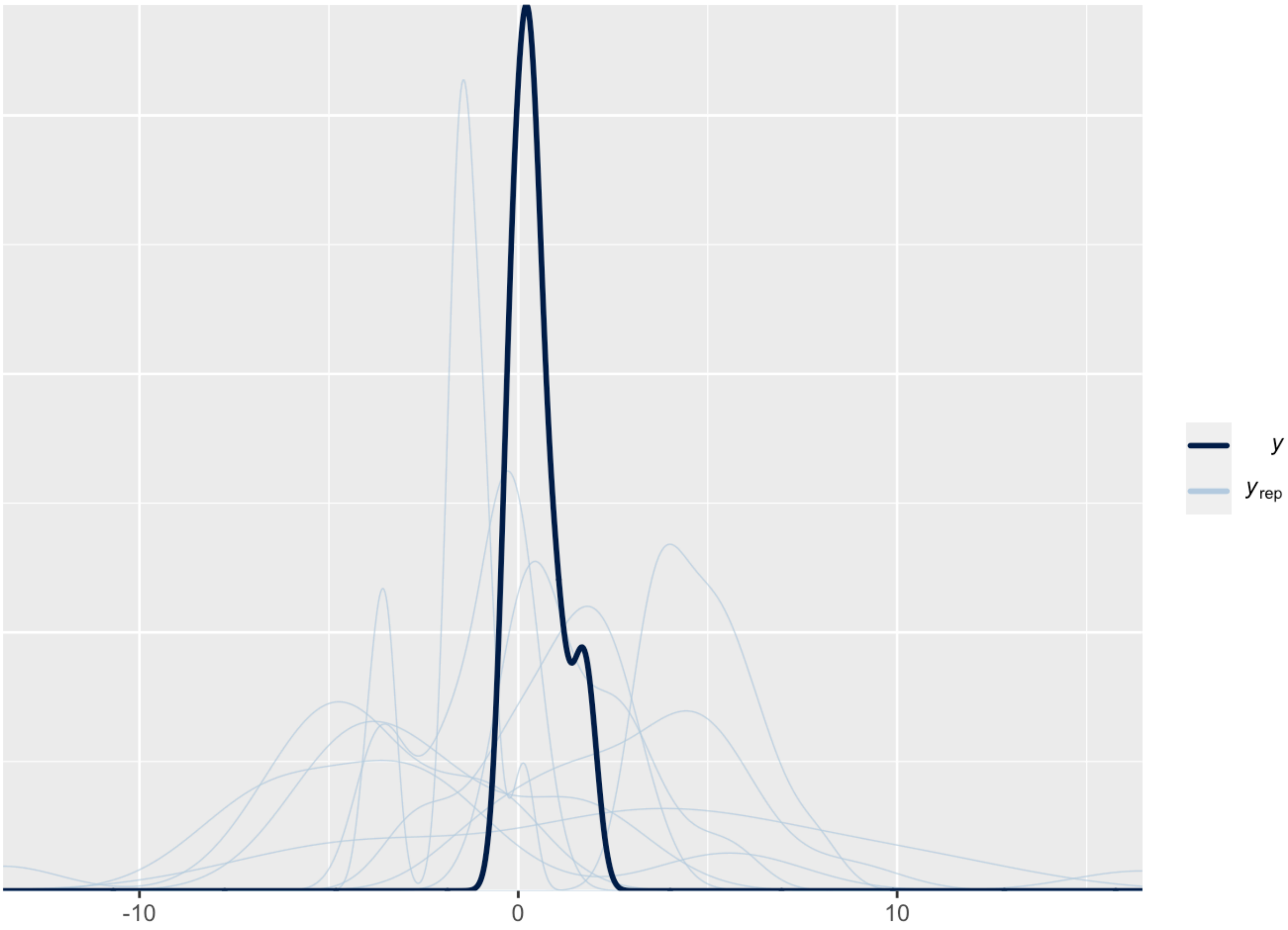
Running the model

```
exclude_matrix_fit <- brm(
  model_matrix,
  data = excluded_matrix,
  prior = matrix_priors,
  family = gaussian,
  refresh=0,
  sample_prior = 'only',
  iter=10000,
  warmup = 1000,
  backend = "cmdstanr",
  threads = threading(2),
  chains = 2,
  cores = 2,
  control = list(
    adapt_delta = 0.99,
    max_treedepth = 20
)
)
```

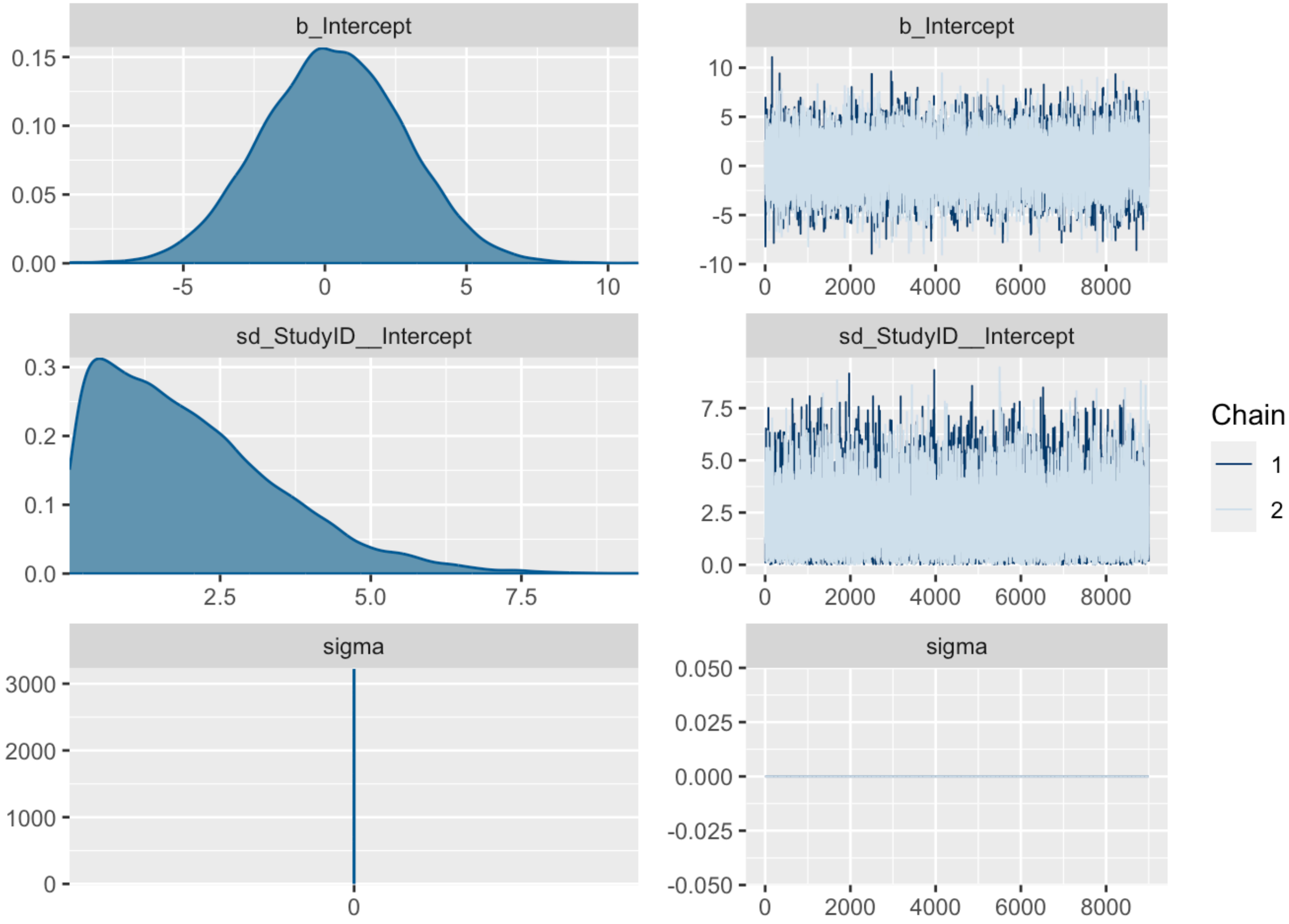
```
## Running MCMC with 2 parallel chains, with 2 thread(s) per chain...
##
## Chain 2 finished in 0.4 seconds.
## Chain 1 finished in 0.5 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 0.5 seconds.
## Total execution time: 0.6 seconds.
```

Visualizing and plotting

```
pp_check(exclude_matrix_fit)
```



```
plot(exclude_matrix_fit)
```



```
summary(exclude_matrix_fit)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: effect_size | se(standard_error) ~ 1 + (1 | StudyID)
##    Data: excluded_matrix (Number of observations: 14)
##   Draws: 2 chains, each with iter = 10000; warmup = 1000; thin = 1;
##          total post-warmup draws = 18000
##
## Group-Level Effects:
## ~StudyID (Number of levels: 11)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     2.00      1.49     0.09     5.60 1.00    14300     7570
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.31      2.52    -4.57     5.20 1.00    22115    13548
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.00      0.00     0.00     0.00   NA       NA       NA
##
## 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).
```