Analyses for the paper: Measuring emotions during learning

Franziska Hirt

19 August 2019

1. PREPARATIONS

```
# load packages
library(tidyverse)
library(rstan)
library(brms)
library(bayesplot)
library(ggmcmc) # for ggs posterior plot

# set rstan options
rstan::rstan_options(auto_write = T)
options(mc.cores = parallel::detectCores())
```

Selection of statistical methods

- Bayesian statistics, as they are more intuitive to interpret.
- Mixed models, because events (level 1) are nested within participants (level 2) and texts (level 2). However, only random intercepts are included, as additional random effects would be hardly identifyable.
- Treat emotional self-reports as ordinal outcome variables. Cummulative family in brms, as we understand the Likert-scales as the categorization of a latent continuous construct (Buerkner & Vuorre, 2019: https://journals.sagepub.com/doi/full/10.1177/2515245918823199 (https://journals.sagepub.com/doi/full/10.1177/2515245918823199)).
- Assummption of equal variances: “If unequal variances are theoretically possible – and they usually are – we also recommend incorporating them into the model” (Buerkner & Vuorre, 2019). However, models allowing for unequal variances did not converge and were therefore omitted.

Choosing link function:

link-distributions available for cummulative models in brms (usually only minor impact on results): logit = logistic, probit = gaussian, cloglog = extreme value distribution http://bayesium.com/which-link-function-logit-probit-or-cloglog/ (http://bayesium.com/which-link-function-logit-probit-or-cloglog/)

The choice should be made based on some combination of: - Knowledge of the response distribution, - Theoretical considerations, and - Empirical fit to the data. https://stats.stackexchange.com/questions/20523/difference-between-logit-and-probit-models (https://stats.stackexchange.com/questions/20523/difference-between-logit-and-probit-models)

interpretation of summary of fitted model

- Estimate is the mean of the posterior distribution, and corresponds to the frequentist point estimate
- Est.Error is the standard deviation of the posterior distribution

- thresholds in ordinal models are called “intercepts” in the output
- Visualisation of marginal effects for ordinal models: https://github.com/paul-buerkner/brms/issues/190 (https://github.com/paul-buerkner/brms/issues/190)

posterior predictive checks (not included in this document)

for ordinal models pp_check not adequate: https://github.com/stan-dev/bayesplot/issues/73 (https://github.com/stan-dev/bayesplot/issues/73) –> use ppc: https://mc-stan.org/bayesplot/articles/graphical-ppcs.html (https://mc-stan.org/bayesplot/articles/graphical-ppcs.html)

load data

```
# load data
df <- read_csv("df_TEEM_final.csv")

# rename some variables
df <- df %>% rename("participant" = "subject_nr", "text" = "text_pic", "valence_post" = "SAM_
LIKERT_POST")
```

standardize predictors (aggregated from FaceReader)

(helps for model convergence and for the interpretation of the interaction effects)

```
df <- df %>%
  mutate(
    mean_interest = scale(mean_interest, center = T, scale = T),
    mean_boredom = scale(mean_boredom, center = T, scale = T),
    mean_valence = scale(mean_valence, center = T), scale = T)

df <- df %>%
  mutate(
    sd_interest = scale(sd_interest, center = T, scale = T),
    sd_boredom = scale(sd_boredom, center = T, scale = T),
    sd_valence = scale(sd_valence, center = T), scale = T)

df <- df %>%
  mutate(
    peak10_interest = scale(peak10_interest, center = T, scale = T),
    peak10_boredom = scale(peak10_boredom, center = T, scale = T),
    peak10_valence_pos = scale(peak10_valence_pos, center = T, scale = T),
    peak10_valence_neg = scale(peak10_valence_neg), center = T, scale = T)
```

2. INTEREST

INTEREST restricted model

```
# complete cases only (drop NAs)
dfsub <- df %>% select(participant, interested_post, mean_interest, sd_interest, peak10_interest, text) %>% drop_na()

# restricted model
m0i_cloglog <- brm(
          interested_post ~ 1 + (1|participant) + (1|text),
          family = cumulative("cloglog"),
          prior = prior(cauchy(0, 10), class = sd),
          iter = 4000, warmup = 2000, chains = 4, cores = 4,
          control = list(adapt_delta = 0.99),
          inits = 0,
          data = dfsub,
          save_all_pars = T) # needed for bayes factor

summary(m0i_cloglog)
```

```
##  Family: cumulative
##   Links: mu = cloglog; disc = identity
## Formula: interested_post ~ 1 + (1 | participant) + (1 | text)
##    Data: dfsub (Number of observations: 205)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~participant (Number of levels: 103)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.52      0.28     1.01     2.10       1663 1.00
##
## ~text (Number of levels: 6)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.73      0.43     0.24     1.83       2457 1.00
##
## Population-Level Effects:
##              Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]    -5.53      0.81    -7.26    -4.09       3740 1.00
## Intercept[2]    -3.07      0.50    -4.11    -2.16       3798 1.00
## Intercept[3]    -1.56      0.43    -2.41    -0.75       4062 1.00
## Intercept[4]     0.98      0.42     0.21     1.89       3773 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# other link functions
m0i_logit <-  update(m0i_cloglog,
                    family = cumulative("logit"))

m0i_probit <-  update(m0i_cloglog,
                      family = cumulative("probit"))
# compare different link functions using assimilation of leave-one-out-cross validation (looi
c)
m0i_logit <- add_criterion(m0i_logit,"loo", reloo = T) #"reloo = T" actually calculates MCMC
 for problematic observations
m0i_probit <- add_criterion(m0i_probit,"loo", reloo = T)
m0i_cloglog <- add_criterion(m0i_cloglog ,"loo", reloo = T)
print(loo_compare(m0i_logit, m0i_probit, m0i_cloglog, criterion="loo"), simplify = F)  # clog
log 1.5-3.5 SD better
```

```
##             elpd_diff se_diff elpd_loo se_elpd_loo p_loo  se_p_loo looic
## m0i_cloglog    0.0       0.0  -239.6     11.8        74.9    5.8    479.1
## m0i_probit    -1.9       1.4  -241.5     11.5        71.5    6.0    483.0
## m0i_logit     -2.0       1.3  -241.5     11.6        76.5    5.7    483.0
##             se_looic
## m0i_cloglog   23.5
## m0i_probit    23.0
## m0i_logit     23.1
```

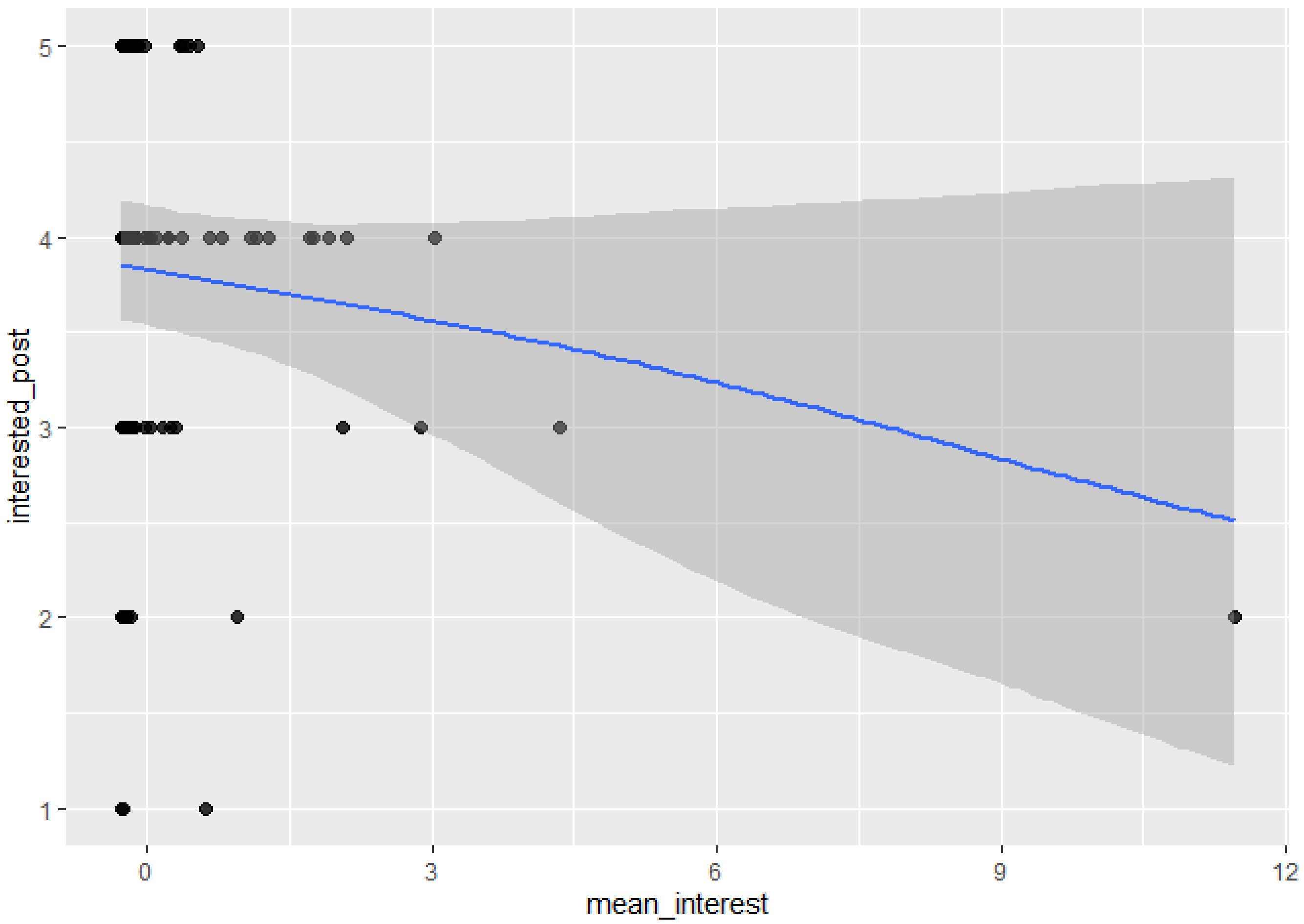
```
# chosen response distribution (link function) for final restricted model
m0i <- m0i_cloglog
```

INTEREST mean

```
# full model including FaceReader's estimate as predictor
m1_imean <-  update(m0i, formula. = ~ . + mean_interest,
                    prior = c(prior(normal(0, 10), class = b),
                              prior(cauchy(0, 10), class = sd)),
                    newdata = dfsub,
                    save_all_pars = T)

## model parameter
summary(m1_imean)
```

```
##  Family: cumulative
##   Links: mu = cloglog; disc = identity
## Formula: interested_post ~ (1 | participant) + (1 | text) + mean_interest
##    Data: dfsub (Number of observations: 205)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~participant (Number of levels: 103)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.49      0.27     0.99     2.06       1475 1.00
##
## ~text (Number of levels: 6)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.70      0.40     0.23     1.69       2715 1.00
##
## Population-Level Effects:
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]     -5.66      0.85    -7.53    -4.17       3859 1.00
## Intercept[2]     -3.08      0.49    -4.13    -2.18       3704 1.00
## Intercept[3]     -1.56      0.41    -2.43    -0.78       4271 1.00
## Intercept[4]      0.97      0.40     0.21     1.83       4362 1.00
## mean_interest    -0.23      0.16    -0.53     0.09       5384 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

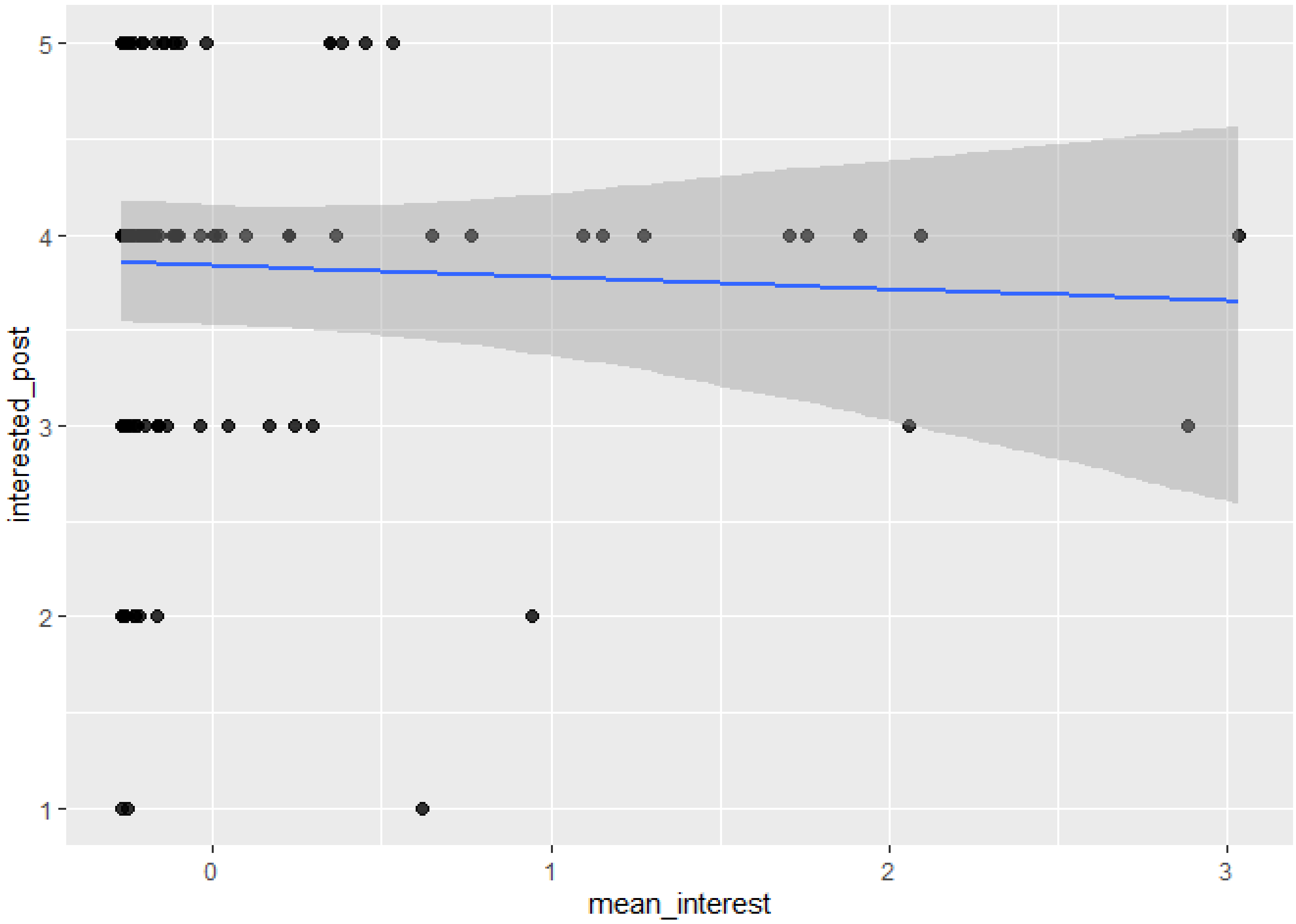
```
# plots
plot(marginal_effects(m1_imean, "mean_interest", categorical = F), points = T, point_args = c
(alpha = 0.8)) # shows the strong influence of two obervations
```



```
# Model without "outliers"
dfsub_out <- dfsub %>%
  filter(mean_interest < (mean(mean_interest) + 4*sd(mean_interest)) & mean_interest > (mean
(mean_interest) - 4*sd(mean_interest)))
# resulting in two observations less
# dfsub %>% filter(mean_interest > (mean(mean_interest) + 4*sd(mean_interest)) | mean_interes
t < (mean(mean_interest) - 4*sd(mean_interest))) # ouliers are from one participant (highly e
xpressive in video)

m1_imean_out <-  update(m0i, formula. = ~ . + mean_interest,
                    prior = c(prior(normal(0, 10), class = b),
                              prior(cauchy(0, 10), class = sd)),
                    newdata = dfsub_out,
                    save_all_pars = T)
summary(m1_imean_out)
```

```
##  Family: cumulative 
##   Links: mu = cloglog; disc = identity 
## Formula: interested_post ~ (1 | participant) + (1 | text) + mean_interest 
##    Data: dfsub_out (Number of observations: 203) 
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
## 
## Group-Level Effects: 
## ~participant (Number of levels: 102) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.52      0.27     1.00     2.08       1748 1.00
## 
## ~text (Number of levels: 6) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.76      0.48     0.25     1.92       2629 1.00
## 
## Population-Level Effects: 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]     -5.48      0.80    -7.22    -4.04       3600 1.00
## Intercept[2]     -3.09      0.50    -4.13    -2.18       3813 1.00
## Intercept[3]     -1.58      0.43    -2.45    -0.77       3964 1.00
## Intercept[4]      0.98      0.43     0.22     1.88       3946 1.00
## mean_interest    -0.15      0.32    -0.77     0.50       5419 1.00
## 
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample 
## is a crude measure of effective sample size, and Rhat is the potential 
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
## plots
plot(marginal_effects(m1_imean_out, "mean_interest", categorical = F), points = T, point_args
= c(alpha = 0.8))
```



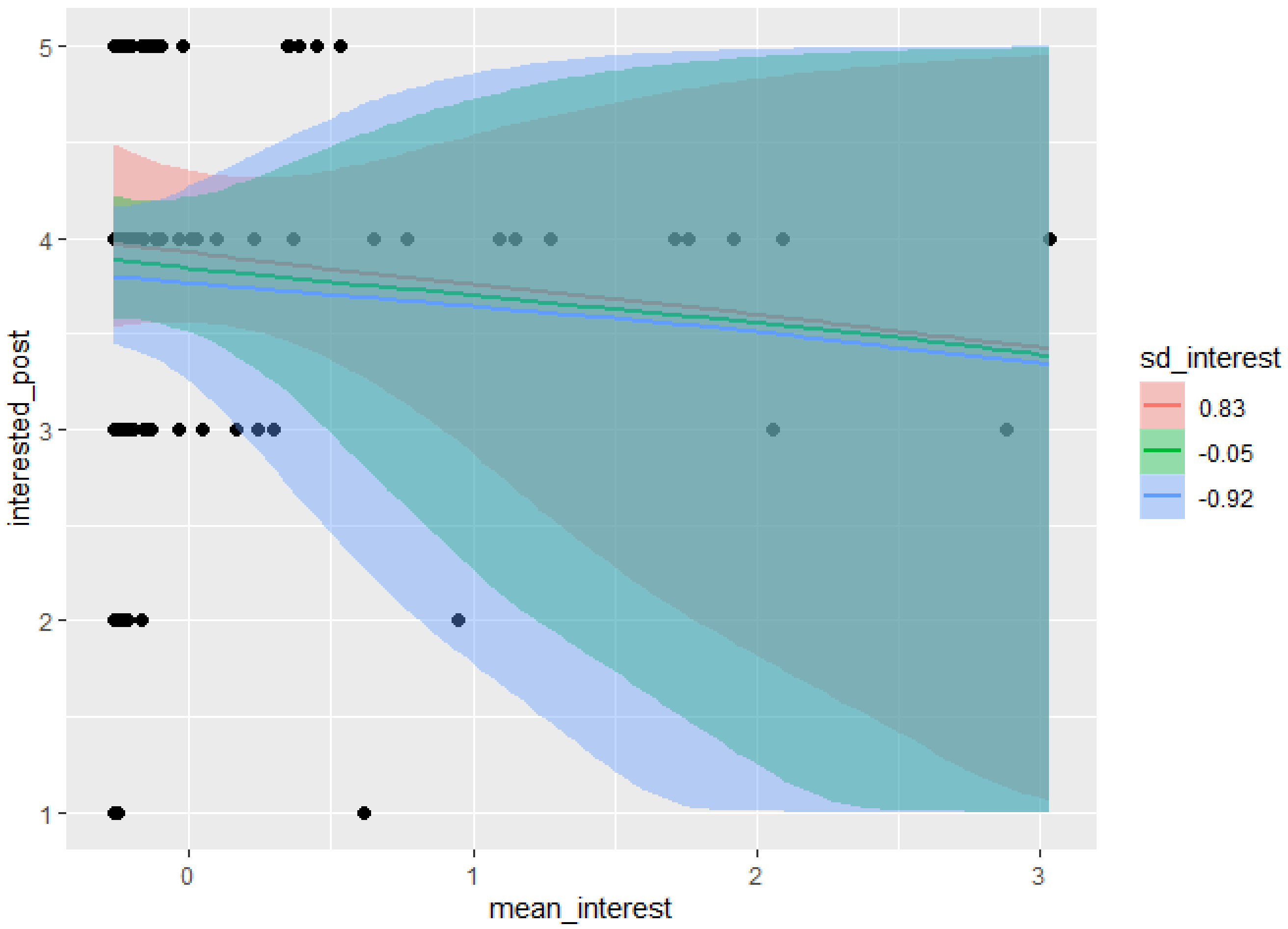
```
# choose final model
m1_imean <- m1_imean_out
```

INTEREST mean*SD

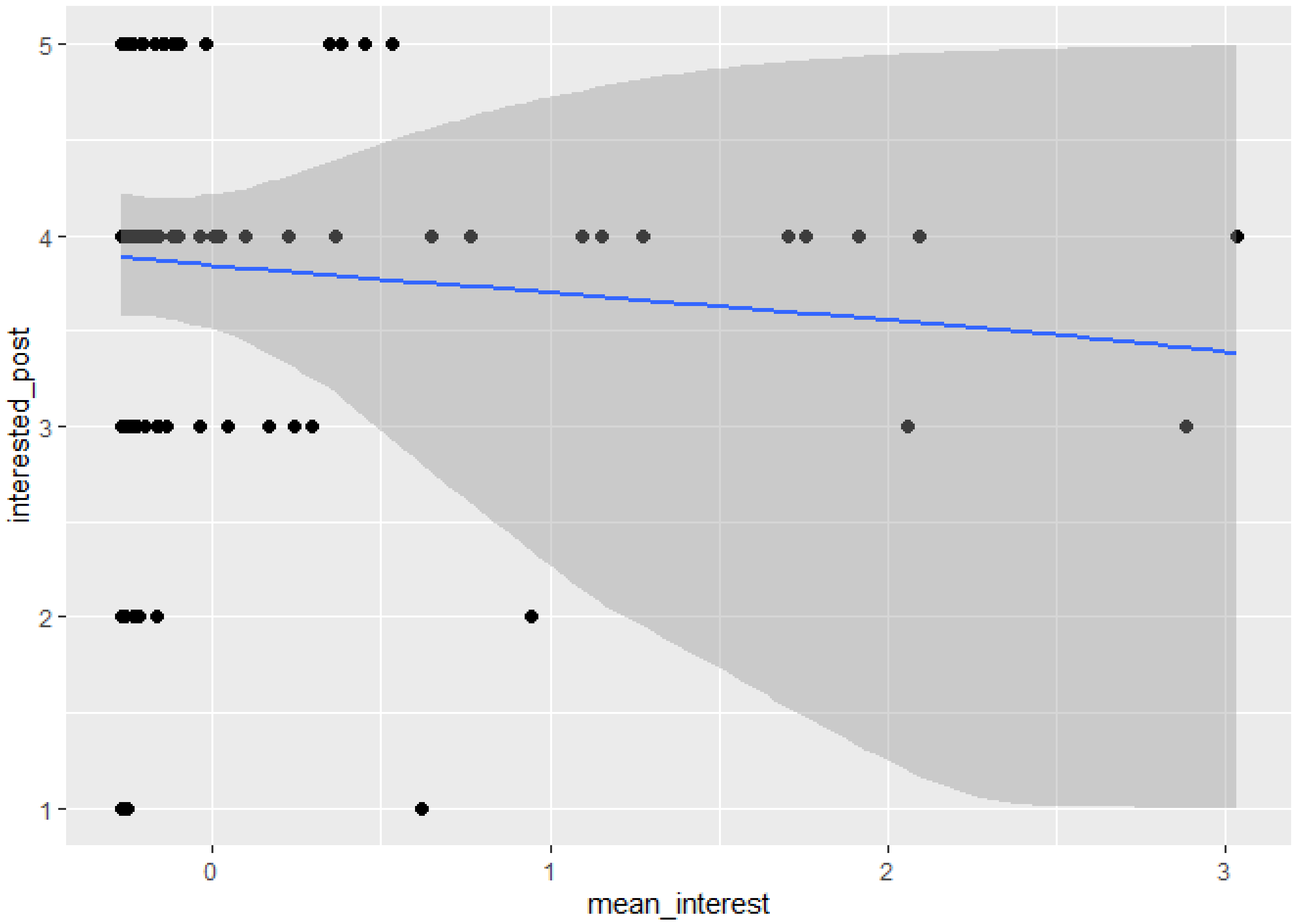
```
# the participant from before also outlier in SD? --> No.
#dfsub_out %>% filter(sd_interest > (mean(sd_interest) + 4*sd(sd_interest)) | sd_interest <
 (mean(sd_interest) - 4*sd(sd_interest)))


# full model including FaceReader's estimates as predictor
m1_imeanxsd <-  update(m0i, formula. = ~ . + mean_interest*sd_interest,
                    prior = c(prior(normal(0, 10), class = b),
                              prior(cauchy(0, 10), class = sd)),
                    newdata = dfsub_out, # without outliers of mean interest
                    save_all_pars = T)

## model indicator
summary(m1_imeanxsd)
```

```
##  Family: cumulative 
##   Links: mu = cloglog; disc = identity 
## Formula: interested_post ~ (1 | participant) + (1 | text) + mean_interest + sd_interest + mean_interest:sd_interest 
##    Data: dfsub_out (Number of observations: 203) 
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
## 
## Group-Level Effects: 
## ~participant (Number of levels: 102) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.54      0.28     1.02     2.14       1702 1.00
## 
## ~text (Number of levels: 6) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.74      0.42     0.25     1.83       2771 1.00
## 
## Population-Level Effects: 
##                           Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## Intercept[1]                 -5.56      0.84    -7.35    -4.09       3422
## Intercept[2]                 -3.15      0.54    -4.29    -2.16       3661
## Intercept[3]                 -1.62      0.47    -2.58    -0.75       3718
## Intercept[4]                  0.98      0.46     0.11     1.95       2842
## mean_interest                -0.41      1.12    -2.70     1.72       3288
## sd_interest                   0.24      0.38    -0.50     1.01       3714
## mean_interest:sd_interest    -0.04      0.33    -0.65     0.66       3842
##                           Rhat
## Intercept[1]              1.00
## Intercept[2]              1.00
## Intercept[3]              1.00
## Intercept[4]              1.00
## mean_interest             1.00
## sd_interest               1.00
## mean_interest:sd_interest 1.00
## 
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample 
## is a crude measure of effective sample size, and Rhat is the potential 
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

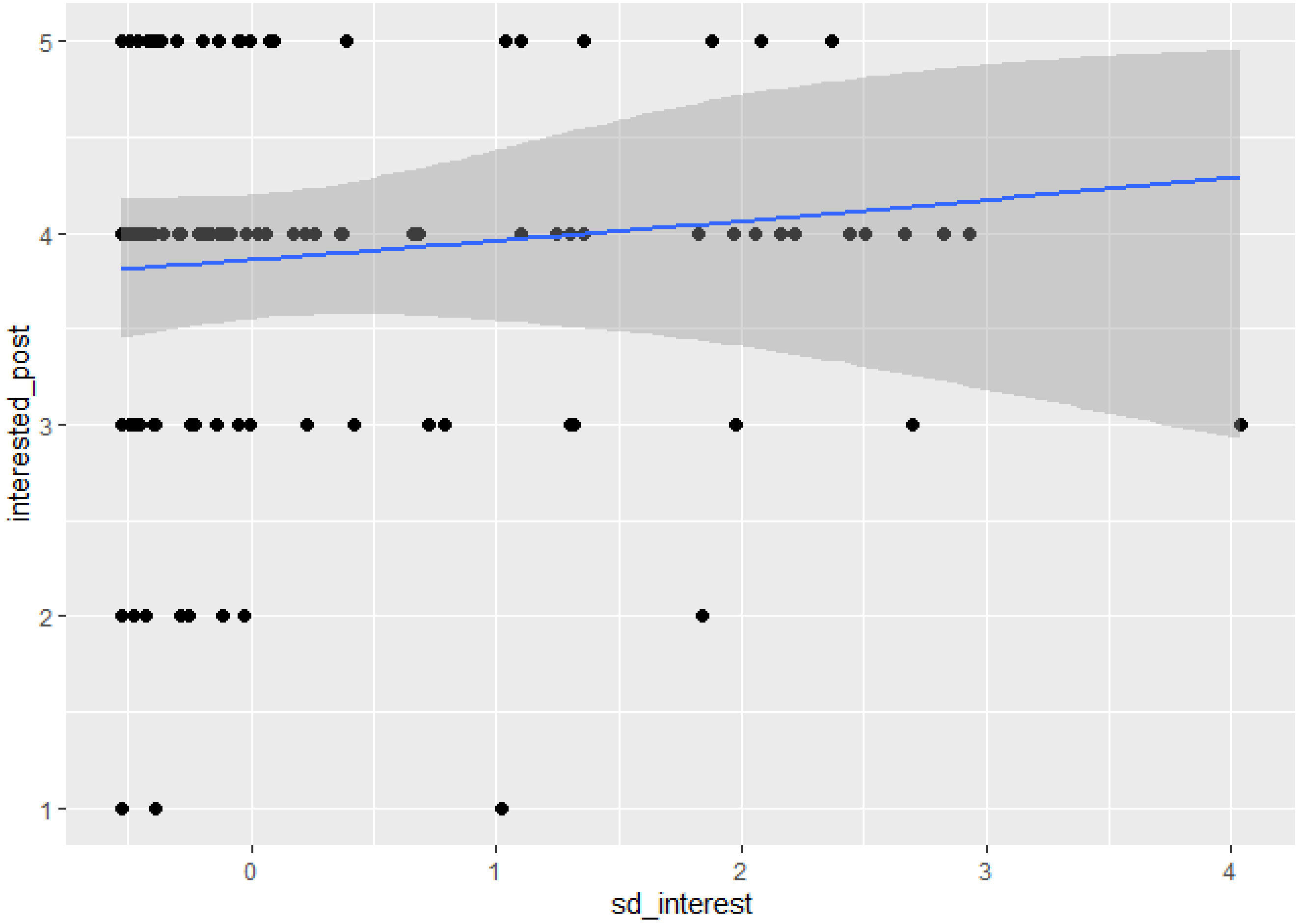
```
# plots
plot(marginal_effects(m1_imeanxsd,"mean_interest:sd_interest", categorical = F), points = T)
```



```
plot(marginal_effects(m1_imeanxsd,"mean_interest", categorical = F), points = T)
```



```
plot(marginal_effects(m1_imeanxsd,"sd_interest", categorical = F), points = T)
```



INTEREST mean of peaks

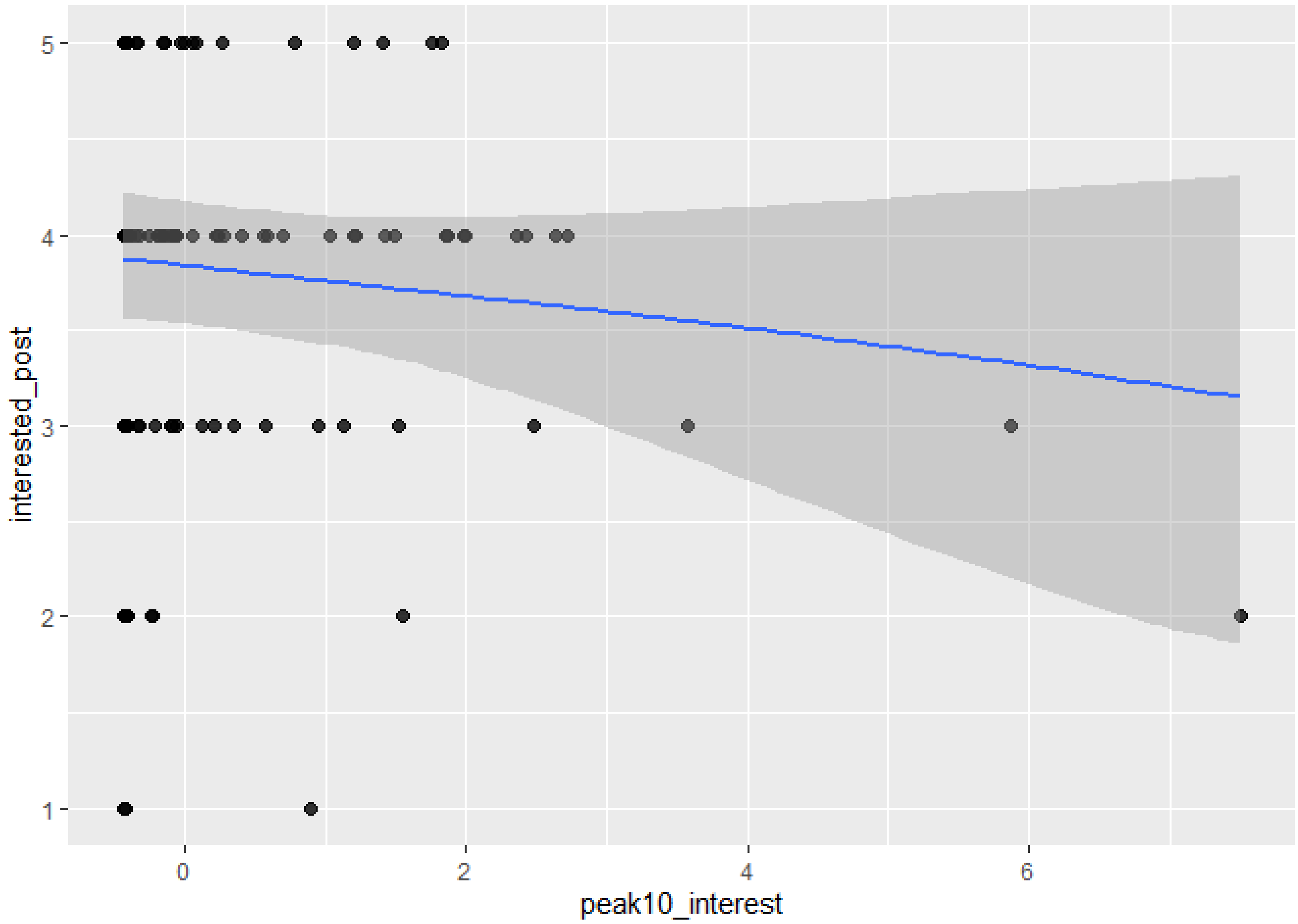
```
# remove outliers also in peak when from the same participant as in mean:
dfsub_outpeak <- dfsub %>% filter(peak10_interest < (mean(peak10_interest) + 4*sd(peak10_inte
rest)) & peak10_interest > (mean(peak10_interest) - 4*sd(peak10_interest)))
# dfsub %>% filter(peak10_interest > (mean(peak10_interest) + 4*sd(peak10_interest)) | peak10
_interest < (mean(peak10_interest) - 4*sd(peak10_interest)))  # 2 ouliers are from the same p
articipant as before

# model including outliers
m1_ipeak <- update(m0i, formula. = ~ . + peak10_interest,
                    prior = c(prior(normal(0, 10), class = b),
                              prior(cauchy(0, 10), class = sd)),
                              newdata = dfsub,
                              save_all_pars = T)
summary(m1_ipeak)
```

```
##  Family: cumulative 
##   Links: mu = cloglog; disc = identity 
## Formula: interested_post ~ (1 | participant) + (1 | text) + peak10_interest 
##    Data: dfsub (Number of observations: 205) 
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
## 
## Group-Level Effects: 
## ~participant (Number of levels: 103) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.54      0.27     1.03     2.12       1773 1.00
## 
## ~text (Number of levels: 6) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.74      0.45     0.25     1.83       2392 1.00
## 
## Population-Level Effects: 
##                 Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]       -5.60      0.83    -7.39    -4.13       4087 1.00
## Intercept[2]       -3.11      0.51    -4.19    -2.18       3378 1.00
## Intercept[3]       -1.58      0.43    -2.48    -0.77       3579 1.00
## Intercept[4]        0.98      0.42     0.21     1.85       3178 1.00
## peak10_interest    -0.21      0.17    -0.55     0.13       3982 1.00
## 
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample 
## is a crude measure of effective sample size, and Rhat is the potential 
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

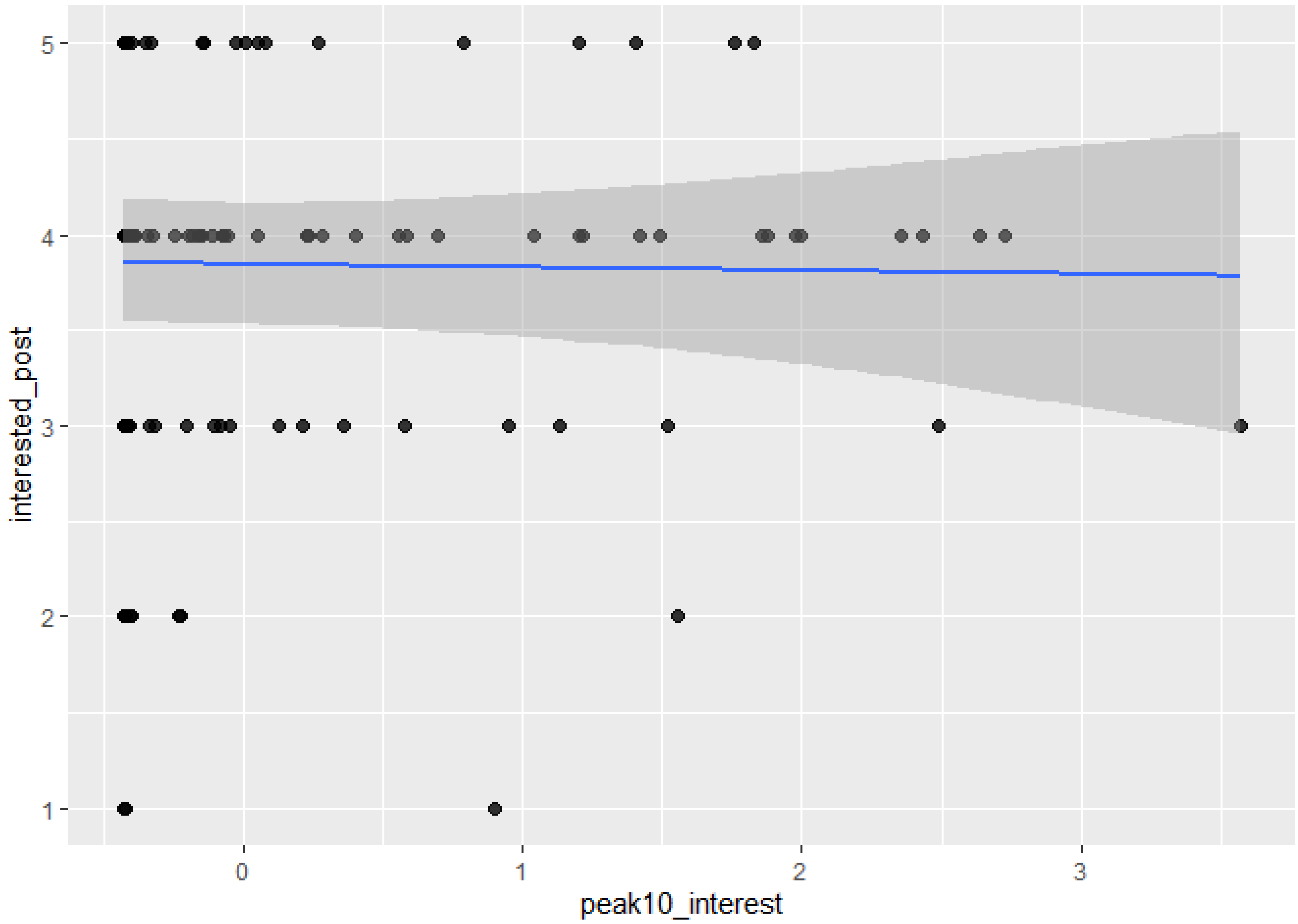
```
# model without outliers
m1_ipeak_out <- update(m0i, formula. = ~ . + peak10_interest,
                    prior = c(prior(normal(0, 10), class = b),
                              prior(cauchy(0, 10), class = sd)),
                              newdata = dfsub_outpeak,
                              save_all_pars = T)

## model parameter
summary(m1_ipeak_out)
```

```
##  Family: cumulative
##   Links: mu = cloglog; disc = identity
## Formula: interested_post ~ (1 | participant) + (1 | text) + peak10_interest
##    Data: dfsub_outpeak (Number of observations: 203)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~participant (Number of levels: 102)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.51      0.28     1.01     2.08       1615 1.00
##
## ~text (Number of levels: 6)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.72      0.40     0.23     1.81       2618 1.00
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]       -5.50      0.80    -7.20    -4.06       4020 1.00
## Intercept[2]       -3.10      0.50    -4.15    -2.19       3856 1.00
## Intercept[3]       -1.60      0.43    -2.45    -0.79       4200 1.00
## Intercept[4]        0.96      0.42     0.20     1.85       4070 1.00
## peak10_interest    -0.04      0.22    -0.48     0.41       4189 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# plots
plot(marginal_effects(m1_ipeak,"peak10_interest", categorical = F), points = T, point_args =
 c(alpha = 0.8))
```



```
plot(marginal_effects(m1_ipeak_out,"peak10_interest", categorical = F), points = T, point_arg
s = c(alpha = 0.8))
```



```
# define final model
m1_peak <- m1_ipeak_out
```

3. BOREDOM

BOREDOM restricted model

```
# complete cases only (drop NAs)
dfsubb <- df %>% select(participant, bored_post, mean_boredom, sd_boredom, peak10_boredom, te
xt) %>% drop_na()

# restricted model
m0b_cloglog <- brm(
          bored_post ~ 1 + (1|participant) + (1|text),
          family = cumulative("cloglog"),
          prior = prior(cauchy(0, 10), class = sd),
          iter = 4000, warmup = 2000, chains = 4, cores = 4,
          control = list(adapt_delta = 0.99),
          inits = 0,
          data = dfsubb,
          save_all_pars = T)

summary(m0b_cloglog)
```

```
##  Family: cumulative 
##   Links: mu = cloglog; disc = identity 
## Formula: bored_post ~ 1 + (1 | participant) + (1 | text) 
##    Data: dfsubb (Number of observations: 204) 
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
## 
## Group-Level Effects: 
## ~participant (Number of levels: 102) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.49      0.36     0.86     2.29       1554 1.00
## 
## ~text (Number of levels: 6) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.41      0.31     0.03     1.17       2204 1.00
## 
## Population-Level Effects: 
##              Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]     0.45      0.31    -0.12     1.13       2614 1.00
## Intercept[2]     1.76      0.41     1.05     2.67       2019 1.00
## Intercept[3]     2.95      0.59     1.93     4.24       2034 1.00
## Intercept[4]     3.53      0.73     2.30     5.16       2163 1.00
## 
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample 
## is a crude measure of effective sample size, and Rhat is the potential 
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# other link functions
m0b_logit <-  update(m0b_cloglog,
                    family = cumulative("logit"))

m0b_probit <-  update(m0b_cloglog,
                      family = cumulative("probit"))

# compare different link functions
m0b_logit <- add_criterion(m0b_logit,"loo")
m0b_probit <- add_criterion(m0b_probit,"loo")
m0b_cloglog <- add_criterion(m0b_cloglog ,"loo")
print(loo_compare(m0b_logit, m0b_probit, m0b_cloglog, criterion="loo"), simplify = F)
```

```
##             elpd_diff se_diff elpd_loo se_elpd_loo p_loo  se_p_loo looic
## m0b_cloglog    0.0       0.0  -157.5     11.7        52.1    5.2    315.0
## m0b_probit    -8.6       1.8  -166.1     13.1        56.6    5.6    332.2
## m0b_logit    -12.4       2.6  -170.0     13.6        61.3    5.9    339.9
##             se_looic
## m0b_cloglog   23.4
## m0b_probit    26.2
## m0b_logit     27.2
```

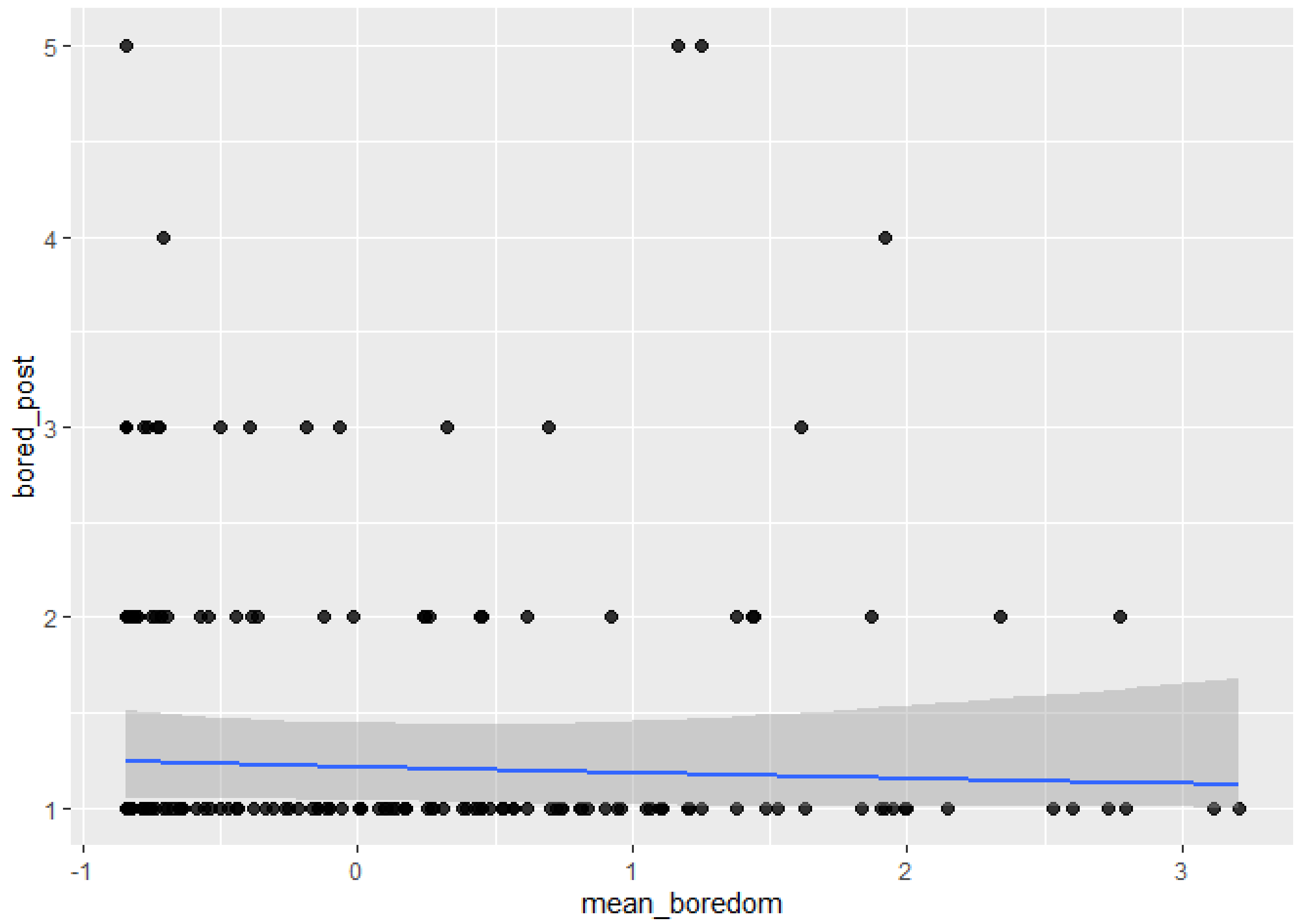
```
# chosen response distribution (link function)
m0b <- m0b_cloglog
```

BOREDOM mean

```
# full model including FaceReader's estimate as predictor
m1_bmean <-  update(m0b, formula. = ~ . + mean_boredom,
                    prior = c(prior(normal(0, 10), class = b),
                              prior(cauchy(0, 10), class = sd)),
                    newdata = dfsubb,
                    save_all_pars = T)

## model parameter
summary(m1_bmean)
```

```
##  Family: cumulative 
##   Links: mu = cloglog; disc = identity 
## Formula: bored_post ~ (1 | participant) + (1 | text) + mean_boredom 
##    Data: dfsubb (Number of observations: 204) 
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
## 
## Group-Level Effects: 
## ~participant (Number of levels: 102) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.56      0.38     0.90     2.41       1451 1.00
## 
## ~text (Number of levels: 6) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.44      0.33     0.03     1.27       2363 1.00
## 
## Population-Level Effects: 
##              Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]     0.48      0.33    -0.11     1.20       2655 1.00
## Intercept[2]     1.83      0.43     1.11     2.81       2025 1.00
## Intercept[3]     3.03      0.61     2.00     4.42       1852 1.00
## Intercept[4]     3.64      0.76     2.36     5.41       1995 1.00
## mean_boredom    -0.10      0.19    -0.50     0.26       3101 1.00
## 
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample 
## is a crude measure of effective sample size, and Rhat is the potential 
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# plots
plot(marginal_effects(m1_bmean, "mean_boredom", categorical = F), points = T, point_args = c(alpha = 0.8))
```

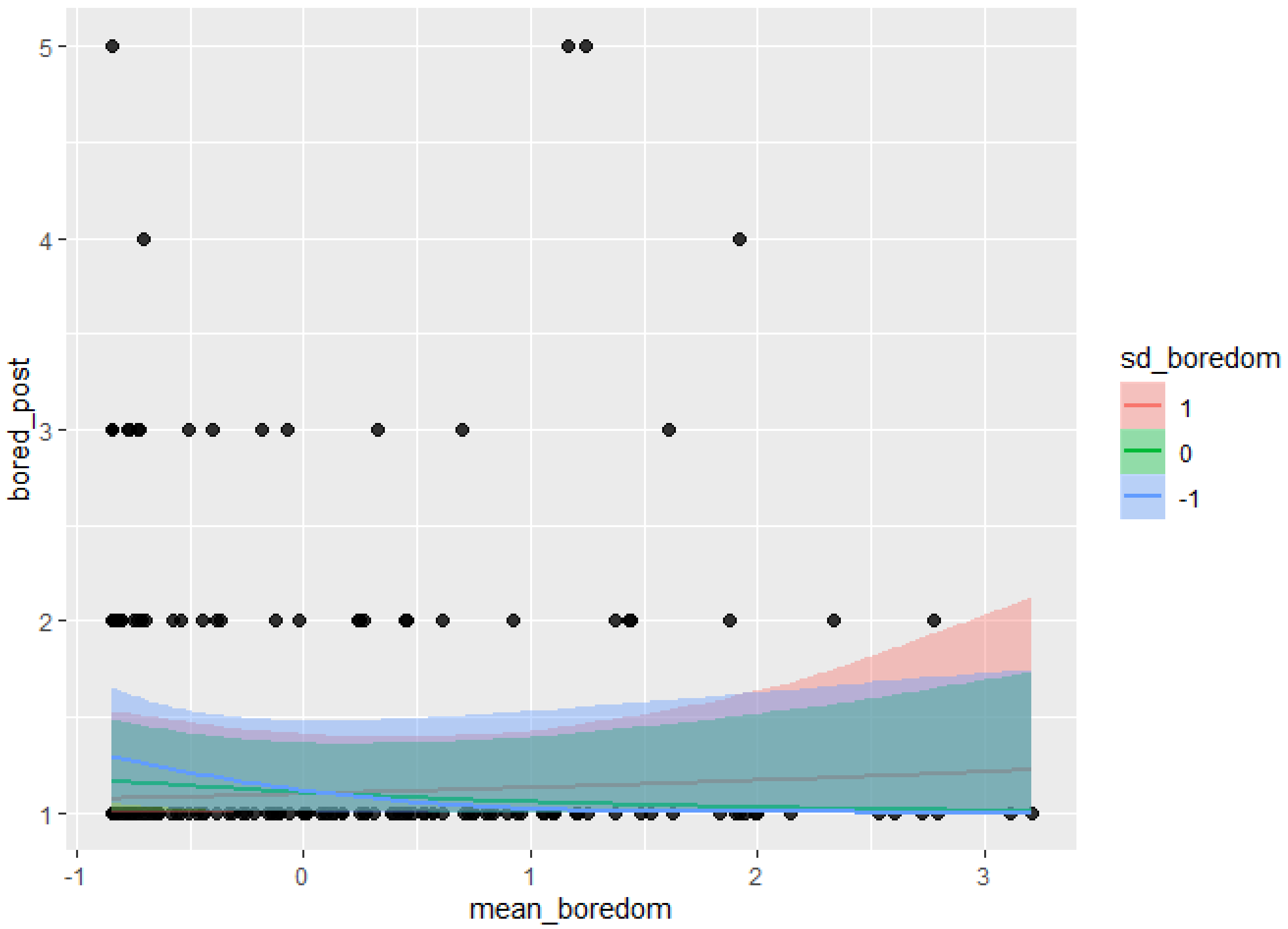


BOREDOM mean*SD

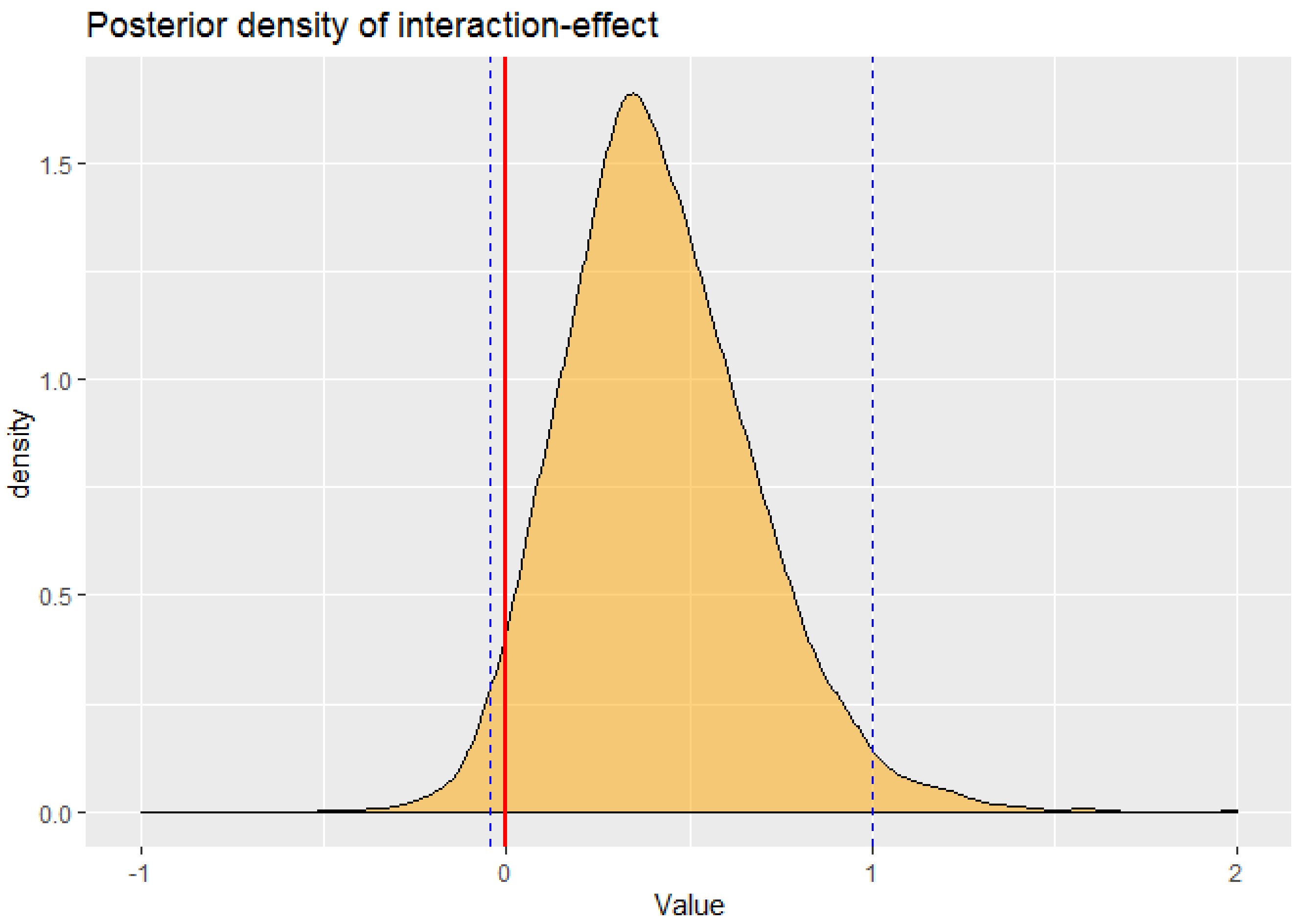
```
# full model including FaceReader's estimates as predictor
m1_bmeanxsd <-  update(m0b, formula. = ~ . + mean_boredom*sd_boredom,
                    prior = c(prior(normal(0, 10), class = b),
                              prior(cauchy(0, 10), class = sd)),
                    newdata = dfsubb,
                    save_all_pars = T)


## model indicators
summary(m1_bmeanxsd)
```

```
##  Family: cumulative
##   Links: mu = cloglog; disc = identity
## Formula: bored_post ~ (1 | participant) + (1 | text) + mean_boredom + sd_boredom + mean_bo
redom:sd_boredom
##    Data: dfsubb (Number of observations: 204)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~participant (Number of levels: 102)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.77      0.45     1.03     2.80       1360 1.00
##
## ~text (Number of levels: 6)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.49      0.42     0.04     1.42       1087 1.00
##
## Population-Level Effects:
##                         Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## Intercept[1]                0.87      0.47     0.07     1.90       1692
## Intercept[2]                2.31      0.60     1.33     3.62       1469
## Intercept[3]                3.61      0.80     2.30     5.37       1496
## Intercept[4]                4.30      0.96     2.70     6.42       1578
## mean_boredom               -0.28      0.38    -1.12     0.38       2566
## sd_boredom                 -0.04      0.32    -0.66     0.62       3397
## mean_boredom:sd_boredom     0.42      0.27    -0.04     1.00       2599
##                         Rhat
## Intercept[1]            1.00
## Intercept[2]            1.01
## Intercept[3]            1.01
## Intercept[4]            1.00
## mean_boredom            1.00
## sd_boredom              1.00
## mean_boredom:sd_boredom 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# plots
plot(marginal_effects(m1_bmeanxsd,"mean_boredom:sd_boredom", categorical = F), points = T, po
int_args = c(alpha = 0.8))
```



```
## plot densities and CIs of interaction-effect
m1_bmeanxsd_ggs <- ggs(m1_bmeanxsd) # transforms the brms output into a longformat tibble (used to make different types of plots)
ggplot(filter(m1_bmeanxsd_ggs, Parameter == "b_mean_boredom:sd_boredom", Iteration>1000), aes(x=value)) +
  geom_density(fill = "orange", alpha = .5) + geom_vline(xintercept = 0, col="red", size=1) +
  scale_x_continuous(name="Value", limits=c(-1, 2)) +
  labs(title="Posterior density of interaction-effect") +
  geom_vline(xintercept = summary(m1_bmeanxsd)$fixed[7,3:4], col="blue", linetype=2) # 95% CrI
```



```
# 10-fold cross validation: interaction model compared to restricted model
m0b <- add_criterion(m0b, criterion =  "kfold", folds = "grouped", group = "participant")
m1_bmeanxsd <- add_criterion(m1_bmeanxsd, criterion = "kfold", folds = "grouped", group = "pa
rticipant")
print(loo_compare(m0b, m1_bmeanxsd, criterion = "kfold"), simplify = T) ## Estimating out-of
 sample predictions (via 10-fold cross validation) of the interaction model, compared to a mo
del with no predictors yielded better results for the model without the interaction. Accordin
gly, we consider this potential interaction effect as irrelevant.
```

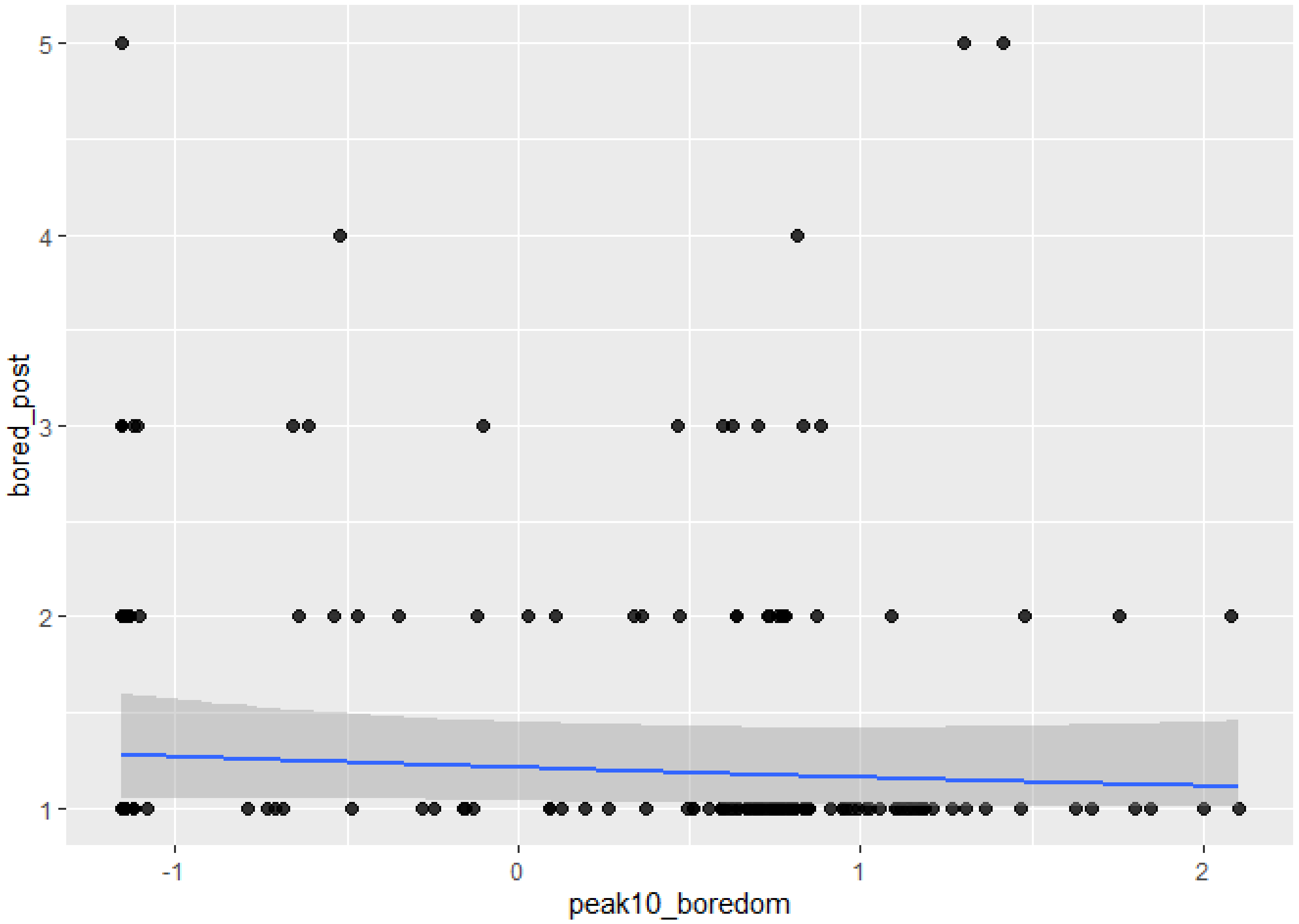
```
##             elpd_diff se_diff
## m0b          0.0       0.0   
## m1_bmeanxsd -0.8       2.2   
```

BOREDOM mean of peaks

```
m1_bpeak <-  update(m0b, formula. = ~ . + peak10_boredom,
                    prior = c(prior(normal(0, 10), class = b),
                              prior(cauchy(0, 10), class = sd)),
                    newdata = dfsubb,
                    save_all_pars = T)

## model parameter
summary(m1_bpeak)
```

```
##  Family: cumulative 
##   Links: mu = cloglog; disc = identity 
## Formula: bored_post ~ (1 | participant) + (1 | text) + peak10_boredom 
##    Data: dfsubb (Number of observations: 204) 
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
## 
## Group-Level Effects: 
## ~participant (Number of levels: 102) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.56      0.38     0.91     2.44       1494 1.00
## 
## ~text (Number of levels: 6) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.43      0.32     0.03     1.23       2184 1.00
## 
## Population-Level Effects: 
##                Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]       0.49      0.33    -0.11     1.22       3405 1.00
## Intercept[2]       1.84      0.44     1.09     2.83       2350 1.00
## Intercept[3]       3.05      0.62     2.00     4.44       2128 1.00
## Intercept[4]       3.66      0.77     2.37     5.37       2257 1.00
## peak10_boredom    -0.16      0.19    -0.56     0.19       4057 1.00
## 
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample 
## is a crude measure of effective sample size, and Rhat is the potential 
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# plots
plot(marginal_effects(m1_bpeak,"peak10_boredom", categorical = F), points = T, point_args = c
(alpha = 0.8))
```



4. VALENCE

stronger priors for valence, as issues with convergence. ## VALENCE restricted model

```
## select complete cases of relevant variables
dfsubv <- df %>% select(participant, valence_post, mean_valence, sd_valence, peak10_valence_p
os, peak10_valence_neg, text) %>% drop_na()

# restricted model
## probit-model
m0v_probit <- brm(
          valence_post ~ 1 + (1|participant) + (1|text),
          family = cumulative("probit"),
          prior = c(prior(normal(0, 1), class = Intercept),
                    prior(cauchy(0, 1), class = sd)),
          iter = 4000, warmup = 2000, chains = 4, cores = 4,
          control = list(adapt_delta = 0.999, max_treedepth = 15),
          inits = 0,
          data = dfsubv,
          save_all_pars = T)
summary(m0v_probit)
```

```
##  Family: cumulative 
##   Links: mu = probit; disc = identity 
## Formula: valence_post ~ 1 + (1 | participant) + (1 | text) 
##    Data: dfsubv (Number of observations: 193) 
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
## 
## Group-Level Effects: 
## ~participant (Number of levels: 97) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.31      0.20     0.94     1.73       1496 1.00
## 
## ~text (Number of levels: 6) 
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.55      0.57     0.01     1.98        761 1.00
## 
## Population-Level Effects: 
##              Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]    -3.12      0.48    -4.07    -2.15       2105 1.00
## Intercept[2]    -2.82      0.43    -3.59    -1.87       1674 1.00
## Intercept[3]    -2.65      0.42    -3.36    -1.72       1437 1.00
## Intercept[4]    -2.35      0.42    -3.02    -1.38       1196 1.00
## Intercept[5]    -1.30      0.43    -1.92    -0.26       1005 1.00
## Intercept[6]    -0.18      0.45    -0.79     0.92        912 1.00
## Intercept[7]     1.33      0.49     0.67     2.51        876 1.00
## Intercept[8]     2.85      0.55     2.01     4.12        921 1.00
## 
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample 
## is a crude measure of effective sample size, and Rhat is the potential 
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

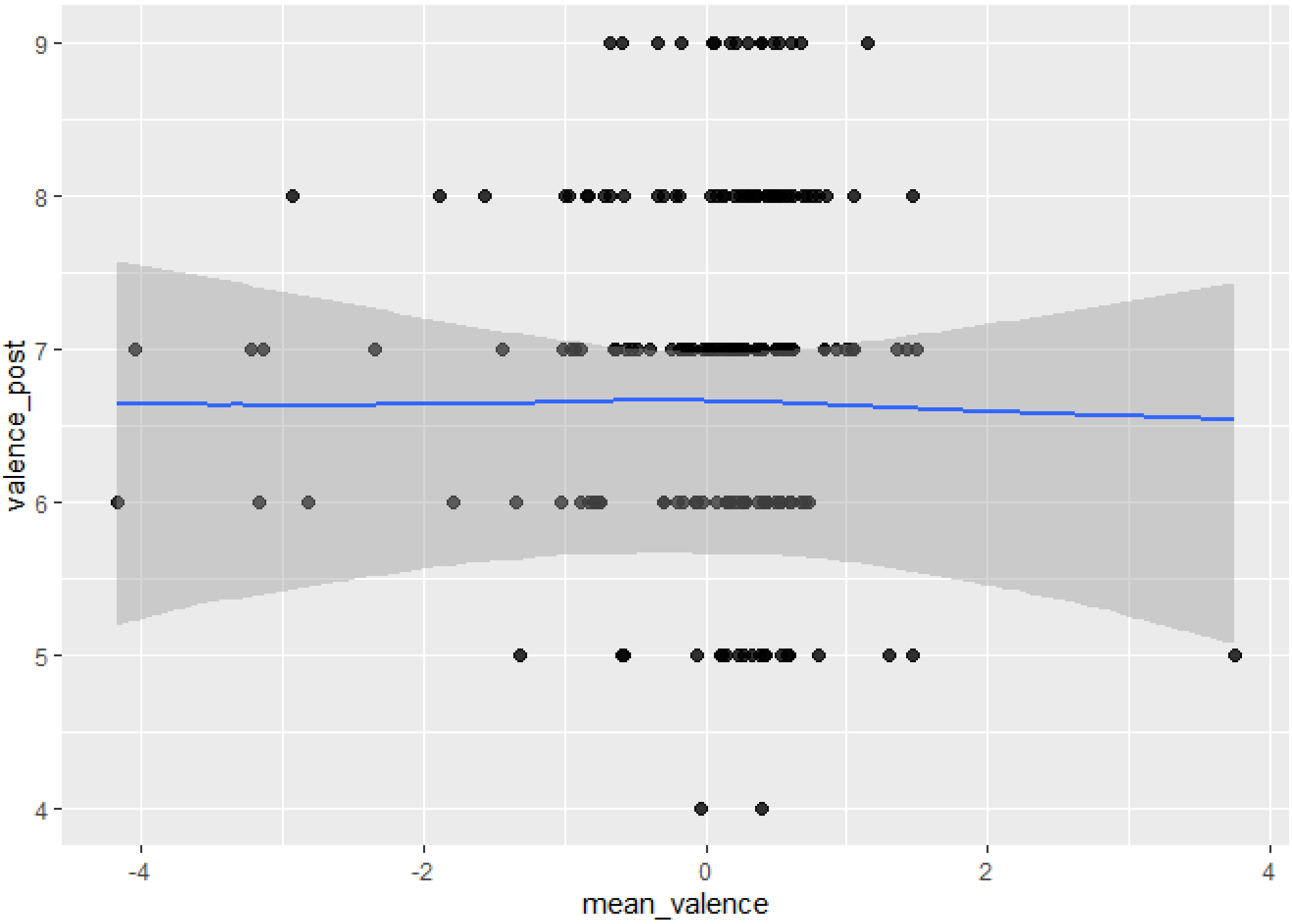
```
# chosen response distribution (link function)
m0v <- m0v_probit
```

VALENCE mean

```
# full model including FaceReader's estimate as predictor
m1_vmean <-  update(m0v, formula. = ~ . + mean_valence,
                    prior = c(prior(normal(0, 1), class = Intercept),
                              prior(normal(0, 1), class = b),
                              prior(cauchy(0, 1), class = sd)),
                    newdata = dfsubv,
                    save_all_pars = T,
                    seed = 19) # for reproducibility (on the same machine) - to avoid "Stan model x does not contain samples." which sometimes occured

## model parameter
summary(m1_vmean)
```

```
##  Family: cumulative
##   Links: mu = probit; disc = identity
## Formula: valence_post ~ (1 | participant) + (1 | text) + mean_valence
##    Data: dfsubv (Number of observations: 193)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~participant (Number of levels: 97)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.34      0.20     0.96     1.75       1812 1.00
##
## ~text (Number of levels: 6)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.55      0.58     0.01     2.07        686 1.00
##
## Population-Level Effects:
##              Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]    -3.13      0.48    -4.08    -2.14       2212 1.00
## Intercept[2]    -2.83      0.43    -3.61    -1.89       1597 1.00
## Intercept[3]    -2.67      0.42    -3.40    -1.75       1400 1.00
## Intercept[4]    -2.36      0.42    -3.04    -1.42       1152 1.00
## Intercept[5]    -1.31      0.43    -1.92    -0.27        886 1.00
## Intercept[6]    -0.19      0.45    -0.78     0.92        797 1.00
## Intercept[7]     1.34      0.49     0.67     2.53        797 1.00
## Intercept[8]     2.86      0.56     2.01     4.16        918 1.00
## mean_valence    -0.02      0.15    -0.31     0.27       3917 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# plots
plot(marginal_effects(m1_vmean, "mean_valence", categorical = F), points = T, point_args = c
(alpha = 0.8))
```

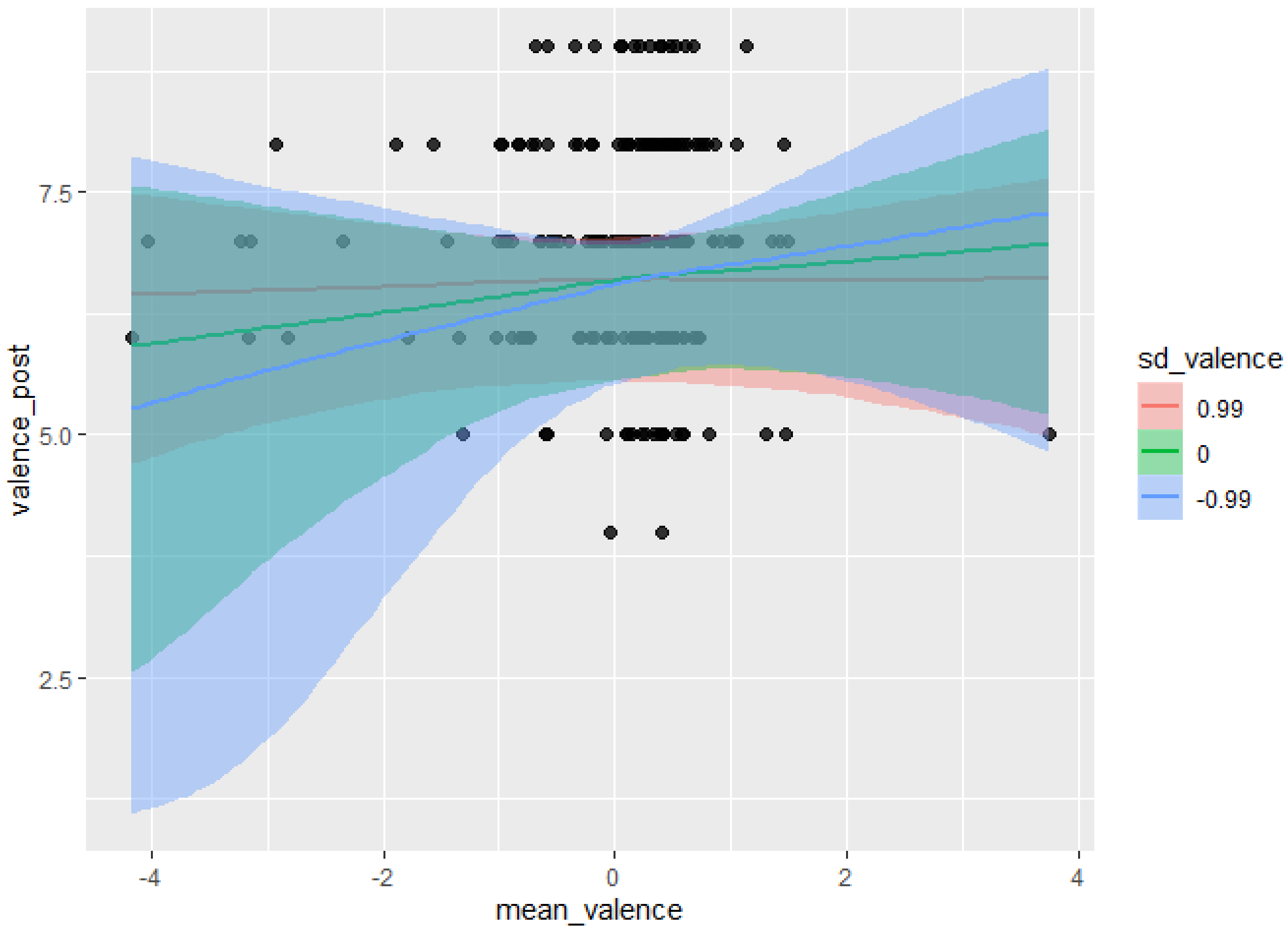


VALENCE mean*SD

```
# full model including FaceReader's estimates as predictor
m1_vmeanxsd <-  update(m0v, formula. = ~ . + mean_valence*sd_valence,
                    prior = c(prior(normal(0, 1), class = Intercept),
                              prior(normal(0, 1), class = b),
                              prior(cauchy(0, 1), class = sd)),
                    control = list(adapt_delta = 0.999, max_treedepth = 15),
                    inits = 0,
                    newdata = dfsubv,
                    save_all_pars = T,
                    seed = 21) # for reproducibility (on the same machine) - to avoid diverge
nt transitions which sometimes occured


## model parameter
summary(m1_vmeanxsd)
```

```
##  Family: cumulative
##   Links: mu = probit; disc = identity
## Formula: valence_post ~ (1 | participant) + (1 | text) + mean_valence + sd_valence + mean_
valence:sd_valence
##    Data: dfsubv (Number of observations: 193)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~participant (Number of levels: 97)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.36      0.21     0.98     1.78       1594 1.00
##
## ~text (Number of levels: 6)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.60      0.62     0.01     2.15        704 1.01
##
## Population-Level Effects:
##                         Estimate Est.Error l-95% CI u-95% CI Eff.Sample
## Intercept[1]               -3.06      0.49    -4.05    -2.08       2020
## Intercept[2]               -2.77      0.44    -3.55    -1.83       1473
## Intercept[3]               -2.60      0.43    -3.34    -1.65       1309
## Intercept[4]               -2.29      0.44    -3.01    -1.30       1119
## Intercept[5]               -1.23      0.46    -1.91    -0.18        866
## Intercept[6]               -0.10      0.48    -0.76     1.03        790
## Intercept[7]                1.44      0.53     0.71     2.66        760
## Intercept[8]                2.99      0.59     2.07     4.32        827
## mean_valence                0.16      0.25    -0.32     0.65       2869
## sd_valence                  0.03      0.14    -0.25     0.31       4093
## mean_valence:sd_valence    -0.14      0.15    -0.44     0.15       3355
##                         Rhat
## Intercept[1]            1.00
## Intercept[2]            1.00
## Intercept[3]            1.00
## Intercept[4]            1.00
## Intercept[5]            1.01
## Intercept[6]            1.01
## Intercept[7]            1.01
## Intercept[8]            1.01
## mean_valence            1.00
## sd_valence              1.00
## mean_valence:sd_valence 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# plots
plot(marginal_effects(m1_vmeanxsd,"mean_valence:sd_valence"), points = T, point_args = c(alph
a = 0.8))
```

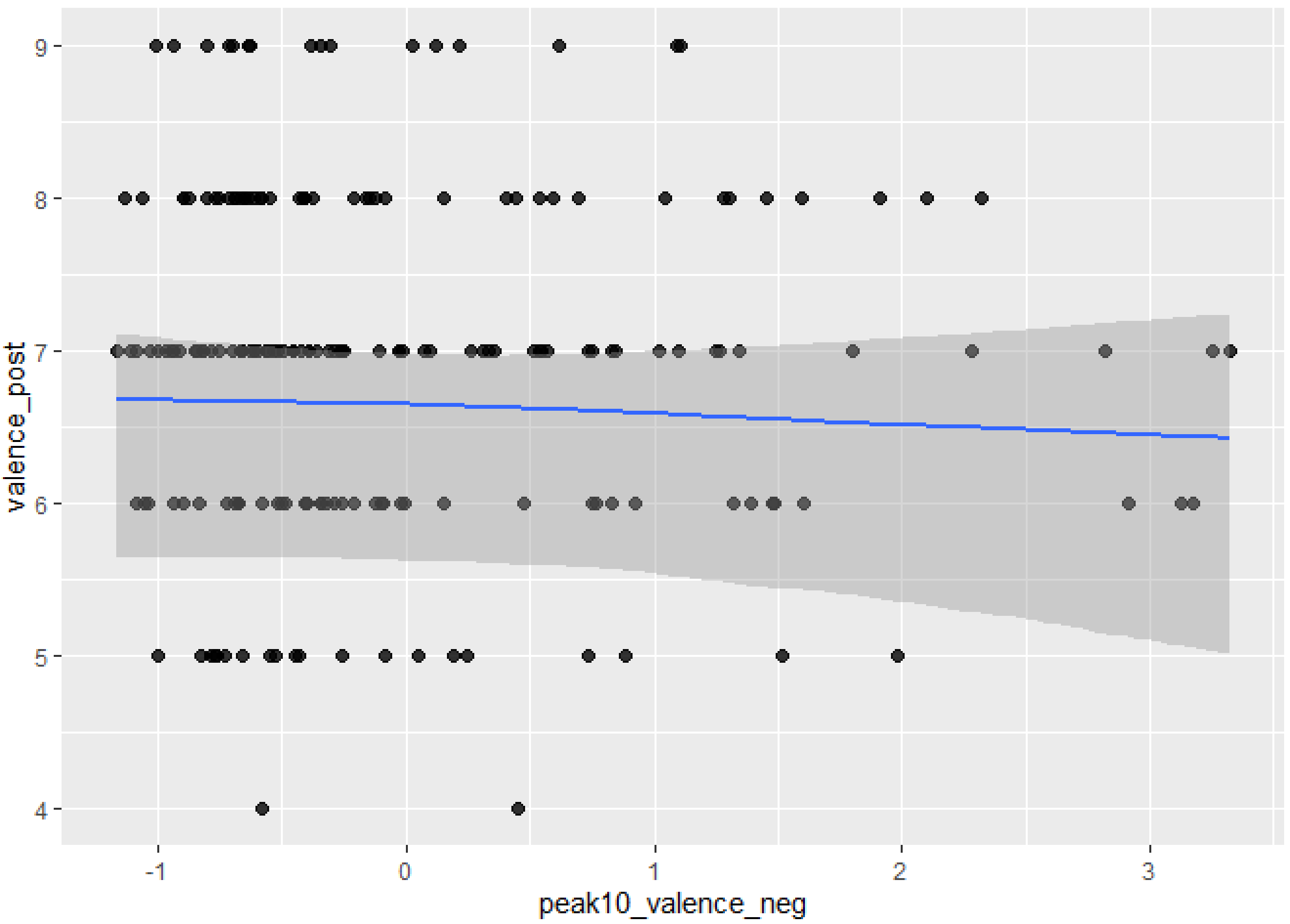


VALENCE mean of peaks

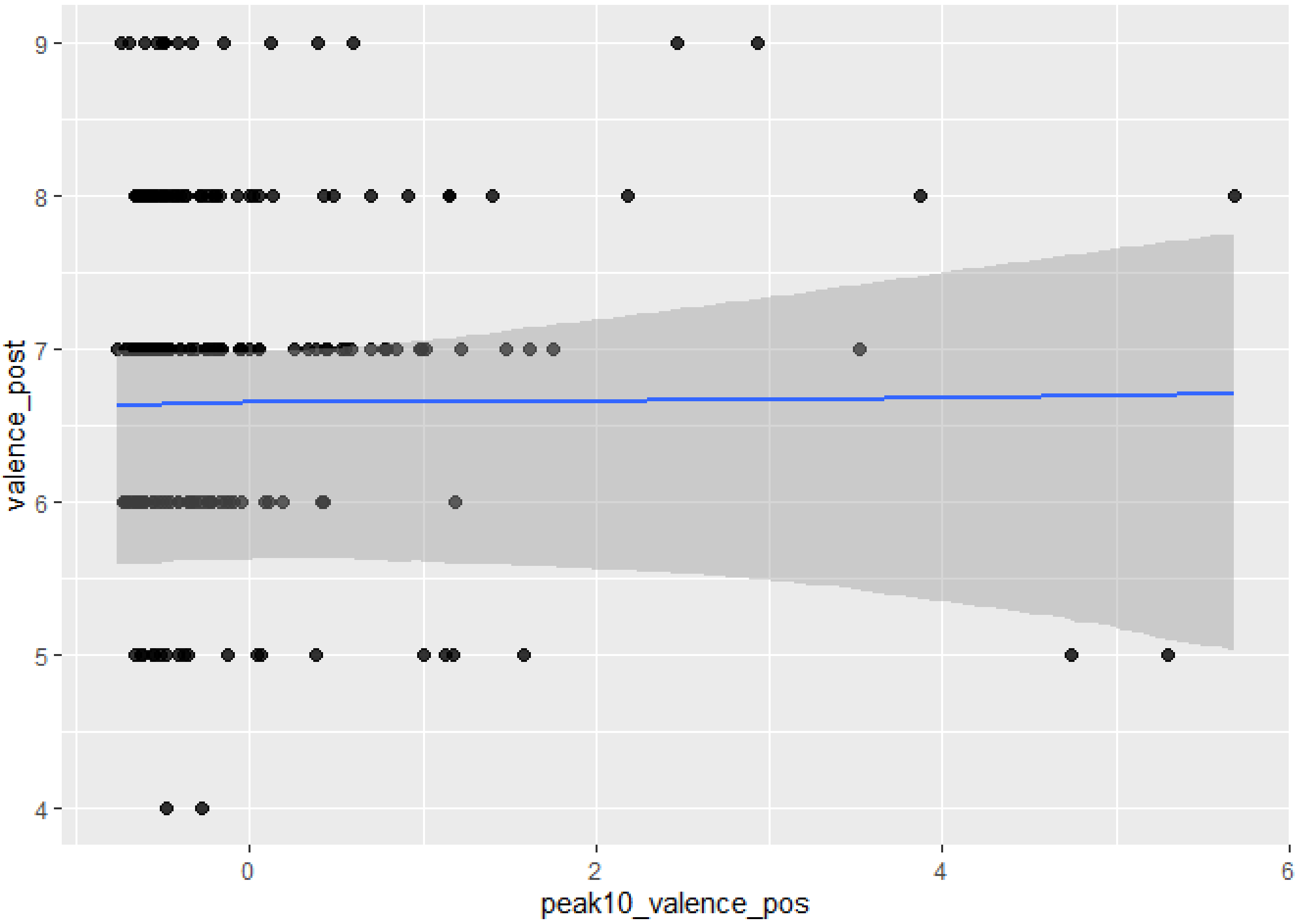
```
m1_vpeak <-  update(m0v, formula. = ~ . + peak10_valence_neg + peak10_valence_pos,
                    prior = c(prior(normal(0, 1), class = Intercept),
                              prior(normal(0, 1), class = b),
                              prior(cauchy(0, 1), class = sd)),
                    newdata = dfsubv,
                    save_all_pars = T)

## model parameter
summary(m1_vpeak)
```

```
##  Family: cumulative
##   Links: mu = probit; disc = identity
## Formula: valence_post ~ (1 | participant) + (1 | text) + peak10_valence_neg + peak10_valen
ce_pos
##    Data: dfsubv (Number of observations: 193)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~participant (Number of levels: 97)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.34      0.20     0.97     1.79       1666 1.00
##
## ~text (Number of levels: 6)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.56      0.58     0.02     2.05        789 1.00
##
## Population-Level Effects:
##                    Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]          -3.13      0.49    -4.09    -2.16       2249 1.00
## Intercept[2]          -2.82      0.43    -3.60    -1.89       1684 1.00
## Intercept[3]          -2.65      0.43    -3.40    -1.71       1541 1.00
## Intercept[4]          -2.35      0.43    -3.05    -1.37       1277 1.00
## Intercept[5]          -1.30      0.45    -1.94    -0.21       1035 1.00
## Intercept[6]          -0.18      0.46    -0.79     0.95        961 1.00
## Intercept[7]           1.36      0.50     0.69     2.56        931 1.00
## Intercept[8]           2.89      0.56     2.04     4.21       1061 1.00
## peak10_valence_neg    -0.06      0.14    -0.34     0.23       3692 1.00
## peak10_valence_pos     0.03      0.12    -0.22     0.27       4991 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# plots
plot(marginal_effects(m1_vpeak,"peak10_valence_neg", categorical = F), points = T, point_args
= c(alpha = 0.8))
```



```
plot(marginal_effects(m1_vpeak,"peak10_valence_pos", categorical = F), points = T, point_args
= c(alpha = 0.8))
```


valence_post
peak10_valence_pos