Statistical Patterns of Sense of School-Belonging

Mariana Nold

```
source("RFiles/packages.r")
```

Load R packages

```
PPath <- "C:/Users/zo95yup/Documents/GitHub/Bayes_for_STRATOS/Task_Truancy"
load(file = file.path(PPath,"Orgdata/IMPDAT_PISA.RData"))
```

Load single imputed PISA data.

```
source(file.path(PPath,"RFiles/01data_preperation.r"))
to.rm <- objects()[!objects() %in% c("PPath","pisa18")]
rm(list = to.rm)
# head(pisa18)
# dim(pisa18)
```

Prepare data, thus build PISA scales

```
pisa18$school.id <- as.factor(pisa18$school.id)
help <- split(pisa18,pisa18$school.id)
nst <- lapply(help, function(x) length(x[[1]]))
tf <- lapply(nst, function(x) x[[1]] > 9)
utf <- unlist(tf)
large_school <- help[utf]
pisa2018 <- list.rbind(large_school)
pisa2018$school.id <- droplevels(pisa2018$school.id)
#table(pisa2018$school.id)

rm(pisa18,tf,nst,help,large_school,utf)

save(pisa2018,file = file.path(PPath,"Files/pisa2018.RData")) #file = file.path(PPath,"File/fits0.RData
#load(file.path(PPath,"Files/pisa2018.RData"))
```

There should be at least ten students per school. Schools with fewer observations are excluded from the analysis.

```
# Compute quartiles of being bullied at school level
pisa2018$ATT4 <- cut_number(pisa2018$ATT01,4)
# model with multilevel structure
model1 <- as.formula(belong ~ female + nld + scie_std + aca  + val  + comp + ndiff + nfof  + native + n
```

Simple random intercept model as starting point: Iteration 1 (Step 1 to 4 in SAP)

```r
a.seed <- 12345
a.iter <-  2000
a.chains <- 4
warmup <- 1000


fit1 <- stan_glmer(model1, data = pisa2018, seed = a.seed, iter = a.iter, chains = a.chains, warmup = wa

save(fit1, file = file.path(PPath,"Files/fit1.RData"))
```

Fit model of iteration 1

Check the convergence criteria (Step 5 in SAP)

post: which diagnostics to use: mcse Rhat < 1.1 n_eff > 1000 and mean_PPD use launch_shinystan

```r
mean(pisa2018$belong)
```

post: how to summarize the prior -> large sample size, check if prior_summary is reasonable

```
## [1] 3.206529
```

```r
sd(pisa2018$belong)
```

```
## [1] 0.654916
```

```r
prior_summary(fit1)
```

```
## Priors for model 'fit1'
## ------
## Intercept (after predictors centered)
##   Specified prior:
##     ~ normal(location = 3.2, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 3.2, scale = 1.6)
##
## Coefficients
##   Specified prior:
##     ~ normal(location = [0,0,0,...], scale = [2.5,2.5,2.5,...])
##   Adjusted prior:
##     ~ normal(location = [0,0,0,...], scale = [3.27,4.02,1.66,...])
##
## Auxiliary (sigma)
##   Specified prior:
##     ~ exponential(rate = 1)
##   Adjusted prior:
##     ~ exponential(rate = 1.5)
##
## Covariance
##  ~ decov(reg. = 1, conc. = 1, shape = 1, scale = 1)
## ------
```

```
## See help('prior_summary.stanreg') for more details
```

```r
#summary(fit1) # mcse Rhat n_eff
# soo_fit1 <- launch_shinystan(fit1)
# save(soo_fit1, file = file.path(PPath,"Files/soo_fit1.RData"))
# launch_shinystan(soo_fit1) # to open it
```

```r
summaryTwoLevel <- tidy(fit1,   conf.int =TRUE, conf.level=.95,
effects = "fixed")
print(summaryTwoLevel, digits = 2, n = 28)
```

Results of fit 1

```
## # A tibble: 28 x 5
##    term                    estimate std.error  conf.low conf.high
##    <chr>                      <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)              3.14      0.0365   3.07        3.21
##  2 female                   0.0357    0.0170   0.00182     0.0671
##  3 nld                      0.0666    0.0259   0.0182      0.118
##  4 scie_std                 0.00478   0.0126  -0.0194      0.0295
##  5 aca                     -0.00700   0.0174  -0.0397      0.0261
##  6 val                      0.0403    0.00814  0.0242      0.0564
##  7 comp                     0.0185    0.00881  0.000865    0.0357
##  8 ndiff                    0.0437    0.00883  0.0275      0.0602
##  9 nfof                     0.0504    0.00755  0.0351      0.0653
## 10 native                  -0.0111    0.0219  -0.0519      0.0315
## 11 nfewbooks                0.0119    0.0200  -0.0267      0.0503
## 12 joyread                 -0.0622    0.00845 -0.0793     -0.0459
## 13 goal                     0.00273   0.00900 -0.0143      0.0200
## 14 mot                      0.0102    0.00846 -0.00655     0.0268
## 15 res                      0.0650    0.00880  0.0486      0.0820
## 16 swbp                     0.108     0.00835  0.0915      0.124
## 17 mean                     0.0297    0.00881  0.0122      0.0475
## 18 parent_sup               0.0276    0.00794  0.0117      0.0433
## 19 ndis_clim               -0.0208    0.00771 -0.0357     -0.00609
## 20 GYM                      0.0577    0.0243   0.00907     0.105
## 21 UNI                      0.0371    0.0493  -0.0595      0.130
## 22 ATT4(0.114,0.176]       -0.0119    0.0220  -0.0575      0.0301
## 23 ATT4(0.176,0.25]        -0.0634    0.0228  -0.108      -0.0175
## 24 ATT4(0.25,0.6]          -0.0659    0.0242  -0.112      -0.0180
## 25 bull                    -0.436     0.0541  -0.546      -0.331
## 26 ATT4(0.114,0.176]:bull   0.110     0.0744  -0.0334      0.259
## 27 ATT4(0.176,0.25]:bull    0.129     0.0729  -0.0119      0.271
## 28 ATT4(0.25,0.6]:bull      0.178     0.0684   0.0440      0.316
```

```r
summaryTwoLevelModelSchools <- tidy(fit1,   conf.int =TRUE, conf.level=.95,
effects = "ran_vals")
print(summaryTwoLevelModelSchools, digits = 2)
```

```
## # A tibble: 221 x 7
##    level   group     term          estimate std.error conf.low conf.high
##    <chr>   <chr>     <chr>            <dbl>     <dbl>    <dbl>     <dbl>
##  1 4000001 school.id (Intercept)  0.000353     0.0115  -0.0366    0.0472
##  2 4000002 school.id (Intercept) -0.00102      0.0121  -0.0488    0.0365
##  3 4000003 school.id (Intercept)  0.000140     0.0117  -0.0388    0.0426
```

```
##  4 4000005 school.id (Intercept)  0.00130      0.0127  -0.0363    0.0556
##  5 4000006 school.id (Intercept)  0.000743     0.0114  -0.0354    0.0474
##  6 4000007 school.id (Intercept) -0.000702     0.0113  -0.0426    0.0374
##  7 4000008 school.id (Intercept) -0.000750     0.0119  -0.0493    0.0358
##  8 4000009 school.id (Intercept) -0.000356     0.0117  -0.0448    0.0378
##  9 4000010 school.id (Intercept)  0.000120     0.0120  -0.0455    0.0402
## 10 4000011 school.id (Intercept) -0.0000500    0.0124  -0.0466    0.0418
## # i 211 more rows
```

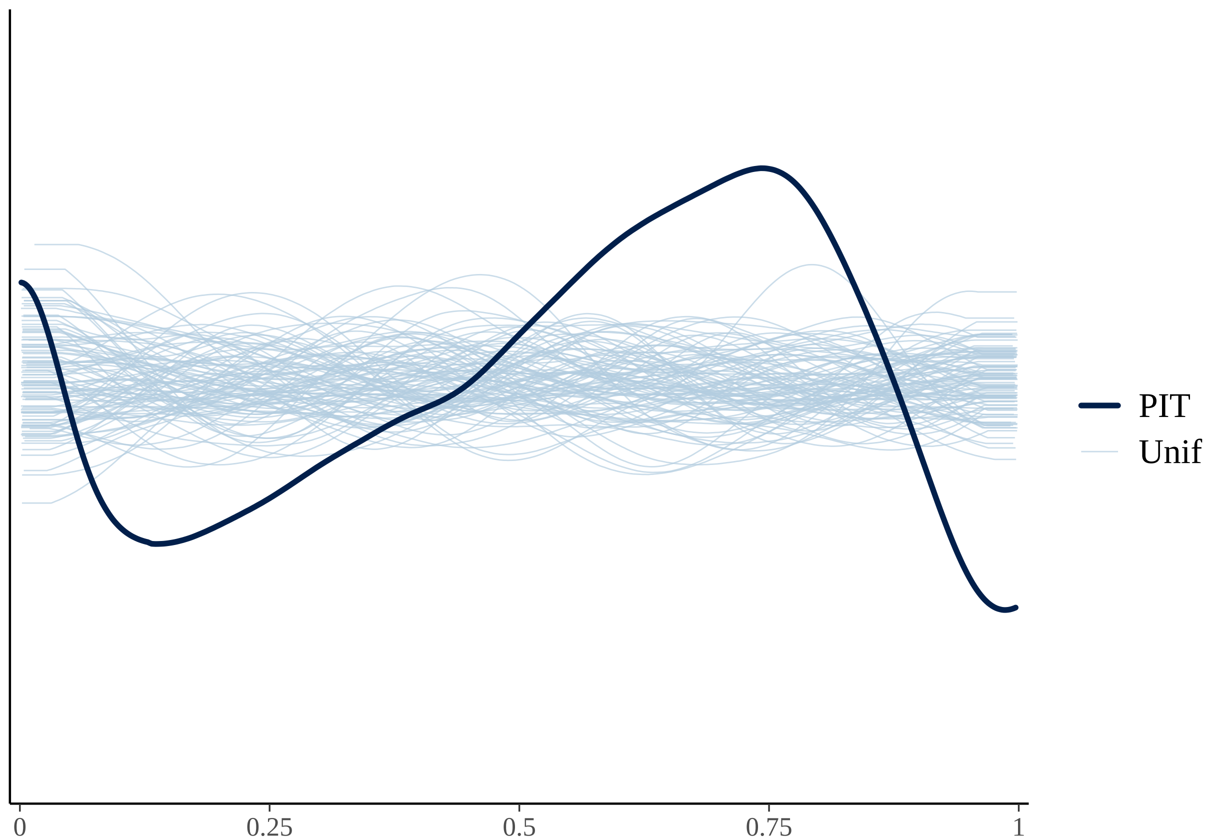
```
summaryTwoLevelModelVar <- tidy(fit1,   conf.int =TRUE, conf.level=.95,
effects = "ran_pars")
print(summaryTwoLevelModelVar, digits = 2)
```

```
## # A tibble: 2 x 3
##   term                       group     estimate
##   <chr>                      <chr>        <dbl>
## 1 sd_(Intercept).school.id school.id   0.0203
## 2 sd_Observation.Residual  Residual    0.589
```

Conclusion Step 5: No indications of convergence problems

Conclusion PPCs in launch_shinystan, in particular histograms and min and max of mean suggest to transform the outcome

Refinement of model building (Step 6 in SAP) fist look a PPCs in launch_shinystan, LOO-PIT

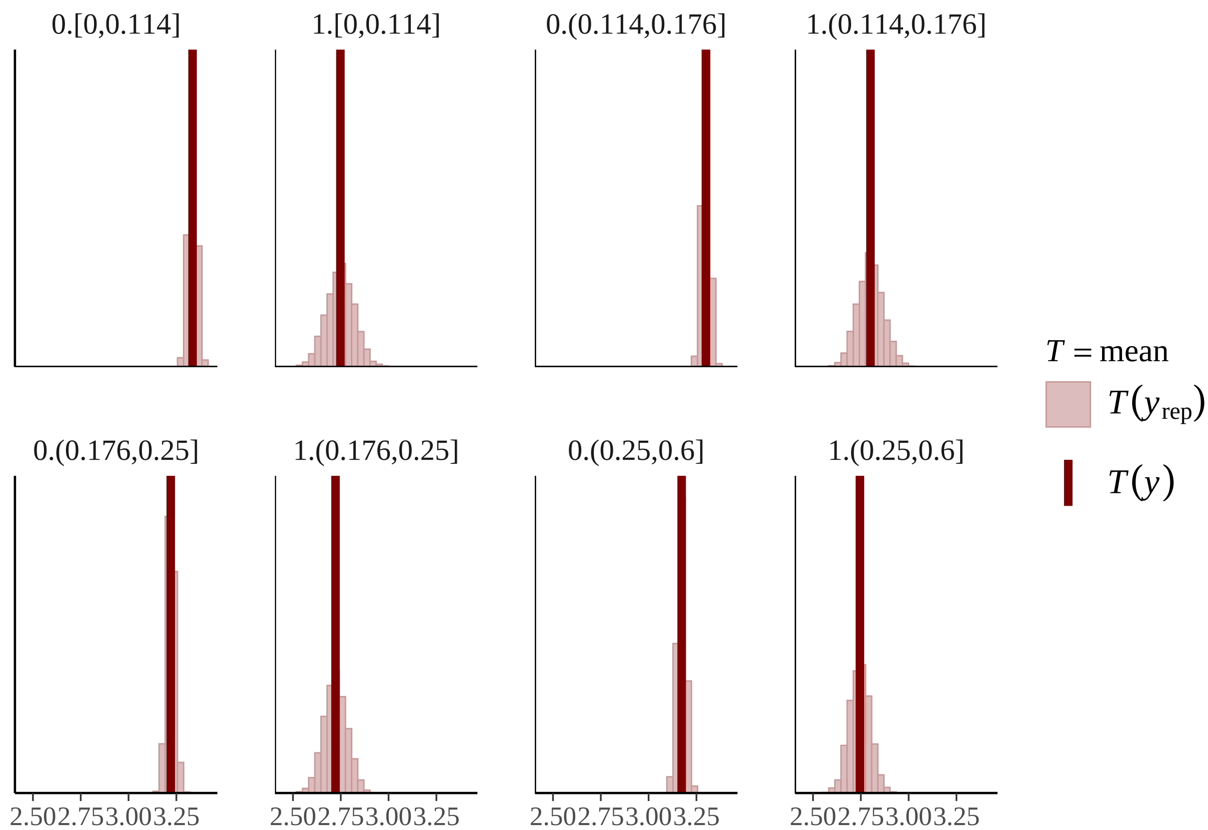


```r
source(file.path(PPath, "RFiles/02functions.r"))
```

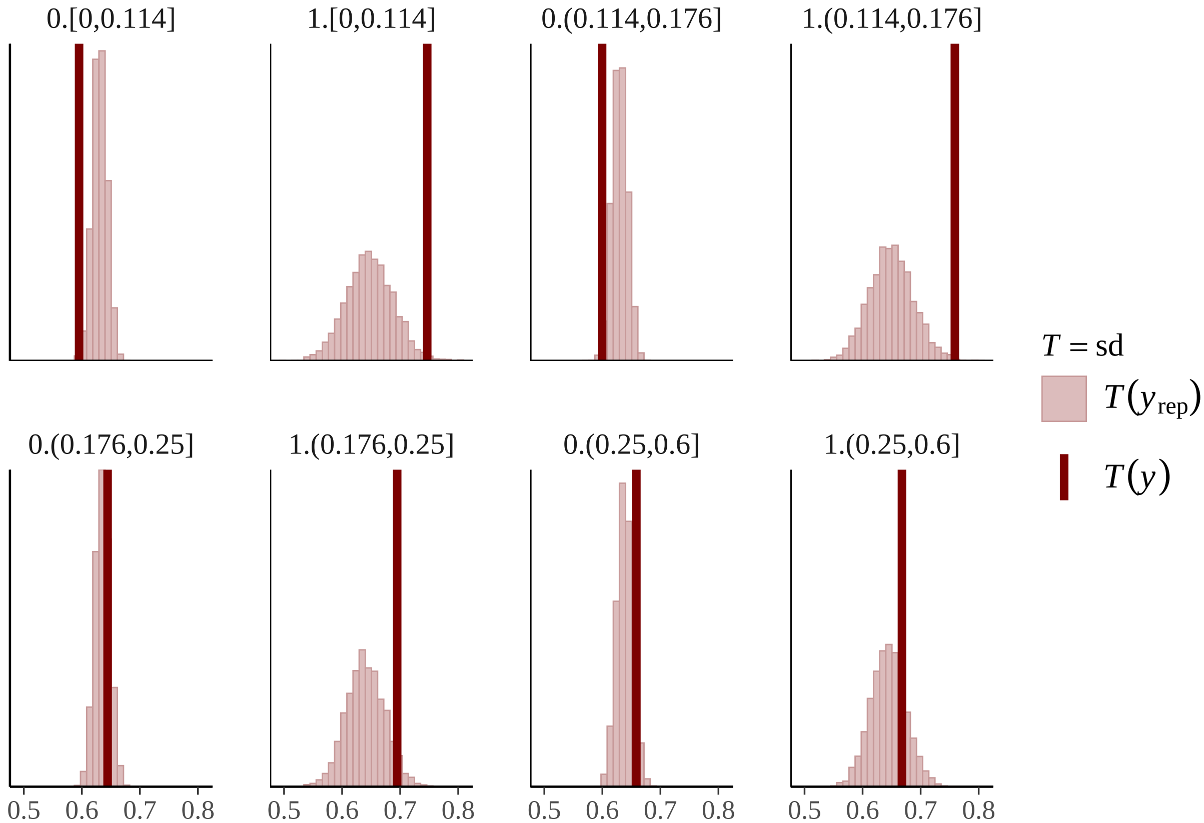
load functions to do more specific PPCs within levels of grouping variables

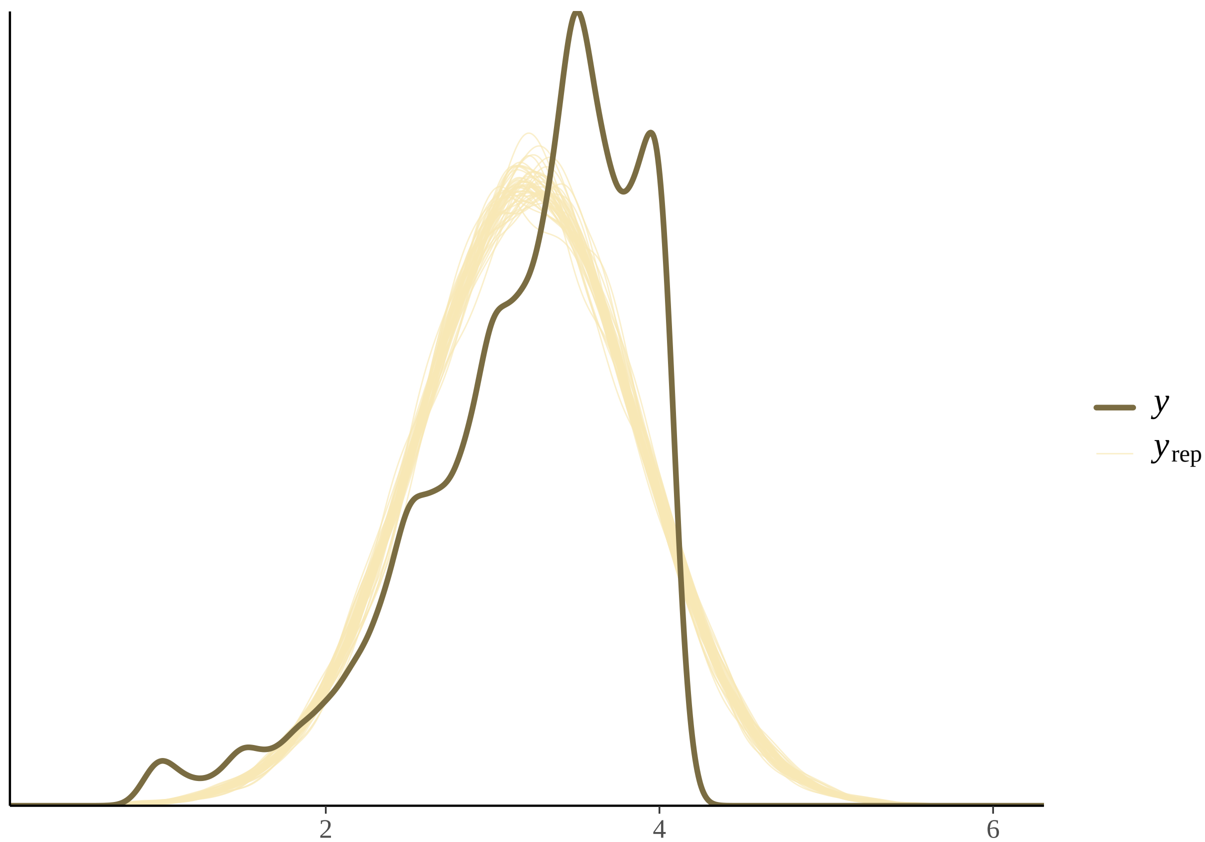
Posterior predictive checks for fit 1, within grouping levels of focal predictors

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

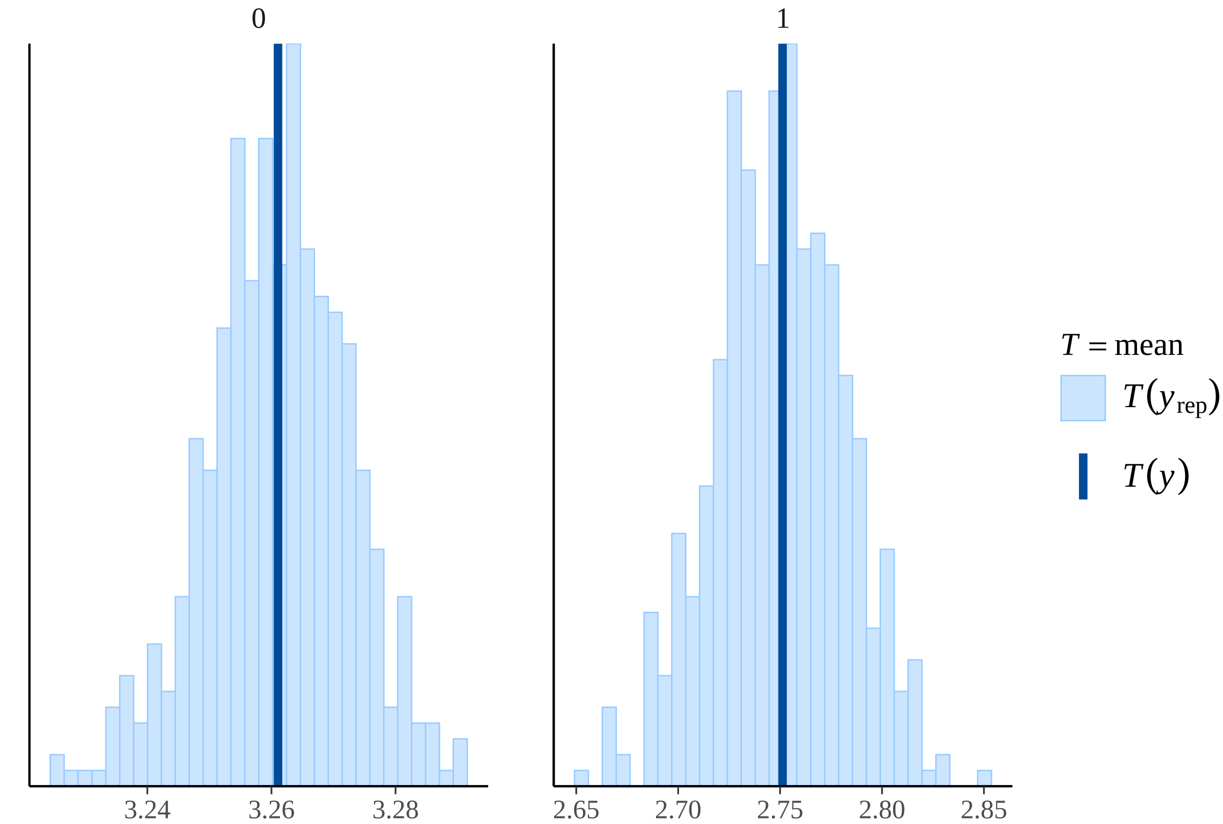



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

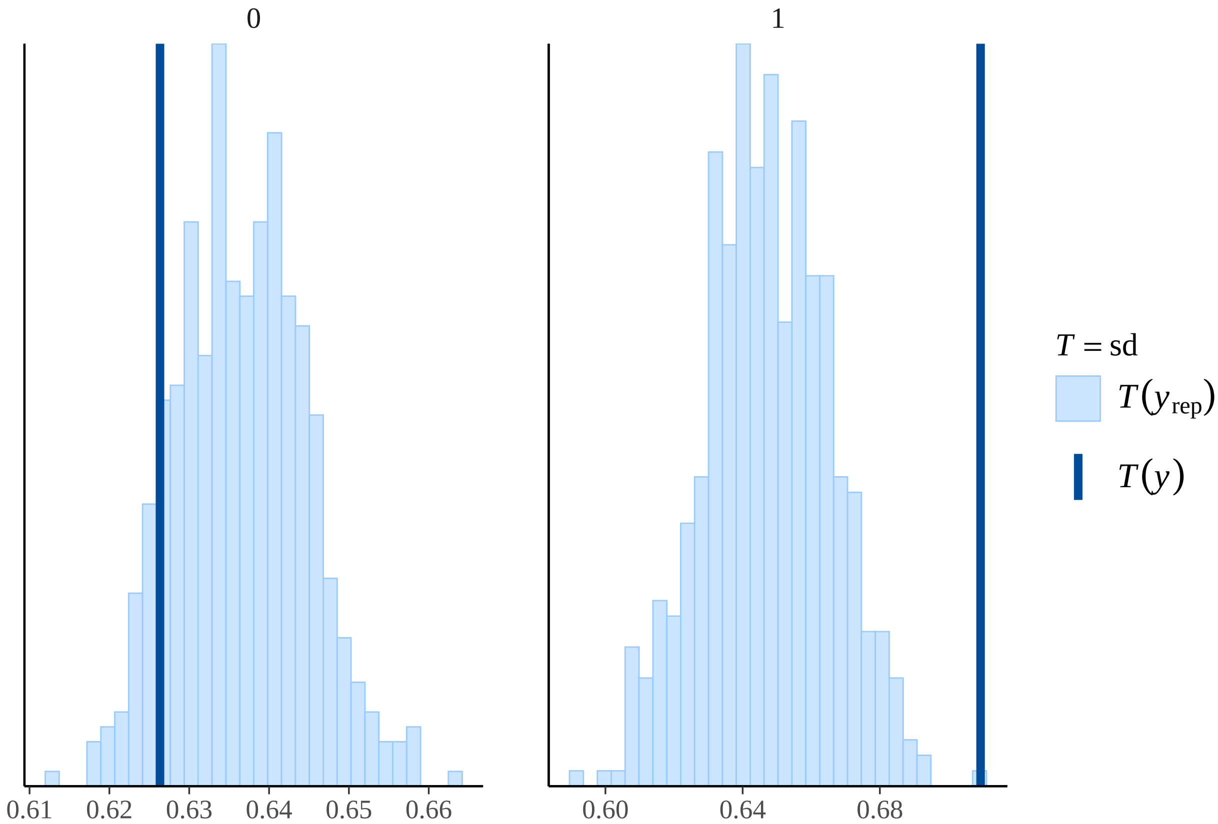

0.[0,0.114]
1.[0,0.114]
0.(0.114,0.176]
1.(0.114,0.176]
0.(0.176,0.25]
1.(0.176,0.25]
0.(0.25,0.6]
1.(0.25,0.6]
0.5
0.6
0.7
0.8
T = sd
T(y_rep)
T(y)



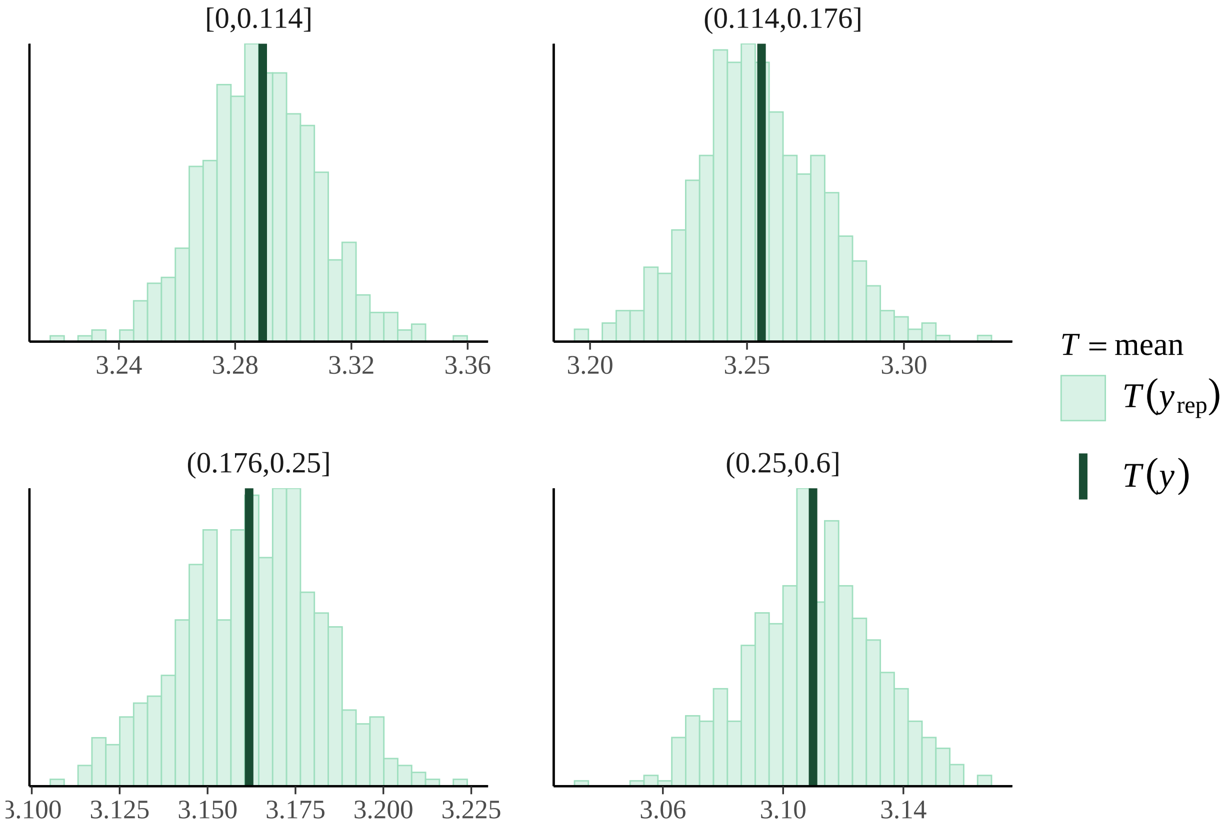

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



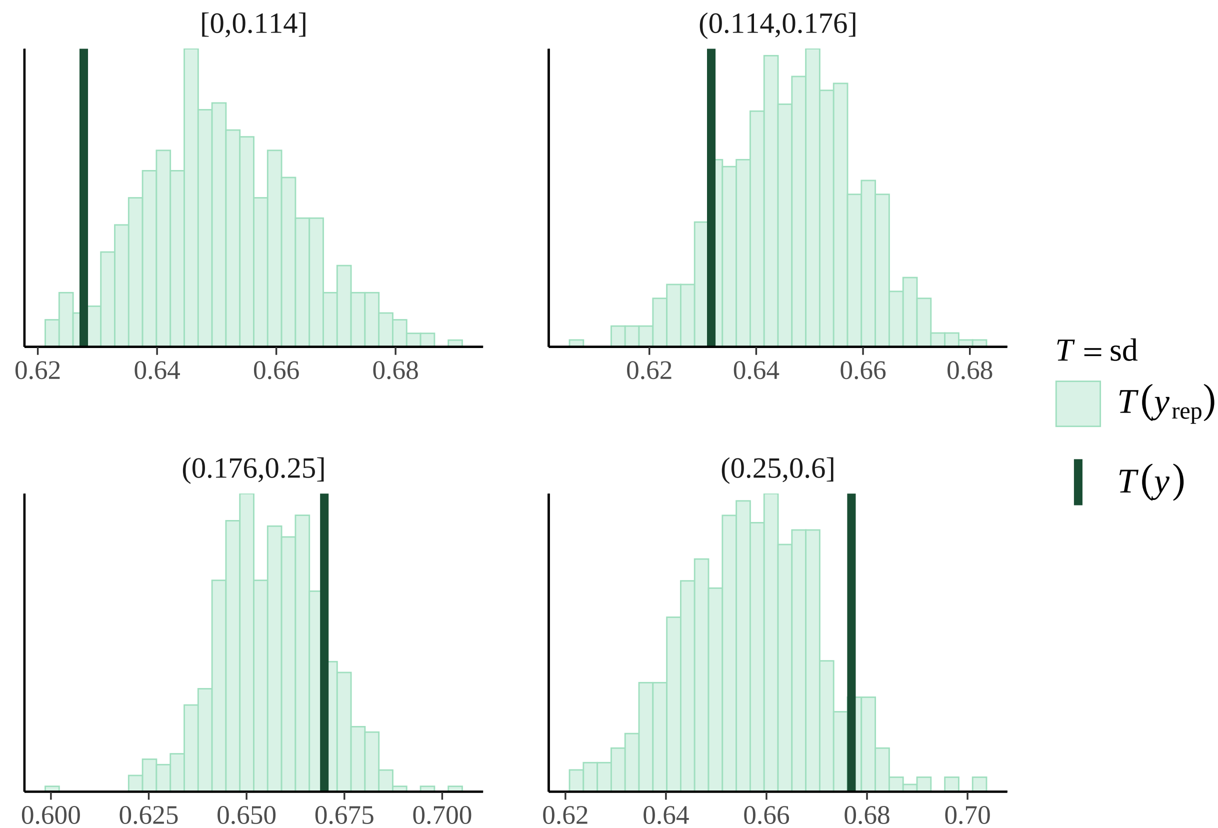

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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




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



Conclusion: LOOPIT indicates bad model fit of iteration 1 -> iteration 2: log-transform outcome

```
# better multiplicative model log(4)/2 = log(2)
# A large proportion of the values are between 3.25 and 4.
# log(5-pisa2018$belong): large prop. between 1 and 1.75
# log(5.25-pisa2018$belong): large prop. between 1.25 and 2

pisa2018$revBelong <- 5-pisa2018$belong
pisa2018$belongLogA <- log(pisa2018$revBelong)
model2A <- update(model1,belongLogA ~ . )
```

```
library("trafo")
```

Check skewness and kurtusis of diffrent log-transformations, post: Use r-package trafo

```
## Warning: Paket 'trafo' wurde unter R Version 4.4.2 erstellt
```

```
model2B <- update(model1,belong ~ . -(1 | school.id)) #intraclass very low,use only Skewness and Kurtos
lm_model2B <- lm(model2B, data = pisa2018)
linMod_trafoB <- trafo_lm(object = lm_model2B , trafo = "logshiftopt",method = "skew")
diagnostics(linMod_trafoB) # bad transformed model, keep 5 -
```

```
## Warning in diagnostics_internal(modOne = modOne, modTwo = modTwo): Number of domains exceeds 5000 or
##                 Shapiro-Wilk test is not applicable for residuals.
```

```
## Diagnostics: Untransformed vs transformed model
##
## Transformation:  logshiftopt
```

```
## Estimation method:  skew
## Optimal Parameter:  2.999958
##
## Residual diagnostics:
##
## Normality:
## Pearson residuals:
##                        Skewness Kurtosis Shapiro_W Shapiro_p
## Untransformed model -1.276765 5.737234        NA        NA
## Transformed model   -1.654021 7.292118        NA        NA
##
## Heteroscedasticity:
##                      BreuschPagan_V BreuschPagan_p
## Untransformed model        74.65283   2.375005e-06
## Transformed model          71.74398   6.303478e-06
```

```
model2C <- update(model1, revBelong ~ . -(1 | school.id))
lm_model2C <- lm(model2C, data = pisa2018)
linMod_trafoC <- trafo_lm(object = lm_model2C , trafo = "logshiftopt",method = "skew")
diagnostics(linMod_trafoC) # good transformed model
```

```
## Warning in diagnostics_internal(modOne = modOne, modTwo = modTwo): Number of domains exceeds 5000 or
##               Shapiro-Wilk test is not applicable for residuals.
```

```
## Diagnostics: Untransformed vs transformed model
##
## Transformation:  logshiftopt
## Estimation method:  skew
## Optimal Parameter:  4.24185e-05
##
## Residual diagnostics:
##
## Normality:
## Pearson residuals:
##                        Skewness Kurtosis Shapiro_W Shapiro_p
## Untransformed model 1.2767645 5.737234        NA        NA
## Transformed model   0.4567051 3.616862        NA        NA
##
## Heteroscedasticity:
##                      BreuschPagan_V BreuschPagan_p
## Untransformed model        74.65283   2.375005e-06
## Transformed model         100.01255   2.561798e-10
```

```
# try ml
linMod_trafoC <- trafo_lm(object = lm_model2C , trafo = "logshiftopt",method = "ml")
diagnostics(linMod_trafoC) # nearly the same as for method = "skew"
```

```
## Warning in diagnostics_internal(modOne = modOne, modTwo = modTwo): Number of domains exceeds 5000 or
##               Shapiro-Wilk test is not applicable for residuals.
```

```
## Diagnostics: Untransformed vs transformed model
##
## Transformation:  logshiftopt
## Estimation method:  ml
## Optimal Parameter:  4.24185e-05
##
```

```
## Residual diagnostics:
##
## Normality:
## Pearson residuals:
##                      Skewness Kurtosis Shapiro_W Shapiro_p
## Untransformed model 1.2767645 5.737234        NA        NA
## Transformed model   0.4567051 3.616862        NA        NA
##
## Heteroscedasticity:
##                     BreuschPagan_V BreuschPagan_p
## Untransformed model       74.65283   2.375005e-06
## Transformed model        100.01255   2.561798e-10
```

```
linMod_trafoC <- trafo_lm(object = lm_model2C , trafo = "logshiftopt",method = "kurt")
diagnostics(linMod_trafoC) # nearly the same as for method = "skew"
```

```
## Warning in diagnostics_internal(modOne = modOne, modTwo = modTwo): Number of domains exceeds 5000 or
##                 Shapiro-Wilk test is not applicable for residuals.
```

```
## Diagnostics: Untransformed vs transformed model
##
## Transformation:  logshiftopt
## Estimation method:  kurt
## Optimal Parameter:  4.24185e-05
##
## Residual diagnostics:
##
## Normality:
## Pearson residuals:
##                      Skewness Kurtosis Shapiro_W Shapiro_p
## Untransformed model 1.2767645 5.737234        NA        NA
## Transformed model   0.4567051 3.616862        NA        NA
##
## Heteroscedasticity:
##                     BreuschPagan_V BreuschPagan_p
## Untransformed model       74.65283   2.375005e-06
## Transformed model        100.01255   2.561798e-10
```

```
# Conclusion: Go on with model model2A
```

```
a.seed <- 12345
a.iter <-  2000
a.chains <- 4
warmup <- 1000

fit2A <- stan_glmer(model2A, data = pisa2018, seed = a.seed,
                      iter = a.iter, chains = a.chains, warmup =  warmup)

save(fit2A, file = file.path(PPath,"Files/fit2A.RData"))

# soo_fit2A <- launch_shinystan(fit2A)
# save(soo_fit2A, file = file.path(PPath,"Files/soo_fit2A.RData"))


 # not perfect, but okay
```

```r
# launch_shinystan(fit2A) # the predicted distribution underestimates the minimum and overestimates the
```

```r
mean(pisa2018$belongLogA)
```

Results of model 2A

```
## [1] 0.522728
```

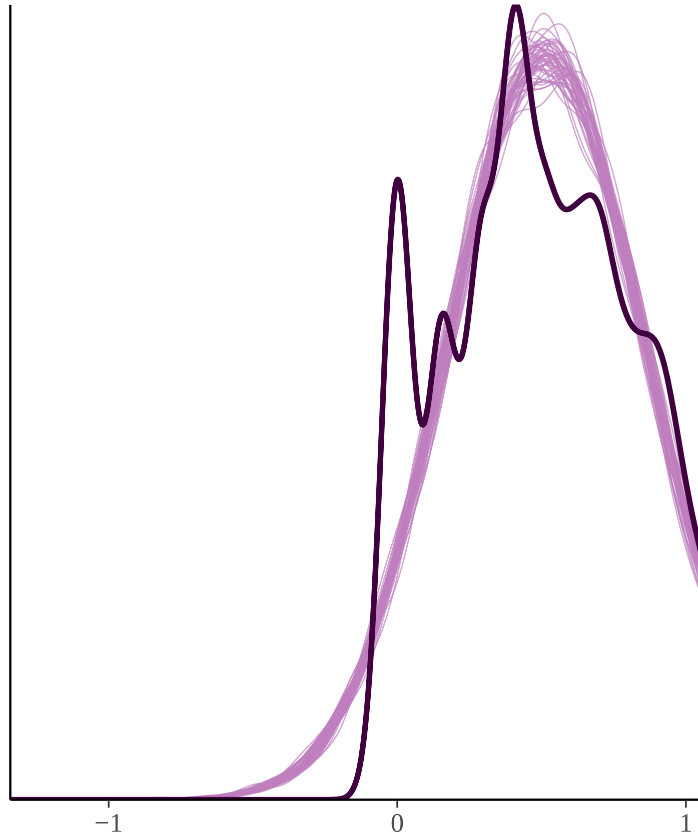
```r
sd(pisa2018$belongLogA)
```

```
## [1] 0.346309
```

```r
prior_summary(fit2A)
```

```
## Priors for model 'fit2A'
## ------
## Intercept (after predictors centered)
##   Specified prior:
##     ~ normal(location = 0.52, scale = 2.5)
##   Adjusted prior:
##     ~ normal(location = 0.52, scale = 0.87)
##
## Coefficients
##   Specified prior:
##     ~ normal(location = [0,0,0,...], scale = [2.5,2.5,2.5,...])
##   Adjusted prior:
##     ~ normal(location = [0,0,0,...], scale = [1.73,2.13,0.88,...])
##
## Auxiliary (sigma)
##   Specified prior:
##     ~ exponential(rate = 1)
##   Adjusted prior:
##     ~ exponential(rate = 2.9)
##
## Covariance
##  ~ decov(reg. = 1, conc. = 1, shape = 1, scale = 1)
## ------
## See help('prior_summary.stanreg') for more details
```

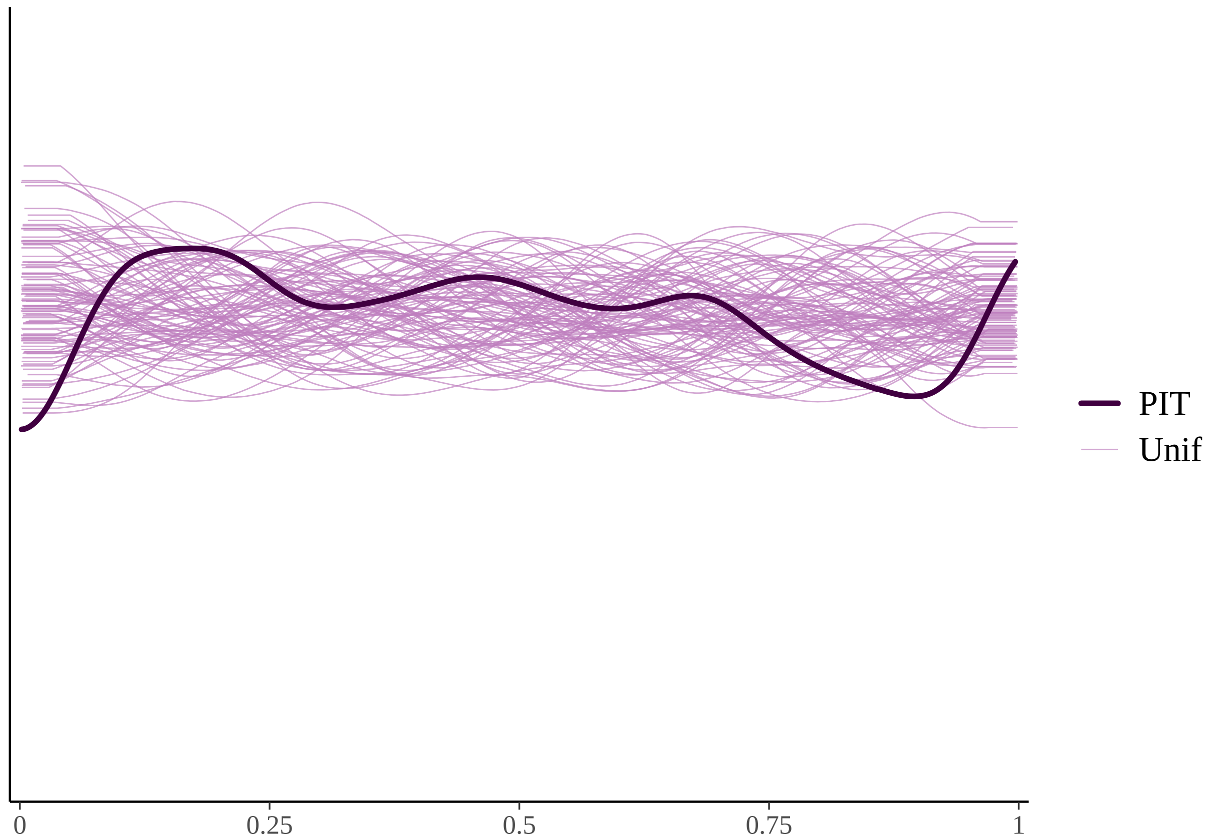
```r
color_scheme_set("purple")
y_rep_logA <- posterior_predict(fit2A)
loo_fit2_logA <- loo(fit2A, save_psis = TRUE, cores = 4)
psis2_logA <- loo_fit2_logA$psis_object
lw_logA <- weights(psis2_logA)
#pdf(file.path(PPath, "Result/LOOPIT_fit2A.pdf"))
pp_check(fit2A)
```



LOOPIT for best model with log-transformed outcome

```
ppc_loo_pit_overlay(y=pisa2018$belongLogA,yrep=y_rep_logA,lw=lw_logA)
```

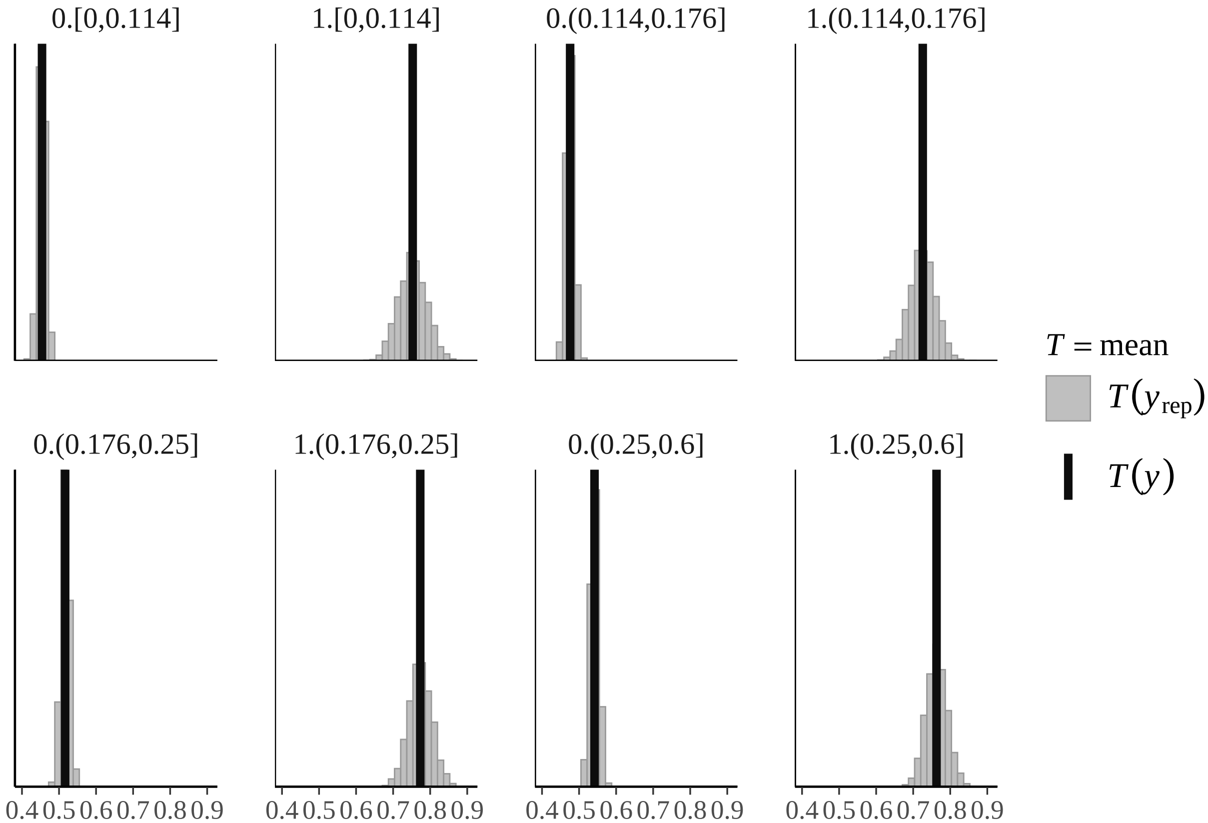
```
## NOTE: The kernel density estimate assumes continuous observations and is not optimal for discrete obs
```




```
y_rep_log <- posterior_predict(fit2A)
color_scheme_set("darkgray")

ppc.plot1_2 <-  ppc_stat_grouped(y = pisa2018$belongLogA,
                                yrep = y_rep_log, stat = "mean",
                     group = interaction(pisa2018$bull,pisa2018$ATT4),
                     facet_args = list(nrow = 2, scales = "fixed"))


ppc.plot2_2 <-  ppc_stat_grouped(y = pisa2018$belongLogA,
                                yrep = y_rep_log, stat = "sd",
                     group = interaction(pisa2018$bull,pisa2018$ATT4),
                     facet_args = list(nrow = 2, scales = "fixed"))


ppc.plot1_2
```

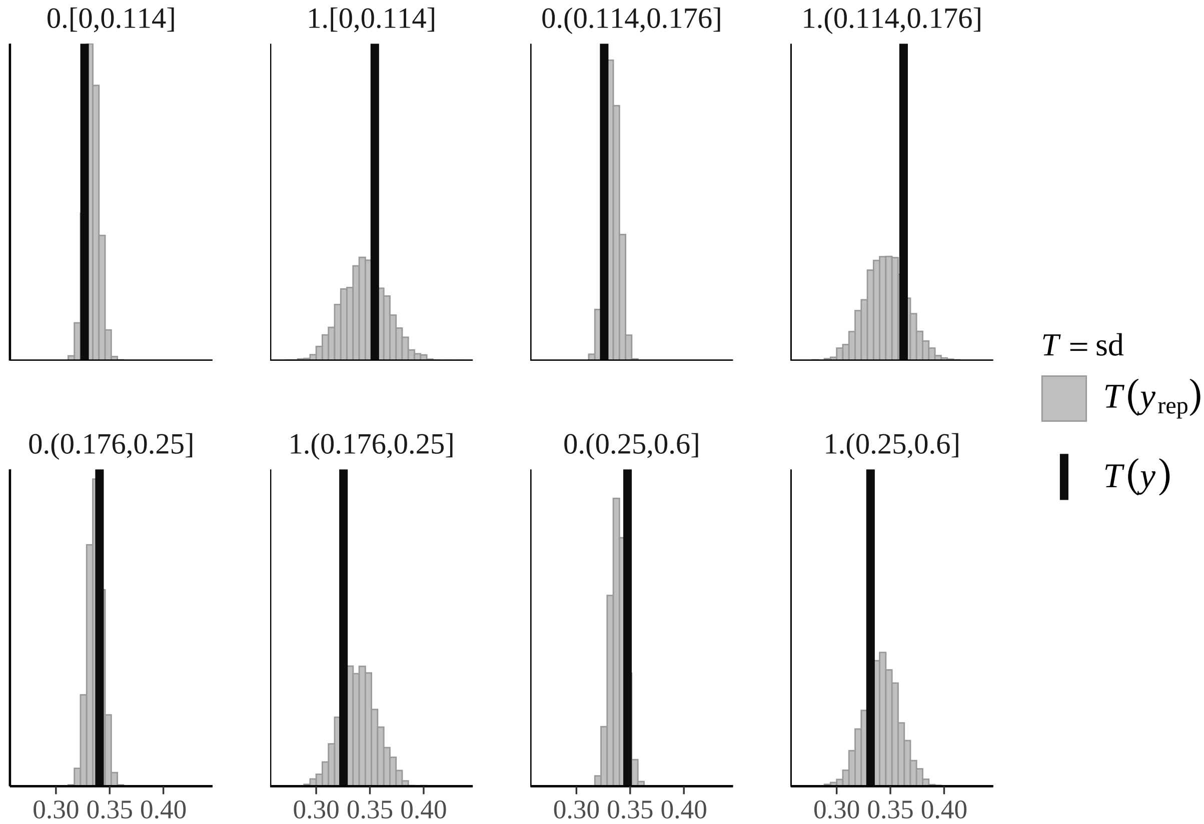
PPC for best model with log-transformed outcome

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



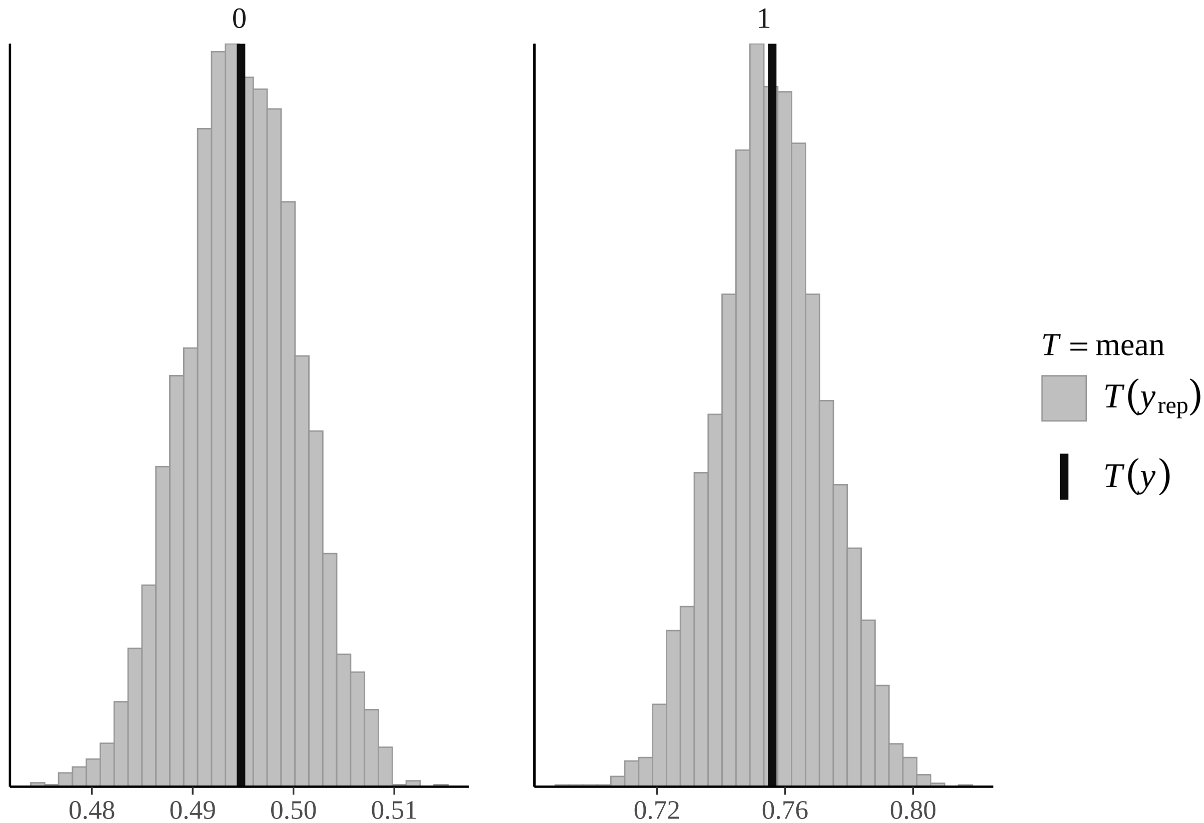

```
ppc.plot2_2
```

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



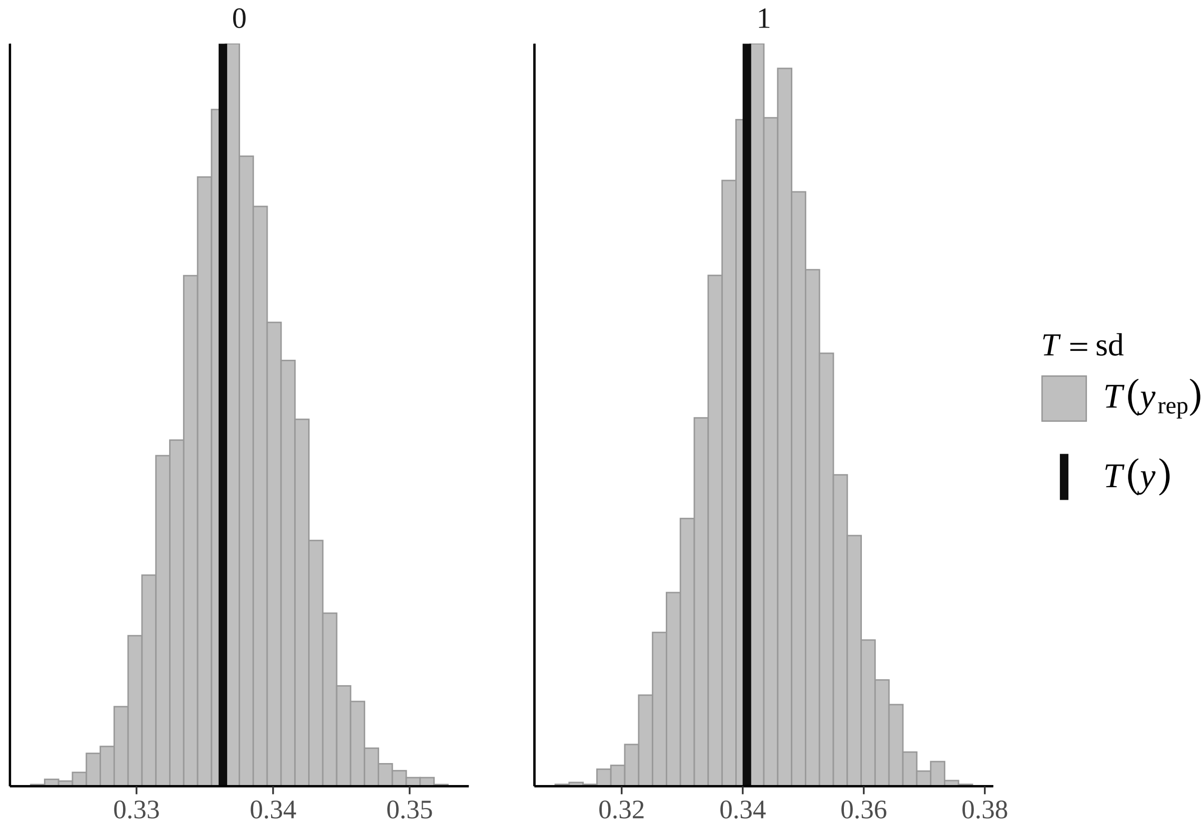

```
y_rep_log  %>% ppc_stat_grouped(y = pisa2018$belongLogA,
                               group = pisa2018$bull,
                               stat = "mean")
```

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



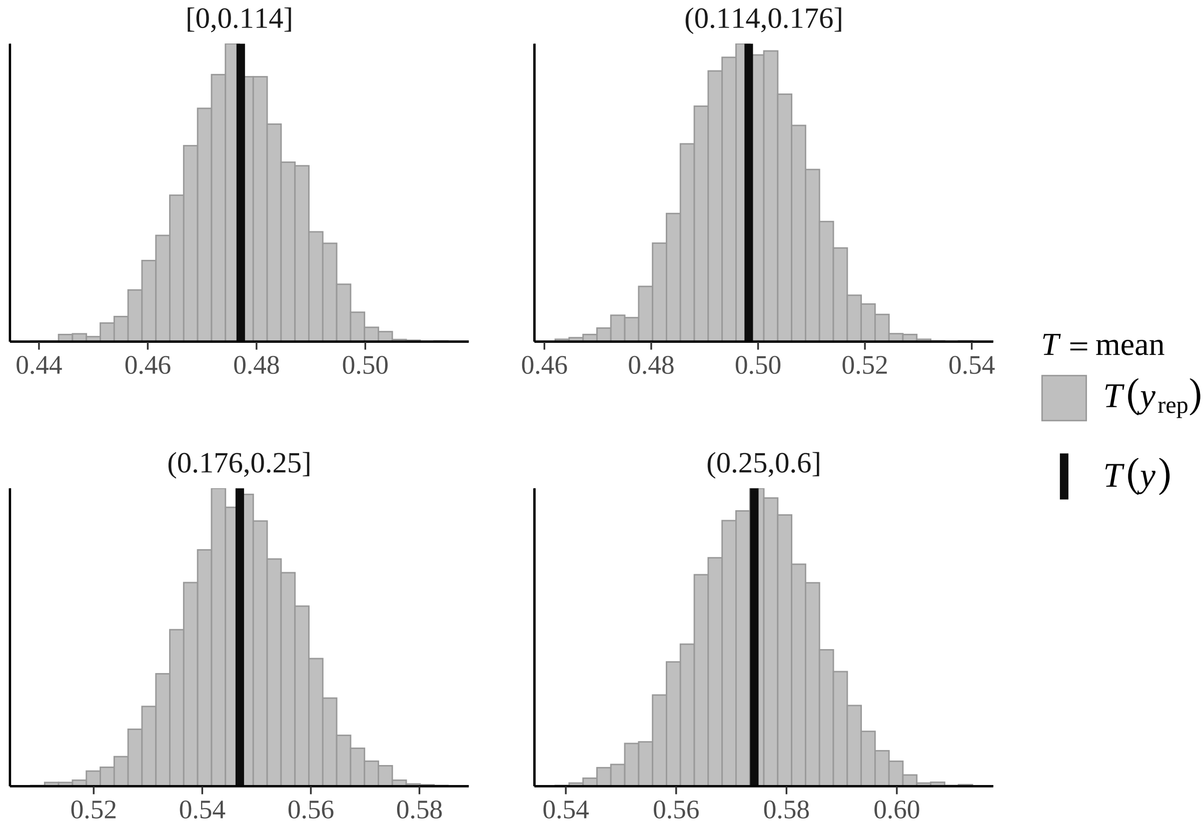

```
y_rep_log  %>% ppc_stat_grouped(y = pisa2018$belongLogA,
                    group = pisa2018$bull,
                    stat = "sd")
```

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



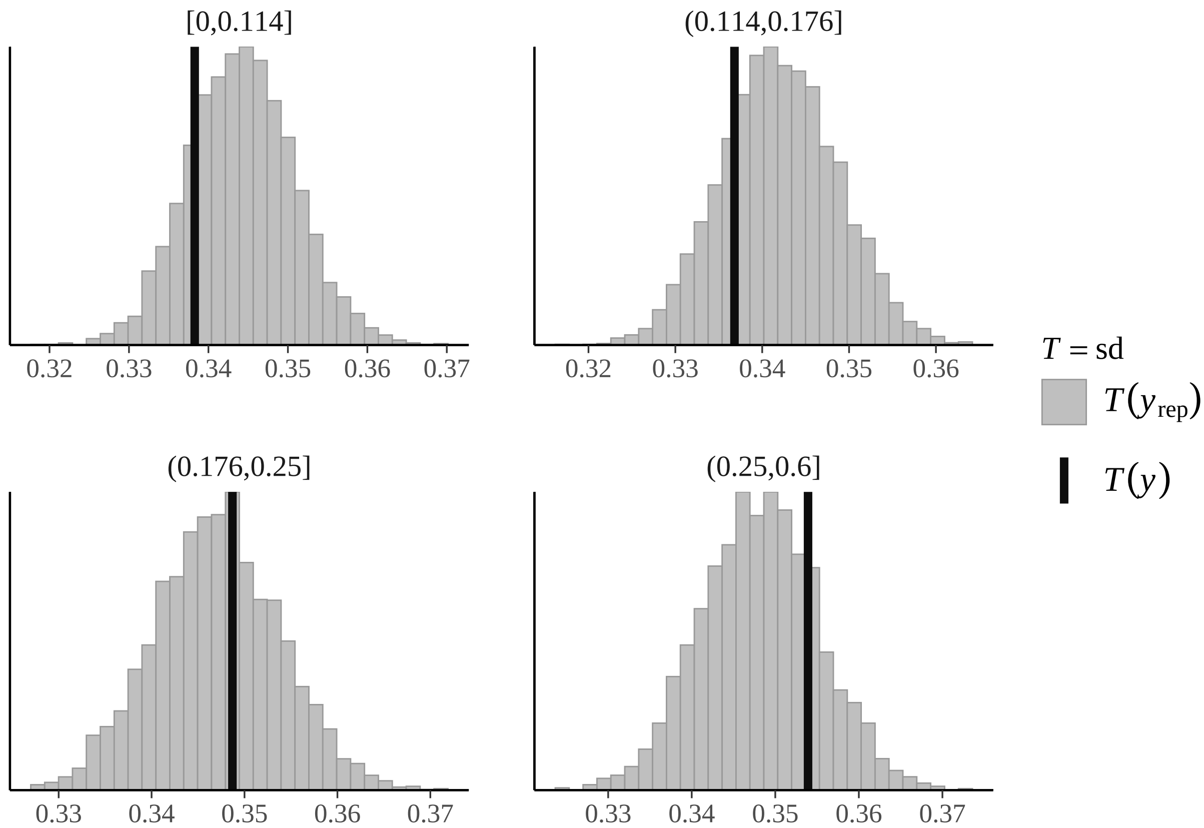

```
y_rep_log  %>% ppc_stat_grouped(y = pisa2018$belongLogA,
                    group = pisa2018$ATT4,
                    stat = "mean")
```

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




```r
y_rep_log  %>% ppc_stat_grouped(y = pisa2018$belongLogA,
                        group = pisa2018$ATT4,
                        stat = "sd")
```

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




```
# sd much better
```

```
summaryTwoLevel <- tidy(fit2A,   conf.int =TRUE, conf.level=.95,
effects = "fixed")
print(summaryTwoLevel, digits = 2, n = 28)
```

Model summary best model with log-transformed outcome

```
## # A tibble: 28 x 5
##    term                        estimate std.error conf.low conf.high
##    <chr>                          <dbl>     <dbl>    <dbl>     <dbl>
##  1 (Intercept)                  0.570     0.0176   0.536    0.605
##  2 female                      -0.0178    0.00873 -0.0346  -0.000866
##  3 nld                         -0.0415    0.0127  -0.0658  -0.0169
##  4 scie_std                     0.00283   0.00617 -0.00954  0.0152
##  5 aca                          0.00153   0.00896 -0.0154   0.0191
##  6 val                         -0.0217    0.00424 -0.0300  -0.0135
##  7 comp                        -0.0140    0.00471 -0.0232  -0.00491
##  8 ndiff                       -0.0263    0.00452 -0.0353  -0.0175
##  9 nfof                        -0.0306    0.00409 -0.0386  -0.0226
## 10 native                       0.00242   0.0109  -0.0186   0.0239
## 11 nfewbooks                   -0.00697   0.00995 -0.0268   0.0122
## 12 joyread                      0.0367    0.00442  0.0282   0.0454
## 13 goal                        -0.00256   0.00465 -0.0116   0.00658
## 14 mot                         -0.00728   0.00441 -0.0160   0.00150
## 15 res                         -0.0382    0.00494 -0.0476  -0.0287
```

```
## 16 swbp                   -0.0623    0.00454 -0.0713  -0.0535
## 17 mean                   -0.0186    0.00470 -0.0276  -0.00925
## 18 parent_sup             -0.0145    0.00421 -0.0227  -0.00646
## 19 ndis_clim               0.00925   0.00401  0.00138  0.0171
## 20 GYM                    -0.0384    0.0133  -0.0643  -0.0130
## 21 UNI                    -0.0175    0.0259  -0.0674   0.0313
## 22 ATT4(0.114,0.176]       0.00877   0.0114  -0.0129   0.0313
## 23 ATT4(0.176,0.25]        0.0348    0.0112   0.0123   0.0572
## 24 ATT4(0.25,0.6]          0.0341    0.0120   0.0103   0.0574
## 25 bull                    0.213     0.0289   0.159    0.268
## 26 ATT4(0.114,0.176]:bull -0.0613    0.0381  -0.135    0.0145
## 27 ATT4(0.176,0.25]:bull  -0.0647    0.0371  -0.137    0.00579
## 28 ATT4(0.25,0.6]:bull    -0.0888    0.0358  -0.158   -0.0183
```

```
summaryTwoLevelModelSchools <- tidy(fit2A,   conf.int =TRUE, conf.level=.95,
effects = "ran_vals")
print(summaryTwoLevelModelSchools, digits = 2)
```

```
## # A tibble: 221 x 7
##    level   group     term           estimate std.error conf.low conf.high
##    <chr>   <chr>     <chr>             <dbl>     <dbl>    <dbl>     <dbl>
##  1 4000001 school.id (Intercept) -0.000116     0.00598  -0.0235    0.0213
##  2 4000002 school.id (Intercept)  0.000764     0.00625  -0.0183    0.0281
##  3 4000003 school.id (Intercept)  0.000318     0.00623  -0.0193    0.0261
##  4 4000005 school.id (Intercept) -0.000929     0.00655  -0.0299    0.0161
##  5 4000006 school.id (Intercept) -0.0000340    0.00627  -0.0228    0.0203
##  6 4000007 school.id (Intercept)  0.000150     0.00583  -0.0193    0.0247
##  7 4000008 school.id (Intercept)  0.000244     0.00571  -0.0184    0.0245
##  8 4000009 school.id (Intercept)  0.0000519    0.00595  -0.0225    0.0245
##  9 4000010 school.id (Intercept)  0.0000990    0.00608  -0.0229    0.0235
## 10 4000011 school.id (Intercept) -0.00000958   0.00596  -0.0233    0.0220
## # i 211 more rows
```

```
summaryTwoLevelModelVar <- tidy(fit2A,   conf.int =TRUE, conf.level=.95,
effects = "ran_pars")
print(summaryTwoLevelModelVar, digits = 2)
```

```
## # A tibble: 2 x 3
##   term                      group     estimate
##   <chr>                     <chr>        <dbl>
## 1 sd_(Intercept).school.id school.id   0.0107
## 2 sd_Observation.Residual  Residual    0.306
```

```
# bull-predictor seems to be the only important predictor, thus consider
# random slope for bull

model2A_sens <- update(model2A,  ~ .+  (1 + bull| school.id) - (1 | school.id))


fit2A_sens <- stan_glmer(model2A_sens, data = pisa2018, seed = a.seed,
                      iter = 2*a.iter, chains = a.chains, warmup =  warmup)

save(fit2A_sens, file = file.path(PPath,"Files/fit2A_sens.RData"))
```

```r
# soo_fit2A_sens <- launch_shinystan(fit2A_sens)
# save(soo_fit2A_sens, file = file.path(PPath,"Files/soo_fit2A_sens.RData"))
```

Sens. analysis: fit model with random slope (part of step 6 in SAP)

```r
# see results
summaryTwoLevel_sens <- tidy(fit2A_sens,   conf.int =TRUE, conf.level=.95,
effects = "fixed")
print(summaryTwoLevel_sens, digits = 2, n = 28)
```

Results of model with random slope: comparison

```
## # A tibble: 28 x 5
##    term                    estimate std.error  conf.low conf.high
##    <chr>                      <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)              0.571     0.0179   0.536      0.607
##  2 female                  -0.0183    0.00871 -0.0352    -0.00142
##  3 nld                     -0.0416    0.0131  -0.0673    -0.0166
##  4 scie_std                 0.00279   0.00615 -0.0100     0.0151
##  5 aca                      0.00169   0.00874 -0.0153     0.0184
##  6 val                     -0.0218    0.00417 -0.0298    -0.0135
##  7 comp                    -0.0142    0.00467 -0.0234    -0.00506
##  8 ndiff                   -0.0262    0.00445 -0.0350    -0.0174
##  9 nfof                    -0.0307    0.00401 -0.0385    -0.0227
## 10 native                   0.00235   0.0111  -0.0191     0.0241
## 11 nfewbooks               -0.00679   0.0104  -0.0268     0.0133
## 12 joyread                  0.0368    0.00447  0.0281     0.0456
## 13 goal                    -0.00257   0.00449 -0.0116     0.00638
## 14 mot                     -0.00734   0.00457 -0.0160     0.00144
## 15 res                     -0.0381    0.00478 -0.0474    -0.0289
## 16 swbp                    -0.0624    0.00447 -0.0709    -0.0537
## 17 mean                    -0.0187    0.00459 -0.0276    -0.00940
## 18 parent_sup              -0.0143    0.00427 -0.0226    -0.00608
## 19 ndis_clim                0.00899   0.00401  0.000993   0.0168
## 20 GYM                     -0.0379    0.0128  -0.0632    -0.0132
## 21 UNI                     -0.0207    0.0253  -0.0701     0.0285
## 22 ATT4(0.114,0.176]        0.00884   0.0118  -0.0137     0.0315
## 23 ATT4(0.176,0.25]         0.0345    0.0118   0.0111     0.0574
## 24 ATT4(0.25,0.6]           0.0335    0.0122   0.00951    0.0578
## 25 bull                     0.215     0.0298   0.157      0.273
## 26 ATT4(0.114,0.176]:bull -0.0625    0.0408  -0.142      0.0171
## 27 ATT4(0.176,0.25]:bull  -0.0690    0.0399  -0.146      0.00852
## 28 ATT4(0.25,0.6]:bull    -0.0885    0.0378  -0.162     -0.0140
```

```r
summaryTwoLevelModelSchools_sens <- tidy(fit2A_sens, conf.int =TRUE, conf.level=.95,
effects = "ran_vals")
print(summaryTwoLevelModelSchools_sens, digits = 2)
```

```
## # A tibble: 442 x 7
##    level   group     term          estimate std.error conf.low conf.high
##    <chr>   <chr>     <chr>            <dbl>     <dbl>    <dbl>     <dbl>
##  1 4000001 school.id (Intercept) -0.000208    0.00798  -0.0263    0.0240
##  2 4000001 school.id bull        -0.000286    0.0350   -0.123     0.119
##  3 4000002 school.id (Intercept)  0.00131     0.00821  -0.0189    0.0304
```

```
##  4 4000002 school.id bull        -0.000463    0.0337   -0.123     0.110
##  5 4000003 school.id (Intercept)  0.000878    0.00842  -0.0207    0.0305
##  6 4000003 school.id bull        -0.0103      0.0371   -0.176     0.0709
##  7 4000005 school.id (Intercept) -0.00244     0.00851  -0.0355    0.0164
##  8 4000005 school.id bull         0.00456     0.0368   -0.0927    0.174
##  9 4000006 school.id (Intercept) -0.0000212   0.00829  -0.0257    0.0249
## 10 4000006 school.id bull        -0.000894    0.0362   -0.133     0.111
## # i 432 more rows
```

```
summaryTwoLevelModelVar_sens <- tidy(fit2A_sens, conf.int =TRUE, conf.level=.95,
effects = "ran_pars")
print(summaryTwoLevelModelVar_sens, digits = 2)
```

```
## # A tibble: 4 x 3
##   term                            group     estimate
##   <chr>                           <chr>        <dbl>
## 1 sd_(Intercept).school.id       school.id   0.0122
## 2 sd_bull.school.id              school.id   0.0604
## 3 cor_(Intercept).bull.school.id school.id  -0.164
## 4 sd_Observation.Residual        Residual    0.306
```

```
fits <- list(fit2A,fit2A_sens)
loo_list <- list()
# Compute loo

loo_list <- lapply(fits, loo, cores = 1)
# Save loo

save("loo_list", file = file.path(PPath,"Files","loo_list.RData"))
#load(file.path("Files2","loo_list.RData"))
n01.models <- 2
#sink(file = file.path(PPath,"Files","loo_info.txt"))
for (i in 1:n01.models) {
  print(paste("******** "))
  print(paste("model ", i))
  print(loo_list[[i]])
  print(paste("******** "))
}
```

```
## [1] "******** "
## [1] "model  1"
##
## Computed from 4000 by 6489 log-likelihood matrix.
##
##          Estimate    SE
## elpd_loo  -1547.2  65.2
## p_loo        38.8   0.9
## looic      3094.4 130.5
## ------
## MCSE of elpd_loo is 0.1.
## MCSE and ESS estimates assume MCMC draws (r_eff in [0.8, 2.4]).
##
## All Pareto k estimates are good (k < 0.7).
## See help('pareto-k-diagnostic') for details.
## [1] "******** "
```

```
## [1] "******** "
## [1] "model  2"
##
## Computed from 12000 by 6489 log-likelihood matrix.
##
##          Estimate    SE
## elpd_loo  -1545.7  65.3
## p_loo        62.4   2.1
## looic      3091.5 130.5
## ------
## MCSE of elpd_loo is 0.1.
## MCSE and ESS estimates assume MCMC draws (r_eff in [0.1, 1.7]).
##
## All Pareto k estimates are good (k < 0.7).
## See help('pareto-k-diagnostic') for details.
## [1] "******** "
```

```
looc <- loo_compare(loo_list)
print(looc)
```

```
##        elpd_diff se_diff
## X[[i]]  0.0       0.0
## X[[i]] -1.5       1.8
```

```
save(looc, file = file.path(PPath,"Files/looc.RData"))
# No notable difference
##### Stacking, only if continous model extension is not possible, not needed here
```

```

squcondPr <- split(pisa2018,f = list(pisa2018$ATT4,pisa2018$bull))
cnt.path <- file.path(PPath,"Files")
#t<-lapply(squcondPr1_gap, function(x) length(x[[1]]))
#
fits <- fit2A
pint_condPr25 <- comp.int(sgr = squcondPr,fit = fits,
                                       cnt.path = cnt.path, p = 0.25)
t25<-read.table(file = file.path(cnt.path,paste(0.25,"group_meds.txt")))

pint_condPr50 <- comp.int(sgr = squcondPr,fit = fits,
                                     cnt.path = cnt.path, p = 0.5)


pint_condPr75 <- comp.int(sgr = squcondPr,fit = fits,
                                       cnt.path = cnt.path, p = 0.75)
#t <- unlist(lapply(squcondPr, function(x) length(x[[1]])))

brE <- 4

condPr1 <- pisa2018 %>%
  group_by(ATT4) %>%
  summarise(mATT = median(ATT01), sd = sd(ATT01),min = min(ATT01),max = max(ATT01))

meancondPr1 <- condPr1$mATT
#h <- round(meancondPr1,2)
#h2 <- c(h[1],h[1],h[2],h[2],h[3],h[3],h[4],h[4])
#meancondPr1 <- h2
```

```r
# c(rep("m.",brE),rep("f.",brE))
df_condPr25 <- data.frame(x = rep(round(meancondPr1,2),2),
                          group = c(rep("nbv.",brE),rep("bv.",brE)),
                          mean =  pint_condPr25[,2],low = pint_condPr25[,1],
                          up = pint_condPr25[,3])#here

df_condPr50 <- data.frame(x = rep(round(meancondPr1,2),2),
                        group = c(rep("nbv.",brE),rep("bv.",brE)),
                        mean =  pint_condPr50[,2],low = pint_condPr50[,1],
                        up = pint_condPr50[,3])

df_condPr75 <- data.frame(x = rep(round(meancondPr1,2),2),
                          group = c(rep("nbv.",brE),rep("bv.",brE)),
                          mean =  pint_condPr75[,2],low = pint_condPr75[,1],
                          up = pint_condPr75[,3])

lowQ <- c(pint_condPr25[,1],pint_condPr50[,1],pint_condPr75[,1])
med <-  c(pint_condPr25[,2],pint_condPr50[,2],pint_condPr75[,2])
upQ <-  c(pint_condPr25[,3],pint_condPr50[,3],pint_condPr75[,3])

df_condPr1 <- data.frame(x = rep(round(meancondPr1,2),6),
                         group = c(rep("25 % nbv.",brE),rep("25 % bv.",brE),
                                   rep("50 % nbv.",brE),rep("50 % bv.",brE),
                                   rep("75 % nbv.",brE),rep("75 % bv.",brE)),
                                   mean =  med,low = lowQ,
                                   up = upQ)

#sink(file = file.path(cnt.path,"interval.txt"))
df_condPr1 # kable(df_condPr1,format = "latex", digits = 3)
```

Result of focal interest: use graphic to summarize posterior knowledge

```
##       x     group     mean       low       up
## 1  0.09 25 % nbv. 3.163284 1.9865605 3.893203
## 2  0.15 25 % nbv. 3.210170 2.0452012 3.922845
## 3  0.21 25 % nbv. 3.070784 1.8357822 3.817942
## 4  0.32 25 % nbv. 2.957749 1.6181208 3.769840
## 5  0.09  25 % bv. 2.484544 0.8654534 3.482914
## 6  0.15  25 % bv. 2.499830 0.8537804 3.475216
## 7  0.21  25 % bv. 2.383142 0.7134909 3.416144
## 8  0.32  25 % bv. 2.490672 0.8837518 3.484522
## 9  0.09 50 % nbv. 3.499062 2.5626851 4.079778
## 10 0.15 50 % nbv. 3.460979 2.4588710 4.072420
## 11 0.21 50 % nbv. 3.397695 2.3647358 4.025642
## 12 0.32 50 % nbv. 3.298535 2.2075068 3.967891
## 13 0.09  50 % bv. 2.929705 1.5701305 3.743598
## 14 0.15  50 % bv. 2.979705 1.6544467 3.773255
## 15 0.21  50 % bv. 2.823578 1.3309783 3.676495
## 16 0.32  50 % bv. 2.752843 1.2960446 3.630608
## 17 0.09 75 % nbv. 3.678571 2.8004309 4.200198
## 18 0.15 75 % nbv. 3.629119 2.7501499 4.173352
## 19 0.21 75 % nbv. 3.623882 2.7169951 4.157295
## 20 0.32 75 % nbv. 3.648124 2.7562964 4.177587
## 21 0.09  75 % bv. 3.285087 2.1603175 3.966743
```

```
## 22 0.15  75 % bv. 3.348921 2.2913188 4.005760
## 23 0.21  75 % bv. 3.284380 2.1473587 3.950089
## 24 0.32  75 % bv. 3.374920 2.2710647 4.027109
```

```r
#sink(file = NULL)
#library("kableExtra")
#sink(file = file.path(cnt.path,"intervalLatex.txt"))
#kable(df_condPr1,format = "latex", digits = 3)
#sink(file = NULL)

p_condPr <- plot.result(df_condPr1)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
p_condPr
```