

Assignment 2 - Meta-analysis of pitch in schizophrenia

Study group 9 - Thomas, Katrine, Caroline, Rikke & Mikkel

16/8/2022

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 discuss what this implies for your model. remember to use at least one plot to visualize your results. BONUS question: do a power/precision analysis: w this kind of sample sizes (participants) how many studies would you need to acquire good precision (e.g. .1 sd in the pop level estimate)
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 - 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: assess the effect of task on the estimates (model comparison with baseline model)

Question 1

```
#Simulating the dataset
set.seed(122)

n_studies <- 100
participants<- round(rnorm(n_studies,20,10),0)

#Testing if any study is too small and reiterating. Note this gives a sharp positively skewed distribut
for(i in 1:n_studies) {
  while(participants[i] <= 10) {
    participants[i]<-round(rnorm(1,20,10),0)
  }
}

#Checking
```

```

#ifelse(participants <= 10, print("Warning"), print("."))

# Investigating participants
#c(mean(d$participants), sd(d$participants), min(d$participants), max(d$participants))

#Predicting the outcome of studies. First the means (mu=0.4, sd=0.4)
effect_size_means<-rnorm(n_studies, .4, .4)
d<-tibble(ID=seq(1:n_studies),participants, effect_size_means, SE=rep(1,n_studies))

#Lean code that doesn't work
d<-d %>%
  mutate(sd=sd(rnorm(participants,effect_size_means,0.8)))

#Standard deviation and standard error
for(i in 1:n_studies){
  d$sd[i]<-sd(rnorm(participants[i],effect_size_means[i],0.8/2))
}

d<- d %>%
  mutate(SE=sd/sqrt(participants))

#Will it get published (90%)
d<-d %>%
  mutate(Published = ifelse(abs(effect_size_means)-2*sd > 0, rbinom(1,1,.9), rbinom(1,1,.1)))

d<-d %>%
  mutate(PublishedPositiveEffect = ifelse(effect_size_means > 0 & Published == 1, 1, 0))

#Making the publication bias dataset for comparison
d_pub_bias <- d %>%
  filter(Published == 1)

### MAKING THE BAYESIAN MODEL ###
ma_f0 <- bf(effect_size_means | se(sd) ~ 1 + (1|ID))

p<-c(
  prior(normal(0, 0.5), class = Intercept),
  prior(normal(0, 0.6), class = sd))

get_prior(ma_f0, data = d)

```

```

##           prior      class      coef group resp dpar nlpar lb ub
## student_t(3, 0.5, 2.5) Intercept
## student_t(3, 0, 2.5)      sd
## student_t(3, 0, 2.5)      sd      ID
## student_t(3, 0, 2.5)      sd Intercept ID
## source
## default
## default
## (vectorized)

```

```
## (vectorized)
```

```
ma_m0_prior <- brm(  
  ma_f0,  
  d,  
  family = gaussian,  
  prior = p,  
  sample_prior = "only",  
  backend = "cmdstanr",  
  chains = 2,  
  cores = 2,  
  control = list(  
    adapt_delta = 0.99,  
    max_treedepth = 20  
  ))
```

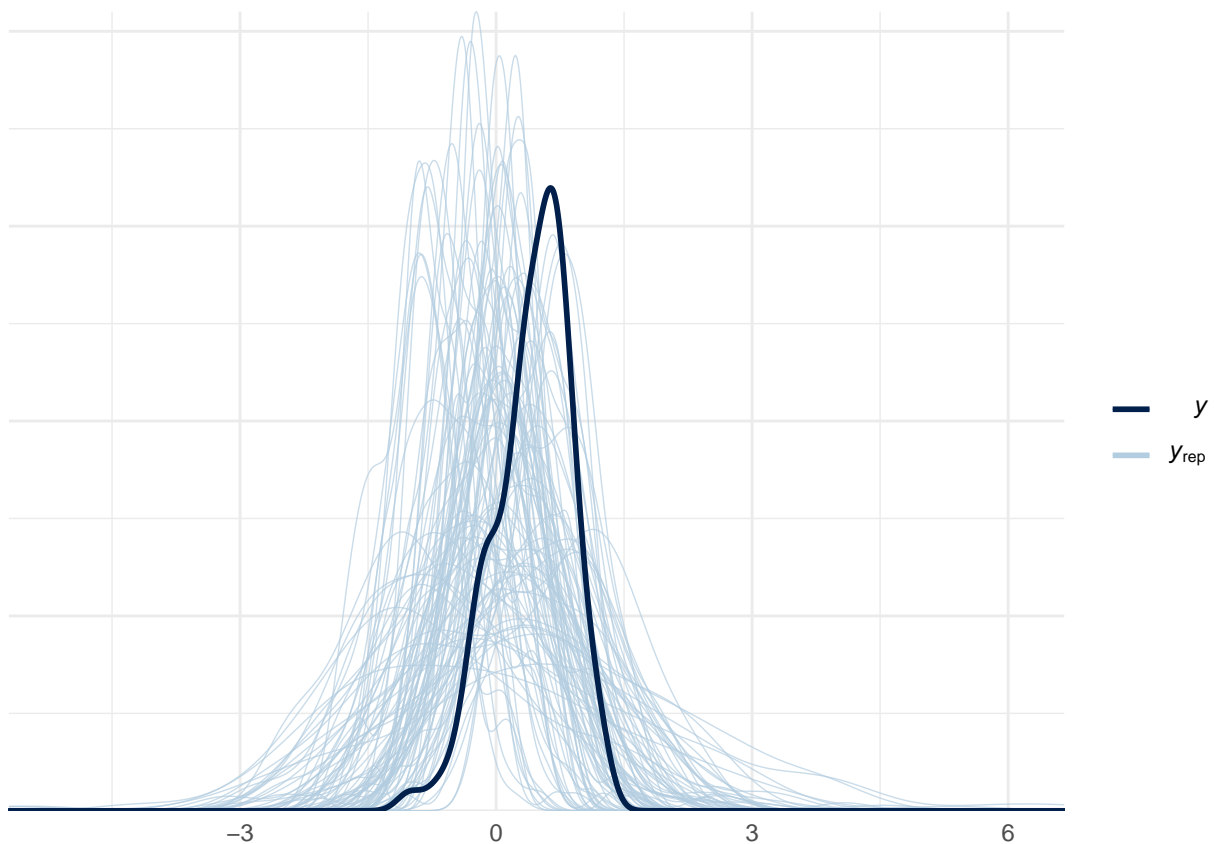
```
## Start sampling
```

```
## Running MCMC with 2 parallel 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 2 Iteration:    1 / 2000 [ 0%] (Warmup)  
## Chain 2 Iteration:   100 / 2000 [ 5%] (Warmup)  
## Chain 2 Iteration:   200 / 2000 [10%] (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 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 1 Iteration:  1300 / 2000 [65%] (Sampling)  
## Chain 1 Iteration:  1400 / 2000 [70%] (Sampling)  
## Chain 1 Iteration:  1500 / 2000 [75%] (Sampling)  
## Chain 2 Iteration:  1300 / 2000 [65%] (Sampling)  
## Chain 2 Iteration:  1400 / 2000 [70%] (Sampling)
```

```
## Chain 1 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 1 Iteration: 1700 / 2000 [ 85%] (Sampling)
## Chain 2 Iteration: 1500 / 2000 [ 75%] (Sampling)
## Chain 2 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 2 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 2 Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 2 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 1 finished in 0.7 seconds.
## Chain 2 Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 2 finished in 0.7 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 0.7 seconds.
## Total execution time: 0.9 seconds.
```

```
pp_check(ma_m0_prior, ndraws = 100)
```



```
ma_m0_all <- brm(
  ma_f0,
  d,
  family = gaussian,
  prior = p,
```

```

sample_prior = T,
backend = "cmdstanr",
chains = 2,
cores = 2,
control = list(
  adapt_delta = 0.99,
  max_treedepth = 20
))

```

```
## Start sampling
```

```
## Running MCMC with 2 parallel chains...
```

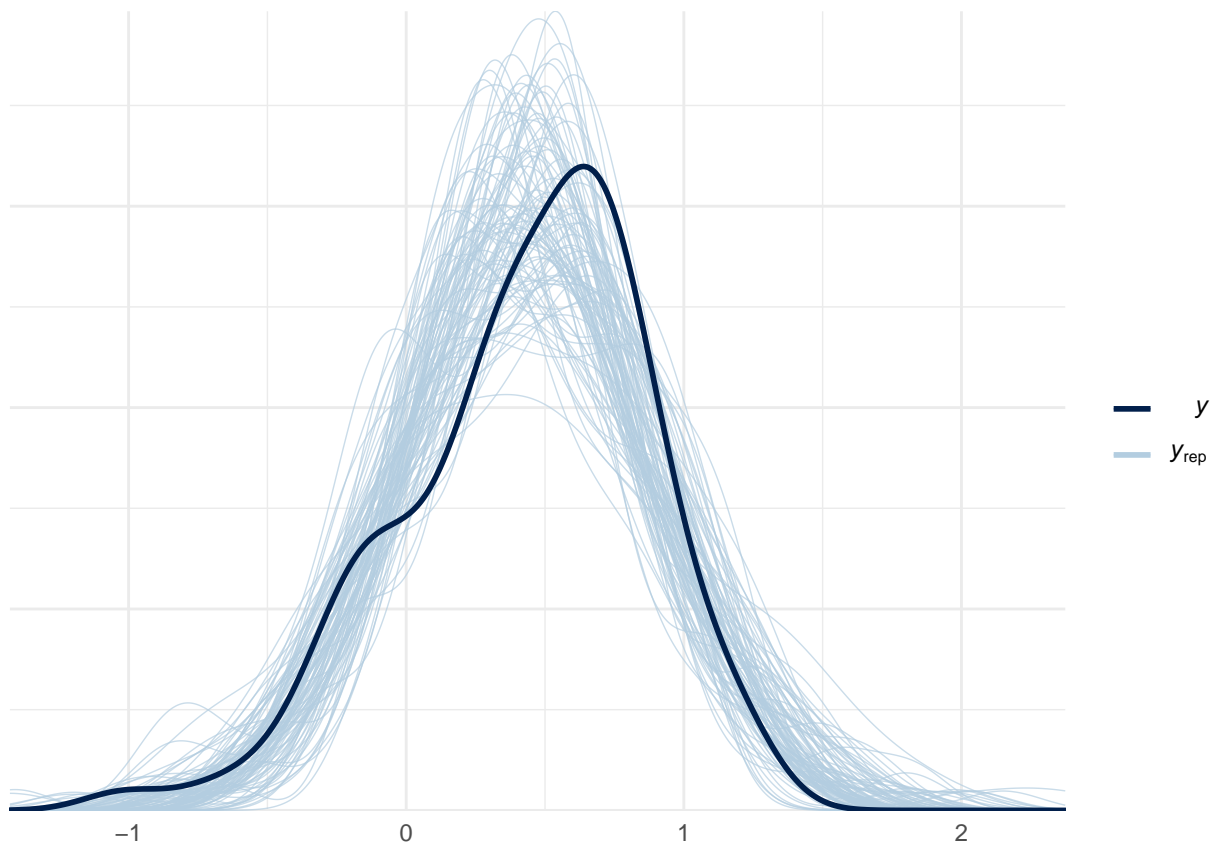
```

##
## Chain 1 Iteration:    1 / 2000 [  0%] (Warmup)
## 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 1 Iteration:  100 / 2000 [  5%] (Warmup)
## Chain 2 Iteration: 1100 / 2000 [ 55%] (Sampling)
## Chain 1 Iteration:  200 / 2000 [ 10%] (Warmup)
## Chain 1 Iteration:  300 / 2000 [ 15%] (Warmup)
## Chain 2 Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 1 Iteration:  400 / 2000 [ 20%] (Warmup)
## Chain 2 Iteration: 1300 / 2000 [ 65%] (Sampling)
## Chain 1 Iteration:  500 / 2000 [ 25%] (Warmup)
## Chain 1 Iteration:  600 / 2000 [ 30%] (Warmup)
## Chain 2 Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 1 Iteration:  700 / 2000 [ 35%] (Warmup)
## Chain 1 Iteration:  800 / 2000 [ 40%] (Warmup)
## Chain 2 Iteration: 1500 / 2000 [ 75%] (Sampling)
## Chain 1 Iteration:  900 / 2000 [ 45%] (Warmup)
## Chain 1 Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 2 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 1 Iteration: 1100 / 2000 [ 55%] (Sampling)
## Chain 2 Iteration: 1700 / 2000 [ 85%] (Sampling)
## Chain 1 Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 1 Iteration: 1300 / 2000 [ 65%] (Sampling)
## Chain 2 Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 1 Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 1 Iteration: 1500 / 2000 [ 75%] (Sampling)
## Chain 2 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 1 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 2 Iteration: 2000 / 2000 [100%] (Sampling)

```

```
## Chain 2 finished in 1.9 seconds.
## 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 2.2 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 2.1 seconds.
## Total execution time: 2.3 seconds.
```

```
pp_check(ma_m0_all, ndraws = 100)
```



```
summary(ma_m0_all)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: effect_size_means | se(sd) ~ 1 + (1 | ID)
## Data: d (Number of observations: 100)
## Draws: 2 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 2000
##
## Group-Level Effects:
## ~ID (Number of levels: 100)
## Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sd(Intercept)      0.16      0.07      0.02      0.30 1.00      360      361
##
## Population-Level Effects:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      0.43      0.04      0.35      0.51 1.00      2491      1540
##
## Family Specific Parameters:
##      Estimate Est.Error 1-95% 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).
```

```
posterior <- as_draws_df(ma_m0_all)
```

```
# Running model on published studies and published studies with positive effect sizes
ma_m0_pub <- update(ma_m0_all, newdata = subset(d, Published == 1))
```

```
## 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 0.6 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 0.6 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 0.6 seconds.
## Total execution time: 1.3 seconds.

```

```

ma_m0_pubpos <- update(ma_m0_all, newdata = subset(d, PublishedPositiveEffect == 1))

```

```

## 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 0.5 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 0.4 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 0.4 seconds.
## Total execution time: 1.1 seconds.
```

```
posterior_pub <- as_draws_df(ma_m0_pub, ndraws = 100)

posterior_pub_pos <- as_draws_df(ma_m0_pubpos, ndraws = 100)

# Prior-posterior update checks
plot1 <- ggplot(posterior) +
  geom_histogram(aes(prior_Intercept), fill = "red", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(b_Intercept), fill = "green", color = "black", alpha = 0.3, bins = 50) +
  geom_vline(xintercept = 0.4) +
  xlab("Prior-posterior update check on the intercepts")

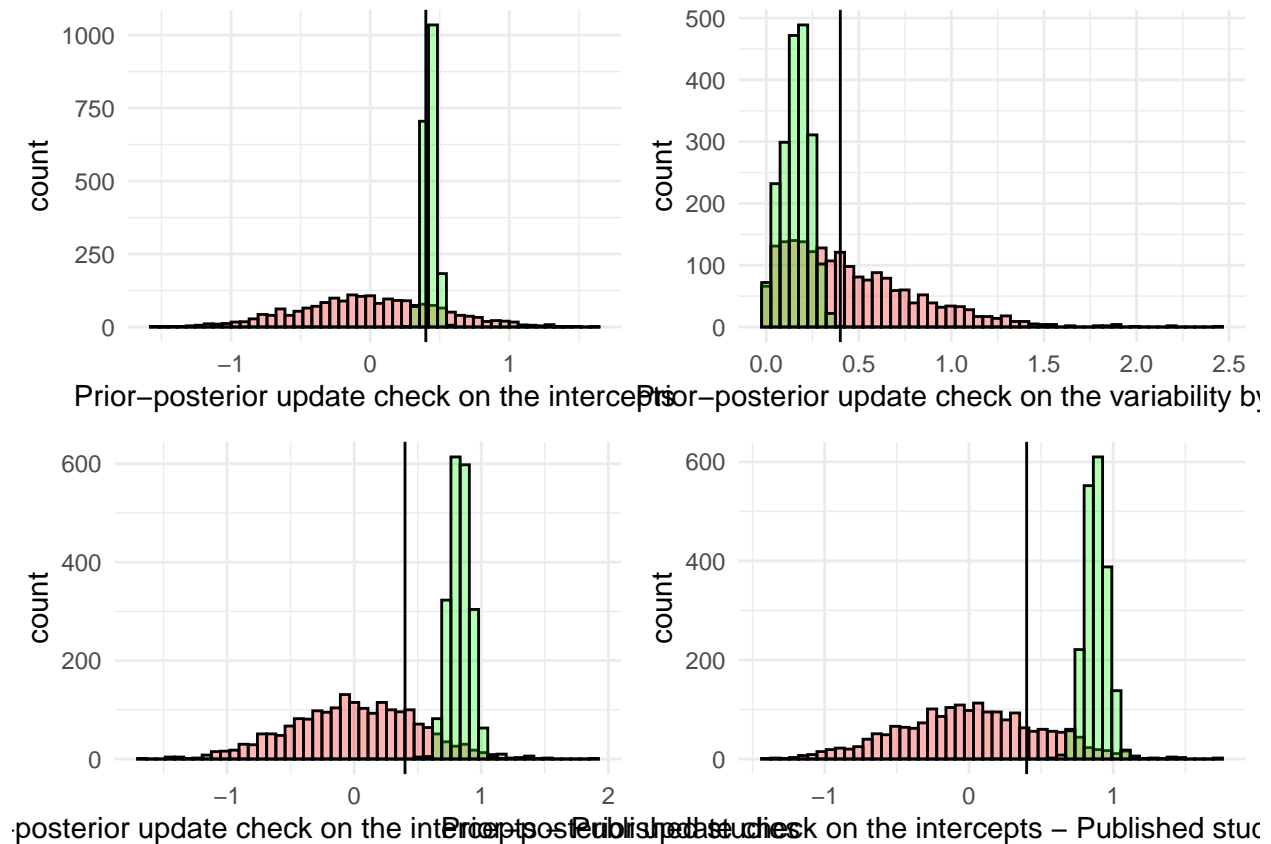
plot2 <- ggplot(posterior) +
  geom_histogram(aes(prior_sd_ID), fill = "red", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(sd_ID__Intercept), fill = "green", color = "black", alpha = 0.3, bins = 50) +
  geom_vline(xintercept = 0.4) +
  xlab("Prior-posterior update check on the variability by study")

# Including publication biases
plot3 <- ggplot(posterior_pub) +
  geom_histogram(aes(prior_Intercept), fill = "red", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(b_Intercept), fill = "green", color = "black", alpha = 0.3, bins = 50) +
  geom_vline(xintercept = 0.4) +
  xlab("Prior-posterior update check on the intercepts - Published studies")

plot4 <- ggplot(posterior_pub_pos) +
  geom_histogram(aes(prior_Intercept), fill = "red", color = "black", alpha = 0.3, bins = 50) +
```

```
geom_histogram(aes(b_Intercept), fill = "green", color = "black", alpha = 0.3, bins = 50) +
geom_vline(xintercept = 0.4) +
xlab("Prior-posterior update check on the intercepts - Published studies positive effectsize")

# GGAranging them plots
ggarrange(plot1, plot2, plot3, plot4, ncol = 2, nrow = 2)
```



Question 2

```
# Load data
df <- read_excel("Matrix_MetaAnalysis_Diagnosis_updated290719.xlsx")

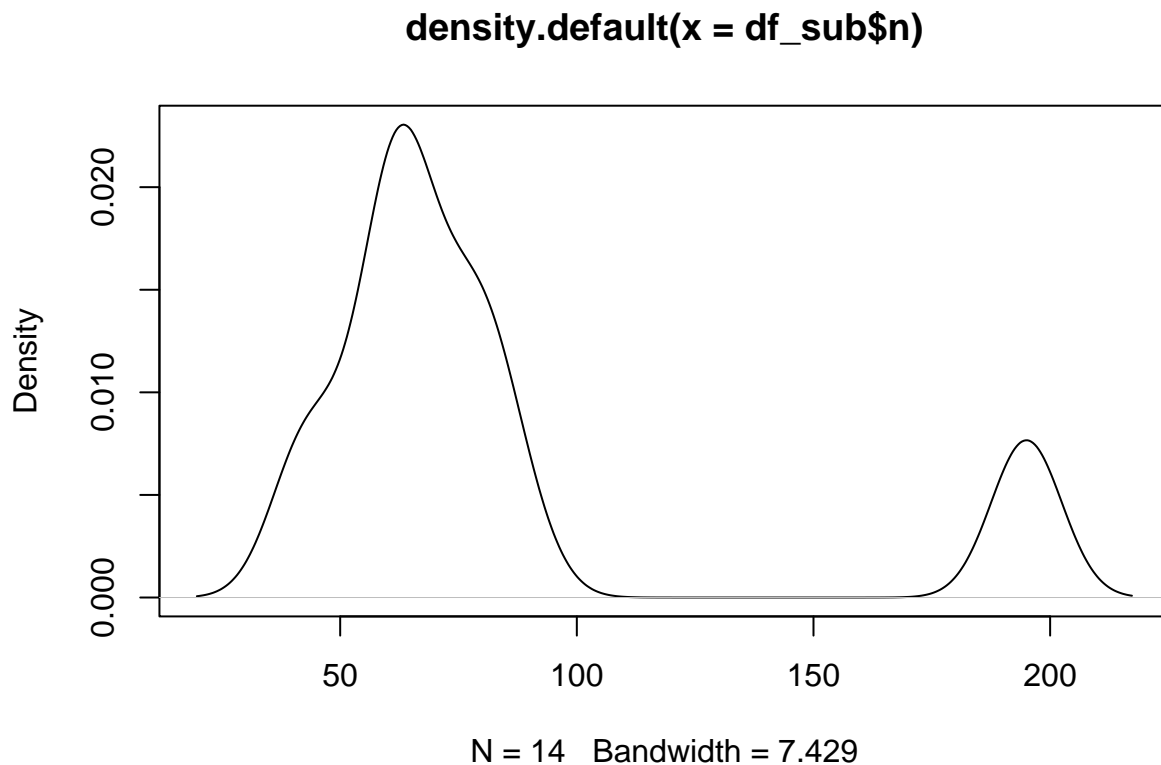
## New names:
## * 'frequency' -> 'frequency...68'
## * 'frequency' -> 'frequency...73'
## * 'frequency' -> 'frequency...78'
## * 'frequency' -> 'frequency...83'
## * 'frequency' -> 'frequency...88'
## * 'frequency' -> 'frequency...93'
## * 'frequency' -> 'frequency...98'
## * 'variability' -> 'variability...108'
## * 'variability' -> 'variability...113'
```

```
## * 'variability' -> 'variability...118'
## * 'variability' -> 'variability...123'
## * 'variability' -> 'variability...128'

# Subset relevant columns and filter out NA's
df_sub <- df %>%
  select(StudyID, MALE_SZ, FEMALE_SZ, MALE_HC, FEMALE_HC, PITCH_FOSD_HC_M, PITCH_FOSD_HC_SD, PITCH_FOSD_
  rename(pitch_hc_m = PITCH_FOSD_HC_M, pitch_hc_sd = PITCH_FOSD_HC_SD, pitch_sz_m = PITCH_FOSD_SZ_M, pi
  filter(pitch_sz_m != "NA") %>%
  filter(pitch_hc_m != "NA") %>%
  filter(MALE_SZ != "NR")

# Summing females and males to n_sz and n_hc
df_sub <- df_sub %>%
  mutate(MALE_HC = as.numeric(MALE_HC), MALE_SZ = as.numeric(MALE_SZ), FEMALE_HC =
    as.numeric(FEMALE_HC), FEMALE_SZ = as.numeric(FEMALE_SZ)) %>%
  mutate(n_sz = MALE_SZ + FEMALE_SZ, n_hc = MALE_HC + FEMALE_HC) %>%
  mutate(n = n_sz + n_hc)

# Investigating participants
dens_n <- density(df_sub$n)
plot(dens_n)
```



```
c(mean(df_sub$n), sd(df_sub$n), min(df_sub$n), max(df_sub$n))
```

```
## [1] 83.64286 48.85553 42.00000 195.00000
```

```

# Standardizing pitch variables
#df_sub <- df_sub %>%
#  mutate(pitch_hc_m_z = (pitch_hc_m-mean(pitch_hc_m))/sd(pitch_hc_m),
#         pitch_hc_sd_z = (pitch_hc_sd-mean(pitch_hc_sd))/sd(pitch_hc_sd),
#         pitch_sz_m_z = (pitch_sz_m-mean(pitch_sz_m))/sd(pitch_sz_m),
#         pitch_sz_sd_z = (pitch_sz_sd-mean(pitch_sz_sd))/sd(pitch_sz_sd)
# )

# Computing PitchMean
PitchMean <- escalc('SMD',
                    n1i = n_sz, n2i = n_hc,
                    m1i = pitch_sz_m, m2i = pitch_hc_m,
                    sd1i = pitch_sz_sd, sd2i = pitch_hc_sd,
                    data = df_sub)

```

```

# Define formula
m_real <- bf(yi | se(vi) ~ 1 + (1|StudyID))

#get_prior(m_real, data = PitchMean)

# Set priors
p_8_jazz<-c(
  prior(normal(0, 0.5), class = Intercept),
  prior(normal(0, 1), class = sd))

# Run model
m_real_prior <- brm(
  m_real,
  PitchMean,
  family = gaussian,
  prior = p_8_jazz,
  sample_prior = "only",
  backend = "cmdstanr",
  chains = 2,
  cores = 2,
  control = list(
    adapt_delta = 0.99,
    max_treedepth = 20
  ))

```

```
## Start sampling
```

```
## Running MCMC with 2 parallel 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)

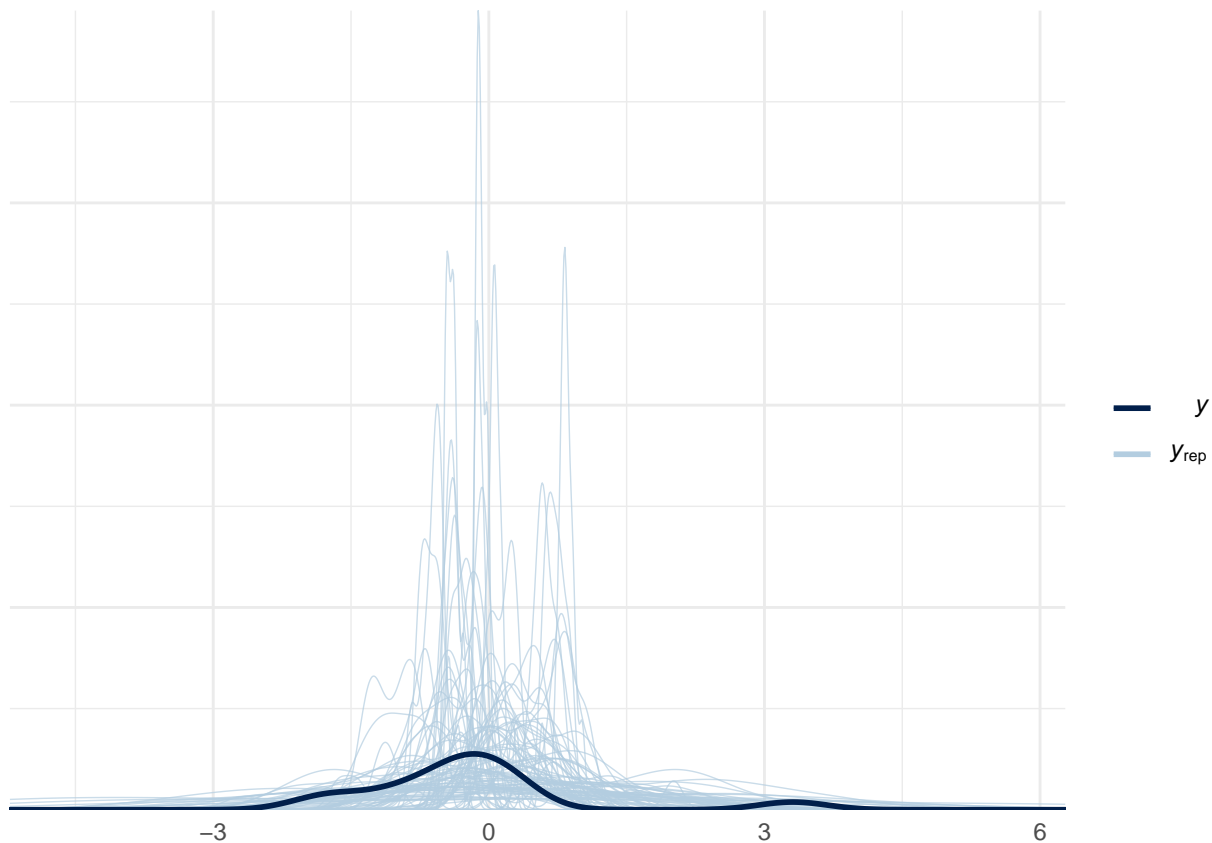
```

```

## 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 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 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 0.2 seconds.
## 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 0.2 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 0.2 seconds.
## Total execution time: 0.3 seconds.

```

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



```
# Fit model
m_real_model <- brm(
  m_real,
  PitchMean,
  family = gaussian,
  prior = p_8_jazz,
  sample_prior = T,
  backend = "cmdstanr",
  chains = 2,
  cores = 2,
  control = list(
    adapt_delta = 0.99,
    max_treedepth = 20
  ))
```

```
## Start sampling
```

```
## Running MCMC with 2 parallel chains...
```

```
##
```

```
## Chain 1 Iteration: 1 / 2000 [ 0%] (Warmup)
```

```
## Chain 2 Iteration: 1 / 2000 [ 0%] (Warmup)
```

```
## Chain 2 Iteration: 100 / 2000 [ 5%] (Warmup)
```

```
## Chain 1 Iteration: 100 / 2000 [ 5%] (Warmup)
```

```
## Chain 2 Iteration: 200 / 2000 [ 10%] (Warmup)
```

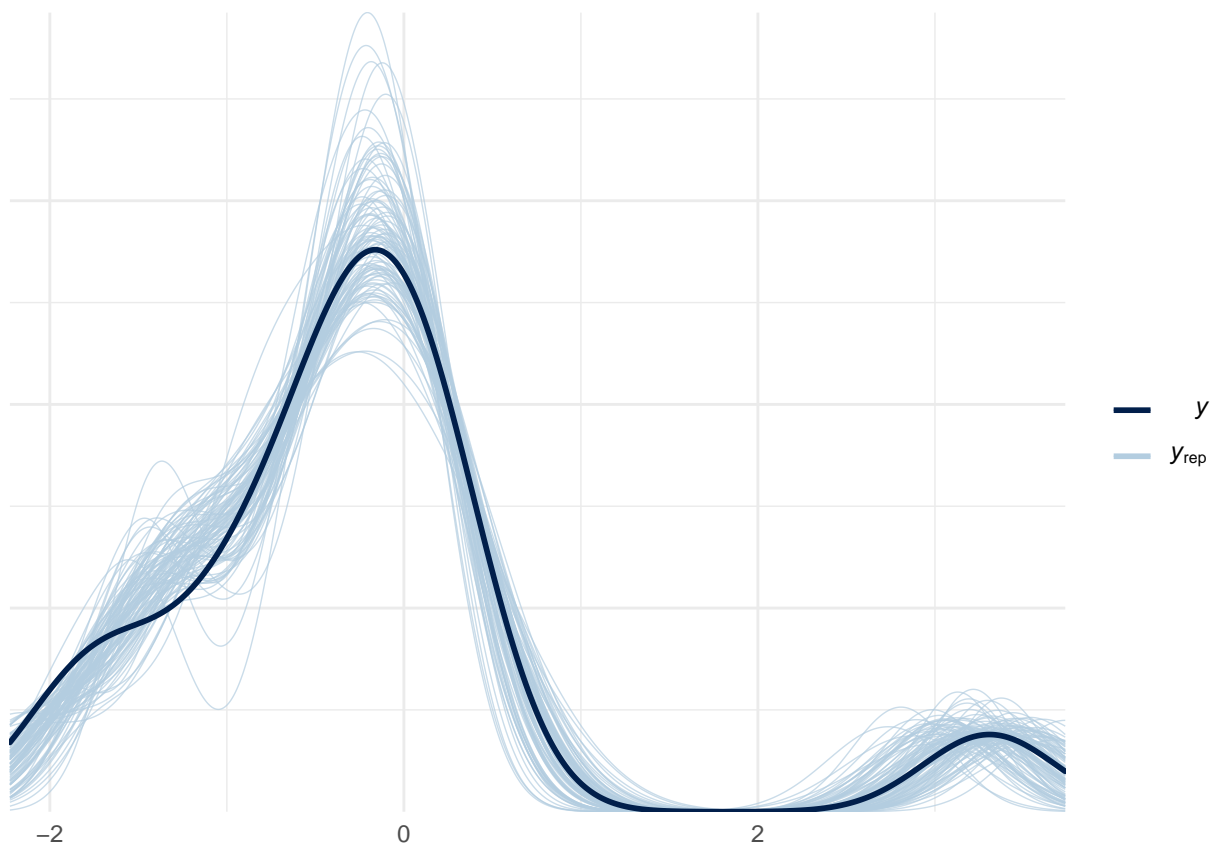
```
## Chain 1 Iteration: 200 / 2000 [ 10%] (Warmup)
```

```

## Chain 2 Iteration: 300 / 2000 [ 15%] (Warmup)
## Chain 1 Iteration: 300 / 2000 [ 15%] (Warmup)
## Chain 2 Iteration: 400 / 2000 [ 20%] (Warmup)
## Chain 1 Iteration: 400 / 2000 [ 20%] (Warmup)
## Chain 2 Iteration: 500 / 2000 [ 25%] (Warmup)
## Chain 1 Iteration: 500 / 2000 [ 25%] (Warmup)
## Chain 1 Iteration: 600 / 2000 [ 30%] (Warmup)
## Chain 2 Iteration: 600 / 2000 [ 30%] (Warmup)
## Chain 2 Iteration: 700 / 2000 [ 35%] (Warmup)
## Chain 1 Iteration: 700 / 2000 [ 35%] (Warmup)
## Chain 2 Iteration: 800 / 2000 [ 40%] (Warmup)
## Chain 1 Iteration: 800 / 2000 [ 40%] (Warmup)
## Chain 2 Iteration: 900 / 2000 [ 45%] (Warmup)
## Chain 1 Iteration: 900 / 2000 [ 45%] (Warmup)
## Chain 2 Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 2 Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 1 Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 2 Iteration: 1100 / 2000 [ 55%] (Sampling)
## Chain 1 Iteration: 1100 / 2000 [ 55%] (Sampling)
## Chain 1 Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 2 Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 1 Iteration: 1300 / 2000 [ 65%] (Sampling)
## Chain 2 Iteration: 1300 / 2000 [ 65%] (Sampling)
## Chain 1 Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 2 Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 1 Iteration: 1500 / 2000 [ 75%] (Sampling)
## Chain 2 Iteration: 1500 / 2000 [ 75%] (Sampling)
## Chain 1 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 2 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 2 Iteration: 1700 / 2000 [ 85%] (Sampling)
## Chain 1 Iteration: 1700 / 2000 [ 85%] (Sampling)
## Chain 2 Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 1 Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 2 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 1 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 2 Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 2 finished in 2.4 seconds.
## Chain 1 Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 1 finished in 2.5 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 2.4 seconds.
## Total execution time: 2.6 seconds.

```

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



```
summary(m_real_model)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: yi | se(vi) ~ 1 + (1 | StudyID)
## Data: PitchMean (Number of observations: 14)
## Draws: 2 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 2000
##
## Group-Level Effects:
## ~StudyID (Number of levels: 11)
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    1.29    0.26    0.88    1.93 1.02    209    362
##
## Population-Level Effects:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -0.09    0.29   -0.67    0.50 1.01    204    268
##
## Family Specific Parameters:
##      Estimate Est.Error 1-95% 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).
```

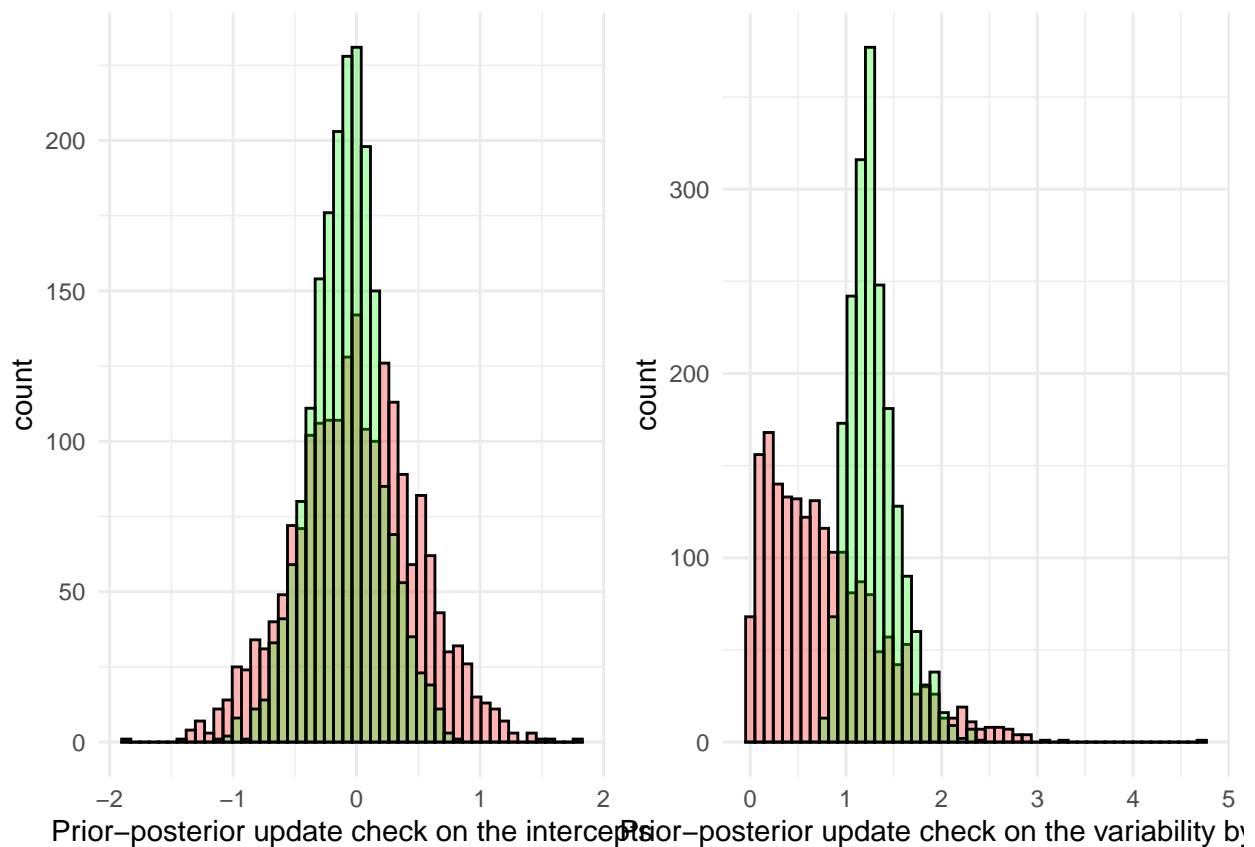


```
real_posterior <- as_draws_df(m_real_model)
```

```
real_plot1 <- ggplot(real_posterior) +
  geom_histogram(aes(prior_Intercept), fill = "red", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(b_Intercept), fill = "green", color = "black", alpha = 0.3, bins = 50) +
  xlab("Prior-posterior update check on the intercepts")
```

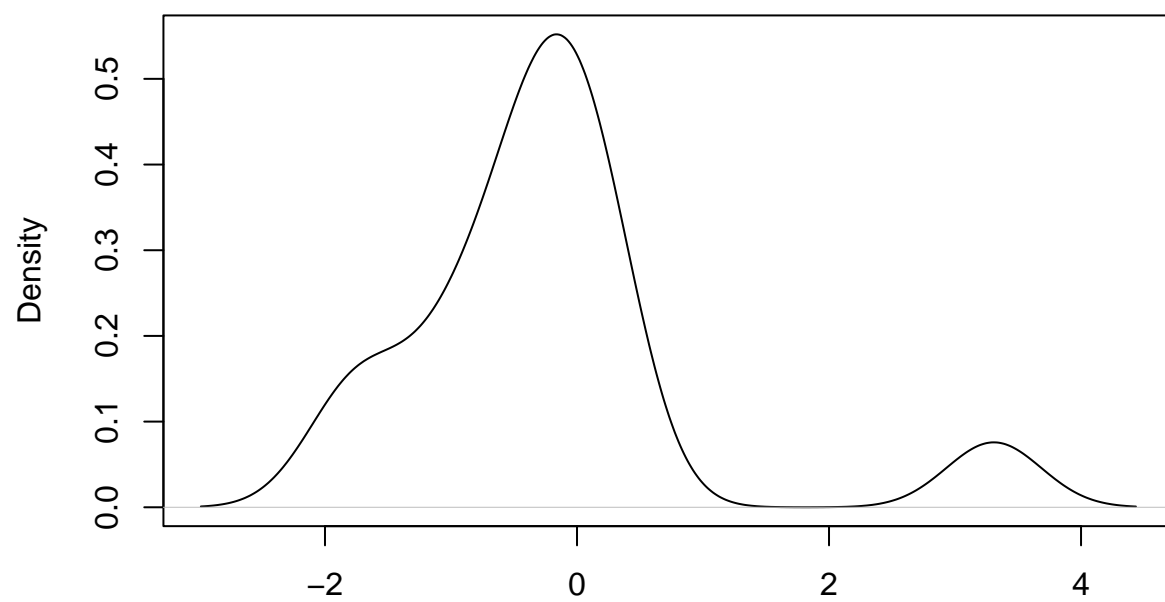
```
real_plot2 <- ggplot(real_posterior) +
  geom_histogram(aes(prior_sd_StudyID), fill = "red", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(sd_StudyID__Intercept), fill = "green", color = "black", alpha = 0.3, bins = 50) +
  xlab("Prior-posterior update check on the variability by study")
```

```
ggarrange(real_plot1, real_plot2, ncol = 2, nrow = 1)
```



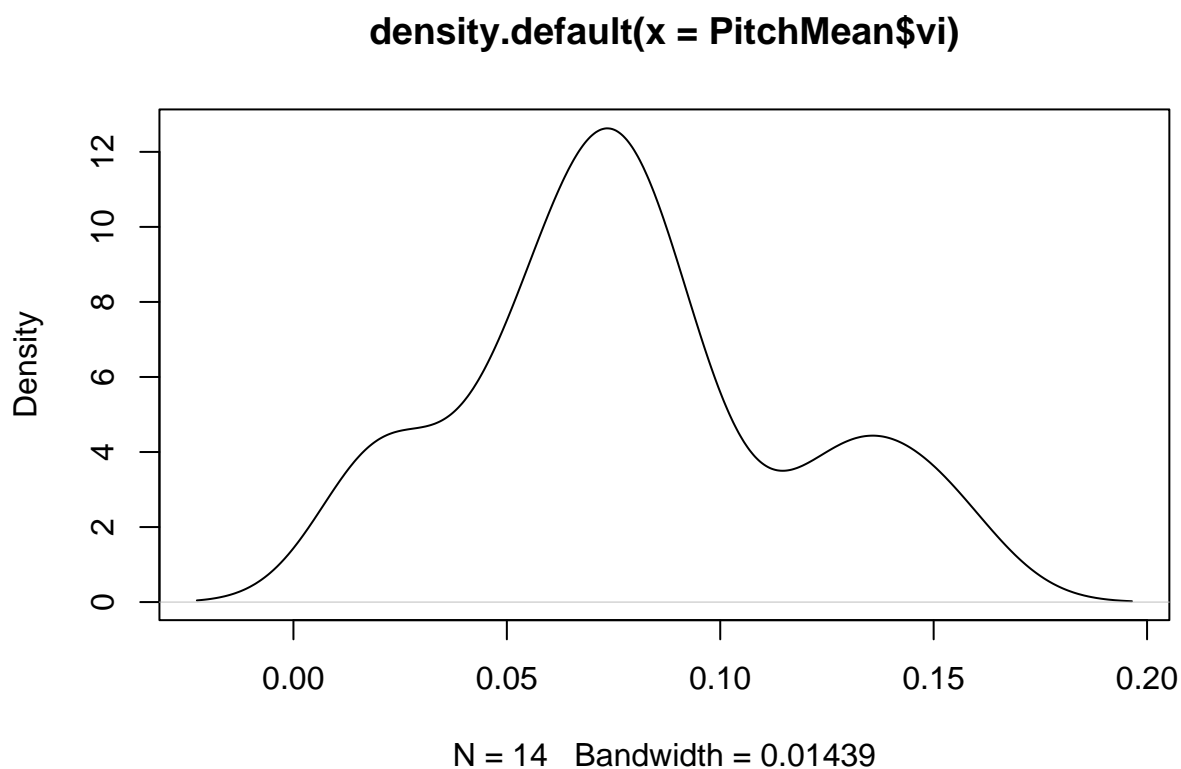
```
dens <- density(PitchMean$yi)
plot(dens)
```

density.default(x = PitchMean\$yi)



N = 14 Bandwidth = 0.3759

```
vary <- density(PitchMean$vi)  
plot(vary)
```



```
test <- PitchMean %>%  
  filter(yi < 2)  
  
# Fit model  
m_real_model_test <- brm(  
  m_real,  
  test,  
  family = gaussian,  
  prior = p_8_jazz,  
  sample_prior = T,  
  backend = "cmdstanr",  
  chains = 2,  
  cores = 2,  
  control = list(  
    adapt_delta = 0.99,  
    max_treedepth = 20  
  ))
```

Start sampling

Running MCMC with 2 parallel chains...

##

Chain 1 Iteration: 1 / 2000 [0%] (Warmup)

Chain 1 Iteration: 100 / 2000 [5%] (Warmup)

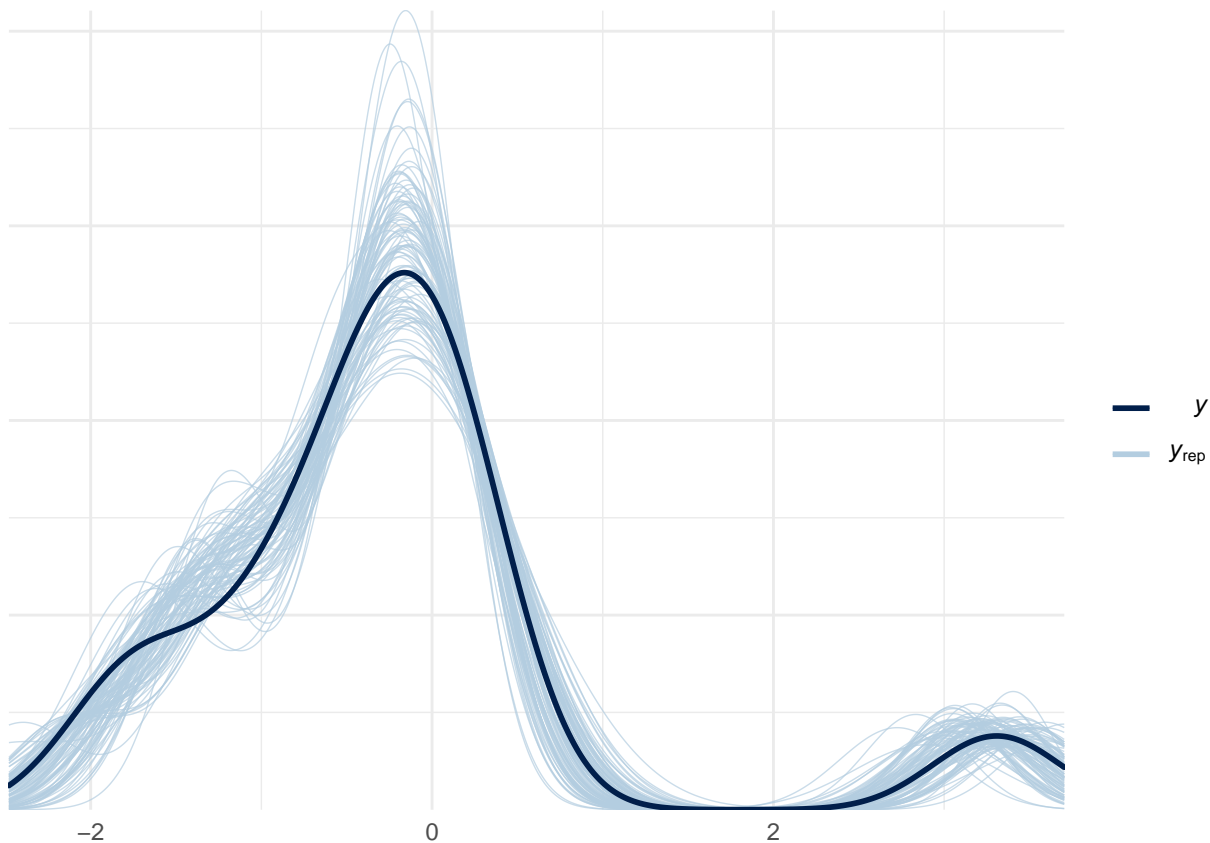
Chain 2 Iteration: 1 / 2000 [0%] (Warmup)

```

## Chain 2 Iteration: 100 / 2000 [ 5%] (Warmup)
## Chain 1 Iteration: 200 / 2000 [ 10%] (Warmup)
## Chain 2 Iteration: 200 / 2000 [ 10%] (Warmup)
## Chain 1 Iteration: 300 / 2000 [ 15%] (Warmup)
## Chain 1 Iteration: 400 / 2000 [ 20%] (Warmup)
## Chain 2 Iteration: 300 / 2000 [ 15%] (Warmup)
## Chain 2 Iteration: 400 / 2000 [ 20%] (Warmup)
## Chain 1 Iteration: 500 / 2000 [ 25%] (Warmup)
## Chain 2 Iteration: 500 / 2000 [ 25%] (Warmup)
## Chain 2 Iteration: 600 / 2000 [ 30%] (Warmup)
## Chain 1 Iteration: 600 / 2000 [ 30%] (Warmup)
## Chain 1 Iteration: 700 / 2000 [ 35%] (Warmup)
## Chain 1 Iteration: 800 / 2000 [ 40%] (Warmup)
## Chain 2 Iteration: 700 / 2000 [ 35%] (Warmup)
## Chain 1 Iteration: 900 / 2000 [ 45%] (Warmup)
## Chain 2 Iteration: 800 / 2000 [ 40%] (Warmup)
## Chain 1 Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 1 Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 2 Iteration: 900 / 2000 [ 45%] (Warmup)
## Chain 2 Iteration: 1000 / 2000 [ 50%] (Warmup)
## Chain 1 Iteration: 1100 / 2000 [ 55%] (Sampling)
## Chain 2 Iteration: 1001 / 2000 [ 50%] (Sampling)
## Chain 1 Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 2 Iteration: 1100 / 2000 [ 55%] (Sampling)
## Chain 1 Iteration: 1300 / 2000 [ 65%] (Sampling)
## Chain 2 Iteration: 1200 / 2000 [ 60%] (Sampling)
## Chain 1 Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 2 Iteration: 1300 / 2000 [ 65%] (Sampling)
## Chain 1 Iteration: 1500 / 2000 [ 75%] (Sampling)
## Chain 2 Iteration: 1400 / 2000 [ 70%] (Sampling)
## Chain 2 Iteration: 1500 / 2000 [ 75%] (Sampling)
## Chain 1 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 2 Iteration: 1600 / 2000 [ 80%] (Sampling)
## Chain 1 Iteration: 1700 / 2000 [ 85%] (Sampling)
## Chain 2 Iteration: 1700 / 2000 [ 85%] (Sampling)
## Chain 1 Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 2 Iteration: 1800 / 2000 [ 90%] (Sampling)
## Chain 1 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 2 Iteration: 1900 / 2000 [ 95%] (Sampling)
## Chain 1 Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 2 Iteration: 2000 / 2000 [100%] (Sampling)
## Chain 1 finished in 2.3 seconds.
## Chain 2 finished in 2.2 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 2.3 seconds.
## Total execution time: 2.4 seconds.

```

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



```
summary(m_real_model)
```

```
## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: yi | se(vi) ~ 1 + (1 | StudyID)
## Data: PitchMean (Number of observations: 14)
## Draws: 2 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 2000
##
## Group-Level Effects:
## ~StudyID (Number of levels: 11)
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    1.29    0.26    0.88    1.93 1.02    209    362
##
## Population-Level Effects:
##      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -0.09    0.29   -0.67    0.50 1.01    204    268
##
## Family Specific Parameters:
##      Estimate Est.Error 1-95% 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).
```

```

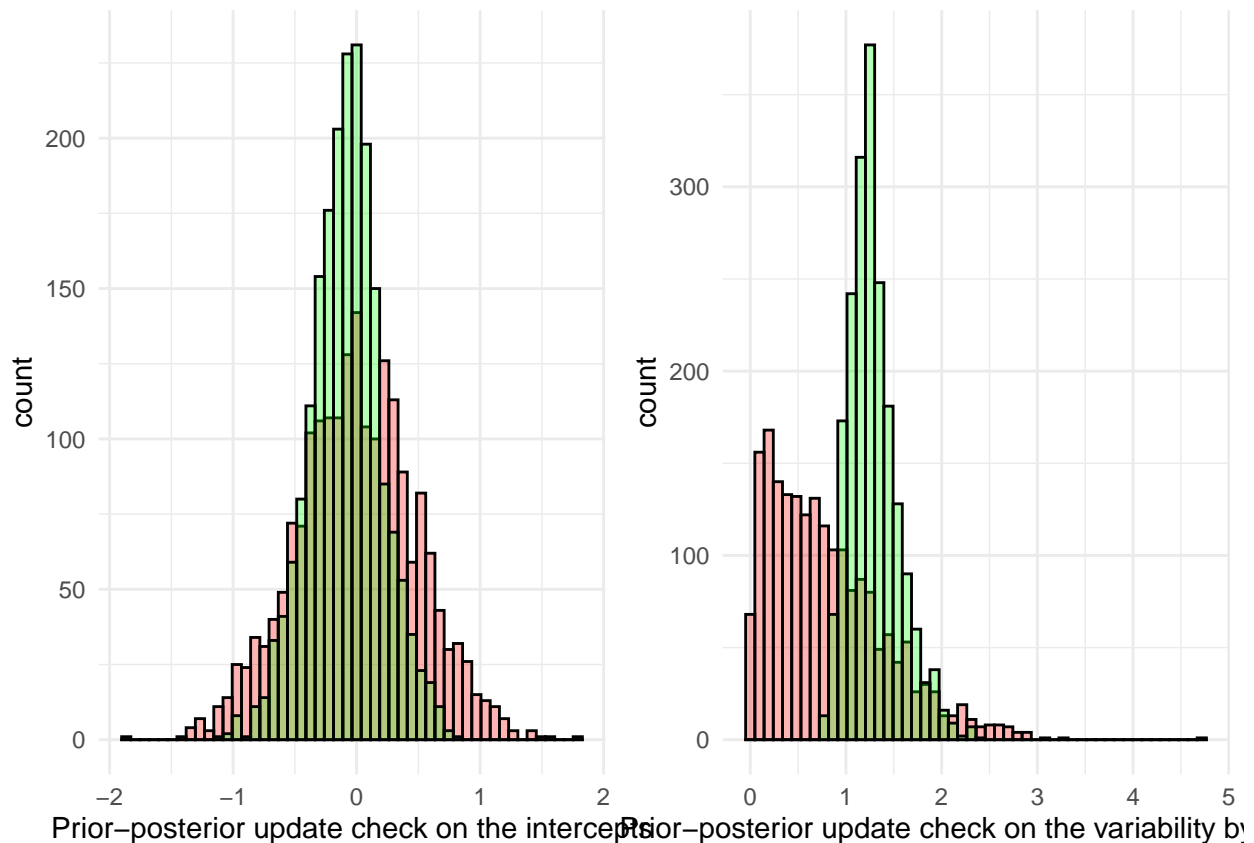
real_posterior_test <- as_draws_df(m_real_model)

# Plotting prior-posterior update checks without extreme effect sizes
real_plot1_test <- ggplot(real_posterior_test) +
  geom_histogram(aes(prior_Intercept), fill = "red", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(b_Intercept), fill = "green", color = "black", alpha = 0.3, bins = 50) +
  xlab("Prior-posterior update check on the intercepts")

real_plot2_test <- ggplot(real_posterior_test) +
  geom_histogram(aes(prior_sd_StudyID), fill = "red", color = "black", alpha = 0.3, bins = 50) +
  geom_histogram(aes(sd_StudyID__Intercept), fill = "green", color = "black", alpha = 0.3, bins = 50) +
  xlab("Prior-posterior update check on the variability by study")

ggarrange(real_plot1_test, real_plot2_test, ncol = 2, nrow = 1)

```



```

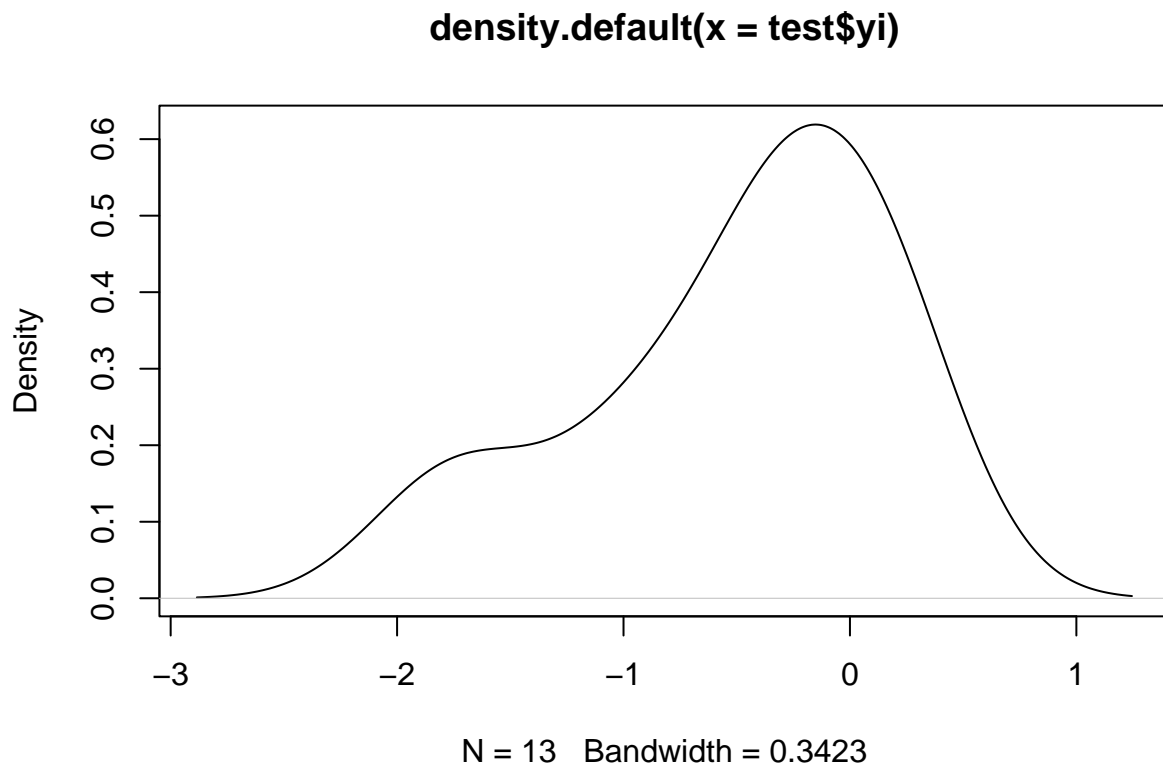
summary(m_real_model_test)

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: yi | se(vi) ~ 1 + (1 | StudyID)
## Data: test (Number of observations: 13)
## Draws: 2 chains, each with iter = 2000; warmup = 1000; thin = 1;
## total post-warmup draws = 2000

```

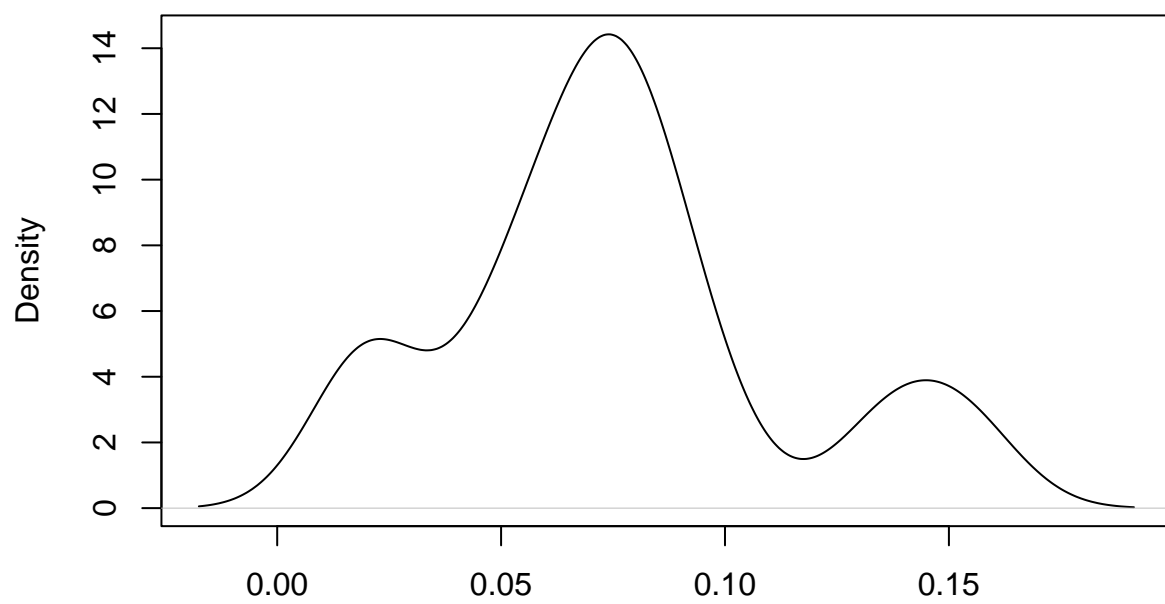
```
##
## Group-Level Effects:
## ~StudyID (Number of levels: 10)
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    0.76     0.21    0.47    1.31 1.01     273     335
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    -0.43     0.23   -0.91    0.02 1.00     263     258
##
## 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).
```

```
dens_test <- density(test$yi)
plot(dens_test)
```



```
vary_test <- density(test$vi)
plot(vary_test)
```

density.default(x = test\$vi)



N = 13 Bandwidth = 0.01268