Supplementary material: Absolute Pitch and Sound-Color-Synesthesia

Beat Meier[1], Andrew W. Ellis[1], and & Solange Glasser[2]

[1] Department of Psychology

University of Bern

Switzerland

[2] University of Melbourne

Australia

Author Note

Andrew W. Ellis https://orcid.org/0000-0002-2788-936X

The authors made the following contributions. Beat Meier: Conceptualization, Writing - Original Draft Preparation, Writing - Review & Editing; Andrew W. Ellis: Writing - Review & Editing, Data analysis; Solange Glasser: Conceptualization, Writing - Review & Editing.

Correspondence concerning this article should be addressed to Beat Meier, Department of Psychology, University of Bern, Fabrikstrasse 8, 3012 Bern, Switzerland. E-mail: beat.meier@unibe.ch

Abstract

One or two sentences providing a **basic introduction** to the field, comprehensible to a scientist in any discipline.

Keywords: keywords

Word count: X

Supplementary material: Absolute Pitch and Sound-Color-Synesthesia

```
theme_clean <- function() {
  theme_minimal(base_family = "Helvetica", base_size = 12) +
    theme(panel.grid.minor = element_blank(),
          plot.title = element_text(face = "bold"),
          axis.title = element_text(face = "bold"),
          strip.text = element_text(face = "bold", size = rel(1), hjust = 0),
          strip.background = element_rect(fill = "grey80", color = NA),
          # strip.background = element_rect(colour="grey80", fill="grey80"),
          legend.title = element_text(face = "bold"))
}
```

```
d <- readxl::read_excel("./data/APfor Modelling.xlsx") |>
  filter(Experimentteil != "nurfarbe") |>
  mutate(
    ID = as_factor(Subject),
    group3 = as_factor(Group),
    syn = as_factor(Syn),
    test = as_factor(Experimentteil),
    time = as.ordered(as.numeric(test)),
    oldnew = as_factor(oldnew),
    item = as_factor(Item),
    rating = ordered(response),
    oldnew = fct_recode(oldnew, old = "alt", new = "new"),
    oldnew = fct_relevel(oldnew, "new"),
```

```
    triplet = ifelse(oldnew == "new", -1, 1),
    syn = fct_recode(syn, syn = "1", nosyn = "0")
  )

d <- d |>
  unite(col = group4, group3, syn) |>
  transmute(
    group4 = as_factor(group4),
    group4 = fct_recode(group4,
      control = "KG_nosyn",
      relpitch = "RP_nosyn",
      abspitch = "AP_nosyn",
      syn = "AP_syn"
    ),
    group4 = fct_relevel(
      group4, "control", "relpitch",
      "abspitch", "syn"
    )
  ) |>
  bind_cols(d) |>
  mutate(
    group3 = fct_recode(group3,
      control = "KG",
      relpitch = "RP",
      abspitch = "AP"
    ),
    group3 = fct_relevel(
```

```
      group3, "control", "relpitch",
      "abspitch"
    ),
    group2 = as_factor(case_when(
      group3 == "abspitch" | group3 == "relpitch" ~ "musician",
      group3 == "control" ~ "control"
    )),
    group2 = fct_relevel(
      group2, "control", "musician"
    ),
    group3alt = as_factor(case_when(
      group4 == "abspitch" | group4 == "relpitch" ~ "musician",
      group4 == "syn" ~ "syn",
      group4 == "control" ~ "control"
    )),
    group3alt = fct_relevel(
      group3alt, "control", "musician", "syn"
    ),
    test = fct_recode(test,
      colors = "nurfarbe2",
      tones = "nurton",
      combined = "beides"
    ),
    test = fct_relevel(
      test, "colors", "tones", "combined"
    )
  ) |>
```

```r
  select(
    ID, group2, group3, group3alt, group4, syn, test,
    time, oldnew, triplet, item, response,
    rating
  )




## remove subject 1305 (control) -> synaesthesia? ----

d <- d |>
  filter(!(ID %in% 10305)) |>
  mutate(ID = droplevels(ID))
```

In this study, subjects in 3 groups (non-musician controls, musicians with relative pitch, musicians with absolute pitch) gave confidence judgments on previously learned or previously unseen (old/new) triplets of stimuli. Confidence ratings were given on a 5-point response scale (1-5). Subjects were tested in three conditions (colours, tones, and colours and tones combined) with both old and new stimuli, resulting in a $3x3x2$ mixed design.

It was subsequently discovered that 7 of the musicians with absolute pitch were also synaesthetes; this subgroup was analyzed separately, as a $4x3x2$ design with the between factor group membership and the within factor experimental condition.

In the original three groups, there were 19, 22 and 24 subjects, respectively.

```r
d |>
  group_by(group3) |>
  summarise(n = n_distinct(ID))
```

```
## # A tibble: 3 x 2
##   group3       n
##   <fct>    <int>
## 1 control     19
## 2 relpitch    22
## 3 abspitch    24
```

```r
# d |>
#   group_by(group3) %>%
#   distinct(ID) %>%
#   count()
```

Out of the 24 subjects with absolute pitch, 7 were synaesthetes.

```r
d |>
  group_by(group4) |>
  summarise(n = n_distinct(ID))
```

```
## # A tibble: 4 x 2
##   group4       n
##   <fct>    <int>
## 1 control     19
## 2 relpitch    22
## 3 abspitch    17
## 4 syn          7
```

Learning Score

In this type of experiment, the traditional approach is to treat the ordinal response as a continuous variable, and to compute the mean response category for old and new items for each combination of test type/group, and then compute a learning score as the difference in mean response to old and new items.

However, treating an ordinal response as a continuous variable is associated with several problems (Liddell & Kruschke, 2018). While the categories have an ordering, but it is unknown what the *psychological distance* between those categories is, and whether distances between categories are the same across subjects. An alternative approach is to use an ordered regression model, in which it is assumed that the observed variable Y originates from the categorization of a latent continuous variable $\tilde{Y}$. There are K thresholds τ_k , which partition $\tilde{Y}$ into $K+1$ observable, ordered categories of Y.

Main points

1) Musicians (with both relative pitch and absolute pitch) are better at learning triplets than non-musicians (controls).

2) There is no difference between absolute pitch and relative pitch. Having absolute pitch confers no advantage over relative pitch.

3) Any advantage in the recognition task is due to synaesthesia.

4) Absolute pitch confers an advantage in associative learning (color-tone), compared to relative pitch.

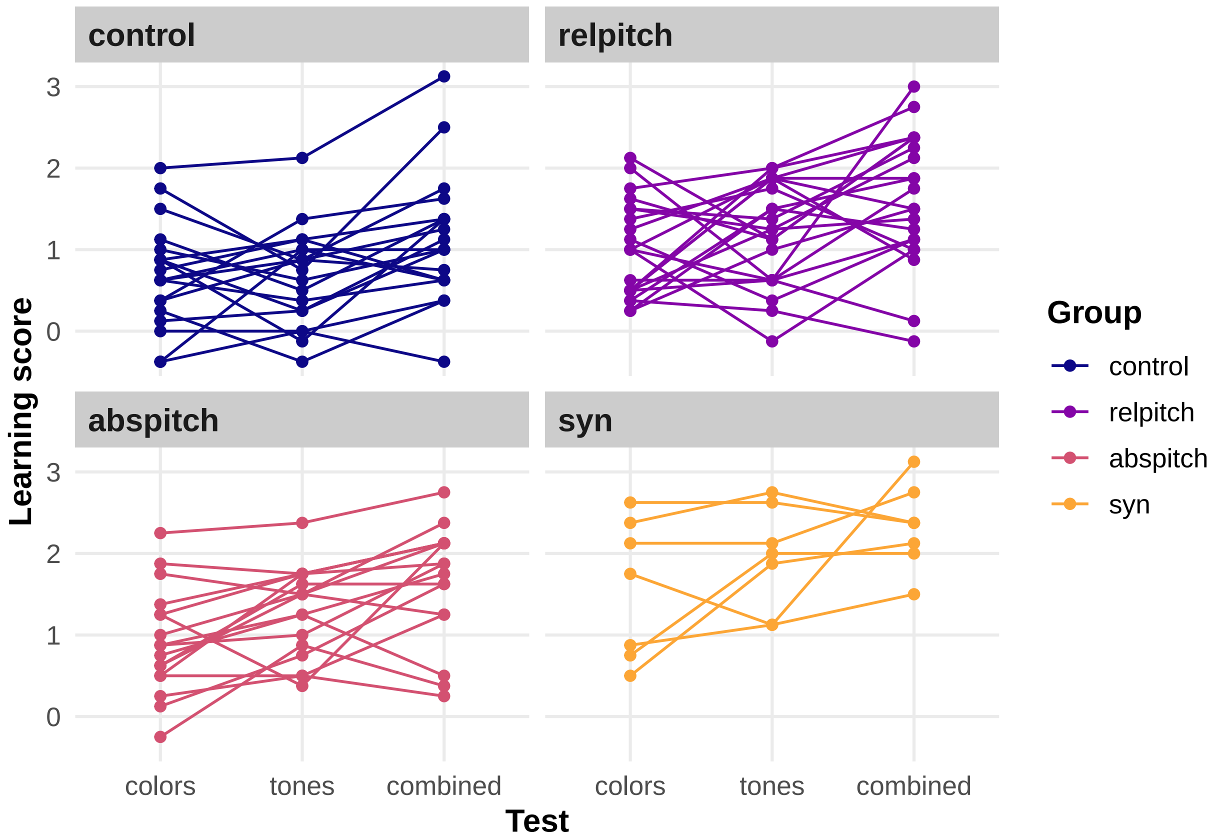
5) Dissociation: There is no difference between abs pitch and synaesthesia in the colour memory task.

6) Memory of colour is very accurate (but it is not synaesthesia that leads to very good color reproducibility).

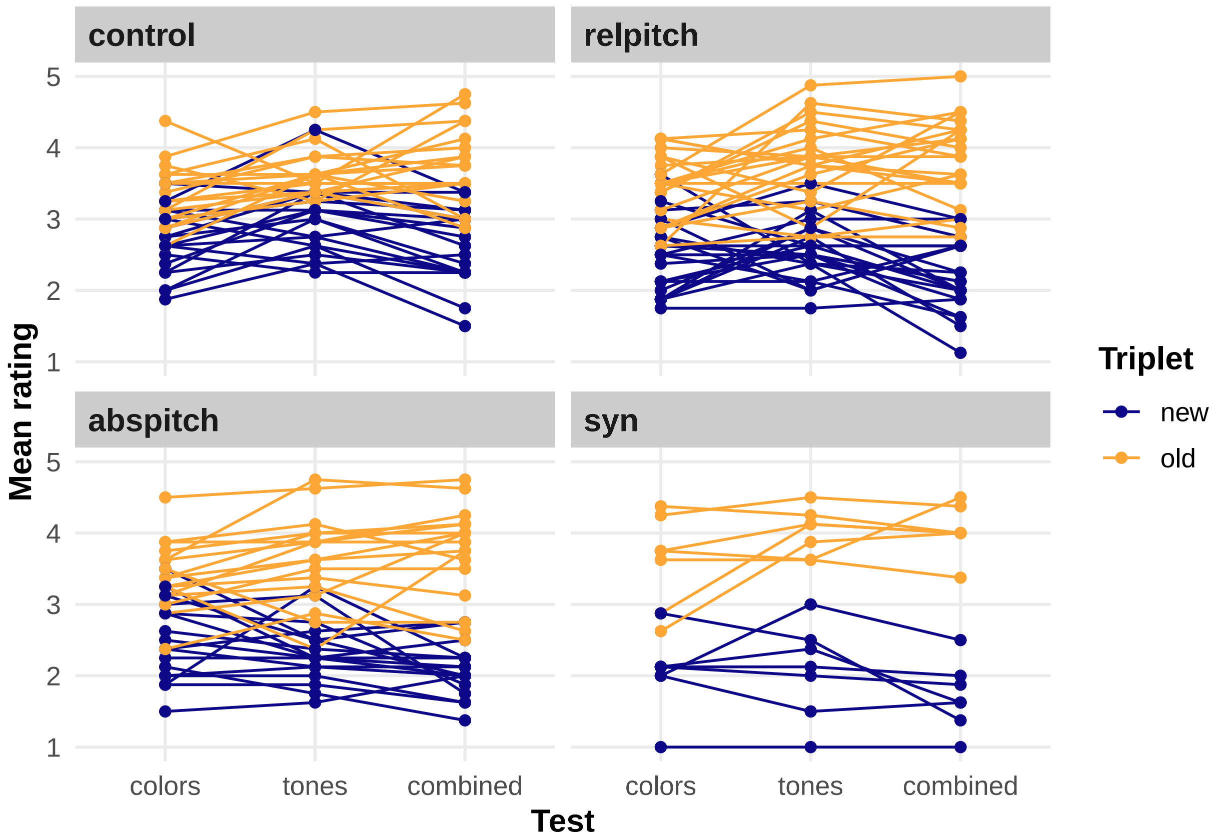
Exploratory Data Analysis

```
se <- function(x) sd(x) / sqrt(length(x))
funs <- list(mean = mean, sd = sd, se = se)
```

```
point_estimates <- d |>
  group_by(ID, group4, test, oldnew) |>
  summarise(mean = mean(response)) |>
  spread(oldnew, mean) |>
  mutate(score = old - new)



point_estimates |>
  ggplot(aes(x = test, y = score, color = group4)) +
  geom_line(aes(group = ID)) +
  geom_point() +
  facet_wrap(~group4) +
  # scale_color_okabe_ito() +
  scale_color_viridis_d(direction = 1, option = "C", end = .80) +
  labs(x = "Test", y = "Learning score", color = "Group") +
  theme_clean()
```




```
d %>%
  group_by(ID, group4, test, oldnew) %>%
  summarise(mean = mean(response)) %>%
  ggplot(aes(x = test, y = mean, color = oldnew)) +
  geom_line(aes(group = interaction(ID, oldnew))) +
  geom_point() +
  facet_wrap(~group4) +
  scale_color_viridis_d(direction = 1, option = "C", end = .80) +
  labs(x = "Test", y = "Mean rating", color = "Triplet") +
  theme_clean()
```



In the following, we compute the learning scores for each individual subject in each test condition as the the difference in mean response to old and new items, `mean(old) - mean(new)`. Positive learning scores thus indicate that the subject gave higher ratings to previously seen triplets than to unseen triplets. Higher scores are interpreted as greater being due to greater learning of the triplets; subjects are able to confidently state that they have previously seen old triplets, whilst being able to reject unseen triplets.

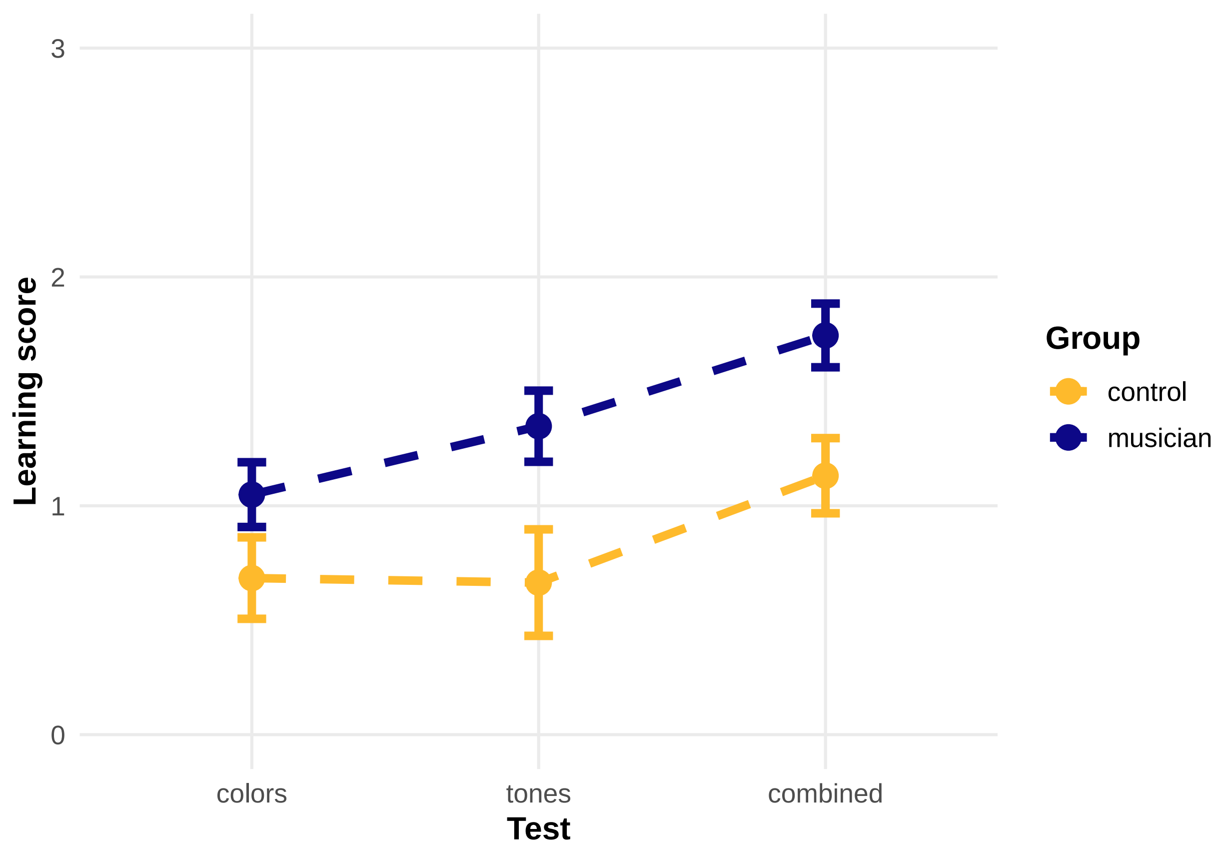
All figures show mean learning scores in all three conditions, aggregated over subjects, with within-subjects confidence intervals (Morey, 2008).

Musicians vs controls

```
sdtdata_2 <- d |>
  group_by(ID, group2, test, oldnew) |>
  summarise(mean = mean(response)) |>
```

```
  pivot_wider(names_from = oldnew, values_from = mean) |>
  mutate(score = old - new)

sdtdata_2_agg <- sdtdata_2 |>
  drop_na() |>
  group_by(group2, test) |>
  summarise(across(score, funs, .names = "{.fn}"))

sdtdata_2_agg_within <- sdtdata_2 |>
  Rmisc::summarySEwithin(
    measurevar = "score",
    betweenvars = "group2",
    withinvars = "test",
    idvar = "ID",
    na.rm = FALSE,
    conf.interval = 0.95
  )

sdtdata_2_agg_within <- sdtdata_2_agg_within |>
  mutate(mean = pull(sdtdata_2_agg, mean))

sdtdata_2_agg_within |>
  ggplot(aes(x = test, y = mean, color = group2)) +
  geom_line(aes(group = group2), linewidth = 1.5, linetype = "dashed") +
  geom_point(size = 4) +
  geom_errorbar(
    aes(
```

```
    ymin = mean - ci,
    ymax = mean + ci
  ),
  width = 0.1, linewidth = 1.5
) +
scale_color_viridis_d(direction = -1, option = "C", end = .85) +
ylim(0, 3) +
labs(x = "Test", y = "Learning score", color = "Group") +
theme_clean() +
guides(color = guide_legend(
  title = "Group",
  title.position = "top"
))
```

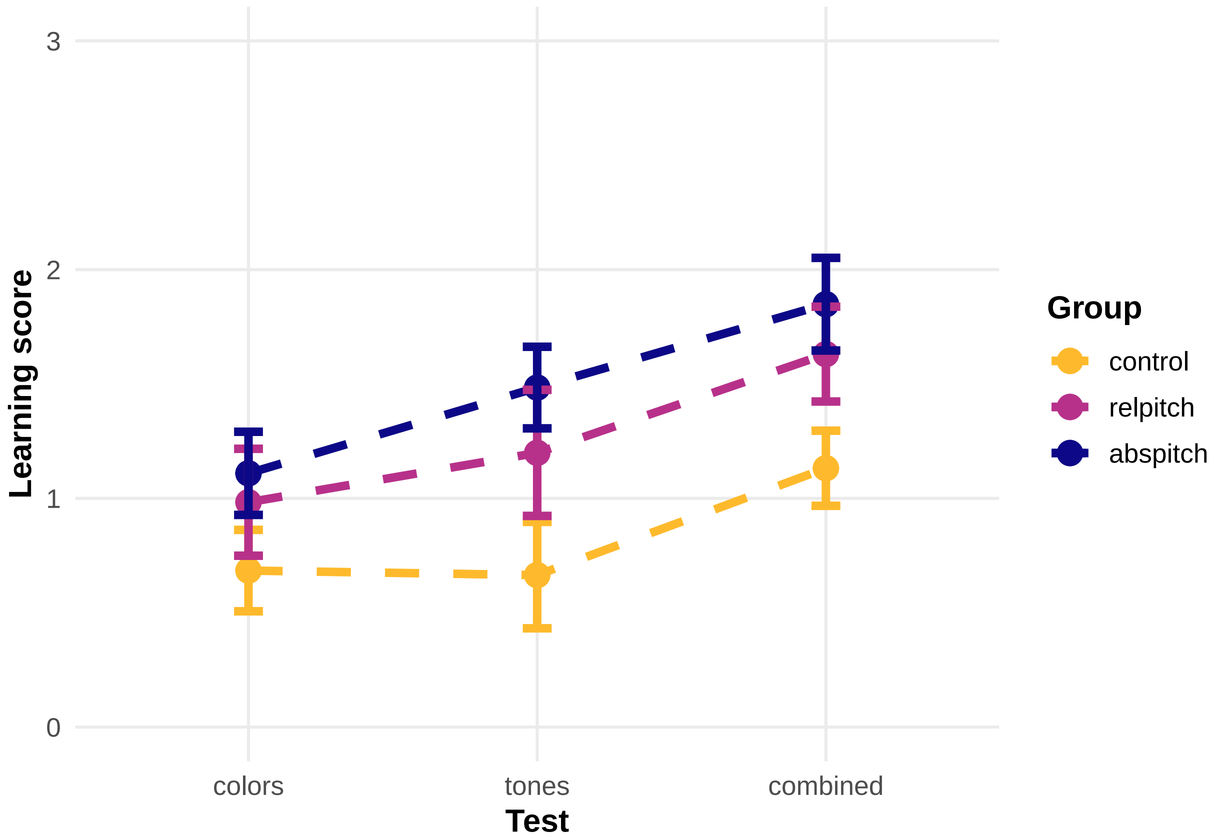


Musicians, consisting of the groups with relative and absolute pitch (also containing the synaesthets) are clearly better at all three recognition task than controls. What is also noticeable is that musicians perform better in the tasks involving tones, whereas controls need both tones and colours combined to perform better.

Absolute pitch, relative pitch and controls

```
sdtdata_3 <- d |>
  group_by(ID, group3, test, oldnew) |>
  summarise(mean = mean(response)) |>
  pivot_wider(names_from = oldnew, values_from = mean) |>
  mutate(score = old - new)

sdtdata_3_agg <- sdtdata_3 |>
  drop_na() |>
  group_by(group3, test) |>
  summarise(across(score, funs,
    .names = "{.fn}"
  ))

sdtdata_3_agg_within <- sdtdata_3 |>
  Rmisc::summarySEwithin(
    measurevar = "score",
    betweenvars = "group3",
    withinvars = "test",
    idvar = "ID",
    na.rm = FALSE,
    conf.interval = 0.95
  )
```

```r
sdtdata_3_agg_within <- sdtdata_3_agg_within |>
  mutate(mean = pull(sdtdata_3_agg, mean))

sdtdata_3_agg_within |>
  ggplot(aes(x = test, y = mean, color = group3)) +
  geom_line(aes(group = group3), linewidth = 1.5, linetype = "dashed") +
  geom_point(size = 4) +
  geom_errorbar(
    aes(
      ymin = mean - ci,
      ymax = mean + ci
    ),
    width = 0.1, linewidth = 1.5
  ) +
  scale_color_viridis_d(direction = -1, option = "C", end = .85) +
  ylim(0, 3) +
  labs(x = "Test", y = "Learning score", color = "Group") +
  theme_clean() +
  guides(color = guide_legend(
    title = "Group",
    title.position = "top"
  ))
```

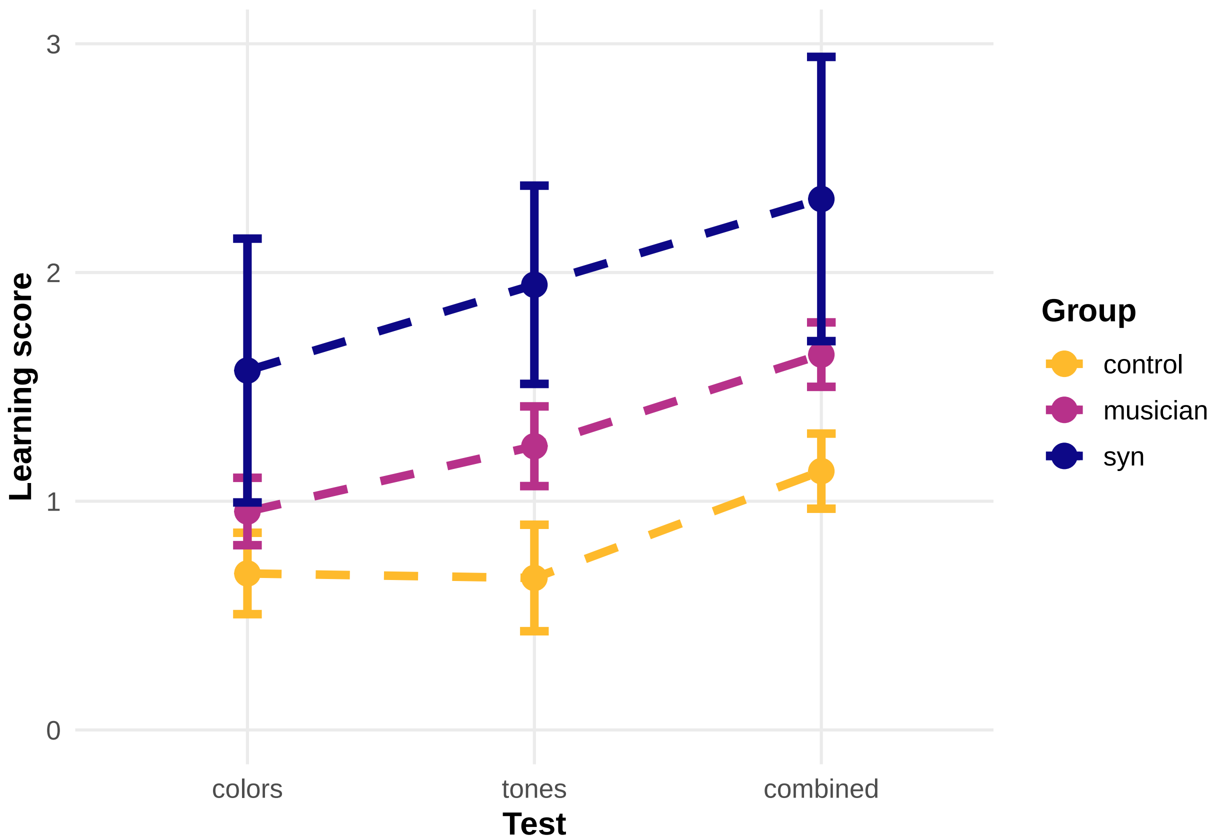



```r
sdtdata_3alt <- d |>
  group_by(ID, group3alt, test, oldnew) |>
  summarise(mean = mean(response)) |>
  pivot_wider(names_from = oldnew, values_from = mean) |>
  mutate(score = old - new)

sdtdata_3alt_agg <- sdtdata_3alt |>
  drop_na() |>
  group_by(group3alt, test) |>
  summarise(across(score, funs,
    .names = "{.fn}"
  ))
```

```
sdtdata_3alt_agg_within <- sdtdata_3alt |>
  Rmisc::summarySEwithin(
    measurevar = "score",
    betweenvars = "group3alt",
    withinvars = "test",
    idvar = "ID",
    na.rm = FALSE,
    conf.interval = 0.95
  )

sdtdata_3alt_agg_within <- sdtdata_3alt_agg_within |>
  mutate(mean = pull(sdtdata_3alt_agg, mean))

sdtdata_3alt_agg_within |>
  ggplot(aes(x = test, y = mean, color = group3alt)) +
  geom_line(aes(group = group3alt), linewidth = 1.5, linetype = "dashed") +
  geom_point(size = 4) +
  geom_errorbar(
    aes(
      ymin = mean - ci,
      ymax = mean + ci
    ),
    width = 0.1, linewidth = 1.5
  ) +
  scale_color_viridis_d(direction = -1, option = "C", end = .85) +
  ylim(0, 3) +
```

```
  labs(x = "Test", y = "Learning score", color = "Group") +
  theme_clean() +
  guides(color = guide_legend(
    title = "Group",
    title.position = "top"
  ))
```

93

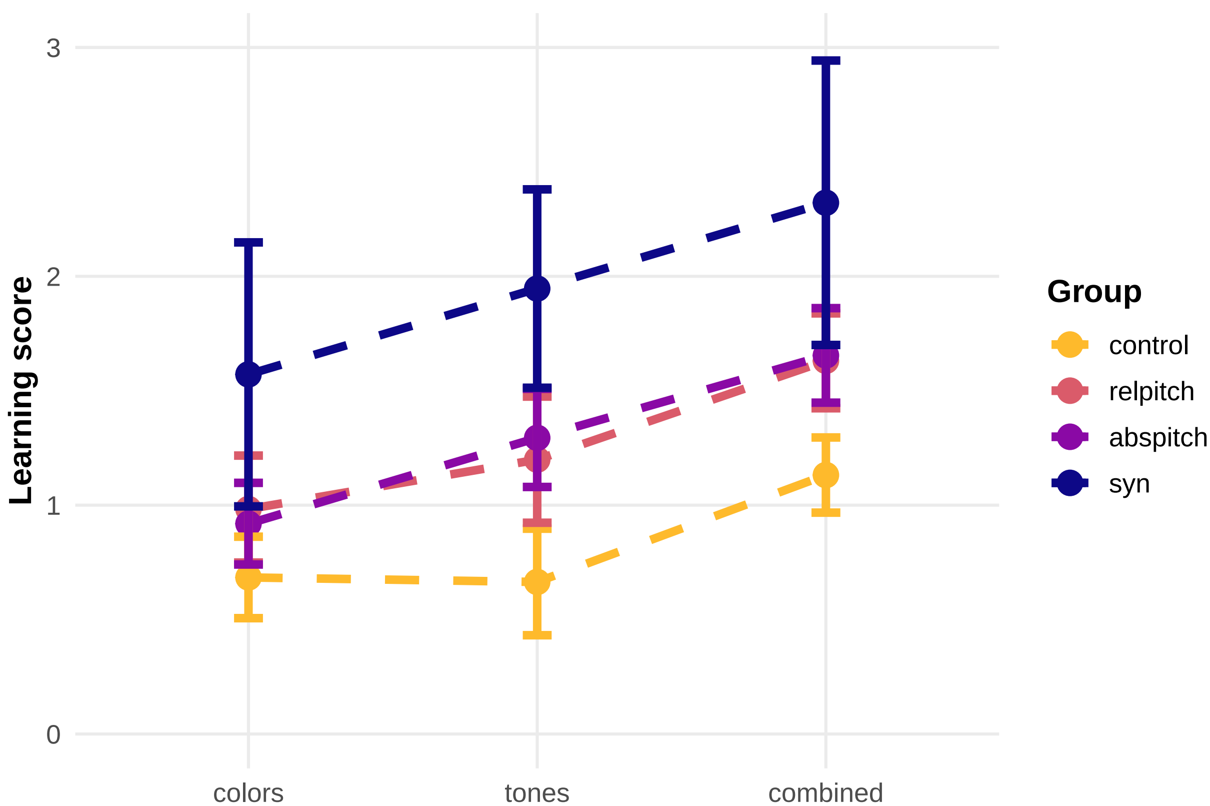


94 ## Synaesthetes, absolute pitch, relative pitch and controls

```
sdtdata_4 <- d |>
  group_by(ID, group4, test, oldnew) |>
  summarise(mean = mean(response)) |>
  pivot_wider(names_from = oldnew, values_from = mean) |>
  mutate(score = old - new)
```

```
sdtdata_4_agg <- sdtdata_4 |>
  drop_na() |>
  group_by(group4, test) |>
  summarise(across(score, funs,
    .names = "{.fn}"
  ))

sdtdata_4_agg_within <- sdtdata_4 |>
  Rmisc::summarySEwithin(
    measurevar = "score",
    betweenvars = "group4",
    withinvars = "test",
    idvar = "ID",
    na.rm = FALSE,
    conf.interval = 0.95
  )

sdtdata_4_agg_within <- sdtdata_4_agg_within |>
  mutate(mean = pull(sdtdata_4_agg, mean))

sdtdata_4_agg_within |>
  ggplot(aes(x = test, y = mean, color = group4)) +
  geom_line(aes(group = group4), linewidth = 1.5, linetype = "dashed") +
  geom_point(size = 4) +
  geom_errorbar(
    aes(
      ymin = mean - ci,
```

```
    ymax = mean + ci
  ),
  width = 0.1, linewidth = 1.5
) +
scale_color_viridis_d(direction = -1, option = "C", end = .85) +
ylim(0, 3) +
labs(x = "", y = "Learning score", color = "Group") +
theme_clean() +
guides(color = guide_legend(
  title = "Group",
  title.position = "top"
))
```



95

Regression models

FIrst, we create a data set by taking the mean rating for each person in each test for both old and new items. Then we compute the *learning score* by subtracting the mean rating for new items from the mean rating for old items.

```
dd <- d |>
  group_by(ID, group4, test, oldnew) |>
  summarise(mean = mean(response)) |>
  ungroup() |>
  pivot_wider(names_from = oldnew, values_from = mean) |>
  mutate(score = old - new)
```

The learning score is bounded in $[-4, 4]$. We will initially use a linear regression to model the outcome, a beta regression would be a better choice for a bounded variable.

```
normalize <- function(x, max_x = 4) {
  min_x <- -max_x
  (x - min_x)/(max_x - min_x)
  }


dd <- dd |>
  mutate(normscore = normalize(score))
```

2 groups

```
priors <- prior(normal(0, 1), class = Intercept) +
  prior(normal(0, 1), class = b) +
  prior(student_t(3, 0, 1), class = sd, group = ID)


fit_ls_2_groups <- brm(score ~ group2 * test + (1 | ID),
```

```
                      family = gaussian,
                      prior = priors,
                      data = sdtdata_2,
                      chains = 4, iter = 2000, cores = 4,
                      backend = "cmdstanr",
                      file = here::here("models/fit_ls_2_groups"),
                      save_model = here::here("stancode/fit_ls_2_groups.stan")
                      ) |>
  add_criterion("loo")
```

```
priors <- prior(normal(0, 1), class = Intercept) +
  prior(normal(0, 1), class = b) +
  prior(student_t(3, 0, 1), class = sd, group = ID)

fit_ls_3_groups <- brm(score ~ group3 * test + (1 | ID),
                      family = gaussian,
                      prior = priors,
                      data = sdtdata_3,
                      chains = 4, iter = 2000, cores = 4,
                      backend = "cmdstanr",
                      file = here::here("models/fit_ls_3_groups"),
                      save_model = here::here("stancode/fit_ls_3_groups.stan")
                      ) |>
  add_criterion("loo")
```

3 groups (controls, musicians, synaesthetes)

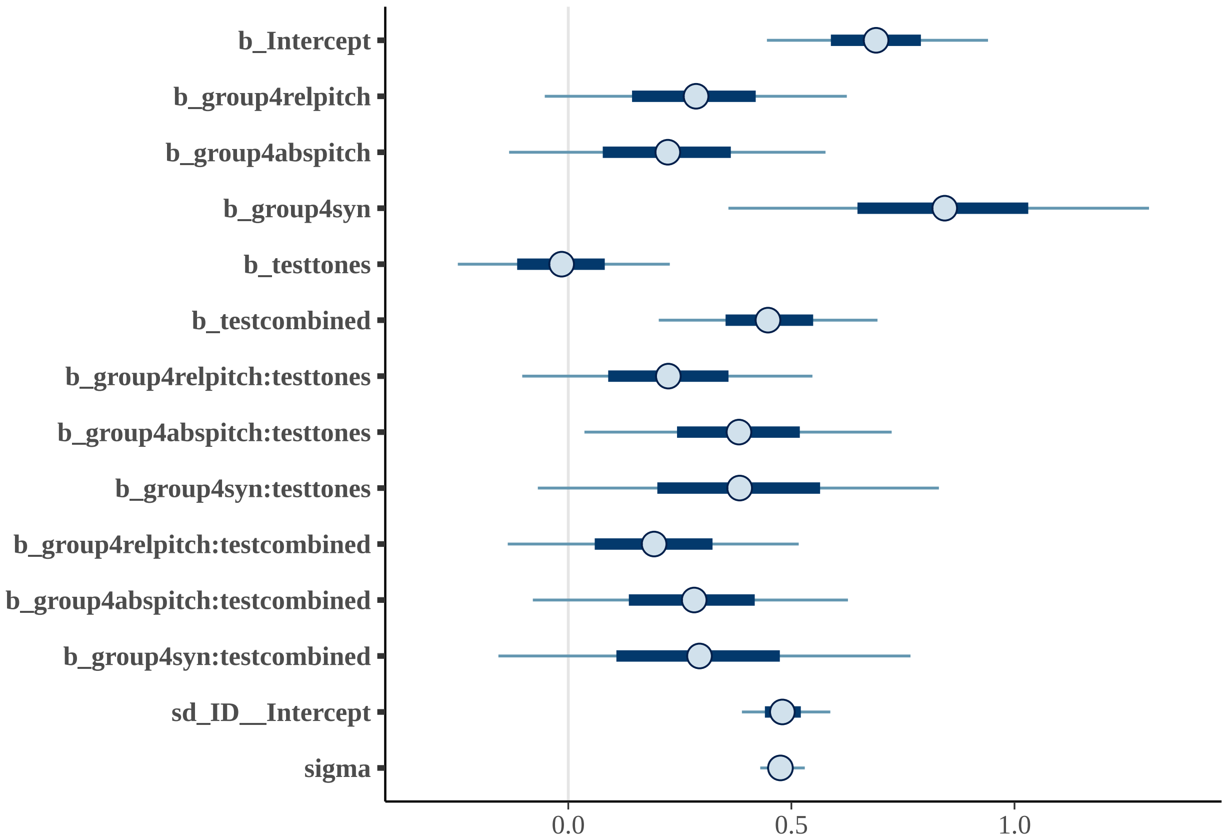
```r
priors <- prior(normal(0, 1), class = Intercept) +
  prior(normal(0, 1), class = b) +
  prior(student_t(3, 0, 1), class = sd, group = ID)

fit_ls_3_groups_alt <- brm(score ~ group3alt * test + (1 | ID),
                      family = gaussian,
                      prior = priors,
                      data = sdtdata_3alt,
                      chains = 4, iter = 2000, cores = 4,
                      backend = "cmdstanr",
                      file = here::here("models/fit_ls_3_groups_alt"),
                      save_model = here::here("stancode/fit_ls_3_groups_alt.stan")
                      ) |>
  add_criterion("loo")
```

4 groups

```r
priors <- prior(normal(0, 1), class = Intercept) +
  prior(normal(0, 1), class = b) +
  prior(student_t(3, 0, 1), class = sd, group = ID)

fit_ls_4_groups <- brm(score ~ group4 * test + (1 | ID),
                      family = gaussian,
                      prior = priors,
                      data = dd,
                      chains = 4, iter = 2000, cores = 4,
                      backend = "cmdstanr",
```

```
                      file = here::here("models/fit_ls_4_groups"),
                      save_model = here::here("stancode/fit_ls_4_groups.stan")
                      ) |>
  add_criterion("loo")
```

```
mcmc_plot(fit_ls_4_groups)
```



Model comparison

The models cannot be distringuished based on their out-of-sample predictive accuracy (loo). In order to perform meaningful model comparisons, i.e. hypothesis tests, we need a different approach, possibly based on posterior predictive checks.

```
loo_compare(fit_ls_2_groups,
            fit_ls_3_groups,
            fit_ls_3_groups_alt,
```

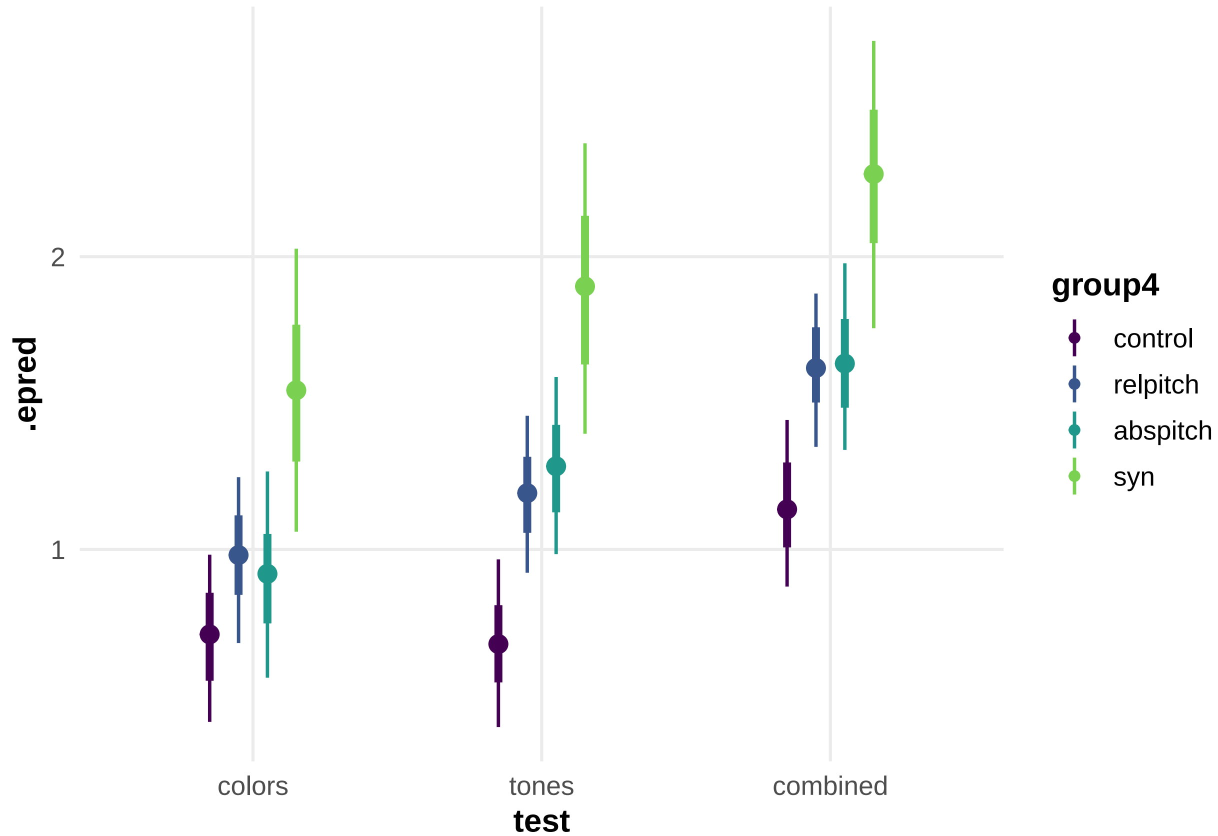
```
            fit_ls_4_groups)
```

```
##                     elpd_diff se_diff
## fit_ls_2_groups      0.0       0.0
## fit_ls_3_groups_alt -1.6       1.7
## fit_ls_3_groups     -1.8       1.3
## fit_ls_4_groups     -4.4       2.0
```

```
epred_ls_4_groups <- dd |>
  data_grid(group4, test) |>
  add_epred_draws(fit_ls_4_groups,
                  re_formula = ~ID,
                  ndraws = 500)
```

```
epred_ls_4_groups |>
  ggplot(aes(x = test, y = .epred, color = group4)) +
  stat_pointinterval(position = position_dodge(width = .4)) +
  scale_size_continuous(guide = FALSE) +
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  theme_clean()
```

```
## Warning: The `guide` argument in `scale_*()` cannot be `FALSE`. This was deprecated i
## ggplot2 3.3.4.
## i Please use "none" instead.
```

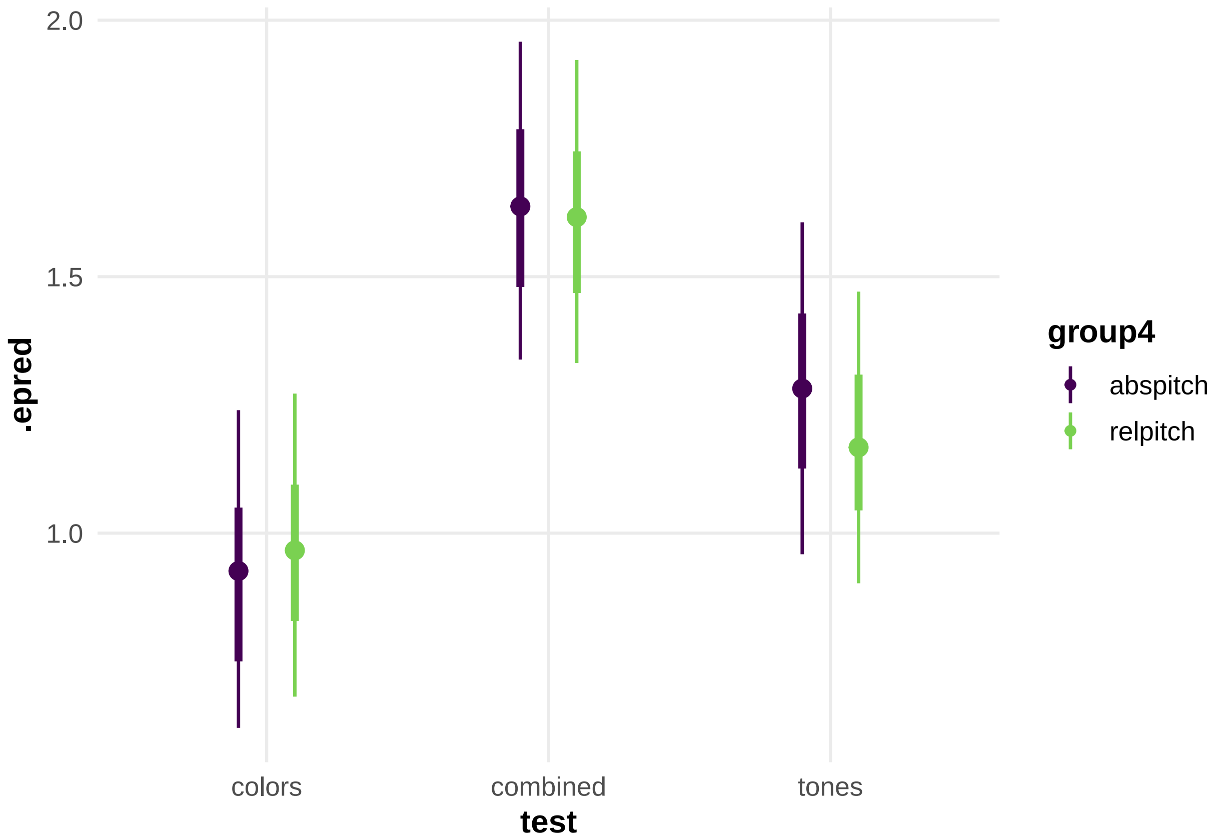


The above figure shoes the expectations of the posterior predictive distributions for each group in each of the three test conditions. THe synaesthesia group has a consistently higher expected learning score over all tests. In this case, however, we are interested in the comparison between musicians with relative and absolute pitch for all three tests.

```
epred_ls_4_groups_contrast <-
  expand_grid(group4 = c("relpitch", "abspitch"),
              test = levels(dd$test)) |>
  add_epred_draws(fit_ls_4_groups,
                  re_formula = ~ID,
                  ndraws = 500)
```

```
epred_ls_4_groups_contrast |>
    ggplot(aes(x = test, y = .epred, color = group4)) +
```

```
  stat_pointinterval(position = position_dodge(width = .4)) +
  scale_size_continuous(guide = FALSE) +
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  theme_clean()
```

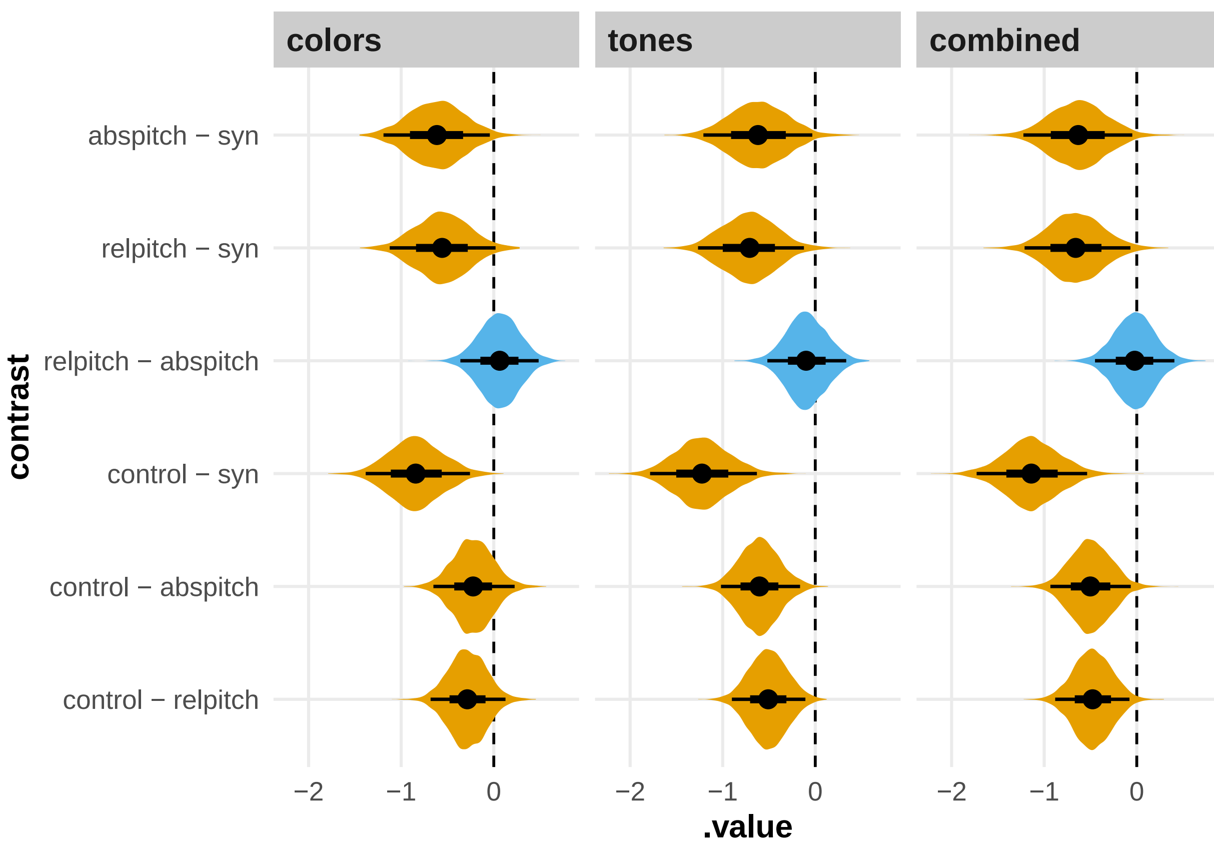



```
contrasts_rel_abs <- fit_ls_4_groups |>
  emmeans::emmeans(~group4 | test) |>
  emmeans::contrast("pairwise") |>
  gather_emmeans_draws() |>
  mutate(indicator = if_else(contrast == "relpitch - abspitch", 1, 0))
```

```
zero_color <- ggokabeito::palette_okabe_ito()[1]


contrasts_rel_abs |>
```

```r
ggplot(aes(y = contrast, x = .value, fill = as_factor(indicator))) +
geom_vline(xintercept = 0, color = "black", linetype = "dashed") +
geom_eye() +
scale_fill_okabe_ito(guide="none") +
# stat_summary(aes(group = NA), fun.y = mean, geom = "line") +
facet_grid(~ test) +
theme_clean()
```

```
## Warning: 'geom_eye' is deprecated.
## Use 'stat_eye' instead.
## See help("Deprecated") and help("tidybayes-deprecated").
```



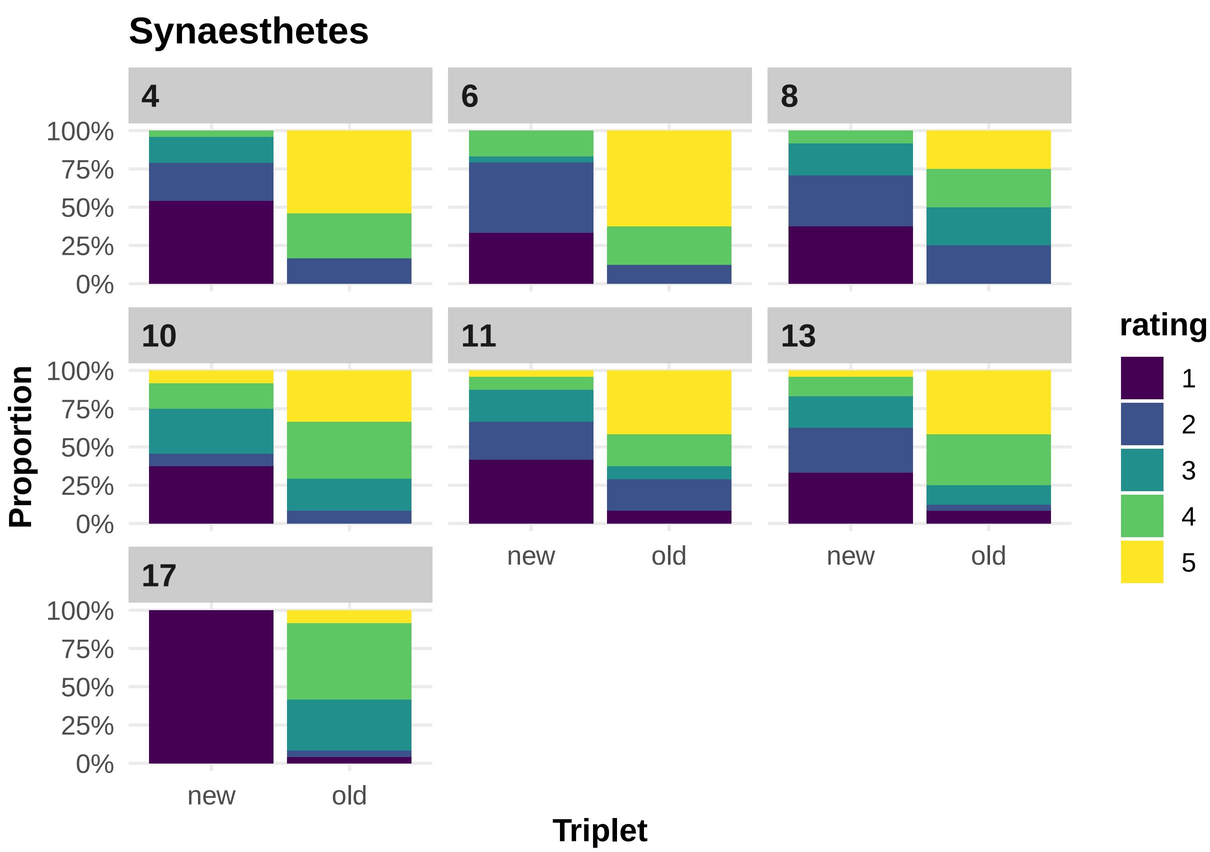
The above figure shows the posterior distributions of the pairwise group differences for each test, in the model allowing for 4 separate groups. When allowing for a separate group for musicians with synaesthesia, the expected difference between musicians with relative and absolute pitch is consistenly centred on zero for all testing conditions (blue distributions).

Individial responses

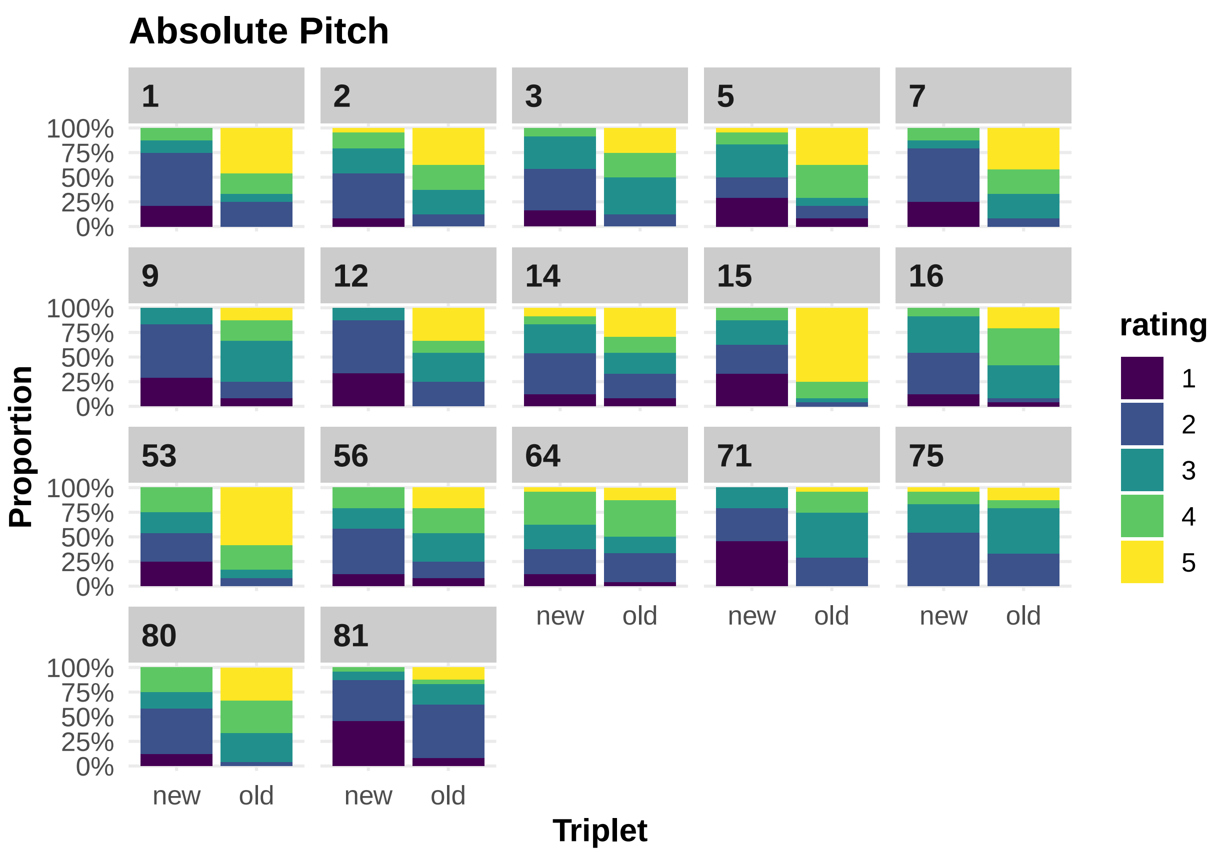
The learning score is computed by averaging over individual ratings. We thus lose information about the subject- and group specific usage of individual ratings. Of particular interest are the extreme rating categories, which denote extreme certainty that either *new* items have never been seen (rating 1) or that *old* items have been previously seen (rating 5).

From visual inspection of individual subjects' ratings it is apparent that synaesthetes tend to use the extreme categories more often; in particular, they more frequently reject previously unseen items with a rating of 1.

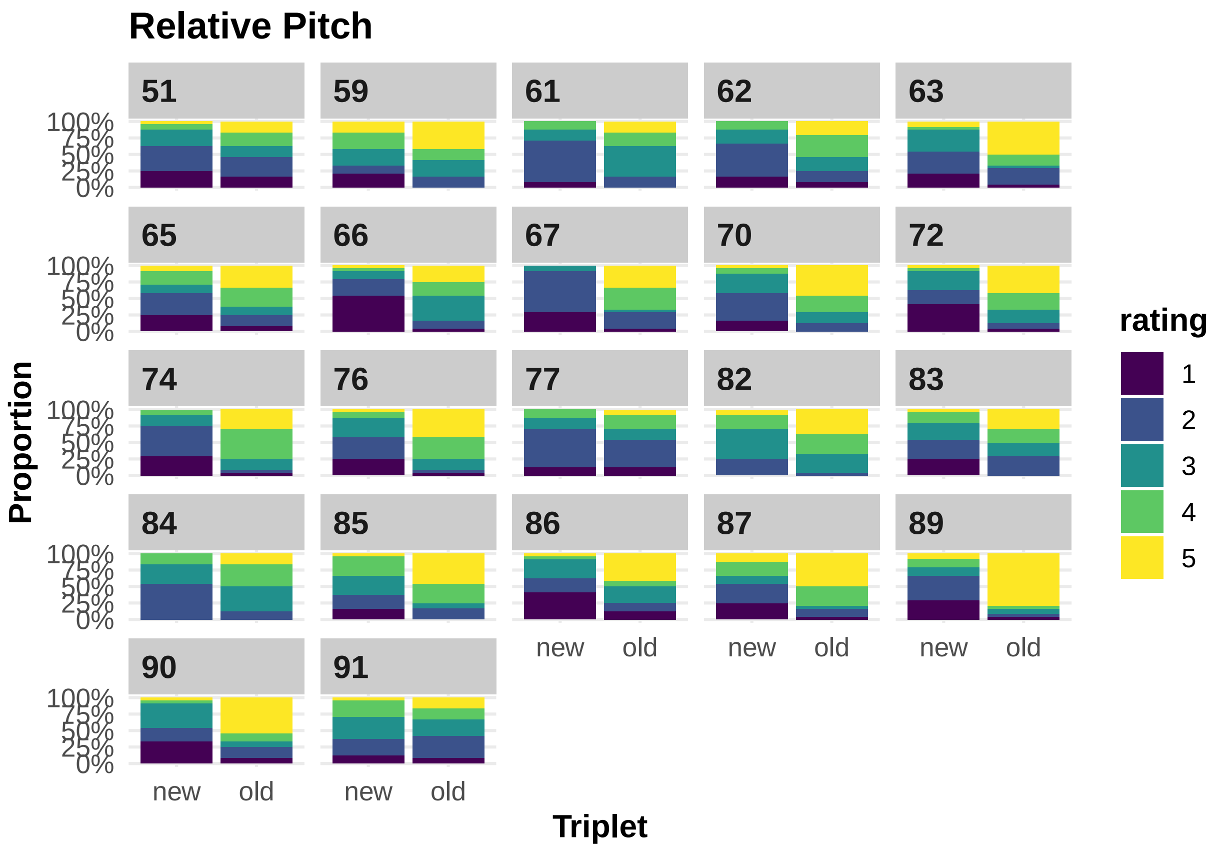
```
d |>
  filter(group4 == "syn") |>
  mutate(ID = fct_drop(ID)) |>
  ggplot(aes(x = oldnew, fill = rating)) +
  geom_bar(position = position_fill(reverse = TRUE)) +
  scale_y_continuous(labels = scales::percent) +
  facet_wrap(~ID) +
  ylab("Proportion") +
  labs(x = "Triplet", y = "Proportion", color = "Rating") +
  theme_clean() +
  ggtitle("Synaesthetes")
```




```r
d |>
  filter(group4 == "abspitch") |>
  mutate(ID = fct_drop(ID)) |>
  ggplot(aes(x = oldnew, fill = rating)) +
  geom_bar(position = position_fill(reverse = TRUE)) +
  scale_y_continuous(labels = scales::percent) +
  facet_wrap(~ID) +
  ylab("Proportion") +
  labs(x = "Triplet", y = "Proportion", color = "Rating") +
  theme_clean() +
  ggtitle("Absolute Pitch")
```

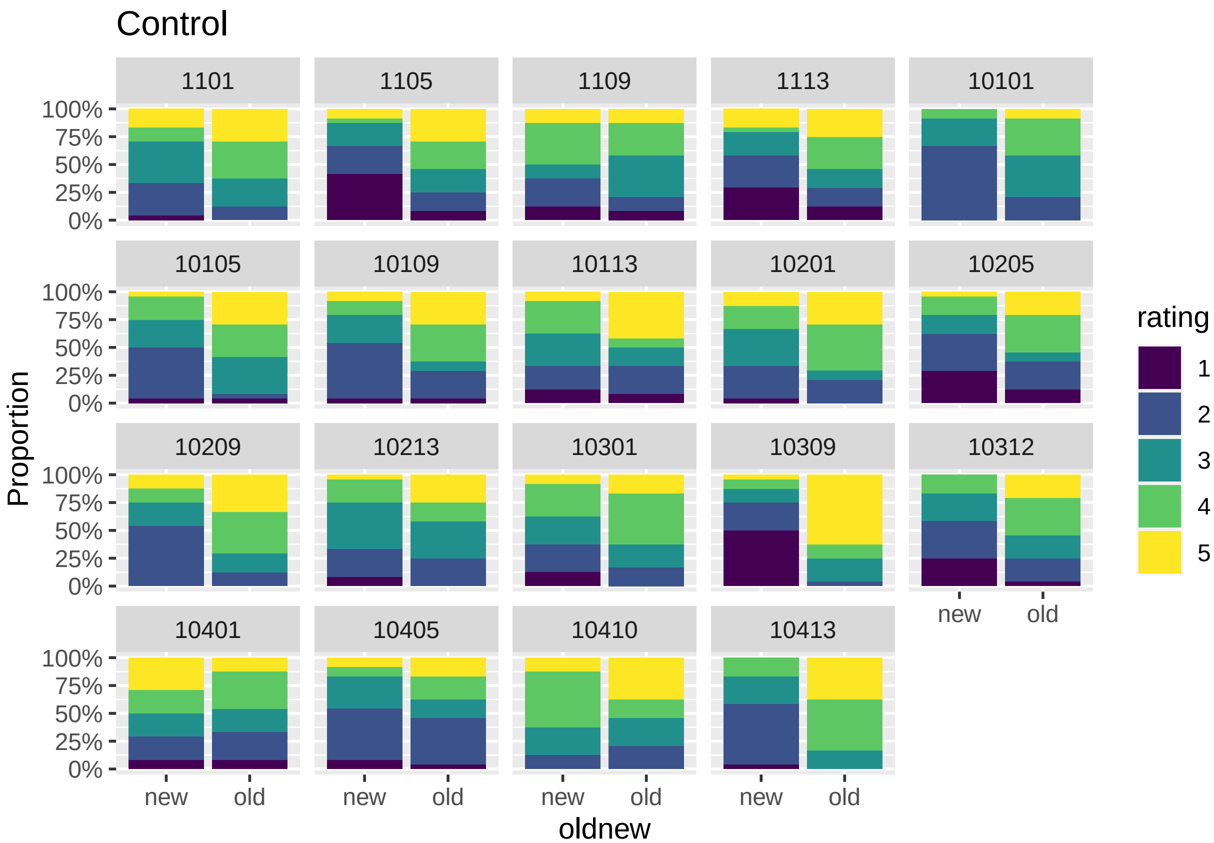



```
d |>
  filter(group4 == "relpitch") |>
  mutate(ID = fct_drop(ID)) |>
  ggplot(aes(x = oldnew, fill = rating)) +
  geom_bar(position = position_fill(reverse = TRUE)) +
  scale_y_continuous(labels = scales::percent) +
  # scale_x_continuous(breaks = 1:5) +
  facet_wrap(~ID) +
  ylab("Proportion") +
  labs(x = "Triplet", y = "Proportion", color = "Rating") +
  theme_clean() +
  ggtitle("Relative Pitch")
```



144

```
d |>
  filter(group4 == "control") |>
  mutate(ID = fct_drop(ID)) |>
  ggplot(aes(x = oldnew, fill = rating)) +
  geom_bar(position = position_fill(reverse = TRUE)) +
  scale_y_continuous(labels = scales::percent) +
  facet_wrap(~ID) +
  ylab("Proportion") +
  ggtitle("Control")
```



Ordinal regression models

Theory

- Although ordinal data are not metric, they are often analyzed using methods that assume metric responses. This practice may lead to serious errors in inference (Liddell & Kruschke, 2018).
- Ordinal variables: categories have an ordering, but it is unknown
  - what the **psychological distance** between them is
  - whether distances between categories are the same across participants

Ordinal regression

- Use the framework of signal detection (unequal variance SDT or logistic model with heteroscedastic error)
- Work with raw responses, instead of summarizing data
- Quantify uncertainty at all levels

- Allows multilevel model (shrinkage could be especially important due to low number of subjects)

Unequal Variance (logistic) SDT Model

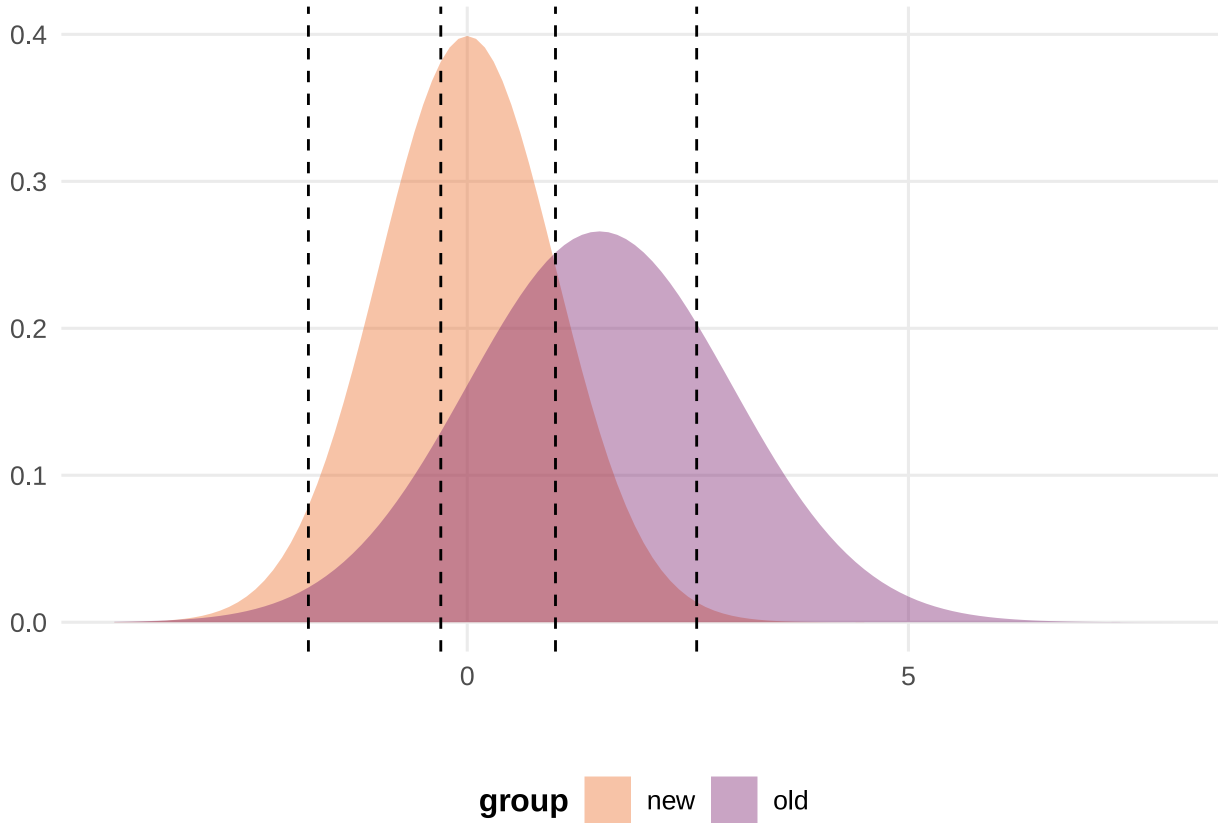
- Item is either old or new
- Subjects do not provide binary old or new responses, but instead give their responses on a 5-point rating scale
- Subjects rate their confidence in whether the item was old or new (actually, how frequently the item was presented)
- Subjects set a number of criteria for the ratings, such that greater evidence is required for 5-responses, than 4-responses, for example.

$$P(Y \leq k|X) = F\left(\frac{c_k - dX}{\sigma_X}\right)$$

for $k = 1$ to $K - 1$, where

- K is the number of response categories
- Y is a response rating, taking on the values $k = 1$ to K
- F is a cumulative distribution function
- c_k are response criteria
- σ_X is the standard deviation of the latent distribution

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



The idea is that each individual sets thresholds on the latent scale, depending on the type of the test. We allow the variance of the internal representation to differ between old and new items. Thresholds are shown above as dashed lines. If the internal representation lies between thresholds τ_k and τ_{k+1}, the corresponding rating is chosen.

BRMS Models

Set initial values for sampling the thresholds, assuming initially that the thresholds are evenly distributed, i.e. each response category has the same probability of being chosen.

```
tibble(rating = 1:5) |>
  mutate(proportion = 1 / 5) |>
  mutate(cumulative_proportion = cumsum(proportion)) |>
  mutate(
    right_hand_threshold = qnorm(cumulative_proportion),
    right_hand_threshold_logit = qlogis(cumulative_proportion)
```

```
)
```

```
## # A tibble: 5 x 5
##   rating proportion cumulative_proportion right_hand_threshold right_hand_thre~1
##    <int>      <dbl>                 <dbl>                <dbl>             <dbl>
## 1      1        0.2                   0.2               -0.842            -1.39
## 2      2        0.2                   0.4               -0.253            -0.405
## 3      3        0.2                   0.6                0.253             0.405
## 4      4        0.2                   0.8                0.842             1.39
## 5      5        0.2                   1                Inf               Inf
## # ... with abbreviated variable name 1: right_hand_threshold_logit
```

2 groups

```
priors <- prior(normal(-1.39, 1), class = Intercept, coef = 1) +
  prior(normal(-0.405, 1), class = Intercept, coef = 2) +
  prior(normal(0.405, 1), class = Intercept, coef = 3) +
  prior(normal(1.39, 1), class = Intercept, coef = 4) +
  prior(normal(0, 1), class = b) +
  prior(normal(0, 1), class = b, dpar = "disc") +
  prior(student_t(3, 0, 1), class = sd, group = ID) +
  prior(lkj(2), class = cor, group = ID) +
  prior(student_t(3, 0, 1), class = sd, group = item)

inits <- list(Intercept = c(-1.39, -0.405, 0.405, 1.39))

formula <- bf(rating ~ oldnew * group2 * test +
  (1 + oldnew | ID) + (1 | item)) +
  lf(disc ~ 0 + oldnew * test +
```

```
    (1 + oldnew | ID) + (1 | item), cmc = FALSE)


fit_2_groups <- brm(formula,
  family = cumulative("logit"),
  data = d,
  prior = priors,
  init = rep(list(inits), 4),
  chains = 4, iter = 2000, cores = 4,
  backend = "cmdstanr",
  file = here::here("models/fit_2_groups"),
  save_model = "stancode/fit_2_groups.stan"
) |>
  add_criterion("loo")
```

```
priors <- prior(normal(-1.39, 1), class = Intercept, coef = 1) +
  prior(normal(-0.405, 1), class = Intercept, coef = 2) +
  prior(normal(0.405, 1), class = Intercept, coef = 3) +
  prior(normal(1.39, 1), class = Intercept, coef = 4) +
  prior(normal(0, 1), class = b) +
  prior(normal(0, 1), class = b, dpar = "disc") +
  prior(student_t(3, 0, 1), class = sd, group = ID) +
  prior(lkj(2), class = cor, group = ID) +
  prior(student_t(3, 0, 1), class = sd, group = item)


inits <- list(Intercept = c(-1.39, -0.405, 0.405, 1.39))
```

```
formula <- bf(rating ~ oldnew * group3 * test +
  (1 + oldnew | ID) + (1 | item)) +
  lf(disc ~ 0 + oldnew * test +
    (1 + oldnew | ID) + (1 | item), cmc = FALSE)

fit_3_groups <- brm(formula,
  family = cumulative("logit"),
  data = d,
  prior = priors,
  init = rep(list(inits), 4),
  chains = 4, iter = 2000, cores = 4,
  backend = "cmdstanr",
  file = here::here("models/fit_3_groups"),
  save_model = "stancode/fit_3_groups.stan"
) |>
  add_criterion("loo")
```

```
priors <- prior(normal(-1.39, 1), class = Intercept, coef = 1) +
  prior(normal(-0.405, 1), class = Intercept, coef = 2) +
  prior(normal(0.405, 1), class = Intercept, coef = 3) +
  prior(normal(1.39, 1), class = Intercept, coef = 4) +
  prior(normal(0, 1), class = b) +
  prior(normal(0, 1), class = b, dpar = "disc") +
  prior(student_t(3, 0, 1), class = sd, group = ID) +
  prior(lkj(2), class = cor, group = ID) +
  prior(student_t(3, 0, 1), class = sd, group = item)


inits <- list(Intercept = c(-1.39, -0.405, 0.405, 1.39))

formula <- bf(rating ~ oldnew * group3alt * test +
  (1 + oldnew | ID) + (1 | item)) +
  lf(disc ~ 0 + oldnew * test +
    (1 + oldnew | ID) + (1 | item), cmc = FALSE)


fit_3_groups_alt <- brm(formula,
  family = cumulative("logit"),
  data = d,
  prior = priors,
  init = rep(list(inits), 4),
  chains = 4, iter = 2000, cores = 4,
  backend = "cmdstanr",
```

```
  file = here::here("models/fit_3_groups_alt"),
  save_model = "stancode/fit_3_groups_alt.stan"
) |>
  add_criterion("loo")
```

3 Alternative Groups.

4 Groups

```
priors <- prior(normal(-1.39, 1), class = Intercept, coef = 1) +
  prior(normal(-0.405, 1), class = Intercept, coef = 2) +
  prior(normal(0.405, 1), class = Intercept, coef = 3) +
  prior(normal(1.39, 1), class = Intercept, coef = 4) +
  prior(normal(0, 1), class = b) +
  prior(normal(0, 1), class = b, dpar = "disc") +
  prior(student_t(3, 0, 1), class = sd, group = ID) +
  prior(lkj(2), class = cor, group = ID) +
  prior(student_t(3, 0, 1), class = sd, group = item)



inits <- list(Intercept = c(-1.39, -0.405, 0.405, 1.39))

formula <- bf(rating ~ oldnew * group4 * test +
  (1 + oldnew | ID) + (1 | item)) +
  lf(disc ~ 0 + oldnew * test +
    (1 + oldnew | ID) + (1 | item), cmc = FALSE)

fit_4_groups <- brm(formula,
  family = cumulative("logit"),
```

```r
  data = d,
  prior = priors,
  init = rep(list(inits), 4),
  chains = 4, iter = 2000, cores = 4,
  backend = "cmdstanr",
  file = here::here("models/fit_4_groups"),
  save_model = here::here("stancode/fit_4_groups.stan")
) |>
  add_criterion("loo")
```

Model comparison

```r
loo_compare(
  fit_2_groups,
  fit_3_groups,
  fit_3_groups_alt,
  fit_4_groups)
```

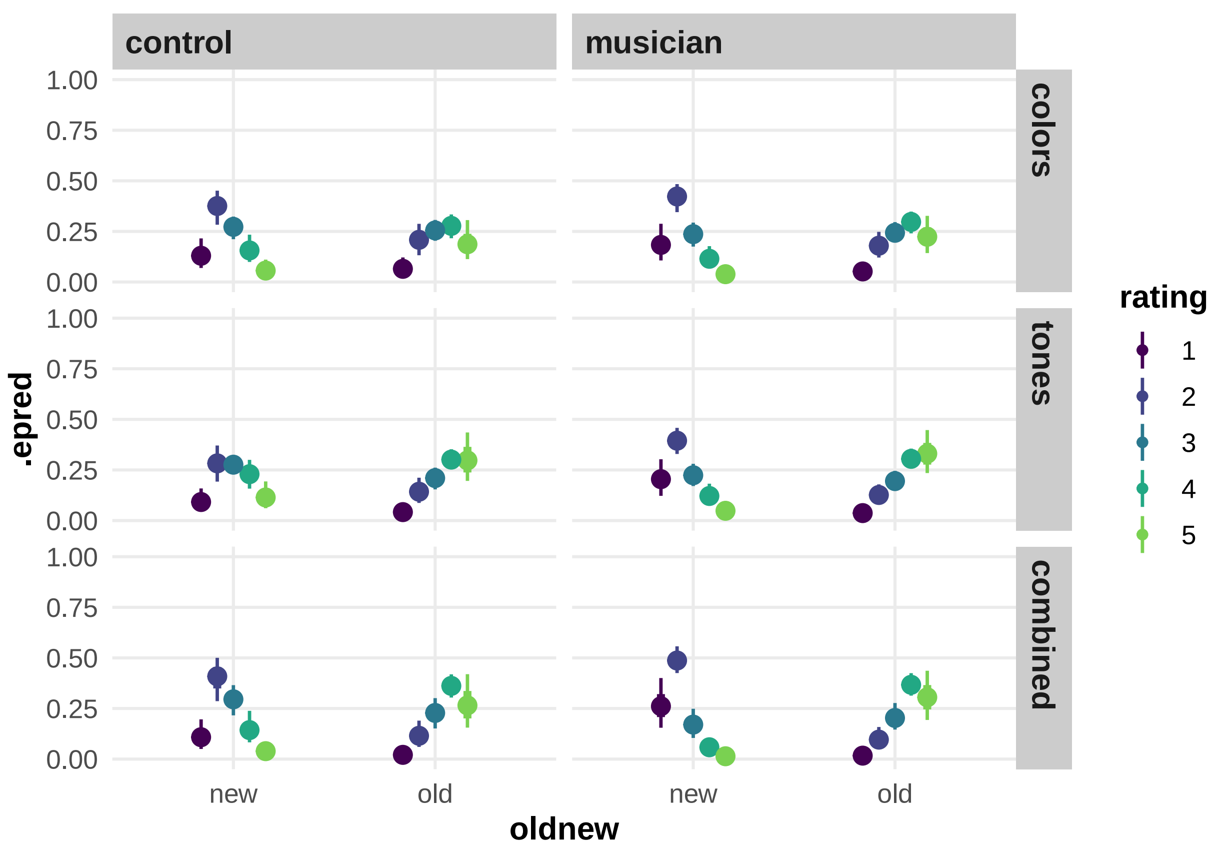
```
##                  elpd_diff se_diff
## fit_2_groups      0.0       0.0
## fit_3_groups     -1.7       2.3
## fit_3_groups_alt -2.0       2.5
## fit_4_groups     -2.9       3.4
```

Expectations of posterior predictive distribution

```r
epred_2_groups <- d |>
  data_grid(group2, test, oldnew) |>
  add_epred_draws(fit_2_groups,
                  category = "rating",
```

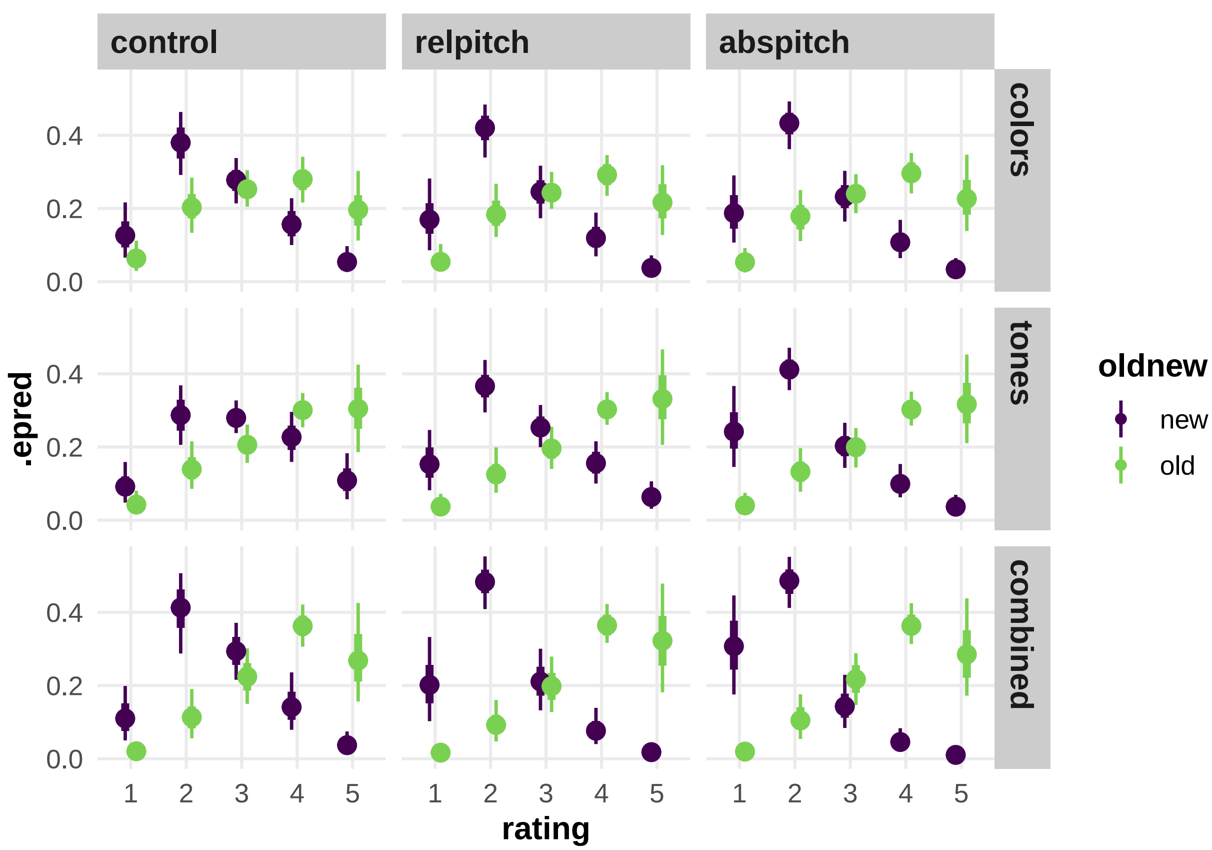
```
                  dpar = TRUE,
                  re_formula = ~ID,
                  ndraws = 500)

epred_3_groups <- d |>
  data_grid(group3, test, oldnew) |>
  add_epred_draws(fit_3_groups,
                  category = "rating",
                  dpar = TRUE,
                  re_formula = ~ID,
                  ndraws = 500)

epred_3_groups_alt <- d |>
  data_grid(group3alt, test, oldnew) |>
  add_epred_draws(fit_3_groups_alt,
                  category = "rating",
                  dpar = TRUE,
                  re_formula = ~ID,
                  ndraws = 500)

epred_4_groups <- d |>
  data_grid(group4, test, oldnew) |>
  add_epred_draws(fit_4_groups,
                  category = "rating",
                  dpar = TRUE,
                  re_formula = ~ID,
                  ndraws = 500)
```

```
epred_2_groups |>
  ggplot(aes(x = oldnew, y = .epred, color = rating)) +
  stat_pointinterval(position = position_dodge(width = .4)) +
  facet_grid(test ~ group2) +
  scale_size_continuous(guide = FALSE) +
  scale_y_continuous(limits = c(0, 1)) +
  # scale_color_brewer(palette = "RdYlBu")
  # scale_size_continuous(guide = "none") +
  # scale_color_manual(values = brewer.pal(6, "Blues")[-c(1)]) +
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  theme_clean()
```

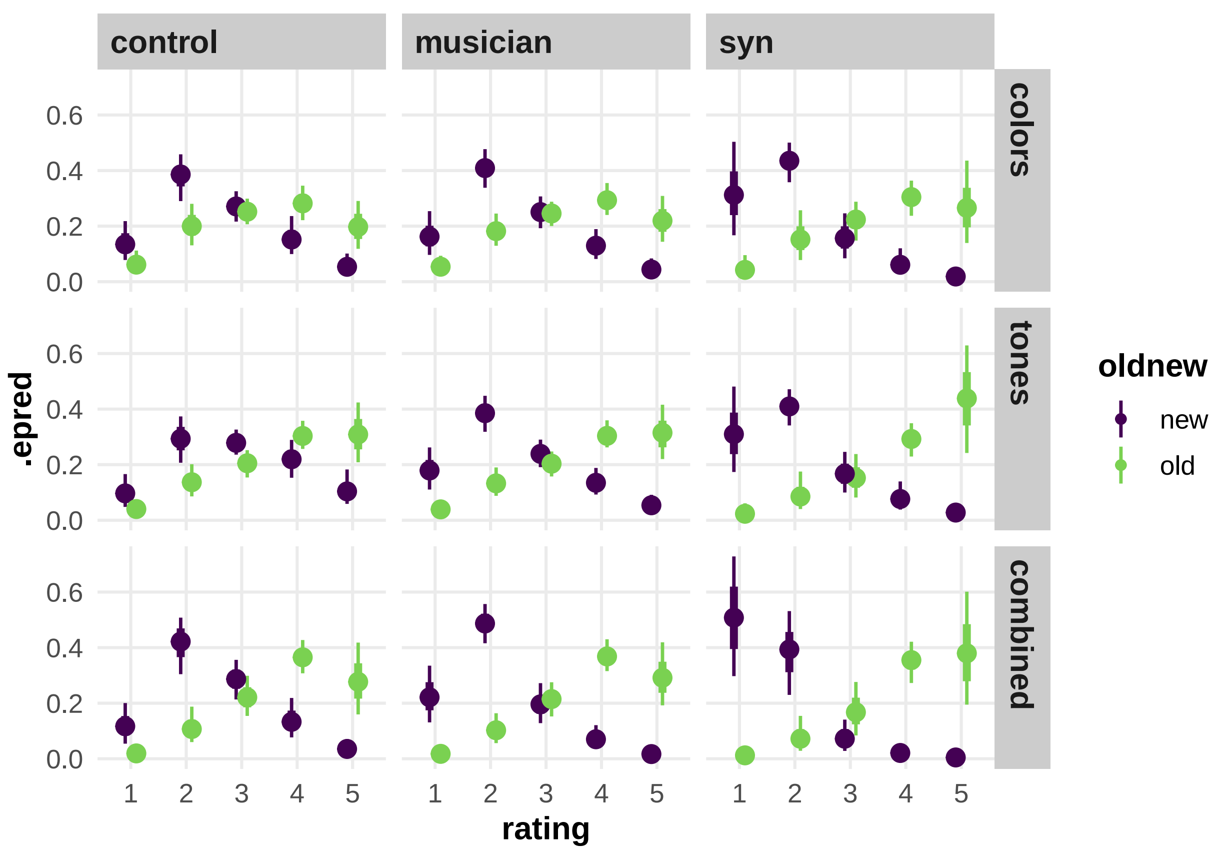


205

```
epred_3_groups |>
  ggplot(aes(x = rating, y = .epred, color = oldnew)) +
  stat_pointinterval(position = position_dodge(width = .4)) +
  # geom_line(aes(group = oldnew)) +
  facet_grid(test ~ group3) +
  scale_size_continuous(guide = FALSE) +
  # scale_y_continuous(limits = c(0, 1)) +
  expand_limits(y = 0) +
  # scale_color_brewer(palette = "RdYlBu")
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  theme_clean()
```

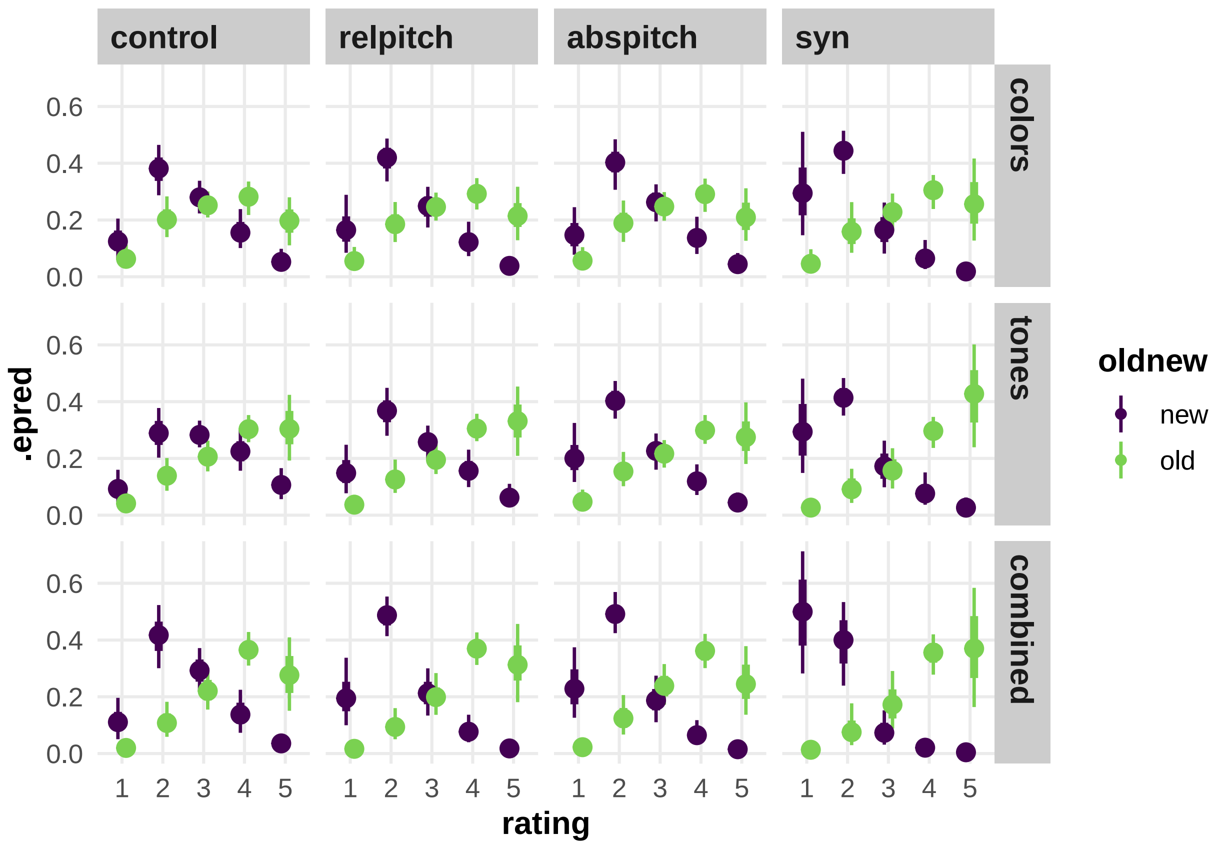


```
epred_3_groups_alt |>
  ggplot(aes(x = rating, y = .epred, color = oldnew)) +
  stat_pointinterval(position = position_dodge(width = .4)) +
  # geom_line(aes(group = oldnew)) +
  facet_grid(test ~ group3alt) +
  scale_size_continuous(guide = FALSE) +
  expand_limits(y = 0) +
  # scale_color_brewer(palette = "RdYlBu")
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  theme_clean()
```




```
epred_4_groups |>
  ggplot(aes(x = rating, y = .epred, color = oldnew)) +
  stat_pointinterval(position = position_dodge(width = .4)) +
```

```r
  # geom_line(aes(group = oldnew)) +
  facet_grid(test ~ group4) +
  scale_size_continuous(guide = FALSE) +
  expand_limits(y = 0) +
  # scale_color_brewer(palette = "RdYlBu")
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  theme_clean()
```



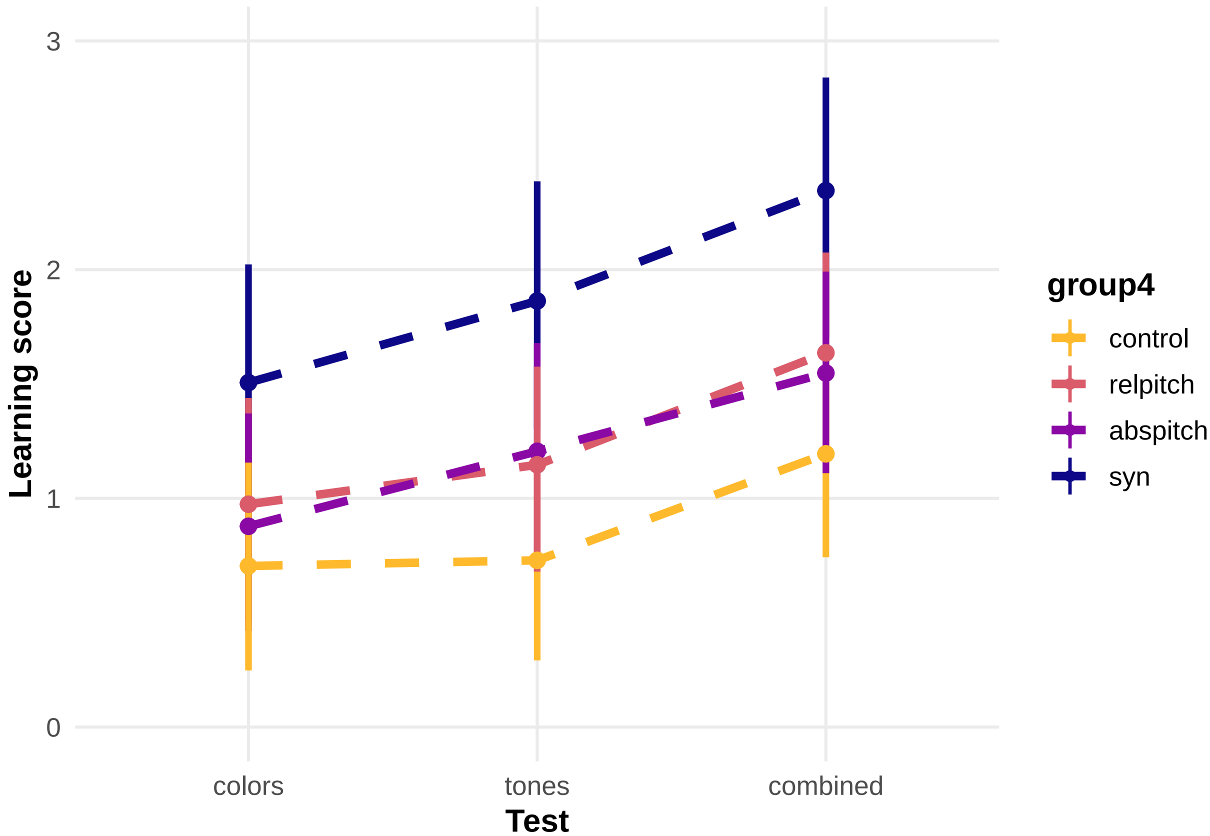
Posterior expectations

The following plots show the exptected learning scores, computed from the models posterior predictive distribution. This plot is shown merely to demonstrate that the models output can be used to create a plot that is similar to the learning score computed from the data. This could serve as a posterior predictive check, i.e. in order to show that the models predictions closely match the empirical data (the model predicts data that are similar to the

215 actual data) according to some desired metric (in this case learning score).

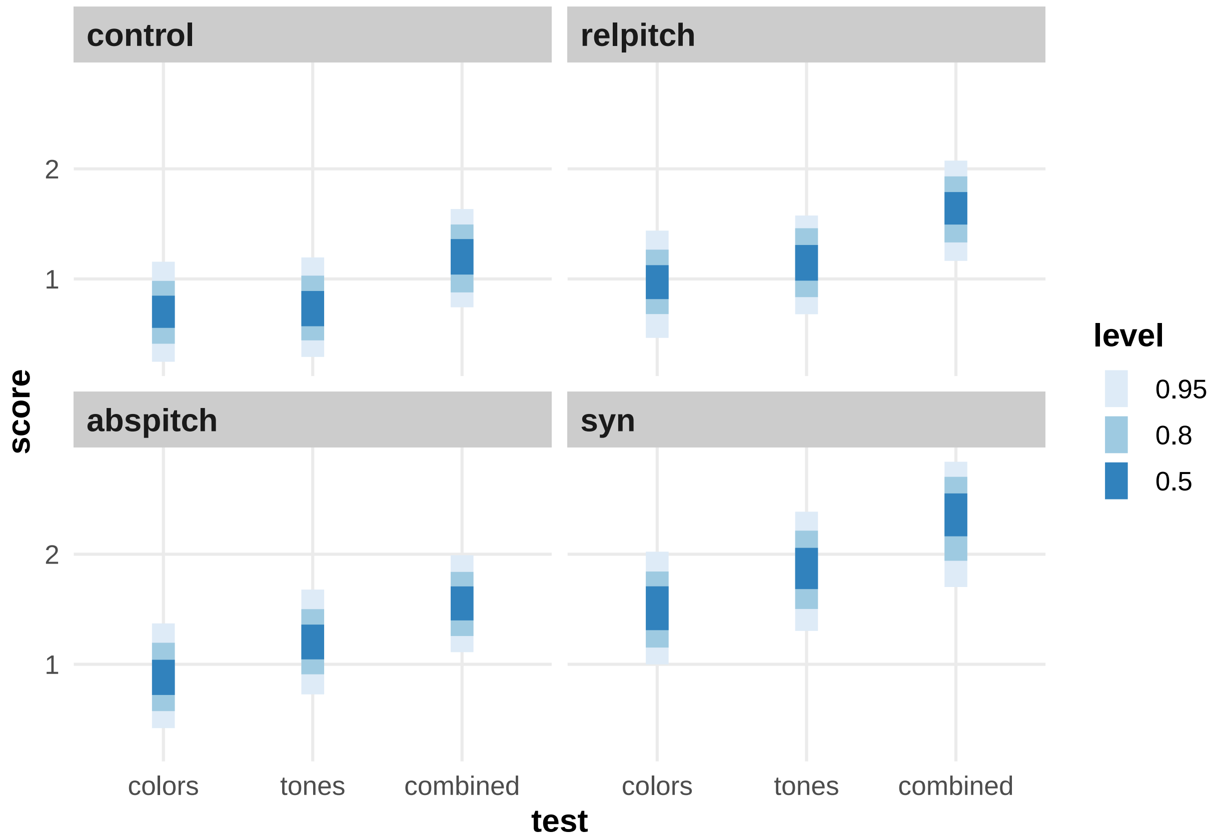
```
posterior_expectations <- epred_4_groups |>
  mutate(product = as.double(rating) * .epred) |>
  # group and convert to the sum-score metric
  group_by(group4, test, oldnew, .draw) |>
  summarise(mean_rating = sum(product)) |>
  pivot_wider(names_from = oldnew, values_from = mean_rating) |>
  mutate(score = old - new)
```

```
posterior_expectations |>
  # summarize
  group_by(group4, test) |>
  mean_qi(score) |>
  ggplot(aes(y = score, x = test, color = group4, ymin = .lower, ymax = .upper)) +
  geom_line(aes(group = group4), linewidth = 1.5, linetype = "dashed") +
  geom_pointinterval() +
  scale_color_viridis_d(direction = -1, option = "C", end = .85) +
  ylab("Learning score") +
  xlab("Test") +
  ylim(0, 3) +
  theme_clean()
```




```
  # facet_grid(. ~ test)
  # facet_wrap(~test)
```

```
posterior_expectations |>
  ggplot(aes(x = test, y = score)) +
  stat_interval(.width = c(.50, .80, .95)) +
  facet_wrap(~group4) +
  scale_color_brewer(palette = "Blues") +
  theme_clean()
```



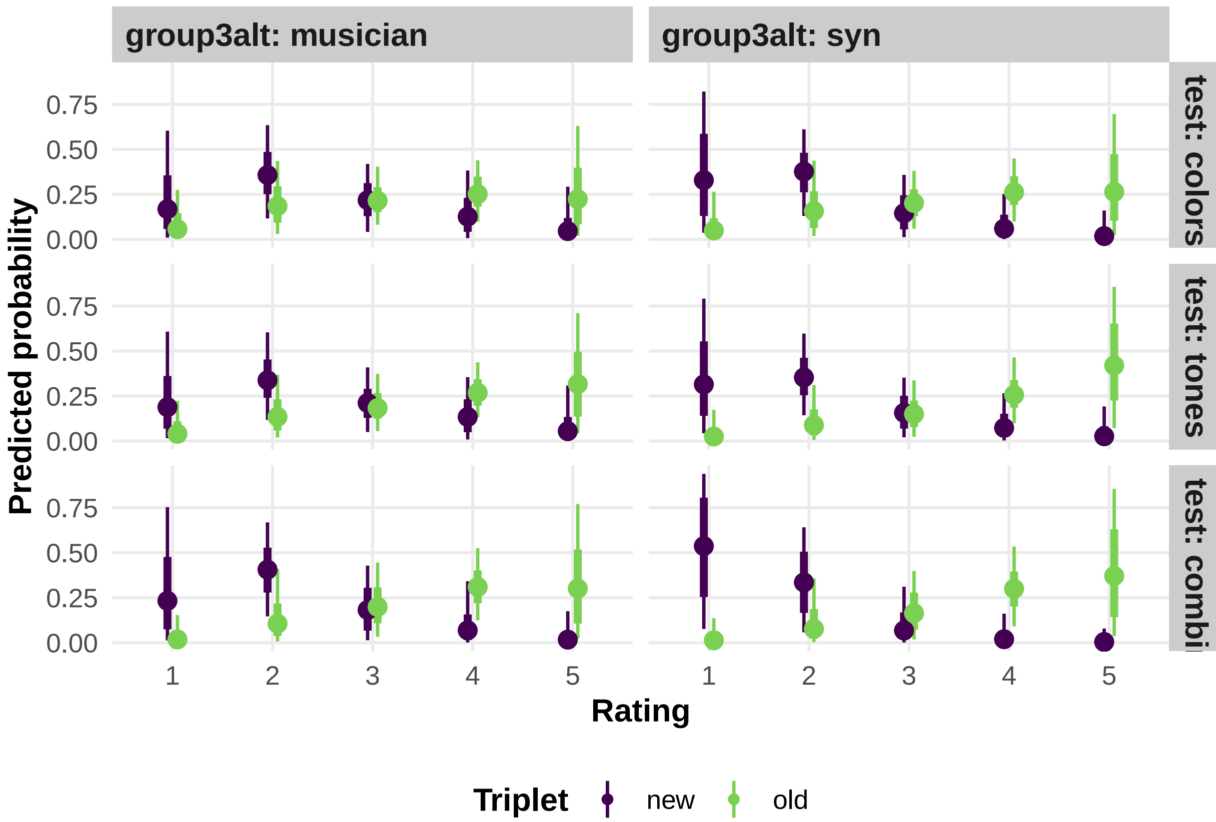
Simulate future data

This is (in my opinion) a very valuable part of the data analysis, because we can use our model to simulate unseen subjects from the 4 groups, according to the estimted random effects structure. Here, I show one simulated subject from the musician group (rel/abs pitch) and one simulated synaesthete.

```
newdata <- expand_grid(group3alt = c("syn", "musician"),
                       oldnew = c("new", "old"),
                       test = c("colors", "tones", "combined"),
                       ID = NA) |>
  mutate(across(where(is_character), as_factor),
         group3alt = fct_relevel(group3alt,
                                 "musician", "syn"),
         test = fct_relevel(test,
```

```
                             "colors", "tones", "combined"))


preds <- fit_3_groups_alt |>
  add_epred_draws(newdata = newdata,
                  category = "rating",
                  dpar = TRUE,
                  re_formula = ~(1 | ID) + (1 | item),
                  ndraws = 500,
                  allow_new_levels = TRUE,
                  sample_new_levels = "gaussian")
```

```
preds |>
  ggplot(aes(x = rating, y = .epred, color = oldnew)) +
  stat_pointinterval(position = position_dodge(width = .2)) +
  facet_grid(test ~ group3alt, labeller = label_both) +
  scale_size_continuous(guide = FALSE) +
  expand_limits(y = 0) +
  # scale_color_okabe_ito() +
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  labs(x = "Rating", y = "Predicted probability", color = "Triplet") +
  theme_clean() +
  theme(legend.position = "bottom")
```



In the next figure, I show 10 simulated synaesthetes in the combined test, and contrast these with ten simulated musicians with either relative or absolute pitch.

```
newdata <- expand_grid(group3alt = "syn",
                       oldnew = c("new", "old"),
                       test = "combined",
                       ID = str_c("Synaesthete ", 1:10)) |>
  mutate(across(where(is_character), as_factor),
         test = fct_relevel(test,
                            "colors", "tones", "combined"))
```

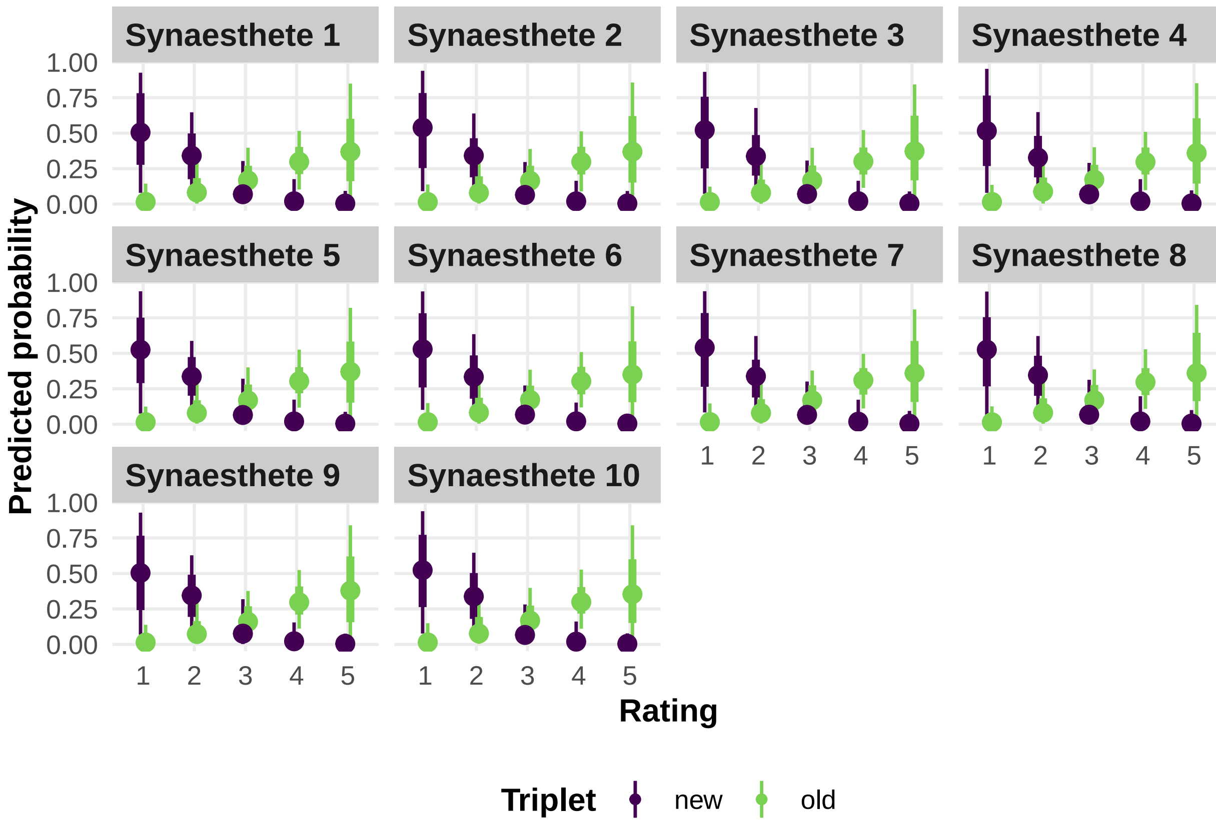
```
## Warning: 2 unknown levels in `f`: colors and tones
```

```
preds <- fit_3_groups_alt |>
  add_epred_draws(newdata = newdata,
```

```
                  category = "rating",
                  dpar = TRUE,
                  re_formula = ~(1 | ID) + (1 | item),
                  ndraws = 500,
                  allow_new_levels = TRUE,
                  sample_new_levels = "gaussian")
```

```
p_syn <- preds |>
  ggplot(aes(x = rating, y = .epred, color = oldnew)) +
  stat_pointinterval(position = position_dodge(width = .2)) +
  facet_wrap(~ ID) +
  scale_size_continuous(guide = FALSE) +
  expand_limits(y = 0) +
  # scale_color_okabe_ito() +
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  labs(x = "Rating", y = "Predicted probability", color = "Triplet") +
  theme_clean() +
  theme(legend.position = "bottom")
```

```
p_syn
```




```r
newdata <- expand_grid(group3alt = "musician",
                       oldnew = c("new", "old"),
                       test = "combined",
                       ID = str_c("Musician ", 1:10)) |>
  mutate(across(where(is_character), as_factor),
         test = fct_relevel(test,
                            "colors", "tones", "combined"))
```

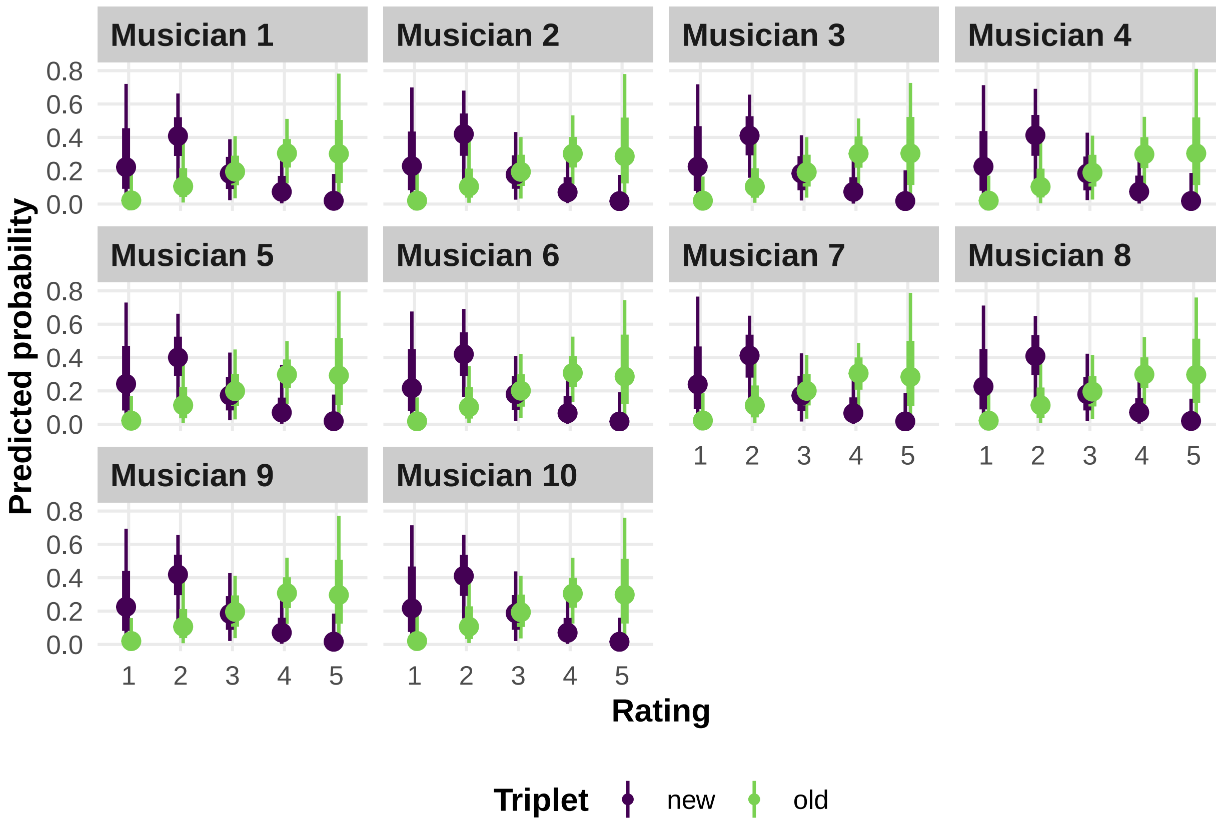
```
## Warning: 2 unknown levels in `f`: colors and tones
```

```r
preds <- fit_3_groups_alt |>
  add_epred_draws(newdata = newdata,
                  category = "rating",
                  dpar = TRUE,
                  re_formula = ~(1 | ID) + (1 | item),
```

```
                  ndraws = 500,
                  allow_new_levels = TRUE,
                  sample_new_levels = "gaussian")
```

```
p_musician <- preds |>
  ggplot(aes(x = rating, y = .epred, color = oldnew)) +
  stat_pointinterval(position = position_dodge(width = .2)) +
  facet_wrap(~ ID) +
  scale_size_continuous(guide = FALSE) +
  expand_limits(y = 0) +
  # scale_color_okabe_ito() +
  scale_color_viridis_d(begin = 0.0, end = 0.8) +
  labs(x = "Rating", y = "Predicted probability", color = "Triplet") +
  theme_clean() +
  theme(legend.position = "bottom")
```

```
p_musician
```



Overall, we can predict from these simulations that synaethetes mainly differ from musicians with relative and absolute pitch in their use of the extreme categories (1 and 5). When presented with previously unseen triplets, synaesthetes consistently use a rating of 1 to reject new items with great confidence, while musicians without synaesthesia are seemingly unable to identify previously unseen triplets with comparable confidence. The differences are less pronounced at the other end of the rating scale (identifying previoulsy seen triplets).

References

Liddell, T. M., & Kruschke, J. K. (2018). Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology*, *79*, 328–348. https://doi.org/10.1016/j.jesp.2018.08.009

Morey, R. D. (2008). Confidence Intervals from Normalized Data: A correction to Cousineau (2005). *Tutorials in Quantitative Methods for Psychology*, *4*(2), 61–64. https://doi.org/10.20982/tqmp.04.2.p061