

S4: behavioural, HGF-based analysis with brms

I. S. Plank

2025-11-06

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Some general settings | 1 |
| 1.2 | Package versions | 2 |
| 1.3 | Preparation | 2 |
| 2 | H3a: phasic volatility | 4 |
| 2.1 | Model setup | 4 |
| 2.2 | Posterior predictive checks | 4 |
| 2.3 | Inferences | 6 |
| 3 | H3b: third level tonic volatility | 8 |
| 3.1 | Model setup | 8 |
| 3.2 | Posterior predictive checks | 8 |
| 3.3 | Inferences | 10 |
| 4 | Exploration of Bernoulli model with HGF parameters | 12 |
| 4.1 | Model setup | 12 |
| 4.2 | Posterior predictive checks | 12 |
| 4.3 | Inferences | 14 |
| 5 | Plots for all HGF parameters | 16 |
| 6 | Learning rate update - volatile to stable | 17 |
| 6.1 | Model setup | 17 |
| 6.2 | Posterior predictive checks | 18 |
| 6.3 | Inferences | 20 |
| 6.4 | Bayesian ANOVA | 24 |
| 7 | Plots for learning rate updates | 25 |
| 8 | Similar plots to Lawson et al. (2017) for comparison | 26 |

1 Introduction

This R Markdown script analyses data from the PAL (probabilistic associative learning) task of the EMBA project. HGF parameters were extracted based on the subject-specific reaction times beforehand in MATLAB.

1.1 Some general settings

```
# number of simulations
nsim = 250

# set number of iterations and warmup for models
iter = 3000
warm = 1000
```

```
# set the seed
set.seed(2468)
```

1.2 Package versions

The following packages are used in this RMarkdown file:

```
## [1] "R version 4.5.1 (2025-06-13)"
## [1] "knitr version 1.50"
## [1] "ggplot2 version 4.0.0"
## [1] "brms version 2.22.0"
## [1] "designr version 0.1.13"
## [1] "bridgesampling version 1.1.2"
## [1] "tidyverse version 2.0.0"
## [1] "ggpubr version 0.6.1"
## [1] "ggrain version 0.0.4"
## [1] "bayesplot version 1.13.0"
## [1] "SBC version 0.3.0.9000"
## [1] "rstatix version 0.7.2"
## [1] "BayesFactor version 0.9.12.4.7"
## [1] "effectsize version 1.0.1"
## [1] "bayestestR version 0.17.0"
```

1.3 Preparation

First, we load the parameters from the winning model.

```
# get HGF parameters
df.hgf = read_csv(file.path("HGF_results/main", "HGF-L17_results.csv")) %>%
  mutate_if(is.character, as.factor)

# get belief state trajectories
df.trj = read_csv(file.path("HGF_results/main", "HGF-L17_traj.csv"))

# extract the absolute changes in learning rate for the phases
df.upd = df.trj %>%
  select(subID, diagnosis, trl, alpha2, alpha3) %>%
  mutate(
    phase = case_when(
      trl < 73 ~ "pre",
      trl > 264 ~ "post",
      trl < 145 ~ "vol1",
      trl > 192 ~ "vol2"
    )
  ) %>%
  drop_na() %>%
  group_by(subID, diagnosis, phase) %>%
  summarise(
    alpha2 = median(alpha2),
    alpha3 = median(alpha3)
  ) %>%
  pivot_wider(names_from = phase, id_cols = c(subID, diagnosis), values_from = starts_with("alpha")) %>%
  group_by(subID, diagnosis) %>%
  summarise(
    alpha2_pre2vol = abs(alpha2_pre - alpha2_vol1),
    alpha2_vol2post = abs(alpha2_post - alpha2_vol2),
    alpha3_pre2vol = abs(alpha3_pre - alpha3_vol1),
    alpha3_vol2post = abs(alpha3_post - alpha3_vol2)
  ) %>%
  pivot_longer(cols = starts_with("alpha")) %>%
```

```

separate(name, into = c("level", "change")) %>%
mutate_if(is.character, as.factor)

# check whether there are LME differences between the diagnostic groups
kable(df.hgf %>% group_by(diagnosis) %>% shapiro_test(LME)) # all normally distributed

```

| diagnosis | variable | statistic | p |
|-----------|----------|-----------|-----------|
| ASD | LME | 0.9534482 | 0.3690192 |
| COMP | LME | 0.9845883 | 0.9717432 |

```

if (!file.exists(file.path(brms_dir, "aov_lme.rds"))) {
  aov = anovaBF(LME ~ diagnosis, data = df.hgf)
  saveRDS(aov, file.path(brms_dir, "aov_lme.rds"))
} else {
  aov = readRDS(file.path(brms_dir, "aov_lme.rds"))
}

aov@bayesFactor

```

```

##           bf      error            time       code
## diagnosis -1.04496 5.810305e-05 Wed Oct 15 12:08:23 2025 144d96dd874c4

```

There is anecdotal evidence against a difference in LME between diagnostic groups. This suggests that the HGF model fit comparably well to the subjects of the different groups. Therefore, we move on to analyse its parameters.

Following Lawson et al. (2017), the following observation model was used:

$$\log RT = \beta_0 + \beta_1 \times surprise_{stimulus} + \beta_2 \times uncertainty_{stimulus} + \beta_3 \times uncertainty_{cue-outcome} + \beta_4 \times volatility_{phasic}$$

Next, we use sum contrast coding for all of our categorical predictors.

```

# set and print the contrasts
contrasts(df.hgf$diagnosis) = contr.sum(2)
contrasts(df.hgf$diagnosis)

```

```

##      [,1]
## ASD     1
## COMP   -1

contrasts(df.upd$diagnosis) = contr.sum(2)
contrasts(df.upd$diagnosis)

```

```

##      [,1]
## ASD     1
## COMP   -1

contrasts(df.upd$change) = contr.sum(2)
contrasts(df.upd$change)

```

```

##      [,1]
## pre2vol    1
## vol2post   -1

contrasts(df.upd$level) = contr.sum(2)
contrasts(df.upd$level)

```

```

##      [,1]
## alpha2    1
## alpha3   -1

```

2 H3a: phasic volatility

2.1 Model setup

```
# model formula
f.vol = brms::bf( be4 ~ diagnosis )

# set weakly informative priors
priors = c(
  prior(normal(0, 4), class = Intercept),
  prior(normal(0, 0.50), class = sigma),
  prior(normal(0, 0.25), class = b)
)

# change Intercept based on empirical priors used in the HGF model
priors = priors %>%
  mutate(
    prior = if_else(
      class == "Intercept",
      gsub("\\\\(.*,", paste0("(", mean(df.hgf$be4mu), ", "), prior), prior),
      prior = if_else(
        class == "Intercept",
        gsub(".*\\\\)", paste0(" ", mean(df.hgf$be4sa), ")"), prior),
        prior)
  )

kable(priors)
```

| prior | class | coef | group | resp | dpar | nlpar | lb | ub | source |
|-----------------------|-----------|------|-------|------|------|-------|----|----|--------|
| normal(0.211, 0.4196) | Intercept | | | | | | NA | NA | user |
| normal(0, 0.5) | sigma | | | | | | NA | NA | user |
| normal(0, 0.25) | b | | | | | | NA | NA | user |

2.2 Posterior predictive checks

As the next step, we fit the model, check whether there are divergence or rhat issues, and then check whether the chains have converged.

```
# fit the model
m.vol = brm(f.vol, seed = 8822,
             df.hgf, prior = priors,
             iter = iter, warmup = warm,
             backend = "cmdstanr", threads = threading(t),
             file = file.path(brms_dir, "m_hgf_vol"),
             save_pars = save_pars(all = TRUE)
           )
rstan::check_hmc_diagnostics(m.vol$fit)

## 
## Divergences:
## 0 of 8000 iterations ended with a divergence.

## 
## Tree depth:
## 0 of 8000 iterations saturated the maximum tree depth of 10.

## 
## Energy:
## E-BFMI indicated no pathological behavior.
```

```

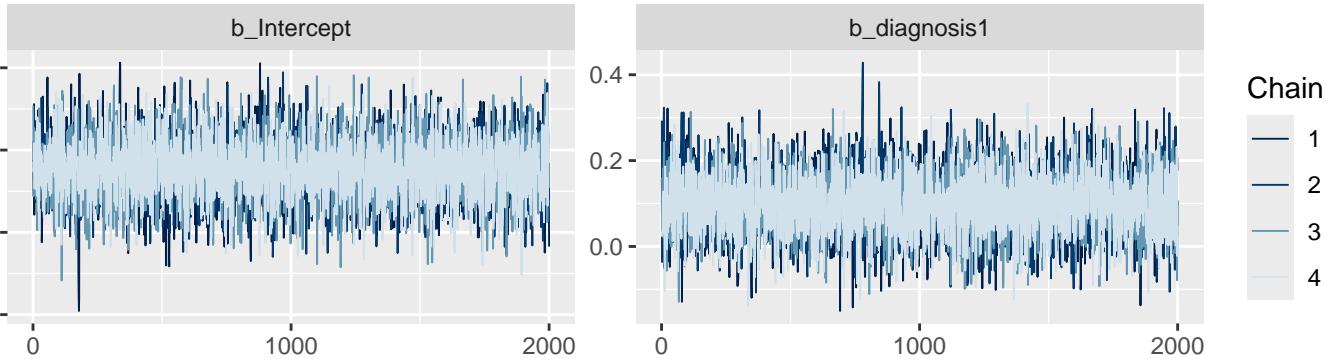
# check that rhats are below 1.01
sum(brms::rhat(m.vol) >= 1.01, na.rm = T)

## [1] 0

# check the trace plots
post.draws = as_draws_df(m.vol)
mcmc_trace(post.draws, regex_pars = "^\b_",
            facet_args = list(ncol = 2)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 3)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 3))

## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.

```



This model has no pathological behaviour with E-BFMI, no divergent sample and no rhats that are higher or equal to 1.01. Therefore, we go ahead and perform our posterior predictive checks.

```

# get posterior predictions
post.pred = posterior_predict(m.vol, ndraws = nsim)

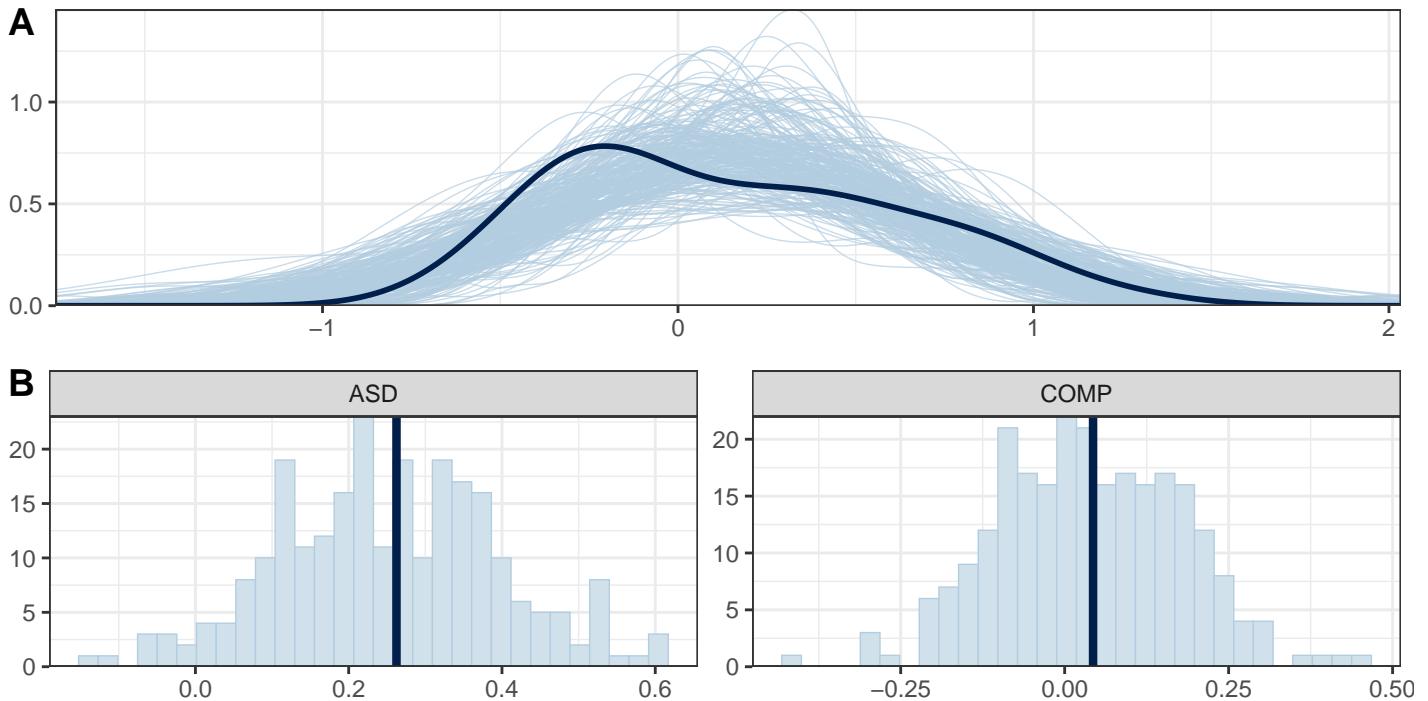
# check the fit of the predicted data compared to the real data
p1 = pp_check(m.vol, ndraws = nsim) +
  theme_bw() + theme(legend.position = "none")

# distributions of means compared to the real values per group
p2 = ppc_stat_grouped(df.hgf$be4, post.pred, df.hgf$diagnosis) +
  theme_bw() + theme(legend.position = "none")

p = ggarrange(p1, p2,
              nrow = 2, ncol = 1, labels = "AUTO")
annotate_figure(p, top = text_grob("Posterior predictive checks",
                                    face = "bold", size = 14))

```

Posterior predictive checks



The overall simulated data fits reasonably well, even though it does not reproduce the shape completely. The mean simulated data based on the model fits well with the real data.

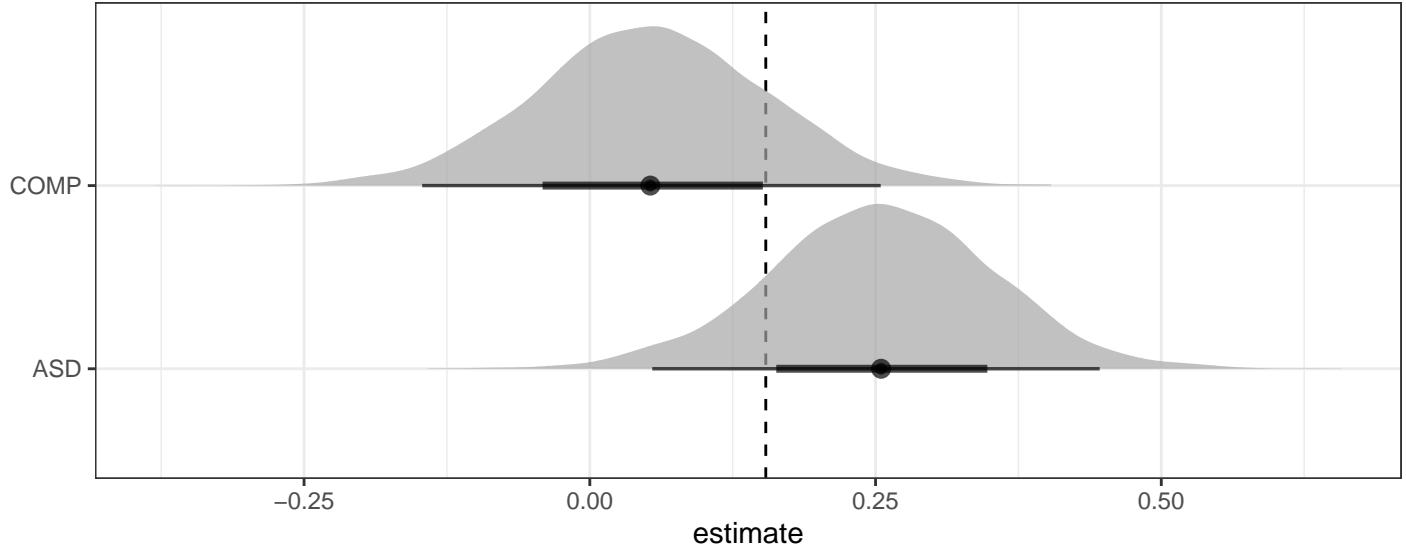
2.3 Inferences

Now that we are convinced that we can trust our model, we have a look at its estimate and use the hypothesis function to assess our hypotheses and perform explorative tests.

```
# print a summary
summary(m.vol)

## Family: gaussian
## Links: mu = identity; sigma = identity
## Formula: be4 ~ diagnosis
## Data: df.hgf (Number of observations: 44)
## Draws: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##        total post-warmup draws = 8000
##
## Regression Coefficients:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.15      0.07    0.01    0.30 1.00    6857    4868
## diagnosis1    0.10      0.07   -0.03    0.24 1.00    7164    5392
##
## Further Distributional Parameters:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma       0.48      0.05    0.39    0.60 1.00    6978    5752
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
# get the estimates and compute group comparisons
df.m.vol = post.draws %>%
  mutate(
    ASD      = b_Intercept + b_diagnosis1,
    COMP    = b_Intercept - b_diagnosis1
  )
```

```
# plot the posterior distributions
df.m.vol %>%
  select(ASD, COMP) %>%
  pivot_longer(cols = everything(), names_to = "coef", values_to = "estimate") %>%
  ggplot(aes(x = estimate, y = coef), fill = c_light) +
  geom_vline(xintercept = mean(df.m.vol$b_Intercept), linetype = 'dashed') +
  ggdist::stat_halfeye(alpha = 0.7) + ylab(NULL) + theme_bw() +
  theme(legend.position = "none")
```



```
# H3a: COMP < ASD
h3a = hypothesis(m.vol, "0 < diagnosis1")
h3a

## Hypothesis Tests for class b:
##          Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio
## 1 (0)-(diagnosis1) < 0     -0.1      0.07    -0.22      0.01     12.4
##   Post.Prob Star
## 1      0.93
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
equivalence_test(m.vol)
```

```
## # Test for Practical Equivalence
##
## ROPE: [-0.05 0.05]
##
## Parameter |      H0 | inside ROPE |      95% HDI
## -----
## Intercept | Undecided |    4.72 % | [9.46e-03, 0.30]
## diagnosis1 | Undecided |  20.59 % | [-0.03, 0.24]
```

```
# get effect sizes (Hedges, 2007)
df.effect = df.m.vol %>%
  mutate(
    group = 2*b_diagnosis1 / sigma
  )
kable(df.effect %>% select(group) %>%
  pivot_longer(cols = everything(), values_to = "estimate") %>%
```

```

group_by(name) %>%
summarise(
  ci.lo = lower_ci(estimate),
  mean = mean(estimate),
  ci.hi = upper_ci(estimate),
  interpret = interpret_cohens_d(mean)
)
)

```

| | name | ci.lo | mean | ci.hi | interpret |
|-------|------------|-----------|----------|-------|-----------|
| group | -0.1411793 | 0.4270947 | 1.015224 | small | |

estimate = -0.1 [-0.22, 0.01], posterior probability = 92.54%

3 H3b: third level tonic volatility

3.1 Model setup

```

# code for filenames
code = "PAL_om3"

# model formula
f.om3 = brms::bf( om3 ~ diagnosis )

# set weakly informative priors
priors = c(
  prior(normal(0, 4), class = Intercept),
  prior(normal(0, 0.50), class = sigma),
  prior(normal(0, 0.25), class = b)
)

# change Intercept based on empirical priors used in the HGF model
priors = priors %>%
  mutate(
    prior = if_else(
      class == "Intercept",
      gsub("\\\\(.*,", paste0("(, mean(df.hgf$om3mu), ", ), prior), prior),
      prior = if_else(
        class == "Intercept",
        gsub(" .*\\\\)", paste0(" ", mean(df.hgf$om3sa), ")"), prior), prior)
    )
  )

kable(priors)

```

| prior | class | coef | group | resp | dpar | nlnpar | lb | ub | source |
|-------------------------|-----------|------|-------|------|------|--------|----|----|--------|
| normal(-2.8218, 6.5532) | Intercept | | | | | | NA | NA | user |
| normal(0, 0.5) | sigma | | | | | | NA | NA | user |
| normal(0, 0.25) | b | | | | | | NA | NA | user |

3.2 Posterior predictive checks

As the next step, we fit the model, check whether there are divergence or rhat issues, and then check whether the chains have converged.

```

# fit the final model
m.om3 = brm(f.om3, seed = 2288,
             df.hgf, prior = priors,

```

```

    iter = iter, warmup = warm,
    backend = "cmdstanr", threads = threading(t),
    file = file.path(brms_dir, "m_hgf_om3"),
    save_pars = save_pars(all = TRUE)
  )
rstan::check_hmc_diagnostics(m.om3$fit)

## 
## Divergences:
## 0 of 8000 iterations ended with a divergence.

## 
## Tree depth:
## 0 of 8000 iterations saturated the maximum tree depth of 10.

## 
## Energy:
## E-BFMI indicated no pathological behavior.

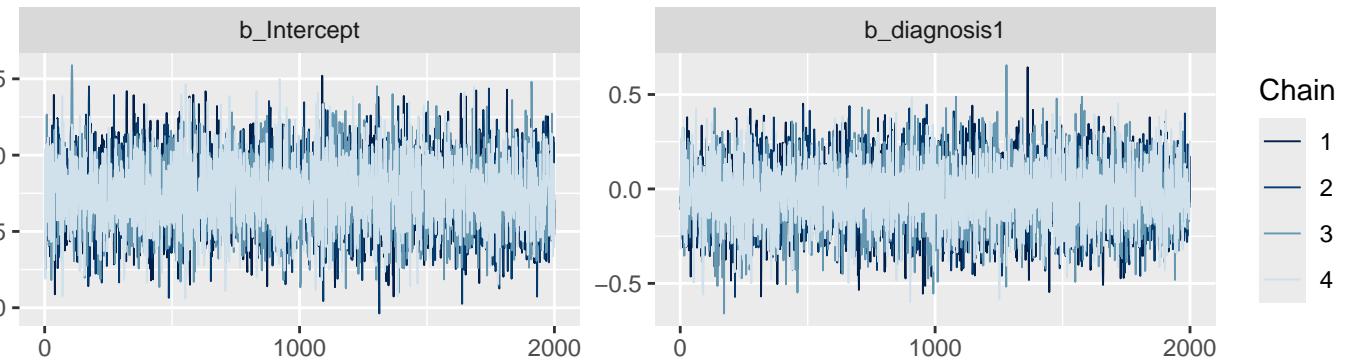
# check that rhats are below 1.01
sum(brms::rhat(m.om3) >= 1.01, na.rm = T)

## [1] 0

# check the trace plots
post.draws = as_draws_df(m.om3)
mcmc_trace(post.draws, regex_pars = "^b_",
            facet_args = list(ncol = 2)) +
  scale_x_continuous(breaks=scales::pretty_breaks(n = 3)) +
  scale_y_continuous(breaks=scales::pretty_breaks(n = 3))

```

Scale for x is already present.
 ## Adding another scale for x, which will replace the existing scale.



This model has no pathological behaviour with E-BFMI, no divergent sample and no rhats that are higher or equal to 1.01. Therefore, we go ahead and perform our posterior predictive checks.

```

# get posterior predictions
post.pred = posterior_predict(m.om3, ndraws = nsim)

# check the fit of the predicted data compared to the real data
p1 = pp_check(m.om3, ndraws = nsim) +
  theme_bw() + theme(legend.position = "none")

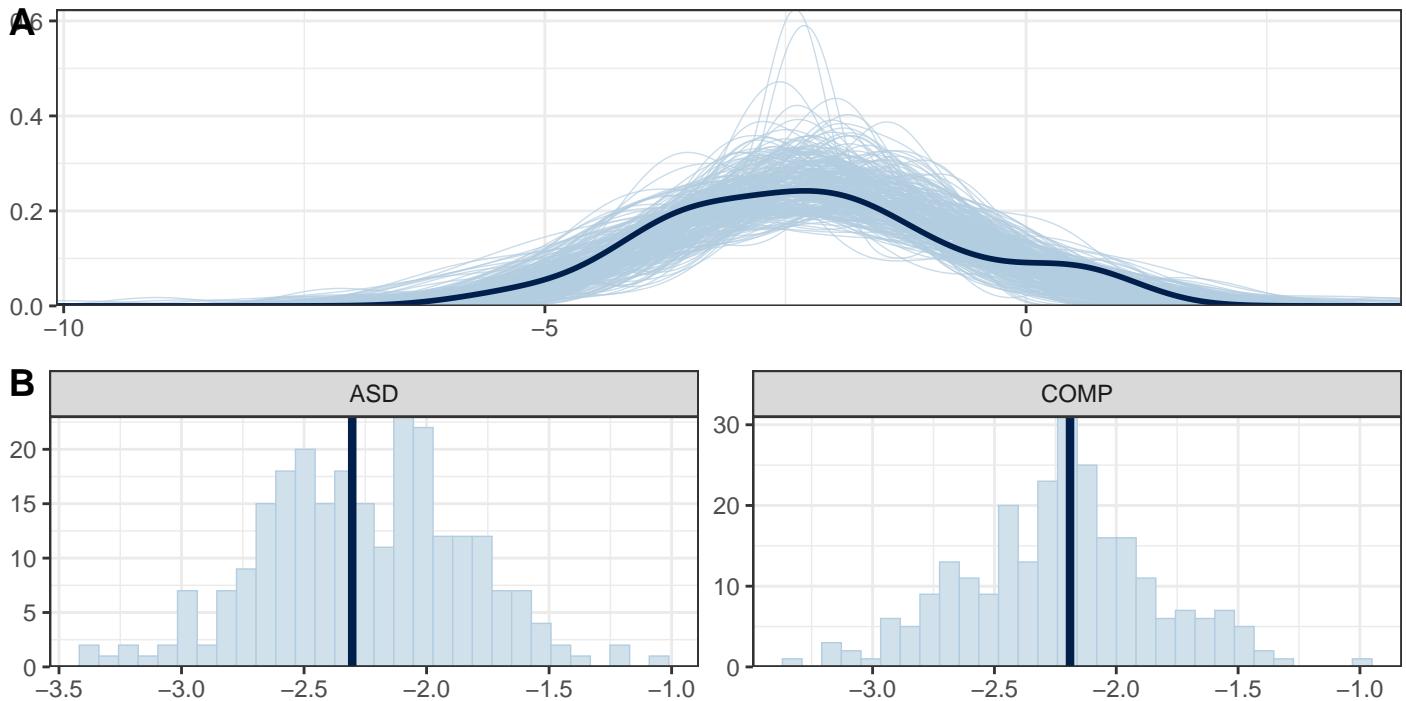
# distributions of means compared to the real values per group
p2 = ppc_stat_grouped(df.hgf$om3, post.pred, df.hgf$diagnosis) +
  theme_bw() + theme(legend.position = "none")

p = ggarrange(p1, p2,
              nrow = 2, ncol = 1, labels = "AUTO")

```

```
annotate_figure(p, top = text_grob("Posterior predictive checks",
  face = "bold", size = 14))
```

Posterior predictive checks



Similar to above, the simulated data based on the model fits well with the real data, although it doesn't reproduce the overall shape.

3.3 Inferences

Now that we are convinced that we can trust our model, we have a look at its estimate and use the hypothesis function to assess our hypotheses and perform explorative tests.

```
# print a summary
summary(m.om3)

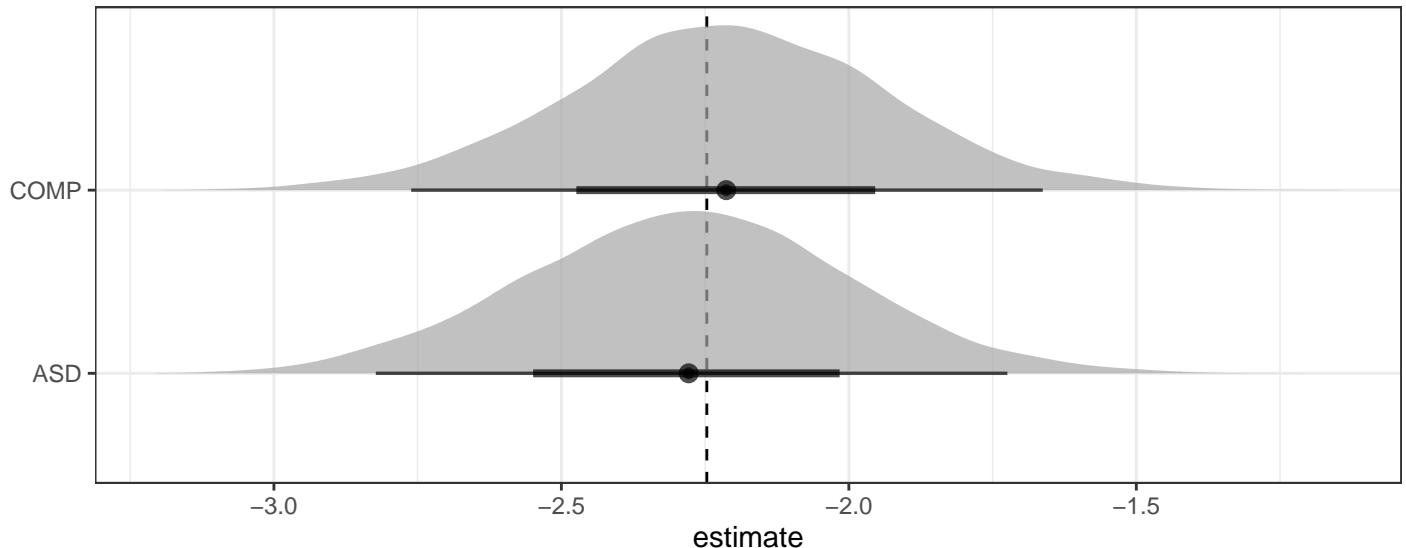
## Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: om3 ~ diagnosis
##   Data: df.hgf (Number of observations: 44)
##   Draws: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##          total post-warmup draws = 8000
##
## Regression Coefficients:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     -2.25      0.22    -2.68    -1.81 1.00     7060     5591
## diagnosis1    -0.03      0.16    -0.36     0.29 1.00     7761     5723
##
## Further Distributional Parameters:
##             Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma        1.46      0.13     1.22     1.74 1.00     7903     5912
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
#
# get the estimates and compute group comparisons
df.m.om3 = post.draws %>%
  mutate(
```

```

ASD      = b_Intercept + b_diagnosis1,
COMP    = b_Intercept - b_diagnosis1
)

# plot the posterior distributions
df.m.om3 %>%
  select(ASD, COMP) %>%
  pivot_longer(cols = everything(), names_to = "coef", values_to = "estimate") %>%
  ggplot(aes(x = estimate, y = coef), fill = c_light) +
  geom_vline(xintercept = mean(df.m.om3$b_Intercept), linetype = 'dashed') +
  ggdist::stat_halfeye(alpha = 0.7) + ylab(NULL) + theme_bw() +
  theme(legend.position = "none")

```



```

# H3b: COMP < ASD
h3b = hypothesis(m.om3, "0 < diagnosis1")
h3b

## Hypothesis Tests for class b:
##          Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio
## 1 (0)-(diagnosis1) < 0     0.03     0.16   -0.24     0.31     0.71
## Post.Prob Star
## 1      0.42
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## *: For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
equivalence_test(m.om3)

```

```

## # Test for Practical Equivalence
##
## ROPE: [-0.16 0.16]
##
## Parameter |      HO | inside ROPE |      95% HDI
## -----
## Intercept | Rejected |      0.00 % | [-2.68, -1.81]
## diagnosis1 | Undecided |    68.79 % | [-0.36, 0.29]

```

```

# get effect sizes (Hedges, 2007)
df.effect = df.m.om3 %>%
  mutate(
    group = 2*b_diagnosis1 / sigma
  )

```

```

kable(df.effect %>% select(group) %>%
  pivot_longer(cols = everything(), values_to = "estimate") %>%
  group_by(name) %>%
  summarise(
    ci.lo = lower_ci(estimate),
    mean = mean(estimate),
    ci.hi = upper_ci(estimate),
    interpret = interpret_cohens_d(mean)
  )
)

```

| | name | ci.lo | mean | ci.hi | interpret |
|-------|------------|------------|-----------|-------|------------|
| group | -0.4986958 | -0.0472644 | 0.3939283 | | very small |

estimate = 0.03 [-0.24, 0.31], *posterior probability* = 41.61%

4 Exploration of Bernoulli model with HGF parameters

4.1 Model setup

```

# recode the order and scale the predictors
df.hgf = df.hgf %>%
  mutate(
    group = if_else(diagnosis == "ASD", 1, 0)
  ) %>% mutate(across(c(be1, be2, be3, be4, ze, om2, om3), scale_this, .names = "s{.col}"))

kable(df.hgf %>% select(diagnosis, group) %>% distinct(),
  caption = "Coding for the order in the Bernoulli model")

```

Table 6: Coding for the order in the Bernoulli model

| diagnosis | group |
|-----------|-------|
| COMP | 0 |
| ASD | 1 |

```

# model formula
f = brms::bf( group ~ sbe1 + sbe2 + sbe3 + sbe4 + sze + som2 + som3 )
f

## group ~ sbe1 + sbe2 + sbe3 + sbe4 + sze + som2 + som3

# Bernoulli
priors = c(
  prior(normal(0,      0.50),   class = Intercept), # because 50:50
  prior(normal(0,      1.00),   class = b)
)

```

4.2 Posterior predictive checks

```

# fit the final model
m = brm(f,
  df.hgf, prior = priors,
  family = bernoulli(link = "logit"),
  iter = iter, warmup = warm,
  backend = "cmdstanr", threads = threading(8),
  file = file.path(brms_dir, "m_hgf_bern"),

```

```

    save_pars = save_pars(all = TRUE),
    seed = 4284
)
rstan::check_hmc_diagnostics(m$fit)

## 
## Divergences:
## 0 of 8000 iterations ended with a divergence.

## 
## Tree depth:
## 0 of 8000 iterations saturated the maximum tree depth of 10.

## 
## Energy:
## E-BFMI indicated no pathological behavior.

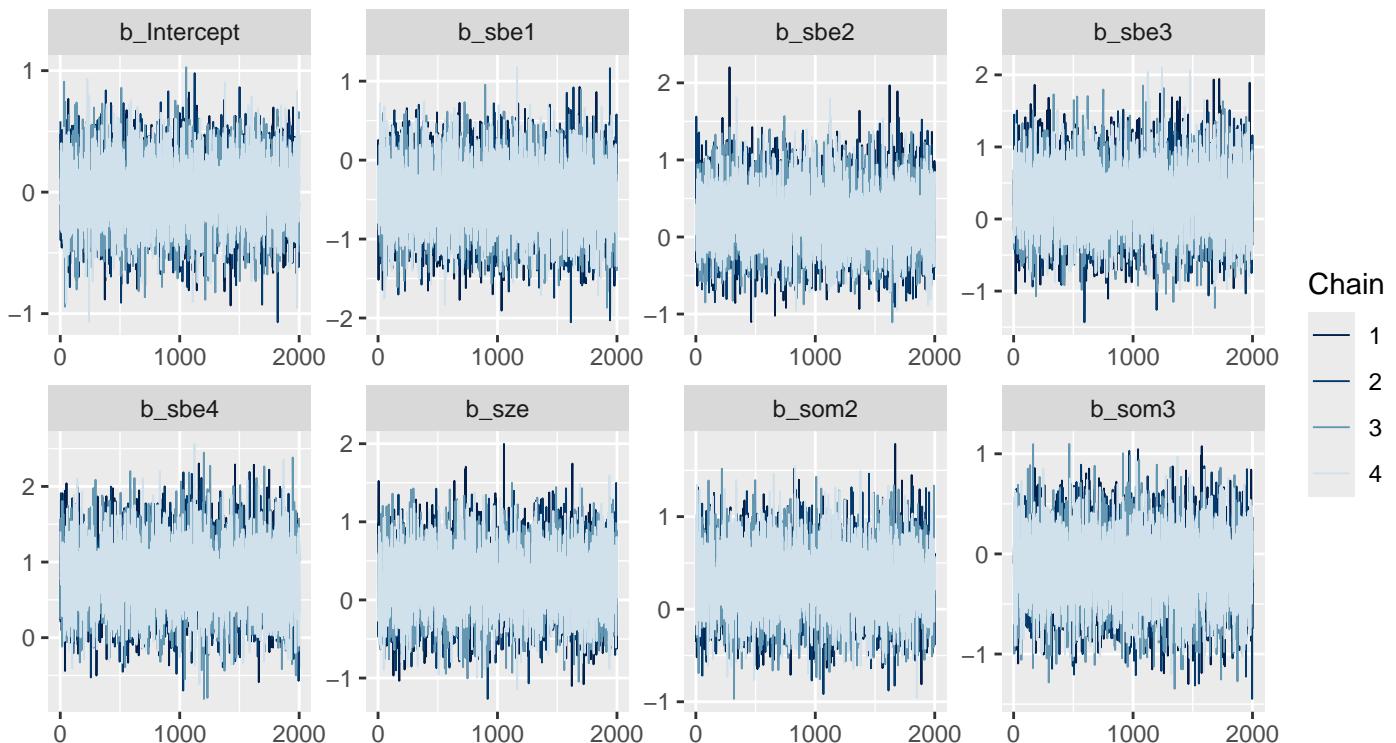
# check that rhats are below 1.01
sum(brms::rhat(m) >= 1.01, na.rm = T)

## [1] 0

# check the trace plots
post.draws = as_draws_df(m)
mcmc_trace(post.draws, regex_pars = "^\b_",
            facet_args = list(ncol = 4)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 3)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 3))

```

Scale for x is already present.
Adding another scale for x, which will replace the existing scale.



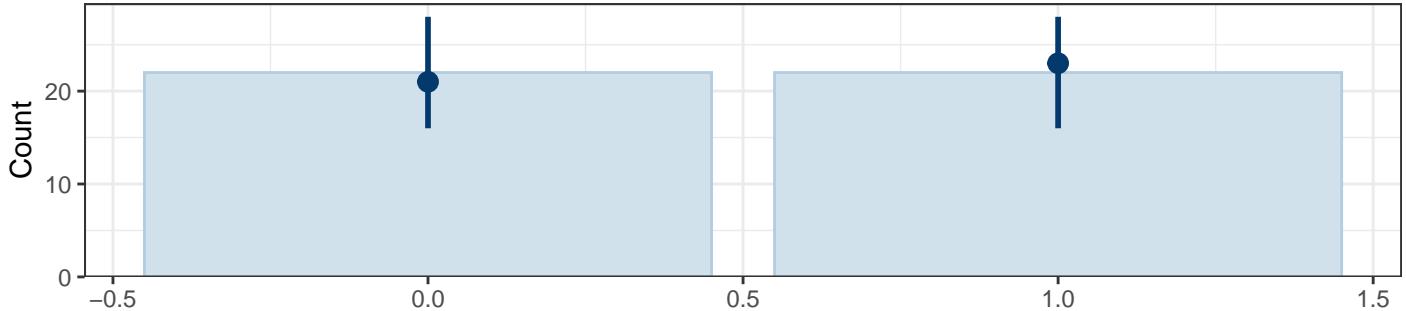
This model has no pathological behaviour with E-BFMI, no divergent sample and no rhats that are higher or equal to 1.01. Therefore, we go ahead and perform our posterior predictive checks.

```
# get posterior predictions
post.pred = posterior_predict(m, ndraws = nsim)
```

```
# check the fit of the predicted data compared to the real data
p = ppc_bars(df.hgf$group, post.pred) +
  theme_bw() + theme(legend.position = "none")

annotate_figure(p, top = text_grob("Posterior predictive checks",
                                    face = "bold", size = 14))
```

Posterior predictive checks



The overall simulated data fits reasonably well. Now that we are convinced that we can trust our model, we have a look at its estimates.

4.3 Inferences

```
# print a summary
summary(m)

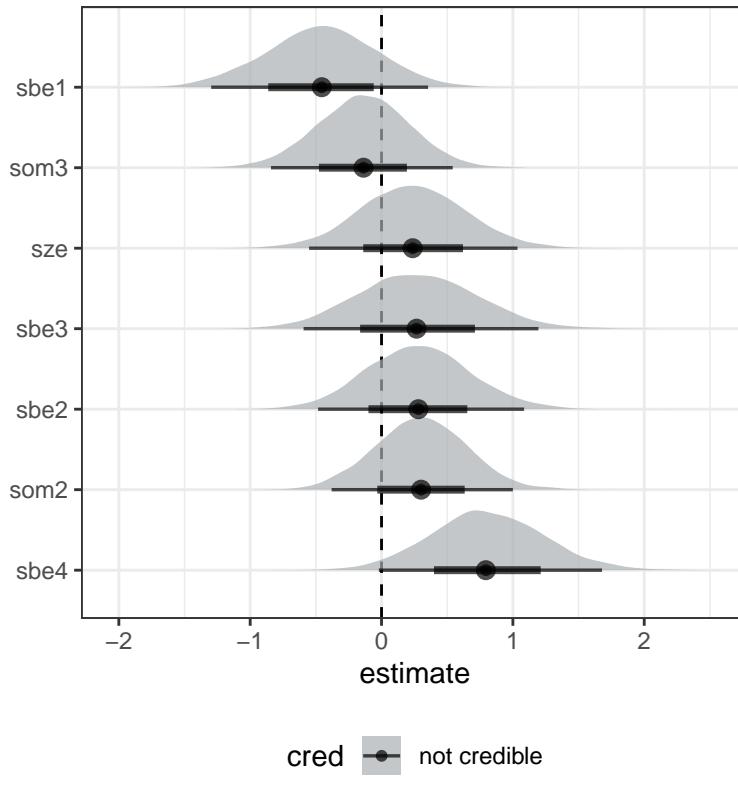
## Family: bernoulli
## Links: mu = logit
## Formula: group ~ sbe1 + sbe2 + sbe3 + sbe4 + sze + som2 + som3
## Data: df.hgf (Number of observations: 44)
## Draws: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##        total post-warmup draws = 8000
##
## Regression Coefficients:
##                               Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept            0.00     0.27   -0.52    0.53 1.00   11653    6220
## sbe1              -0.46     0.42   -1.30    0.35 1.00    7619    6607
## sbe2               0.28     0.40   -0.48    1.09 1.00   8016    6758
## sbe3               0.28     0.46   -0.59    1.19 1.00   6374    5607
## sbe4               0.81     0.43   -0.02    1.68 1.00   7819    6003
## sze                0.24     0.40   -0.55    1.04 1.00   7223    6128
## som2               0.30     0.35   -0.38    1.00 1.00   10326   6234
## som3              -0.14     0.35   -0.84    0.54 1.00   8173    6191
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
# plot the posterior distributions
post.draws %>%
  select(starts_with("b_") & !starts_with("b_Int")) %>%
  pivot_longer(cols = starts_with("b_"), names_to = "coef", values_to = "estimate") %>%
  mutate(
    coef = substr(coef, 3, nchar(coef)),
    coef = fct_reorder(coef, desc(estimate))
  ) %>%
  group_by(coef) %>%
  mutate(
    cred = case_when(
```

```

    (mean(estimate) < 0 & quantile(estimate, probs = 0.975) < 0) |
    (mean(estimate) > 0 & quantile(estimate, probs = 0.025) > 0) ~ "credible",
  T ~ "not credible"
)
) %>% ungroup() %>%
ggplot(aes(x = estimate, y = coef, fill = cred)) +
geom_vline(xintercept = 0, linetype = 'dashed') +
ggdist::stat_halfeye(alpha = 0.7) + ylab(NULL) +
scale_fill_manual(values = c("credible" = c_dark, "not credible" = c_light)) +
theme_bw() + theme(legend.position = "bottom", legend.direction = "horizontal")

```



cred ● not credible

```

e = hypothesis(m, "sbe4 > 0", alpha = 0.025)
e$hypothesis

```

```

##   Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio Post.Prob Star
## 1 (sbe4) > 0  0.8058745  0.4319043 -0.02068792  1.678584    34.39823   0.97175

```

```
equivalence_test(m)
```

```

## # Test for Practical Equivalence
## 
## ROPE: [-0.18 0.18]
## 
## Parameter |      H0 | inside ROPE |      95% HDI
## -----
## Intercept | Undecided |  52.00 % | [-0.52, 0.53]
## sbe1      | Undecided | 19.89 % | [-1.30, 0.35]
## sbe2      | Undecided | 29.95 % | [-0.48, 1.09]
## sbe3      | Undecided | 28.01 % | [-0.59, 1.19]
## sbe4      | Undecided |  4.93 % | [-0.02, 1.68]
## sze       | Undecided | 31.72 % | [-0.55, 1.04]
## som2      | Undecided | 29.75 % | [-0.38, 1.00]
## som3      | Undecided | 39.25 % | [-0.84, 0.54]

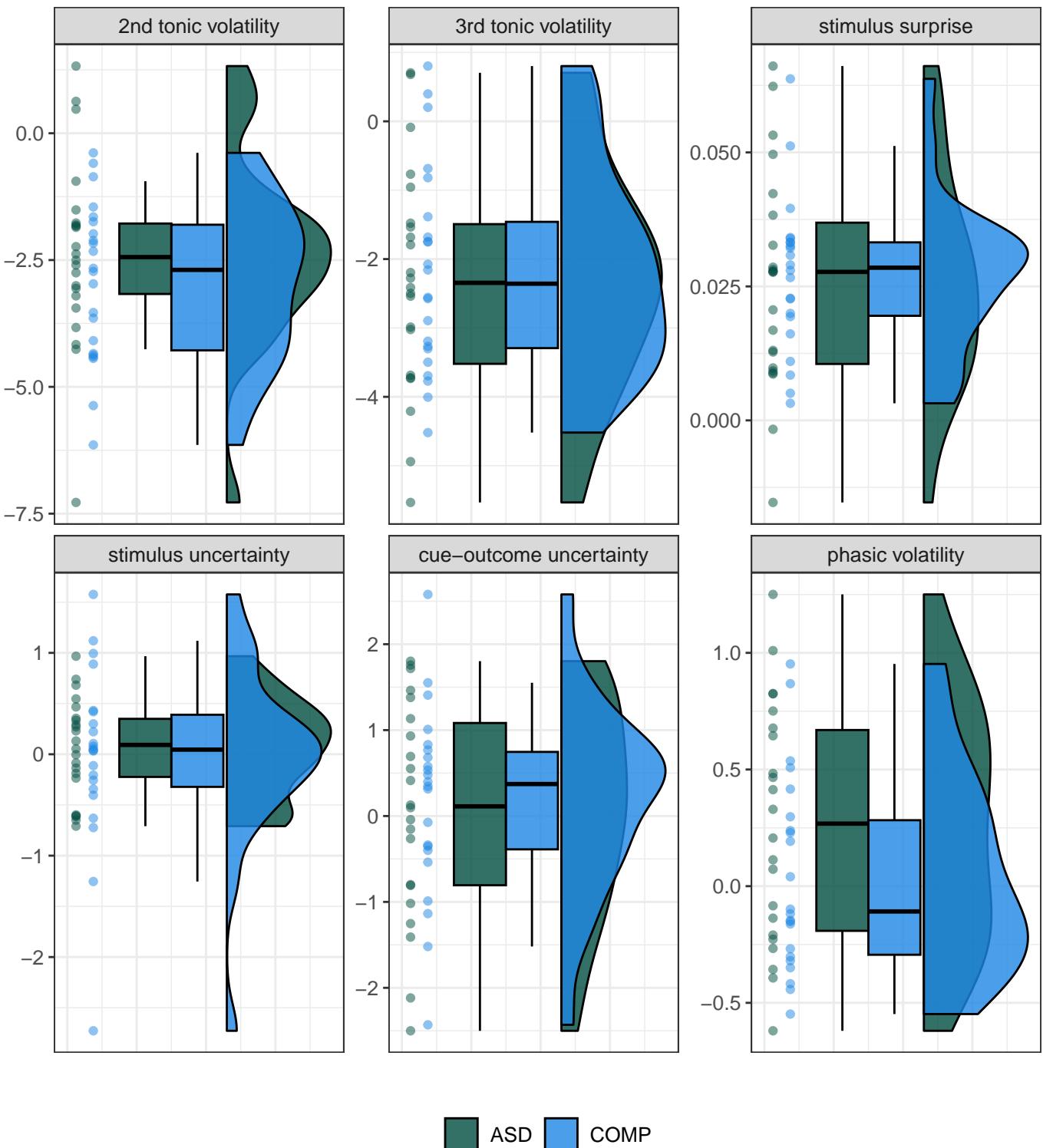
```

5 Plots for all HGF parameters

```

df.hgf %>%
  select(subID, diagnosis, be0, be1, be2, be3, be4, ze, om2, om3) %>% #
  pivot_longer(cols = c(be0, be1, be2, be3, be4, ze, om2, om3),
               names_to = "parameter") %>%
  mutate(
    parameter = factor(case_match(parameter,
                                    "be0" ~ "predicted log RT",
                                    "be1" ~ "stimulus surprise",
                                    "be2" ~ "stimulus uncertainty",
                                    "be3" ~ "cue-outcome uncertainty",
                                    "be4" ~ "phasic volatility",
                                    "ze" ~ "Sigma (decision noise)",
                                    "om2" ~ "2nd tonic volatility",
                                    "om3" ~ "3rd tonic volatility"
                                ), levels = c("2nd tonic volatility",
                                              "3rd tonic volatility",
                                              "predicted log RT",
                                              "stimulus surprise",
                                              "stimulus uncertainty",
                                              "cue-outcome uncertainty",
                                              "phasic volatility",
                                              "Sigma (decision noise)"))
  ) %>%
  filter(!(parameter %in% c("predicted log RT", "Sigma (decision noise)"))) %>%
  ggplot(aes(x = 1, y = value, fill = diagnosis, colour = diagnosis)) + #
  geom_rain(rain.side = 'r',
  boxplot.args = list(color = "black", outlier.shape = NA, show.legend = FALSE, alpha = .8),
  violin.args = list(color = "black", outlier.shape = NA, alpha = .8),
  boxplot.args.pos = list(
    position = ggpp::position_dodgegenudge(x = 0, width = 0.3), width = 0.3
  ),
  point.args = list(show.legend = FALSE, alpha = .5),
  violin.args.pos = list(
    width = 0.6, position = position_nudge(x = 0.16)),
  point.args.pos = list(position = ggpp::position_dodgegenudge(x = -0.25, width = 0.1))) +
  scale_fill_manual(values = col.grp) +
  scale_color_manual(values = col.grp) +
  facet_wrap(. ~ parameter, scales = "free", ncol = 3) +
  labs(title = "HGF parameter", x = "", y = "") +
  theme_bw() +
  theme(legend.position = "bottom", plot.title = element_blank(),
        text = element_text(size = 13), axis.text.x=element_blank(),
        axis.ticks.x=element_blank(), legend.direction = "horizontal",
        legend.title = element_blank())

```



ASD COMP

```
ggsave("plots/FigHGF.svg", units = "cm", width = 27, height = 13.5)
```

6 Learning rate update - volatile to stable

6.1 Model setup

```
# model formula
f.alpha = brms::bf( value ~ diagnosis * level * change + (level + change | subID) )

# set weakly informative priors taking Lawson 2017 into consideration
priors = c(
```

```

prior(normal(-5, 2),    class = Intercept),
prior(normal(0.5, 0.5), class = sigma),
prior(normal(0.5, 0.5), class = sd),
prior(lkj(2),           class = cor),
prior(normal(0,   1.0),  class = b) # probably big difference between levels
)

```

6.2 Posterior predictive checks

As the next step, we fit the model, check whether there are divergence or rhat issues, and then check whether the chains have converged.

```

# fit the final model
m.alpha = brm(f.alpha, family = lognormal,
               df.upd, prior = priors, seed = 6688,
               iter = iter, warmup = warm,
               backend = "cmdstanr", threads = threading(t),
               file = file.path(brms_dir, "m_hgf_alpha"),
               save_pars = save_pars(all = TRUE)
)
rstan::check_hmc_diagnostics(m.alpha$fit)

## 
## Divergences:
## 0 of 8000 iterations ended with a divergence.

## 
## Tree depth:
## 0 of 8000 iterations saturated the maximum tree depth of 10.

## 
## Energy:
## E-BFMI indicated no pathological behavior.

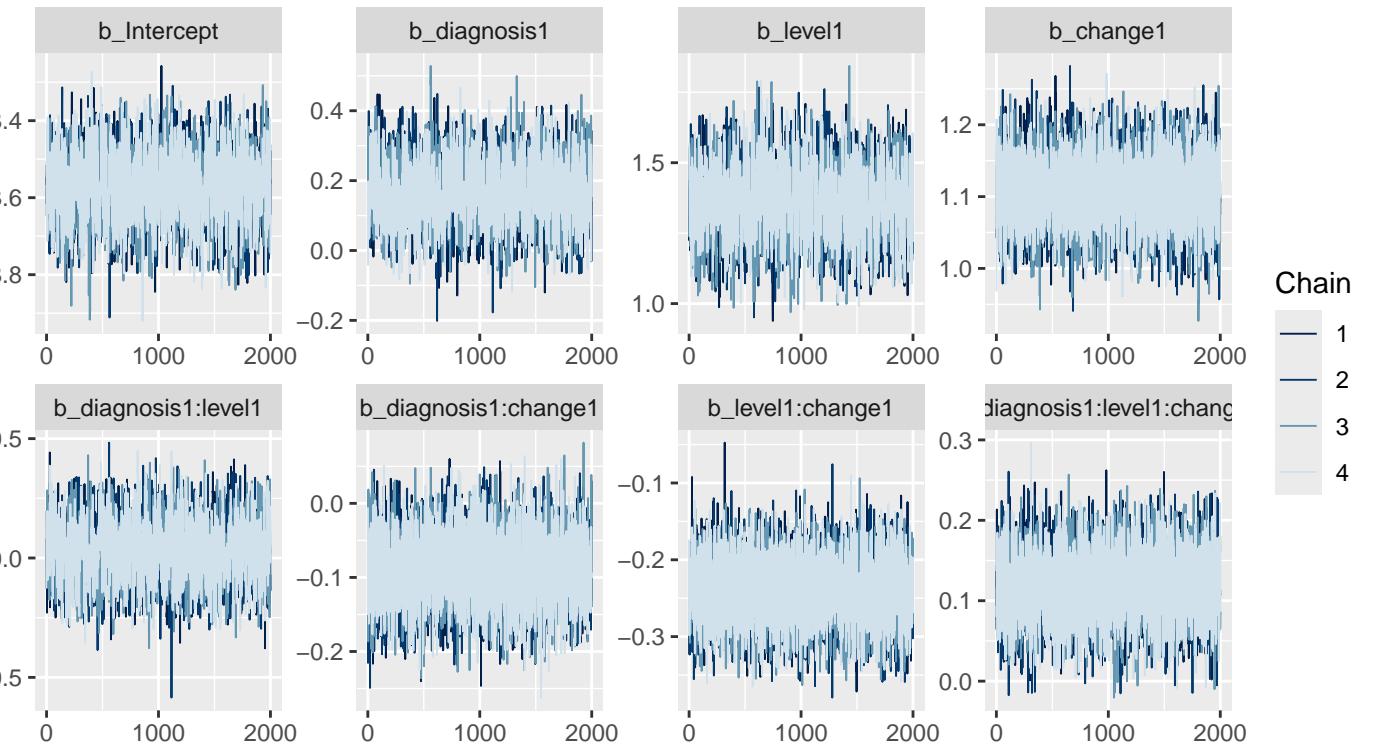
# check that rhats are below 1.01
sum(brms::rhat(m.alpha) >= 1.01, na.rm = T)

## [1] 0

# check the trace plots
post.draws = as_draws_df(m.alpha)
mcmc_trace(post.draws, regex_pars = "^b_",
            facet_args = list(ncol = 4)) +
  scale_x_continuous(breaks=scales::pretty_breaks(n = 3)) +
  scale_y_continuous(breaks=scales::pretty_breaks(n = 3))

## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.

```



This model has no pathological behaviour with E-BFMI, no divergent sample and no rhats that are higher or equal to 1.01. Therefore, we go ahead and perform our posterior predictive checks.

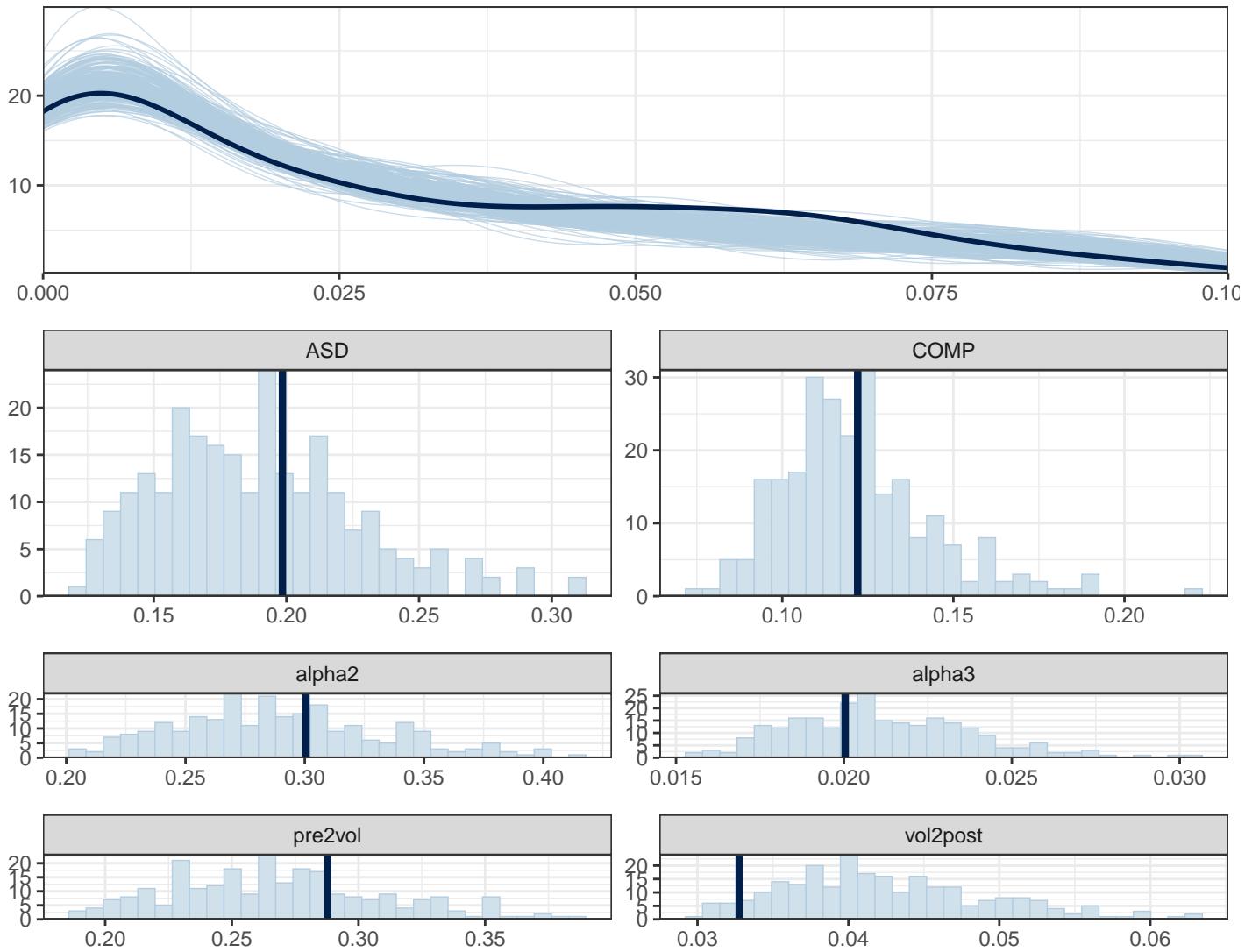
```
# get posterior predictions
post.pred = posterior_predict(m.alpha, ndraws = nsim)

# check the fit of the predicted data compared to the real data
p1 = ppc_check(m.alpha, ndraws = nsim) +
  theme_bw() + theme(legend.position = "none") + xlim(0, 0.10)

# distributions of means compared to the real values per group
p2 = ppc_stat_grouped(df.upd$value, post.pred, df.upd$diagnosis) +
  theme_bw() + theme(legend.position = "none")
p3 = ppc_stat_grouped(df.upd$value, post.pred, df.upd$level) +
  theme_bw() + theme(legend.position = "none")
p4 = ppc_stat_grouped(df.upd$value, post.pred, df.upd$change) +
  theme_bw() + theme(legend.position = "none")

p = ggarrange(p1, p2, ggarrange(p3, p4, nrow = 2),
              nrow = 3, ncol = 1)
annotate_figure(p, top = text_grob("Posterior predictive checks",
                                  face = "bold", size = 14))
```

Posterior predictive checks



This model does not quite capture the different changes. We'll still use the estimates for our hypothesis testing, since we are not interested in the specific changes. However, we also do model comparison to back the results up.

6.3 Inferences

Now that we are convinced that we can trust our model, we have a look at its estimate and use the hypothesis function to assess our hypotheses and perform explorative tests.

```
# print a summary
summary(m.alpha)

## Family: lognormal
## Links: mu = identity; sigma = identity
## Formula: value ~ diagnosis * level * change + (level + change | subID)
## Data: df.upd (Number of observations: 176)
## Draws: 4 chains, each with iter = 3000; warmup = 1000; thin = 1;
##        total post-warmup draws = 8000
##
## Multilevel Hyperparameters:
## ~subID (Number of levels: 44)
##                               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)              0.48     0.07    0.35    0.64 1.00    2751
## sd(level1)                 0.74     0.09    0.57    0.94 1.00    2582
## sd(change1)                0.11     0.06    0.01    0.22 1.00    2400
## cor(Intercept,level1)      0.40     0.16    0.08    0.68 1.00    1321
```

```

## cor(Intercept,change1)      0.51      0.32     -0.31      0.92 1.00      4714
## cor(level1,change1)        0.15      0.33     -0.54      0.72 1.00      8925
##                                     Tail_ESS
## sd(Intercept)                4852
## sd(level1)                  4058
## sd(change1)                 1948
## cor(Intercept,level1)        2778
## cor(Intercept,change1)       4106
## cor(level1,change1)         5669
##
## Regression Coefficients:
##                               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                   -3.57      0.08    -3.74     -3.41 1.00      2132
## diagnosis1                  0.17      0.09     0.00      0.33 1.00      2328
## level1                      1.36      0.12     1.13      1.60 1.00      2131
## change1                      1.11      0.05     1.02      1.20 1.00      7150
## diagnosis1:level1           0.03      0.12    -0.21      0.27 1.00      2149
## diagnosis1:change1          -0.09      0.05    -0.18     -0.00 1.00      7153
## level1:change1              -0.24      0.04    -0.32     -0.16 1.00     10154
## diagnosis1:level1:change1   0.12      0.04     0.04      0.20 1.00     10027
##                                     Tail_ESS
## Intercept                     3428
## diagnosis1                    3860
## level1                        3391
## change1                       5410
## diagnosis1:level1            3702
## diagnosis1:change1           5459
## level1:change1                6174
## diagnosis1:level1:change1    5466
##
## Further Distributional Parameters:
##                               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma             0.55      0.05     0.47      0.64 1.00      2647      4808
##
## Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).

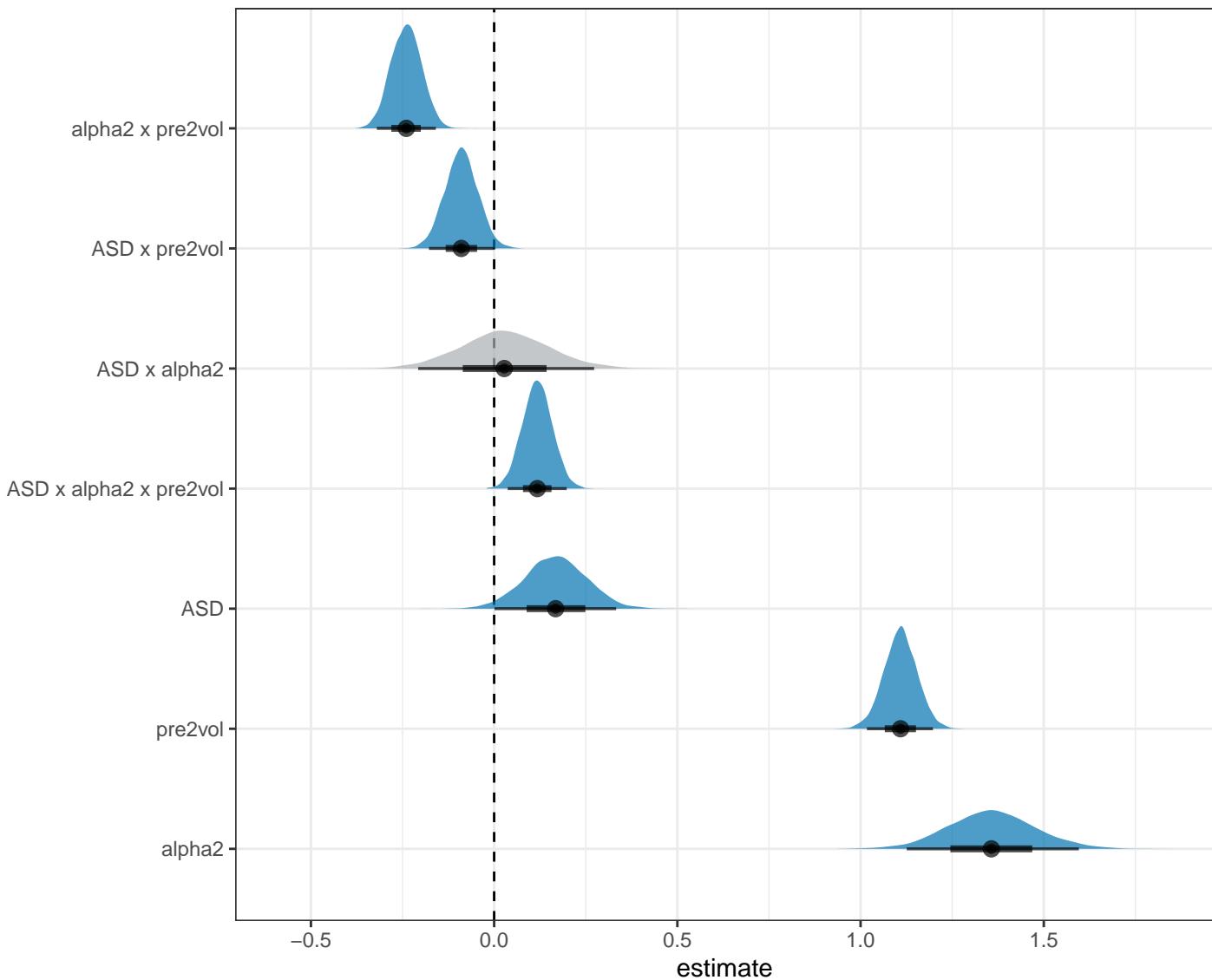
# plot the posterior distributions
post.draws %>%
  select(starts_with("b_")) %>%
  pivot_longer(cols = starts_with("b_"), names_to = "coef", values_to = "estimate") %>%
  subset(!startsWith(coef, "b_Int")) %>%
  mutate(
    coef = substr(coef, 3, nchar(coef)),
    coef = str_replace_all(coef, ":" , " x "),
    coef = str_replace_all(coef, "diagnosis1", "ASD"),
    coef = str_replace_all(coef, "level1", "alpha2"),
    coef = str_replace_all(coef, "change1", "pre2vol"),
    coef = fct_reorder(coef, desc(estimate))
  ) %>%
  group_by(coef) %>%
  mutate(
    cred = case_when(
      (mean(estimate) < 0 & quantile(estimate, probs = 0.975) < 0) |
        (mean(estimate) > 0 & quantile(estimate, probs = 0.025) > 0) ~ "credible",
      T ~ "not credible"
    )
  ) %>% ungroup() %>%
  ggplot(aes(x = estimate, y = coef, fill = cred)) +

```

```

geom_vline(xintercept = 0, linetype = 'dashed') +
ggdist::stat_halfeye(alpha = 0.7) + ylab(NULL) + theme_bw() +
scale_fill_manual(values = c(c_dark, c_light)) + theme(legend.position = "none")

```



```

# get the design matrix to figure out how to set the contrasts
df.des = cbind(df.upd,
  model.matrix(~ diagnosis * level * change, data = df.upd)) %>%
ungroup() %>%
select(-subID, -value) %>% distinct()

# H4a: alpha3 ASD > COMP
t(df.des %>%
  filter(level == "alpha3") %>%
  group_by(diagnosis) %>%
  summarise(across(where(is.numeric), ~ mean(.x))) %>%
  arrange(diagnosis) %>%
  select(where(is.numeric)) %>%
  map_df(~ diff(.x))) # COMP - ASD

## [,1]
## (Intercept) 0
## diagnosis1 -2
## level1 0
## change1 0

```

```

## diagnosis1:level1      2
## diagnosis1:change1     0
## level1:change1         0
## diagnosis1:level1:change1  0

h4a = hypothesis(m.alpha, "0 > - diagnosis1 + diagnosis1:level1")
h4a

## Hypothesis Tests for class b:
##                               Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio
## 1 (0)-(-diagnosis1+... > 0      0.14      0.12    -0.07     0.35      6.7
##   Post.Prob Star
## 1      0.87
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.

# H4b: alpha2 ASD < COMP
t(df.des %>%
  filter(level == "alpha2") %>%
  group_by(diagnosis) %>%
  summarise(across(where(is.numeric), ~ mean(.x))) %>%
  arrange(diagnosis) %>%
  select(where(is.numeric)) %>%
  map_df(~ diff(.x))) # COMP - ASD

##                                     [,1]
## (Intercept)                  0
## diagnosis1                   -2
## level1                       0
## change1                      0
## diagnosis1:level1            -2
## diagnosis1:change1           0
## level1:change1                0
## diagnosis1:level1:change1    0

h4b = hypothesis(m.alpha, "0 < -diagnosis1 - diagnosis1:level1")
h4b

## Hypothesis Tests for class b:
##                               Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio
## 1 (0)-(-diagnosis1-... < 0      0.2       0.17    -0.08     0.47      0.13
##   Post.Prob Star
## 1      0.12
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.

# get effect sizes (Hedges, 2007)
df.effect = post.draws %>%
  mutate(
    sumvar = sqrt(sigma^2 + sd_subID_Intercept^2 +
                  sd_subID_level1^2 + sd_subID_change1^2),
    group = 2*b_diagnosis1 / sumvar,
    h4a   = (-2*b_diagnosis1 + 2*b_diagnosis1:level1) / sumvar,
    h4b   = (-2*b_diagnosis1 - 2*b_diagnosis1:level1) / sumvar
  )

kable(df.effect %>% select(group, h4a, h4b) %>

```

```

pivot_longer(cols = everything(), values_to = "estimate") %>%
group_by(name) %>%
summarise(
  ci.lo = lower_ci(estimate),
  mean = mean(estimate),
  ci.hi = upper_ci(estimate),
  interpret = interpret_cohens_d(mean)
)
)

```

| | name | ci.lo | mean | ci.hi | interpret |
|--|-------|------------|------------|-----------|-----------|
| | group | 0.0020368 | 0.3213977 | 0.6435737 | small |
| | h4a | -0.7374934 | -0.2663236 | 0.2049889 | small |
| | h4b | -1.0105097 | -0.3764719 | 0.2570543 | small |

h4a ASD alpha3: *estimate* = 0.14 [-0.07, 0.35], *posterior probability* = 87.01%

h4b ASD alpha2: *estimate* = 0.2 [-0.08, 0.47], *posterior probability* = 11.89%

6.4 Bayesian ANOVA

```

df.upd = df.upd %>% ungroup() %>%
  mutate(rvalue = rank(value))

if (!file.exists(file.path(brms_dir, "aov_alpha.rds"))) {
  aov = anovaBF(rvalue ~ diagnosis * level * change, data = df.upd)
  saveRDS(aov, file.path(brms_dir, "aov_alpha.rds"))
} else {
  aov = readRDS(file.path(brms_dir, "aov_alpha.rds"))
}

kable(aov@bayesFactor %>% arrange(desc(bf)) %>%
  select(bf) %>% mutate(bf.diff = abs(lead(bf)-bf),
                         bf.int = interpret_bf(bf.diff, log = T)), digits = 3)

```

| | bf | bf.diff | bf.int |
|--|---------|---------|---------------------------------|
| level + change | 105.308 | 0.034 | anecdotal evidence in favour of |
| level + change + level:change | 105.274 | 0.565 | anecdotal evidence in favour of |
| diagnosis + level + change + level:change | 104.709 | 0.078 | anecdotal evidence in favour of |
| diagnosis + level + change | 104.631 | 0.888 | anecdotal evidence in favour of |
| diagnosis + level + change + diagnosis:change | 103.743 | 0.110 | anecdotal evidence in favour of |
| diagnosis + level + change + diagnosis:change + level:change | 103.633 | 0.109 | anecdotal evidence in favour of |
| diagnosis + level + diagnosis:level + change + level:change | 103.524 | 0.053 | anecdotal evidence in favour of |
| diagnosis + level + diagnosis:level + change | 103.471 | 1.119 | moderate evidence in favour of |
| diagnosis + level + diagnosis:level + change + diagnosis:change + level:change | 102.352 | 0.001 | anecdotal evidence in favour of |
| diagnosis + level + diagnosis:level + change + diagnosis:change | 102.351 | 0.804 | anecdotal evidence in favour of |

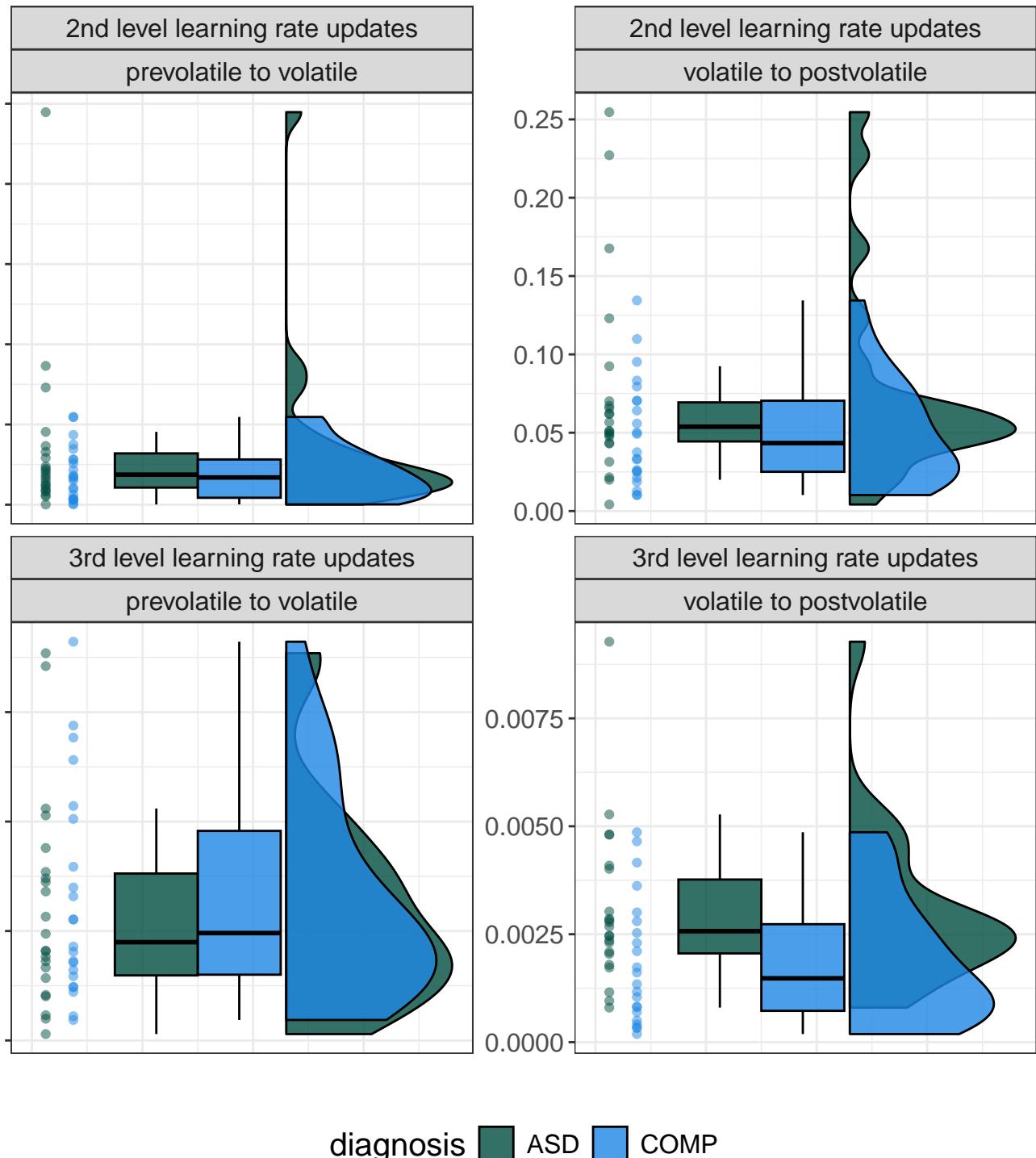
| | bf | bf.diff | bf.int |
|--|---------|---------|--------------------------------|
| diagnosis + level + diagnosis:level + change + diagnosis:change + level:change + diagnosis:level:change | 101.547 | 50.328 | extreme evidence in favour of |
| level | 51.219 | 1.229 | moderate evidence in favour of |
| diagnosis + level | 49.990 | 1.346 | moderate evidence in favour of |
| diagnosis + level + diagnosis:level | 48.643 | 25.339 | extreme evidence in favour of |
| change | 23.304 | 1.419 | moderate evidence in favour of |
| diagnosis + change | 21.886 | 1.348 | moderate evidence in favour of |
| diagnosis + change + diagnosis:change | 20.537 | 22.032 | extreme evidence in favour of |
| diagnosis | -1.494 | NA | |

7 Plots for learning rate updates

```
# rain cloud plot

df.upd %>%
  mutate(
    change = case_match(change,
                         "pre2vol" ~ "prevolatile to volatile",
                         "vol2post" ~ "volatile to postvolatile"),
    level = case_match(level,
                        "alpha2" ~ "2nd level learning rate updates",
                        "alpha3" ~ "3rd level learning rate updates")
  ) %>%
  ggplot(aes(1, value, fill = diagnosis, colour = diagnosis)) +
  geom_rain(rain.side = 'r',
            boxplot.args = list(color = "black", outlier.shape = NA, show.legend = FALSE, alpha = .8),
            violin.args = list(color = "black", outlier.shape = NA, alpha = .8),
            boxplot.args.pos = list(
              position = ggpp::position_dodgenudge(x = 0, width = 0.3), width = 0.3
            ),
            point.args = list(show.legend = FALSE, alpha = .5),
            violin.args.pos = list(
              width = 0.6, position = position_nudge(x = 0.16)),
            point.args.pos = list(position = ggpp::position_dodgenudge(x = -0.25, width = 0.1))) +
  scale_fill_manual(values = col.grp) +
  scale_color_manual(values = col.grp) +
  facet_wrap(level ~ change, scales = "free") +
  labs(title = "Learning rate updates", x = "", y = "") +
  theme_bw() +
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5),
        legend.direction = "horizontal", text = element_text(size = 15),
        axis.text.x = element_blank(), axis.ticks.x = element_blank())
```

Learning rate updates



```
ggsave("plots/FigHGF_LR.svg", units = "cm", width = 27, height = 13.5)
```

8 Similar plots to Lawson et al. (2017) for comparison

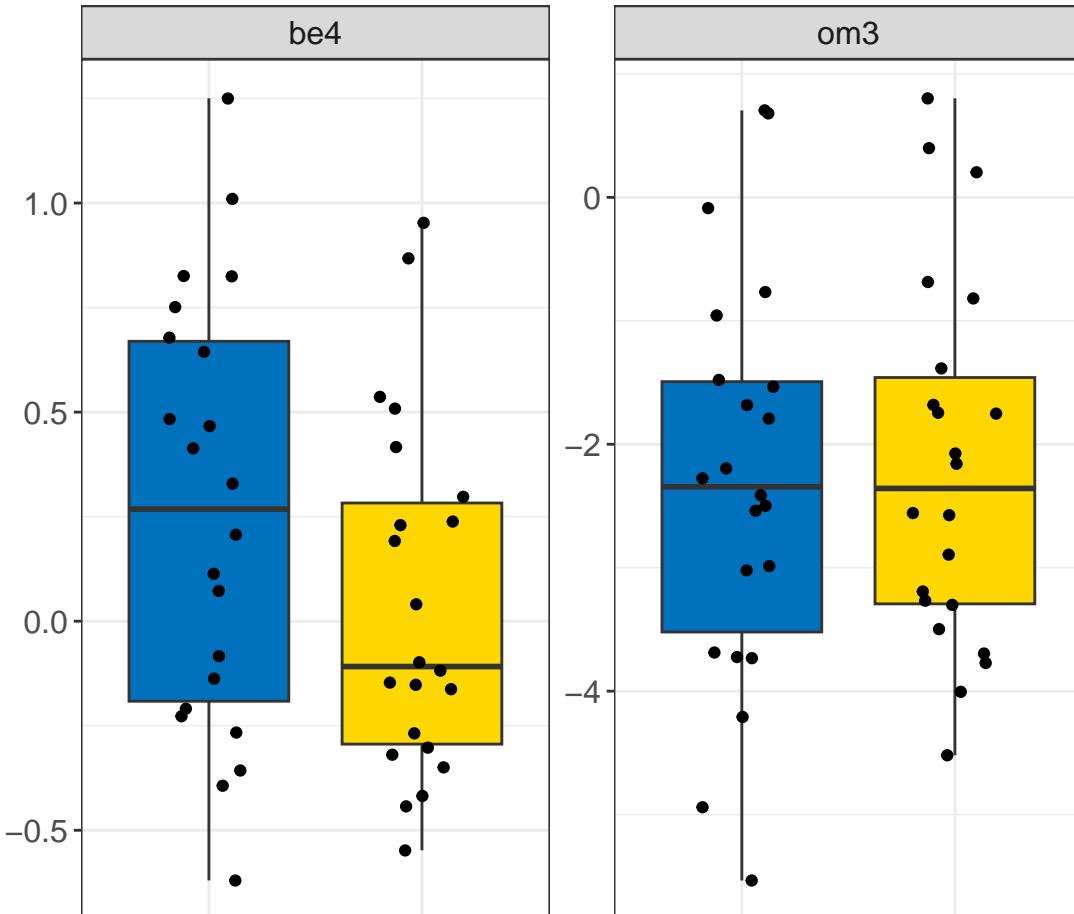
```
df.hgf %>%
  select(subID, diagnosis, be4, om3) %>% #
  pivot_longer(cols = c(be4, om3), names_to = "parameter") %>%
  ggplot(aes(x = diagnosis, y = value, fill = diagnosis)) +
  geom_boxplot() +
  geom_jitter(width = 0.2) +
  scale_fill_manual(values = c("#0072bd", "#ffd700")) +
```

```

facet_wrap(. ~ parameter, scales = "free", ncol = 3) +
  labs(title = "HGF parameter", x = "", y = "") +
  theme_bw() +
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5),
        text = element_text(size = 15), axis.text.x=element_blank(),
        axis.ticks.x=element_blank(), legend.direction = "horizontal")

```

HGF parameter



diagnosis ASD COMP

```

# two-way interaction
df.upd %>%
  filter(diagnosis %in% c("ASD", "COMP") & change == "pre2vol") %>%
  ggplot(aes(y = value, fill = diagnosis, x = level)) +
  geom_boxplot() +
  geom_point(position=position_jitterdodge(dodge.width=0.9)) +
  scale_fill_manual(values = c("#0072bd", "#ffd700")) +
  labs(x = "diagnosis", y = "Delta(alpha)") + # "\u0394 \u03b1" not working for now
  theme_bw() +
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5))

```

