

Test Analysis

```
test_data <- read_csv("https://raw.githubusercontent.com/Theaang/so-sp-coupling/main/Data/

eszcor <- escalc(measure="ZCOR", ri=es, ni=n, data=test_data)
test_data$eszcor <- eszcor$yi
test_data$vi <- eszcor$vi
test_data$se <- sqrt(eszcor$vi)
priors <- c(prior(normal(0,1), class = b, coef = Intercept),
prior(cauchy(0,0.5), class = sd))

seed <- 2023
iter <- 5000
chains <- 4

priors <- c(prior(normal(0,1), class = b, coef = Intercept),
prior(cauchy(0,0.5), class = sd))

meta_prob <-
  brm(
    eszcor|se(se) ~ 0 + Intercept + (1|esid),
    data = test_data,
    family = gaussian(),
    prior = priors,
    sample_prior = T,
    iter = iter,
    save_pars = save_pars(all = TRUE),
    control = list(adapt_delta = 0.99, max_treedepth = 15),
    seed = seed,
    chains = chains
  )
```

Compiling Stan program...

Trying to compile a simple C file

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen:
namespace Eigen {
~
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen:
namespace Eigen {
~
;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen:
#include <complex>
~~~~~~
3 errors generated.
make: *** [foo.o] Error 1
```

Start sampling

SAMPLING FOR MODEL '1d0f2777348f6b2e21c93c848f58a532' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 3.9e-05 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.39 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 5000 [0%] (Warmup)

Chain 1: Iteration: 500 / 5000 [10%] (Warmup)

Chain 1: Iteration: 1000 / 5000 [20%] (Warmup)

Chain 1: Iteration: 1500 / 5000 [30%] (Warmup)

Chain 1: Iteration: 2000 / 5000 [40%] (Warmup)

Chain 1: Iteration: 2500 / 5000 [50%] (Warmup)

Chain 1: Iteration: 2501 / 5000 [50%] (Sampling)

Chain 1: Iteration: 3000 / 5000 [60%] (Sampling)

```

Chain 1: Iteration: 3500 / 5000 [ 70%] (Sampling)
Chain 1: Iteration: 4000 / 5000 [ 80%] (Sampling)
Chain 1: Iteration: 4500 / 5000 [ 90%] (Sampling)
Chain 1: Iteration: 5000 / 5000 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 0.890596 seconds (Warm-up)
Chain 1:           1.27887 seconds (Sampling)
Chain 1:           2.16947 seconds (Total)
Chain 1:

```

SAMPLING FOR MODEL '1d0f2777348f6b2e21c93c848f58a532' NOW (CHAIN 2).

```

Chain 2:
Chain 2: Gradient evaluation took 1.8e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.18 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 5000 [  0%] (Warmup)
Chain 2: Iteration:   500 / 5000 [ 10%] (Warmup)
Chain 2: Iteration:  1000 / 5000 [ 20%] (Warmup)
Chain 2: Iteration:  1500 / 5000 [ 30%] (Warmup)
Chain 2: Iteration:  2000 / 5000 [ 40%] (Warmup)
Chain 2: Iteration:  2500 / 5000 [ 50%] (Warmup)
Chain 2: Iteration: 2501 / 5000 [ 50%] (Sampling)
Chain 2: Iteration:  3000 / 5000 [ 60%] (Sampling)
Chain 2: Iteration:  3500 / 5000 [ 70%] (Sampling)
Chain 2: Iteration:  4000 / 5000 [ 80%] (Sampling)
Chain 2: Iteration:  4500 / 5000 [ 90%] (Sampling)
Chain 2: Iteration:  5000 / 5000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 0.958916 seconds (Warm-up)
Chain 2:           0.721962 seconds (Sampling)
Chain 2:           1.68088 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL '1d0f2777348f6b2e21c93c848f58a532' NOW (CHAIN 3).

```

Chain 3:
Chain 3: Gradient evaluation took 1.1e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.11 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 5000 [  0%] (Warmup)

```

```

Chain 3: Iteration: 500 / 5000 [ 10%] (Warmup)
Chain 3: Iteration: 1000 / 5000 [ 20%] (Warmup)
Chain 3: Iteration: 1500 / 5000 [ 30%] (Warmup)
Chain 3: Iteration: 2000 / 5000 [ 40%] (Warmup)
Chain 3: Iteration: 2500 / 5000 [ 50%] (Warmup)
Chain 3: Iteration: 2501 / 5000 [ 50%] (Sampling)
Chain 3: Iteration: 3000 / 5000 [ 60%] (Sampling)
Chain 3: Iteration: 3500 / 5000 [ 70%] (Sampling)
Chain 3: Iteration: 4000 / 5000 [ 80%] (Sampling)
Chain 3: Iteration: 4500 / 5000 [ 90%] (Sampling)
Chain 3: Iteration: 5000 / 5000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 0.963621 seconds (Warm-up)
Chain 3: 0.689415 seconds (Sampling)
Chain 3: 1.65304 seconds (Total)
Chain 3:

```

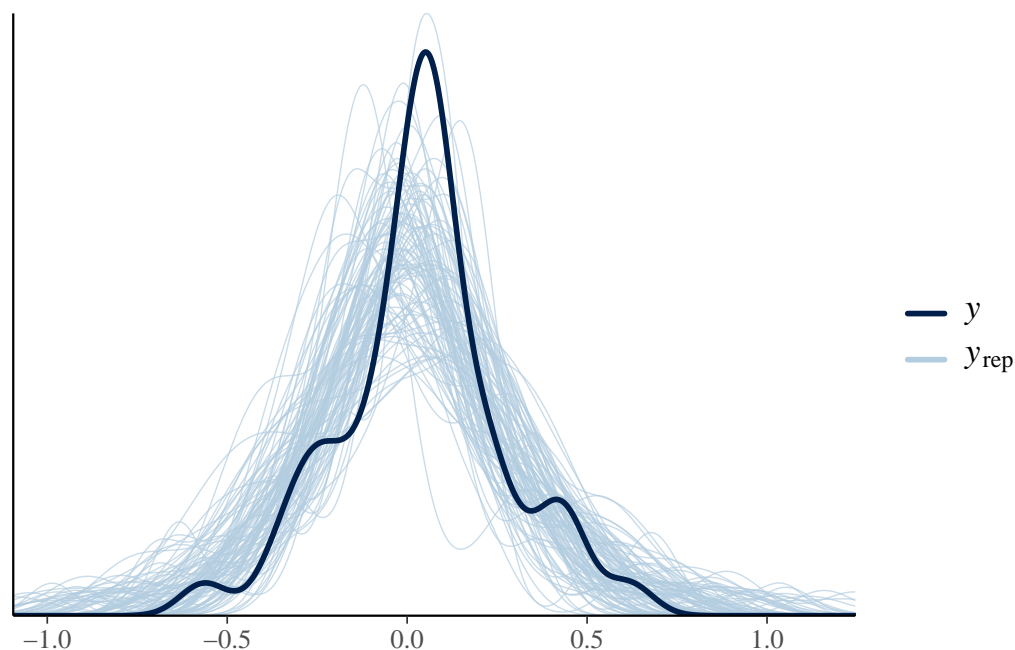
SAMPLING FOR MODEL '1d0f2777348f6b2e21c93c848f58a532' NOW (CHAIN 4).

```

Chain 4:
Chain 4: Gradient evaluation took 1.6e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.16 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration: 1 / 5000 [ 0%] (Warmup)
Chain 4: Iteration: 500 / 5000 [ 10%] (Warmup)
Chain 4: Iteration: 1000 / 5000 [ 20%] (Warmup)
Chain 4: Iteration: 1500 / 5000 [ 30%] (Warmup)
Chain 4: Iteration: 2000 / 5000 [ 40%] (Warmup)
Chain 4: Iteration: 2500 / 5000 [ 50%] (Warmup)
Chain 4: Iteration: 2501 / 5000 [ 50%] (Sampling)
Chain 4: Iteration: 3000 / 5000 [ 60%] (Sampling)
Chain 4: Iteration: 3500 / 5000 [ 70%] (Sampling)
Chain 4: Iteration: 4000 / 5000 [ 80%] (Sampling)
Chain 4: Iteration: 4500 / 5000 [ 90%] (Sampling)
Chain 4: Iteration: 5000 / 5000 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 1.3146 seconds (Warm-up)
Chain 4: 0.641053 seconds (Sampling)
Chain 4: 1.95566 seconds (Total)
Chain 4:

```

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



```
summary(meta_prob)
```

```
Family: gaussian
Links: mu = identity; sigma = identity
Formula: eszcor | se(se) ~ 0 + Intercept + (1 | esid)
Data: test_data (Number of observations: 42)
Draws: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
       total post-warmup draws = 10000

Group-Level Effects:
~esid (Number of levels: 42)
      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)    0.05    0.04    0.00    0.14 1.00    4350    4801

Population-Level Effects:
      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    0.00    0.03   -0.06    0.06 1.00    17614    7616

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 sampling(NUTS). 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).

```
get_variables(meta_prob)
```

```
[1] "b_Intercept"          "sd_esid__Intercept"  "sigma"
[4] "r_esid[1,Intercept]"  "r_esid[2,Intercept]" "r_esid[3,Intercept]"
[7] "r_esid[4,Intercept]"  "r_esid[5,Intercept]" "r_esid[6,Intercept]"
[10] "r_esid[7,Intercept]"  "r_esid[8,Intercept]" "r_esid[9,Intercept]"
[13] "r_esid[10,Intercept]" "r_esid[11,Intercept]" "r_esid[12,Intercept]"
[16] "r_esid[13,Intercept]" "r_esid[14,Intercept]" "r_esid[15,Intercept]"
[19] "r_esid[16,Intercept]" "r_esid[17,Intercept]" "r_esid[18,Intercept]"
[22] "r_esid[19,Intercept]" "r_esid[20,Intercept]" "r_esid[21,Intercept]"
[25] "r_esid[22,Intercept]" "r_esid[23,Intercept]" "r_esid[24,Intercept]"
[28] "r_esid[25,Intercept]" "r_esid[26,Intercept]" "r_esid[27,Intercept]"
[31] "r_esid[28,Intercept]" "r_esid[29,Intercept]" "r_esid[30,Intercept]"
[34] "r_esid[31,Intercept]" "r_esid[32,Intercept]" "r_esid[33,Intercept]"
[37] "r_esid[34,Intercept]" "r_esid[35,Intercept]" "r_esid[36,Intercept]"
[40] "r_esid[37,Intercept]" "r_esid[38,Intercept]" "r_esid[39,Intercept]"
[43] "r_esid[40,Intercept]" "r_esid[41,Intercept]" "r_esid[42,Intercept]"
[46] "prior_b_Intercept"    "prior_sd_esid"       "lprior"
[49] "lp__"                 "z_1[1,1]"            "z_1[1,2]"
[52] "z_1[1,3]"             "z_1[1,4]"            "z_1[1,5]"
[55] "z_1[1,6]"             "z_1[1,7]"            "z_1[1,8]"
[58] "z_1[1,9]"             "z_1[1,10]"           "z_1[1,11]"
[61] "z_1[1,12]"            "z_1[1,13]"           "z_1[1,14]"
[64] "z_1[1,15]"            "z_1[1,16]"           "z_1[1,17]"
[67] "z_1[1,18]"            "z_1[1,19]"           "z_1[1,20]"
[70] "z_1[1,21]"            "z_1[1,22]"           "z_1[1,23]"
[73] "z_1[1,24]"            "z_1[1,25]"           "z_1[1,26]"
[76] "z_1[1,27]"            "z_1[1,28]"           "z_1[1,29]"
[79] "z_1[1,30]"            "z_1[1,31]"           "z_1[1,32]"
[82] "z_1[1,33]"            "z_1[1,34]"           "z_1[1,35]"
[85] "z_1[1,36]"            "z_1[1,37]"           "z_1[1,38]"
[88] "z_1[1,39]"            "z_1[1,40]"           "z_1[1,41]"
[91] "z_1[1,42]"            "accept_stat__"        "stepsize__"
[94] "treedepth__"          "n_leapfrog__"         "divergent__"
[97] "energy__"
```

```
fixef(meta_prob)
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	0.002241	0.03155	-0.05906	0.06494

```
analysis <- metafor::rma.mv(yi = eszcor, V = vi, data = test_data, random = ~ 1 | esid/stu  
summary(analysis)
```

Multivariate Meta-Analysis Model (k = 42; method: REML)

logLik	Deviance	AIC	BIC	AICc
5.0810	-10.1620	-4.1620	0.9787	-3.5133

Variance Components:

	estim	sqrt	nlvls	fixed	factor
sigma^2.1	0.0000	0.0000	42	no	esid
sigma^2.2	0.0000	0.0000	42	no	esid/studyid

Test for Heterogeneity:

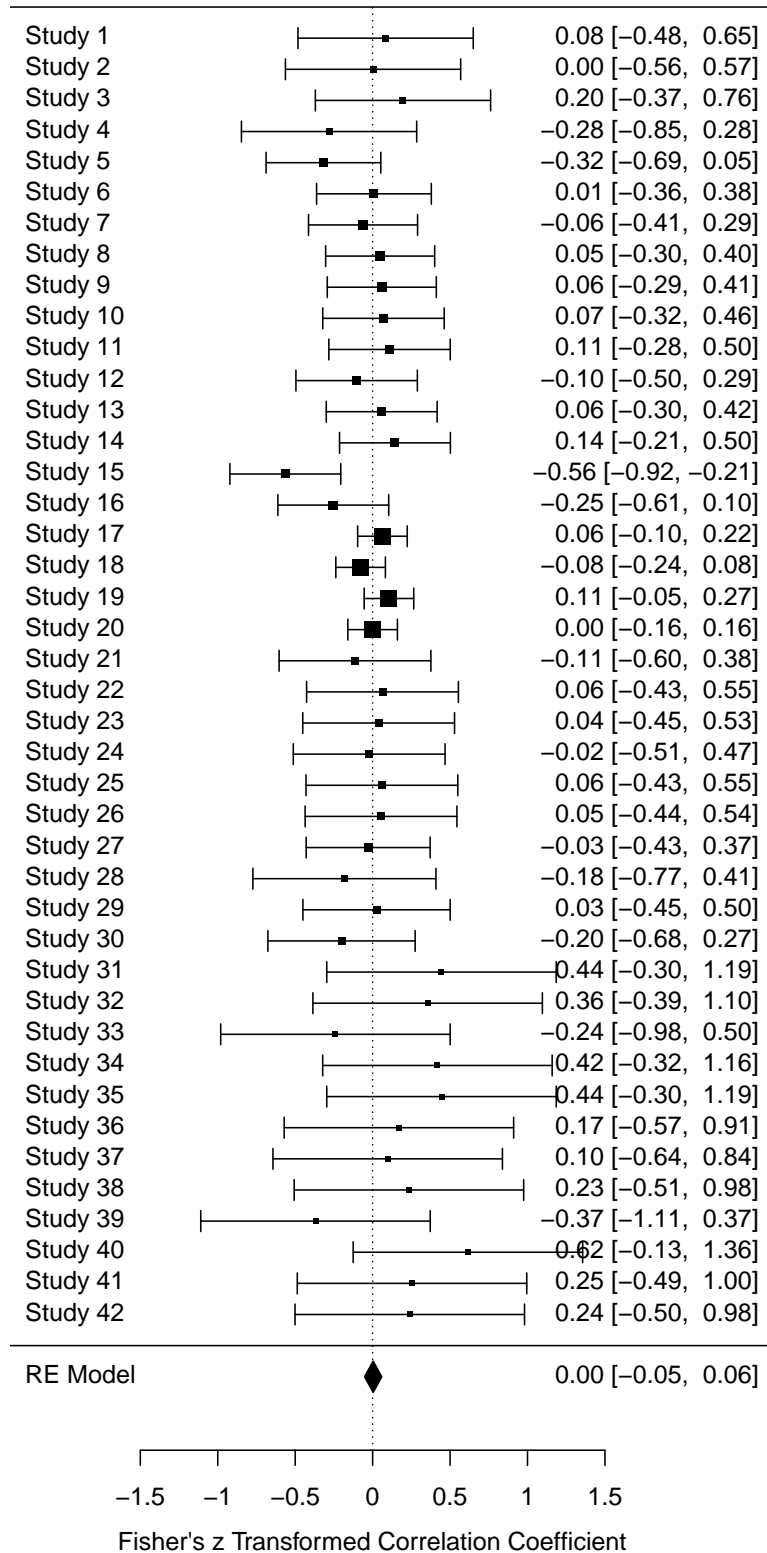
Q(df = 41) = 32.5378, p-val = 0.8245

Model Results:

estimate	se	zval	pval	ci.lb	ci.ub
0.0033	0.0284	0.1164	0.9073	-0.0523	0.0589

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
metafor::forest(analysis)
```



SAMPLING FOR MODEL '8630b7bdb90dfccf834b55a121fecbd8' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 7.3e-05 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.73 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 5000 [0%] (Warmup)

Chain 1: Iteration: 500 / 5000 [10%] (Warmup)

Chain 1: Iteration: 1000 / 5000 [20%] (Warmup)

Chain 1: Iteration: 1500 / 5000 [30%] (Warmup)

Chain 1: Iteration: 2000 / 5000 [40%] (Warmup)

Chain 1: Iteration: 2500 / 5000 [50%] (Warmup)

Chain 1: Iteration: 2501 / 5000 [50%] (Sampling)

Chain 1: Iteration: 3000 / 5000 [60%] (Sampling)

Chain 1: Iteration: 3500 / 5000 [70%] (Sampling)

Chain 1: Iteration: 4000 / 5000 [80%] (Sampling)

Chain 1: Iteration: 4500 / 5000 [90%] (Sampling)

Chain 1: Iteration: 5000 / 5000 [100%] (Sampling)

Chain 1:

Chain 1: Elapsed Time: 1.36795 seconds (Warm-up)

Chain 1: 1.60654 seconds (Sampling)

Chain 1: 2.97449 seconds (Total)

Chain 1:

SAMPLING FOR MODEL '8630b7bdb90dfccf834b55a121fecbd8' NOW (CHAIN 2).

Chain 2:

Chain 2: Gradient evaluation took 1.5e-05 seconds

Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.15 seconds.

Chain 2: Adjust your expectations accordingly!

Chain 2:

Chain 2:

Chain 2: Iteration: 1 / 5000 [0%] (Warmup)

Chain 2: Iteration: 500 / 5000 [10%] (Warmup)

Chain 2: Iteration: 1000 / 5000 [20%] (Warmup)

Chain 2: Iteration: 1500 / 5000 [30%] (Warmup)

Chain 2: Iteration: 2000 / 5000 [40%] (Warmup)

Chain 2: Iteration: 2500 / 5000 [50%] (Warmup)

Chain 2: Iteration: 2501 / 5000 [50%] (Sampling)

Chain 2: Iteration: 3000 / 5000 [60%] (Sampling)

Chain 2: Iteration: 3500 / 5000 [70%] (Sampling)

Chain 2: Iteration: 4000 / 5000 [80%] (Sampling)

Chain 2: Iteration: 4500 / 5000 [90%] (Sampling)
Chain 2: Iteration: 5000 / 5000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 1.3778 seconds (Warm-up)
Chain 2: 1.39262 seconds (Sampling)
Chain 2: 2.77042 seconds (Total)
Chain 2:

SAMPLING FOR MODEL '8630b7bdb90dfccf834b55a121fecbd8' NOW (CHAIN 3).

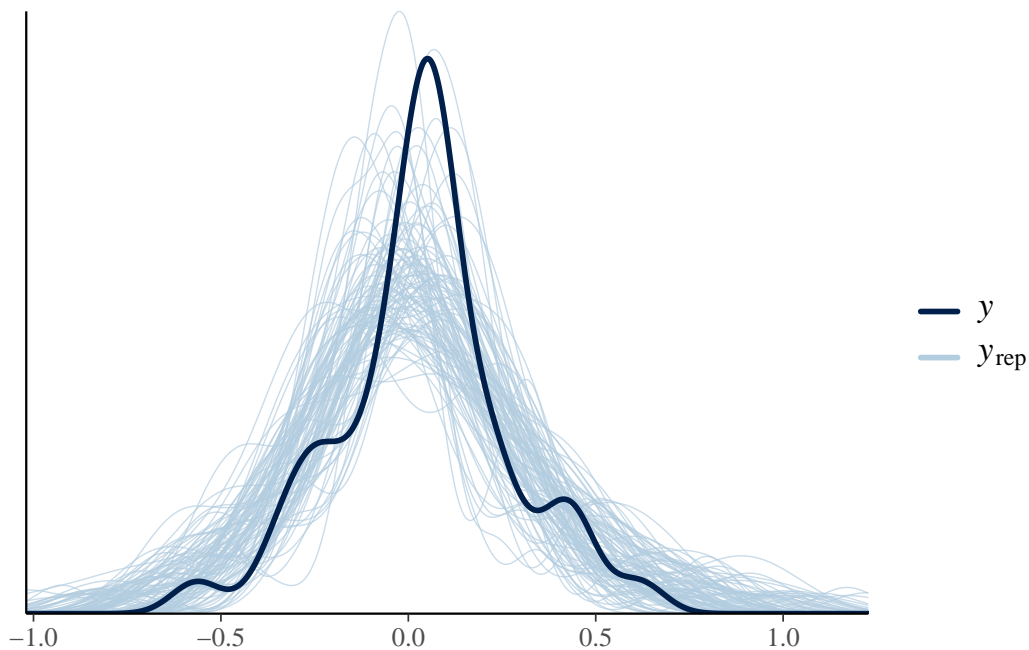
Chain 3:
Chain 3: Gradient evaluation took 1.6e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.16 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration: 1 / 5000 [0%] (Warmup)
Chain 3: Iteration: 500 / 5000 [10%] (Warmup)
Chain 3: Iteration: 1000 / 5000 [20%] (Warmup)
Chain 3: Iteration: 1500 / 5000 [30%] (Warmup)
Chain 3: Iteration: 2000 / 5000 [40%] (Warmup)
Chain 3: Iteration: 2500 / 5000 [50%] (Warmup)
Chain 3: Iteration: 2501 / 5000 [50%] (Sampling)
Chain 3: Iteration: 3000 / 5000 [60%] (Sampling)
Chain 3: Iteration: 3500 / 5000 [70%] (Sampling)
Chain 3: Iteration: 4000 / 5000 [80%] (Sampling)
Chain 3: Iteration: 4500 / 5000 [90%] (Sampling)
Chain 3: Iteration: 5000 / 5000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 1.55195 seconds (Warm-up)
Chain 3: 1.58028 seconds (Sampling)
Chain 3: 3.13223 seconds (Total)
Chain 3:

SAMPLING FOR MODEL '8630b7bdb90dfccf834b55a121fecbd8' NOW (CHAIN 4).

Chain 4:
Chain 4: Gradient evaluation took 1.4e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.14 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration: 1 / 5000 [0%] (Warmup)
Chain 4: Iteration: 500 / 5000 [10%] (Warmup)
Chain 4: Iteration: 1000 / 5000 [20%] (Warmup)

```
Chain 4: Iteration: 1500 / 5000 [ 30%] (Warmup)
Chain 4: Iteration: 2000 / 5000 [ 40%] (Warmup)
Chain 4: Iteration: 2500 / 5000 [ 50%] (Warmup)
Chain 4: Iteration: 2501 / 5000 [ 50%] (Sampling)
Chain 4: Iteration: 3000 / 5000 [ 60%] (Sampling)
Chain 4: Iteration: 3500 / 5000 [ 70%] (Sampling)
Chain 4: Iteration: 4000 / 5000 [ 80%] (Sampling)
Chain 4: Iteration: 4500 / 5000 [ 90%] (Sampling)
Chain 4: Iteration: 5000 / 5000 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 1.33213 seconds (Warm-up)
Chain 4:                1.61514 seconds (Sampling)
Chain 4:                2.94727 seconds (Total)
Chain 4:
```

```
pp_check(meta_prob2, ndraws = 100)
```



```
summary(meta_prob2)
```

```
Family: gaussian
Links: mu = identity; sigma = identity
```

Formula: eszcor | se(se) ~ 0 + Intercept + (1 | studyid/esid)
 Data: test_data (Number of observations: 42)
 Draws: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
 total post-warmup draws = 10000

Group-Level Effects:

~studyid (Number of levels: 9)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	0.07	0.05	0.00	0.20	1.00	3605	5324

~studyid:esid (Number of levels: 42)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	0.05	0.04	0.00	0.13	1.00	4275	4203

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-0.00	0.05	-0.10	0.08	1.00	6919	6222

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 sampling(NUTS). 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).

```
study.draws <- spread_draws(meta_prob, r_esid[id, ], b_Intercept) %>%
  mutate(b_Intercept = r_esid + b_Intercept)
pooled.effect.draws <- spread_draws(meta_prob, b_Intercept) %>%
  mutate(Author = "Pooled Effect")
forest.data <- bind_rows(study.draws,
  pooled.effect.draws) %>%
  ungroup() %>%
  mutate(id = str_replace_all(id, "[.]", " ")) %>%
  mutate(id = reorder(id, b_Intercept))

meta_prob %>%
  spread_draws(b_Intercept, r_esid[id,]) %>%
  # add the grand mean to the group-specific deviations
  mutate(b_Intercept = b_Intercept + r_esid) %>%
  ungroup() %>%
```

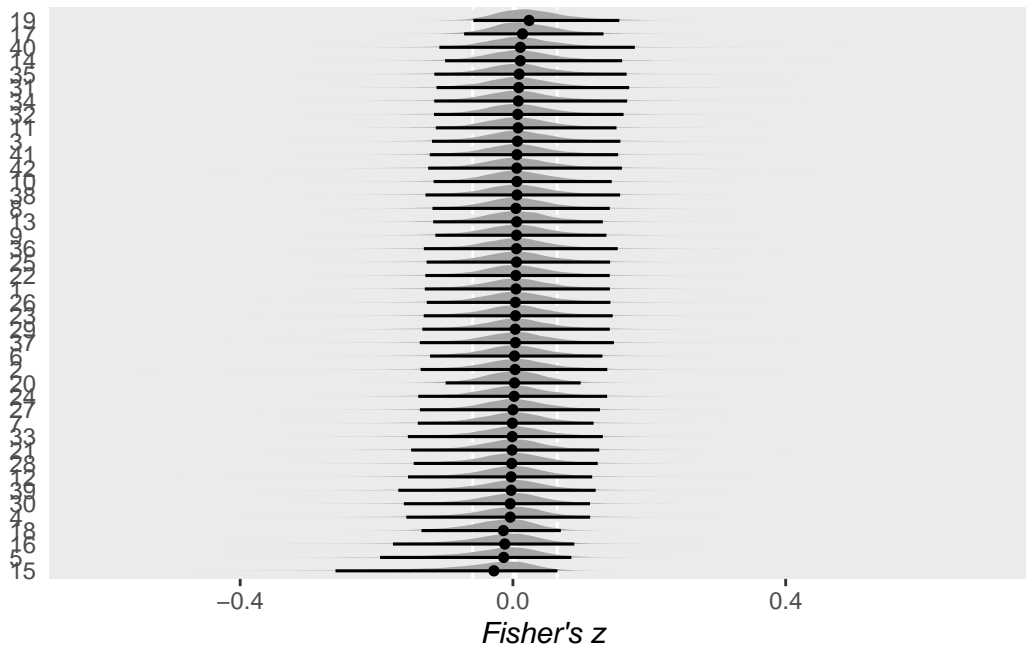
```

mutate(id = str_replace_all(id, "[.]", " ")) %>%

ggplot(aes(x = b_Intercept, y = reorder(id, b_Intercept))) +
  geom_vline(xintercept = fixef(meta_prob)[1, 1], color = "white", size = 1) +
  geom_vline(xintercept = fixef(meta_prob)[1, 3:4], color = "white", linetype = 2) +
  stat_halfeye(.width = .95, size = 2/3) +
  labs(x = expression(italic("Fisher's z")),
       y = NULL) +
  theme(panel.grid    = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y   = element_text(hjust = 0))

```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
 i Please use `linewidth` instead.



```
get_variables(meta_prob2)
```

```

[1] "b_Intercept"                "sd_studyid__Intercept"
[3] "sd_studyid:esid__Intercept" "sigma"
[5] "r_studyid[1,Intercept]"    "r_studyid[2,Intercept]"
[7] "r_studyid[3,Intercept]"    "r_studyid[4,Intercept]"

```

[9] "r_studyid[5,Intercept]"	"r_studyid[6,Intercept]"
[11] "r_studyid[7,Intercept]"	"r_studyid[8,Intercept]"
[13] "r_studyid[9,Intercept]"	"r_studyid:esid[1_1,Intercept]"
[15] "r_studyid:esid[1_2,Intercept]"	"r_studyid:esid[1_3,Intercept]"
[17] "r_studyid:esid[1_4,Intercept]"	"r_studyid:esid[2_5,Intercept]"
[19] "r_studyid:esid[2_6,Intercept]"	"r_studyid:esid[3_10,Intercept]"
[21] "r_studyid:esid[3_11,Intercept]"	"r_studyid:esid[3_12,Intercept]"
[23] "r_studyid:esid[3_7,Intercept]"	"r_studyid:esid[3_8,Intercept]"
[25] "r_studyid:esid[3_9,Intercept]"	"r_studyid:esid[4_13,Intercept]"
[27] "r_studyid:esid[4_14,Intercept]"	"r_studyid:esid[4_15,Intercept]"
[29] "r_studyid:esid[4_16,Intercept]"	"r_studyid:esid[5_17,Intercept]"
[31] "r_studyid:esid[5_18,Intercept]"	"r_studyid:esid[5_19,Intercept]"
[33] "r_studyid:esid[5_20,Intercept]"	"r_studyid:esid[6_21,Intercept]"
[35] "r_studyid:esid[6_22,Intercept]"	"r_studyid:esid[6_23,Intercept]"
[37] "r_studyid:esid[6_24,Intercept]"	"r_studyid:esid[6_25,Intercept]"
[39] "r_studyid:esid[6_26,Intercept]"	"r_studyid:esid[7_27,Intercept]"
[41] "r_studyid:esid[7_28,Intercept]"	"r_studyid:esid[8_29,Intercept]"
[43] "r_studyid:esid[8_30,Intercept]"	"r_studyid:esid[9_31,Intercept]"
[45] "r_studyid:esid[9_32,Intercept]"	"r_studyid:esid[9_33,Intercept]"
[47] "r_studyid:esid[9_34,Intercept]"	"r_studyid:esid[9_35,Intercept]"
[49] "r_studyid:esid[9_36,Intercept]"	"r_studyid:esid[9_37,Intercept]"
[51] "r_studyid:esid[9_38,Intercept]"	"r_studyid:esid[9_39,Intercept]"
[53] "r_studyid:esid[9_40,Intercept]"	"r_studyid:esid[9_41,Intercept]"
[55] "r_studyid:esid[9_42,Intercept]"	"prior_b_Intercept"
[57] "prior_sd_studyid"	"prior_sd_studyid:esid"
[59] "lprior"	"lp__"
[61] "z_1[1,1]"	"z_1[1,2]"
[63] "z_1[1,3]"	"z_1[1,4]"
[65] "z_1[1,5]"	"z_1[1,6]"
[67] "z_1[1,7]"	"z_1[1,8]"
[69] "z_1[1,9]"	"z_2[1,1]"
[71] "z_2[1,2]"	"z_2[1,3]"
[73] "z_2[1,4]"	"z_2[1,5]"
[75] "z_2[1,6]"	"z_2[1,7]"
[77] "z_2[1,8]"	"z_2[1,9]"
[79] "z_2[1,10]"	"z_2[1,11]"
[81] "z_2[1,12]"	"z_2[1,13]"
[83] "z_2[1,14]"	"z_2[1,15]"
[85] "z_2[1,16]"	"z_2[1,17]"
[87] "z_2[1,18]"	"z_2[1,19]"
[89] "z_2[1,20]"	"z_2[1,21]"
[91] "z_2[1,22]"	"z_2[1,23]"
[93] "z_2[1,24]"	"z_2[1,25]"

```

[95] "z_2[1,26]" "z_2[1,27]"
[97] "z_2[1,28]" "z_2[1,29]"
[99] "z_2[1,30]" "z_2[1,31]"
[101] "z_2[1,32]" "z_2[1,33]"
[103] "z_2[1,34]" "z_2[1,35]"
[105] "z_2[1,36]" "z_2[1,37]"
[107] "z_2[1,38]" "z_2[1,39]"
[109] "z_2[1,40]" "z_2[1,41]"
[111] "z_2[1,42]" "accept_stat_"
[113] "stepsize_" "treedepth_"
[115] "n_leapfrog_" "divergent_"
[117] "energy_"

```

```
fixef(meta_prob2)
```

```

      Estimate Est.Error    Q2.5   Q97.5
Intercept -0.00408    0.04613 -0.09611 0.08326

```

```
#extract data from the intercepts-only model:
```

```

study.draws <- spread_rvars(meta_prob2, `r_studyid`[`esid`, ], b_Intercept) %>%
  mutate(b_Intercept = `r_studyid` + b_Intercept) %>%
  mutate(esid = 1:9) %>%
  mutate(esid = as.factor(esid)) %>%
  select(esid, b_Intercept)

```

```

pooled.effect.draws <- spread_rvars(meta_prob2, b_Intercept) %>%
  mutate(esid = "Pooled Effect") %>%
  mutate(esid = as.factor(esid)) %>%
  select(esid, b_Intercept)

```

```

forest.data <- rbind(study.draws, pooled.effect.draws) %>%
  ungroup() %>%
  mutate(esid = reorder(esid, b_Intercept))

```

```

forest.data.summary <- group_by(forest.data, esid) %>%
  mean_qi(b_Intercept)

```

```
#plot the results:
```

```

forest_plot_F0 <- ggplot(aes(dist = b_Intercept, y = releval(esid, "Pooled Effect", after
  data = forest.data) +

```

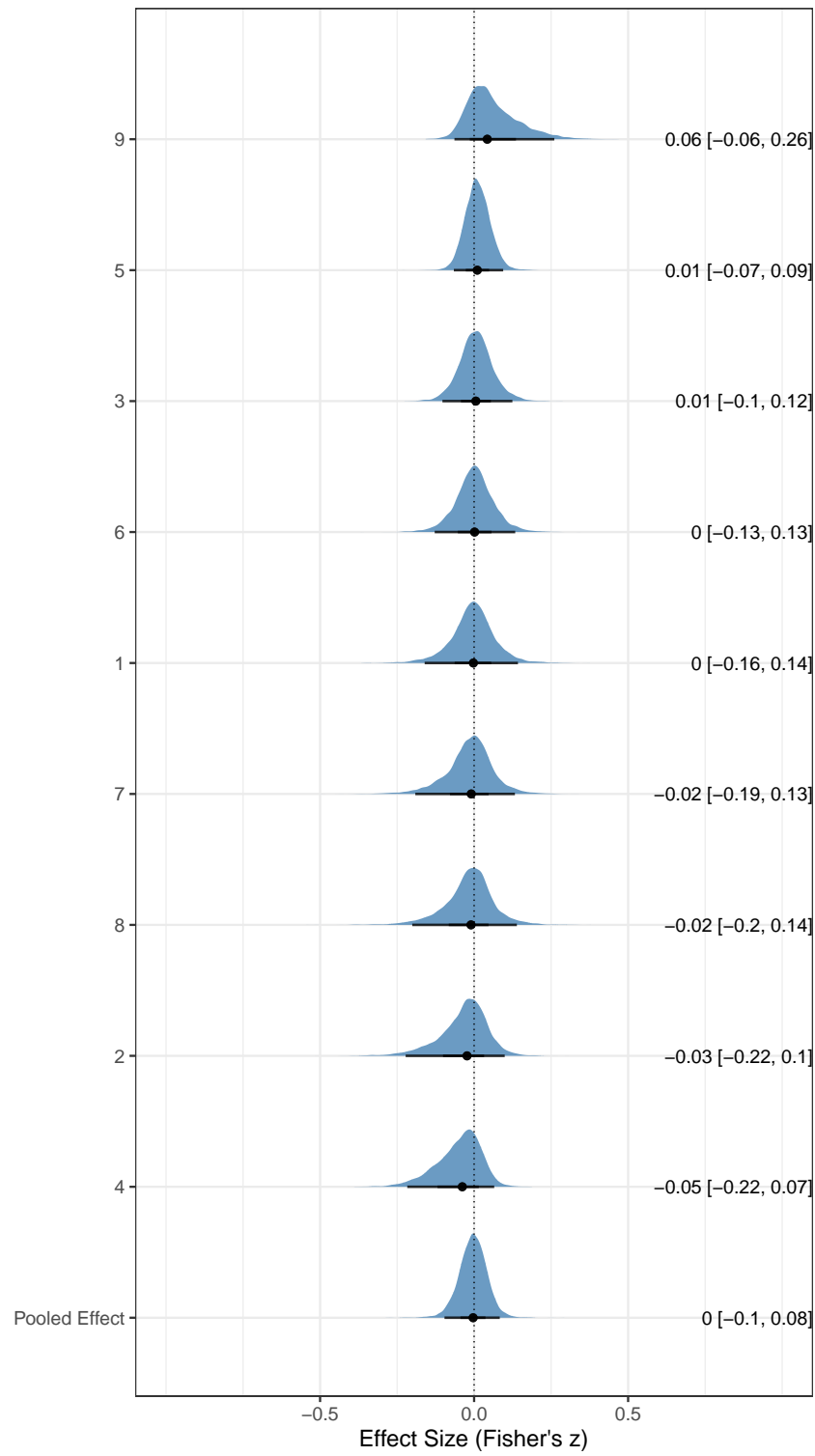


```

stat_dist_halfeye(fill = "steelblue", scale = 0.7, alpha = 0.8, size = 0.8, show_point =
geom_vline(xintercept = 0.0, color = "black", size = 0.3, linetype="dotted") +
geom_text(data = mutate_if(forest.data.summary, is.numeric, round, 2), aes(label = glue(
labs(x = "placeholder", y = element_blank()) +
scale_x_continuous("Effect Size (Fisher's z)", limits = c(-1, 1), breaks = c(-0.5, 0, 0.
ggtitle(expression(paste("Forest Plot for Studies on ", italic(f)[o])))) +
theme_bw()
forest_plot_F0 <- forest_plot_F0 + theme(plot.title = element_text(hjust = 0.5, size=15))
forest_plot_F0

```

Forest Plot for Studies on f_0



```

meta_prob3 <-
  brm(
    eszcor|se(se) ~ 0 + Intercept + (1|studyid/esid),
    data = test_data,
    family = gaussian(),
    sample_prior = T,
    iter = iter,
    prior = priors,
    save_pars = save_pars(all = TRUE),
    control = list(adapt_delta = 0.99, max_treedepth = 15),
    seed = seed,
    chains = chains
  )

```

Compiling Stan program...

recompiling to avoid crashing R session

Trying to compile a simple C file

```

Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
namespace Eigen {
~
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
namespace Eigen {
~
;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/S
In file included from /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/library/RcppEigen/include/Eigen,
#include <complex>
~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

```

Start sampling

SAMPLING FOR MODEL '8630b7bdb90dfccf834b55a121fecbd8' NOW (CHAIN 1).

Chain 1:

Chain 1: Gradient evaluation took 8.9e-05 seconds

Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.89 seconds.

Chain 1: Adjust your expectations accordingly!

Chain 1:

Chain 1:

Chain 1: Iteration: 1 / 5000 [0%] (Warmup)

Chain 1: Iteration: 500 / 5000 [10%] (Warmup)

Chain 1: Iteration: 1000 / 5000 [20%] (Warmup)

Chain 1: Iteration: 1500 / 5000 [30%] (Warmup)

Chain 1: Iteration: 2000 / 5000 [40%] (Warmup)

Chain 1: Iteration: 2500 / 5000 [50%] (Warmup)

Chain 1: Iteration: 2501 / 5000 [50%] (Sampling)

Chain 1: Iteration: 3000 / 5000 [60%] (Sampling)

Chain 1: Iteration: 3500 / 5000 [70%] (Sampling)

Chain 1: Iteration: 4000 / 5000 [80%] (Sampling)

Chain 1: Iteration: 4500 / 5000 [90%] (Sampling)

Chain 1: Iteration: 5000 / 5000 [100%] (Sampling)

Chain 1:

Chain 1: Elapsed Time: 1.43314 seconds (Warm-up)

Chain 1: 1.59289 seconds (Sampling)

Chain 1: 3.02603 seconds (Total)

Chain 1:

SAMPLING FOR MODEL '8630b7bdb90dfccf834b55a121fecbd8' NOW (CHAIN 2).

Chain 2:

Chain 2: Gradient evaluation took 1.5e-05 seconds

Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.15 seconds.

Chain 2: Adjust your expectations accordingly!

Chain 2:

Chain 2:

Chain 2: Iteration: 1 / 5000 [0%] (Warmup)

Chain 2: Iteration: 500 / 5000 [10%] (Warmup)

Chain 2: Iteration: 1000 / 5000 [20%] (Warmup)

Chain 2: Iteration: 1500 / 5000 [30%] (Warmup)

Chain 2: Iteration: 2000 / 5000 [40%] (Warmup)

Chain 2: Iteration: 2500 / 5000 [50%] (Warmup)

Chain 2: Iteration: 2501 / 5000 [50%] (Sampling)

Chain 2: Iteration: 3000 / 5000 [60%] (Sampling)
Chain 2: Iteration: 3500 / 5000 [70%] (Sampling)
Chain 2: Iteration: 4000 / 5000 [80%] (Sampling)
Chain 2: Iteration: 4500 / 5000 [90%] (Sampling)
Chain 2: Iteration: 5000 / 5000 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 1.40761 seconds (Warm-up)
Chain 2: 1.37556 seconds (Sampling)
Chain 2: 2.78317 seconds (Total)
Chain 2:

SAMPLING FOR MODEL '8630b7bdb90dfccf834b55a121fecbd8' NOW (CHAIN 3).

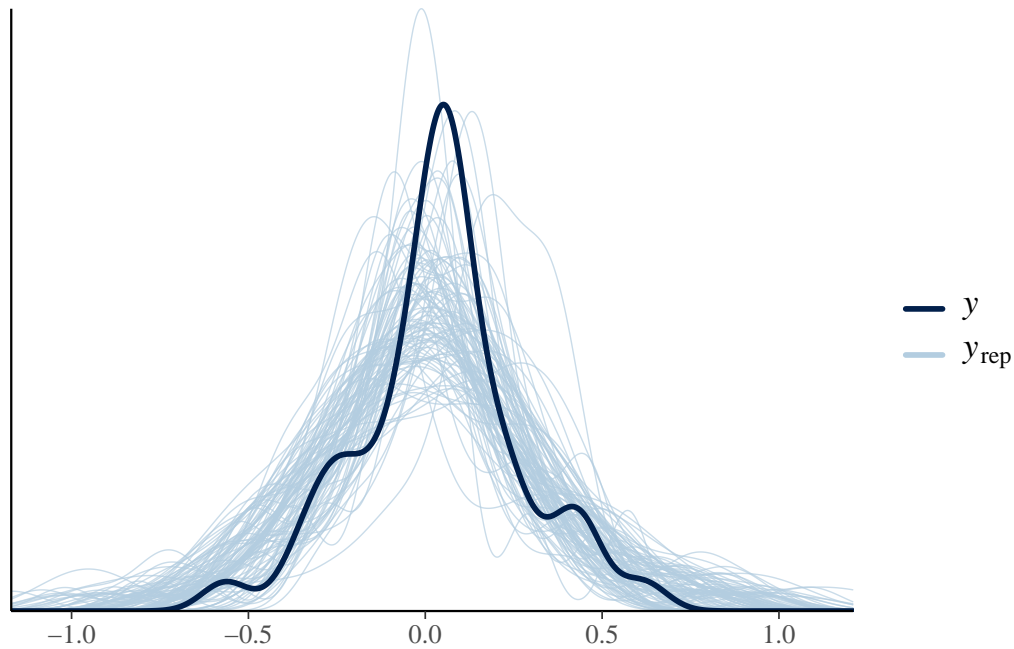
Chain 3:
Chain 3: Gradient evaluation took 1.9e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.19 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration: 1 / 5000 [0%] (Warmup)
Chain 3: Iteration: 500 / 5000 [10%] (Warmup)
Chain 3: Iteration: 1000 / 5000 [20%] (Warmup)
Chain 3: Iteration: 1500 / 5000 [30%] (Warmup)
Chain 3: Iteration: 2000 / 5000 [40%] (Warmup)
Chain 3: Iteration: 2500 / 5000 [50%] (Warmup)
Chain 3: Iteration: 2501 / 5000 [50%] (Sampling)
Chain 3: Iteration: 3000 / 5000 [60%] (Sampling)
Chain 3: Iteration: 3500 / 5000 [70%] (Sampling)
Chain 3: Iteration: 4000 / 5000 [80%] (Sampling)
Chain 3: Iteration: 4500 / 5000 [90%] (Sampling)
Chain 3: Iteration: 5000 / 5000 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 1.60766 seconds (Warm-up)
Chain 3: 1.61346 seconds (Sampling)
Chain 3: 3.22111 seconds (Total)
Chain 3:

SAMPLING FOR MODEL '8630b7bdb90dfccf834b55a121fecbd8' NOW (CHAIN 4).

Chain 4:
Chain 4: Gradient evaluation took 1.6e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.16 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:

```
Chain 4: Iteration:    1 / 5000 [ 0%] (Warmup)
Chain 4: Iteration:   500 / 5000 [10%] (Warmup)
Chain 4: Iteration:  1000 / 5000 [20%] (Warmup)
Chain 4: Iteration:  1500 / 5000 [30%] (Warmup)
Chain 4: Iteration:  2000 / 5000 [40%] (Warmup)
Chain 4: Iteration:  2500 / 5000 [50%] (Warmup)
Chain 4: Iteration:  2501 / 5000 [50%] (Sampling)
Chain 4: Iteration:  3000 / 5000 [60%] (Sampling)
Chain 4: Iteration:  3500 / 5000 [70%] (Sampling)
Chain 4: Iteration:  4000 / 5000 [80%] (Sampling)
Chain 4: Iteration:  4500 / 5000 [90%] (Sampling)
Chain 4: Iteration:  5000 / 5000 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 1.36081 seconds (Warm-up)
Chain 4:                1.61777 seconds (Sampling)
Chain 4:                2.97858 seconds (Total)
Chain 4:
```

```
pp_check(meta_prob3, ndraws = 100)
```



```
summary(meta_prob3)
```

```

Family: gaussian
Links: mu = identity; sigma = identity
Formula: eszcor | se(se) ~ 0 + Intercept + (1 | studyid/esid)
Data: test_data (Number of observations: 42)
Draws: 4 chains, each with iter = 5000; warmup = 2500; thin = 1;
       total post-warmup draws = 10000

Group-Level Effects:
~studyid (Number of levels: 9)
      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)    0.07     0.05    0.00    0.20 1.00     3605     5324

~studyid:esid (Number of levels: 42)
      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)    0.05     0.04    0.00    0.13 1.00     4275     4203

Population-Level Effects:
      Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept   -0.00     0.05   -0.10    0.08 1.00     6919     6222

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 sampling(NUTS). 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).

```

#extract data from the intercepts-only model:

study.draws1 <- spread_rvars(meta_prob3, `r_studyid`[`esid1`, ], `r_studyid:esid`[`esid2`,

study.draws2 <- study.draws1 %>%
  mutate(esid3 = as.integer(sub("_.*", "", esid2))) %>%
  filter(esid3 == esid1)

study.draws <- study.draws2 %>%
  mutate(b_Intercept = `r_studyid` + `r_studyid:esid` + b_Intercept) %>%
  mutate(esid = 1:42) %>%
  mutate(esid = as.factor(esid)) %>%
  select(esid, b_Intercept)

```

```

pooled.effect.draws <- spread_rvars(meta_prob3, b_Intercept) %>%
  mutate(esid = "Pooled Effect") %>%
  mutate(esid = as.factor(esid)) %>%
  select(esid, b_Intercept)

forest.data <- rbind(study.draws, pooled.effect.draws) %>%
  ungroup() %>%
  mutate(esid = reorder(esid, b_Intercept))

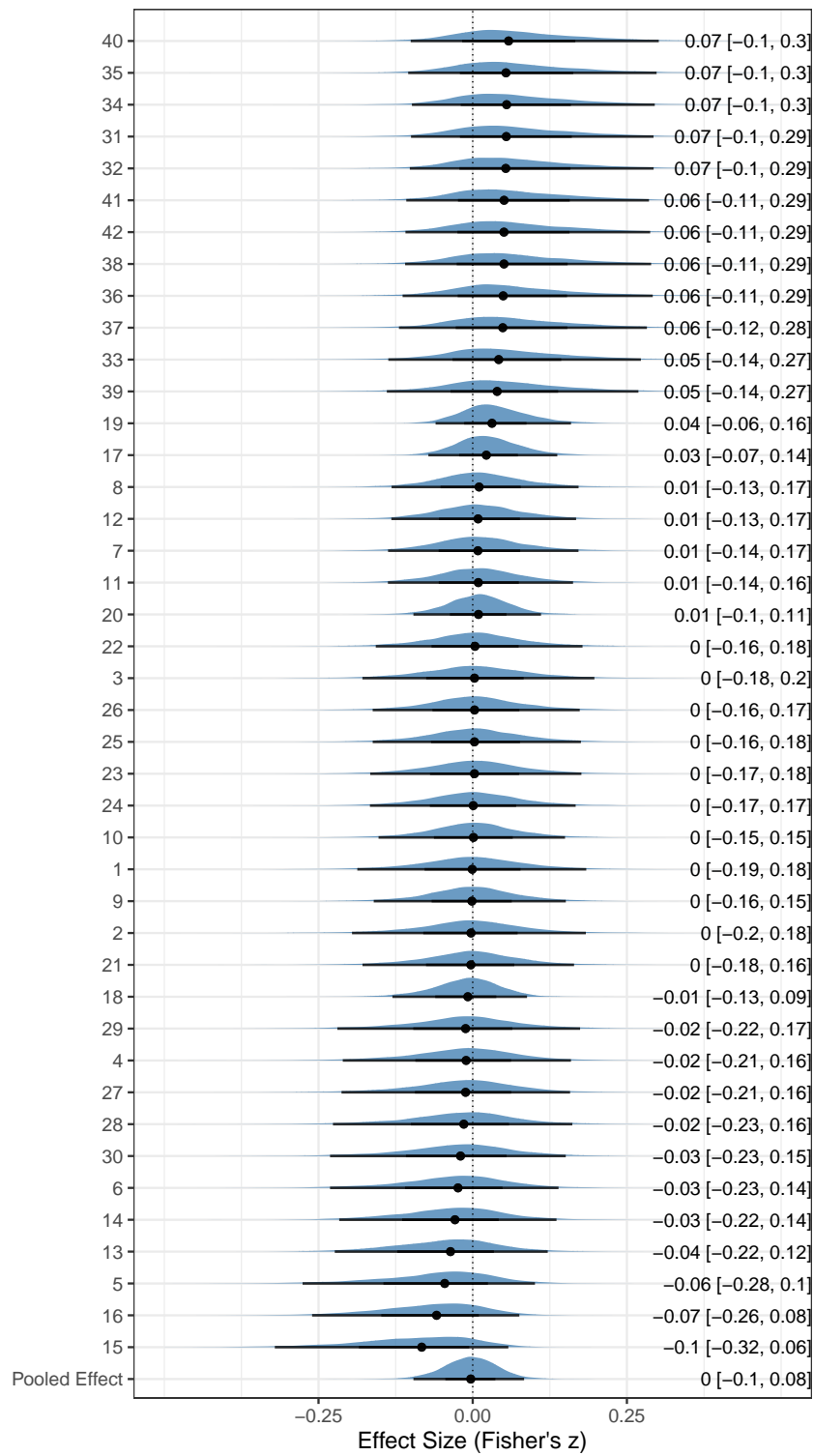
forest.data.summary <- group_by(forest.data, esid) %>%
  mean_qi(b_Intercept)

#plot the results:
forest_plot_F0 <- ggplot(aes(dist = b_Intercept, y = relevel(esid, "Pooled Effect", after
  data = forest.data) +
  stat_dist_halfeye(fill = "steelblue", scale = 0.7, alpha = 0.8, size = 0.8, show_point =
  geom_vline(xintercept = 0.0, color = "black", size = 0.3, linetype="dotted") +
  geom_text(data = mutate_if(forest.data.summary, is.numeric, round, 2), aes(label = glue(
  labs(x = "placeholder", y = element_blank()) +
  scale_x_continuous("Effect Size (Fisher's z)", limits = c(-0.5, 0.5), breaks = c(-0.25,
  ggtitle(expression(paste("Forest Plot for Coupling Percentage Test Data")))) +
  theme_bw()
forest_plot_F0 <- forest_plot_F0 + theme(plot.title = element_text(hjust = 0.5, size=15))
forest_plot_F0

```

Warning: Removed 863 rows containing missing values (`geom_slabinterval()`).

Forest Plot for Coupling Percentage Test Data



```

#extract data from the intercepts-only model:

study.draws1 <- spread_rvars(meta_prob3, `r_studyid`[`esid1`, ], `r_studyid:esid`[`esid2`,

study.draws2 <- study.draws1 %>%
  mutate(esid3 = as.integer(sub("_.*", "", esid2))) %>%
  filter(esid3 == esid1)

study.draws <- study.draws2 %>%
  mutate(b_Intercept = `r_studyid` + `r_studyid:esid` + b_Intercept) %>%
  mutate(esid = 1:42) %>%
  mutate(esid = as.factor(esid)) %>%
  select(esid, b_Intercept)

pooled.effect.draws <- spread_rvars(meta_prob3, b_Intercept) %>%
  mutate(esid = "Pooled Effect") %>%
  mutate(esid = as.factor(esid)) %>%
  select(esid, b_Intercept)

forest.data <- rbind(study.draws, pooled.effect.draws) %>%
  ungroup() %>%
  mutate(esid = reorder(esid, b_Intercept))

forest.data.summary <- group_by(forest.data, esid) %>%
  mean_qi(b_Intercept)

#plot the results:
forest_plot_F0 <- ggplot(aes(dist = b_Intercept, y = relevel(esid, "Pooled Effect", after
                        data = forest.data) +
  stat_dist_halfeye(fill = "steelblue", scale = 0.7, alpha = 0.8, size = 0.8, show_point =
  geom_vline(xintercept = 0.0, color = "black", size = 0.3, linetype="dotted") +
  geom_text(data = mutate_if(forest.data.summary, is.numeric, round, 2), aes(label = glue(
  labs(x = "placeholder", y = element_blank()) +
  scale_x_continuous("Effect Size (Fisher's z)", limits = c(-1, 1), breaks = c(-0.5, 0, 0.
  ggtitle(expression(paste("Forest Plot for Coupling Percentage Test Data")))) +
  theme_bw()
forest_plot_F0 <- forest_plot_F0 + theme(plot.title = element_text(hjust = 0.5, size=15))
forest_plot_F0

```

Forest Plot for Coupling Percentage Test Data

