Towards reproducibility in small-N treatment research in aphasiology: a tutorialy

Introduction

This document details the code batch calculate effect sizes.

Setup

Load packages and functions

```
library(here)            # for locating files
library(tidyverse)       # data wrangling
library(SingleCaseES)    # calculating SMD, Tau-U
library(lme4)            # frequentist mixed-effects models
library(emmeans)         # estimating effect sizes from lme4
library(brms)            # bayesian mixed-effects models
library(tidybayes)       # estimating effect sizes from brms
library(ggdist)          # Visualizing posterior distributions

# set a seed for reproducibility
set.seed(42)

source(here("R", "effect-size-functions.R"))
```

Read in data

Note that the current setup uses RStudio R projects (https://support.rstudio.com/hc/en-us/articles/200526207-Using-RStudio-Projects). One of the features of R projects is that the working directory is automatically set to the project root (the folder with the .Rproj). A discussion of R projects can be found at https://www.tidyverse.org/blog/2017/12/workflow-vs-script/. In this case `here("study-data")` refers to the /study-data folder inside the project.

```
# create a list of files
files <- list.files(
                 here("study-data"), # look in the study-data folder
                 full.names = TRUE,  # use the full paths of the files
                 pattern = ".csv",   # only read in .csv files
                 recursive = TRUE)   # include files within subfolders

# read in the files and combine them together
# map_df takes a function, in this case read_csv().
# show_col_types suppresses output sinc we're reading in many files
df <- files %>%
  map_dfr(read_csv, show_col_types = FALSE)
```

Calculate effect sizes

$d_{\mathbf{BR}}$

```
df_smd = df %>%
  filter(spt2017 == "pre" | spt2017 == "post" ) %>%
  group_by(participant, phase, condition, itemType, session,
           spt2017, trials, phoneme) %>%
  summarize(correct = sum(response),
            trials = unique(trials), .groups = "drop") %>%
  nest_by(participant, condition, itemType) %>%
  summarize(SMD_br(data), .groups = "drop")
```

PMG

```
df_pmg = df %>%
  filter(spt2017 == "pre" | spt2017 == "post" ) %>%
  group_by(participant, phase, condition, itemType, session,
           spt2017, trials) %>%
  summarize(correct = sum(response),
            trials = unique(trials)*2, .groups = "drop") %>%
  group_by(participant, condition, itemType) %>%
  summarize(PMG(outcome = correct, phase = spt2017, nitems = trials,
                bl_phase = "pre", tx_phase = "post"), .groups = "drop")
```

Tau-U

```
df_tau = df %>%
  filter(phase == "baseline" | phase == "treatment") %>%
  group_by(participant, phase, condition, itemType, session) %>%
  summarize(correct = sum(response), .groups = "drop") %>%
  group_by(participant, condition, itemType) %>%
  summarize(Tau_custom(outcome = correct, phase = phase,
                       bl_phase = "baseline", tx_phase = "treatment",
                       session = session), .groups = "drop")
```

Bayesian Mixed-effects models

Setup data

```
df_itts_group = df %>%
  filter(phase == "baseline" | phase == "treatment") %>%
  mutate(baseline_slope = session,
         level_change = ifelse(phase == "baseline", 0, 1),
         slope_change = (session - (n_baselines+2))*level_change) %>%
  select(response, participant, condition, itemType, phase,
         phoneme, item, baseline_slope, level_change, slope_change)
```

Blocked Treated

```
  mod_tx_bl <- brm(
    response ~ 0 + Intercept + baseline_slope + level_change + slope_change +
               (1 + baseline_slope + level_change + slope_change | participant) +
               (1| item),
             data = df_itts_group %>% filter(condition == "blocked",
                                             itemType == "tx"),
             family = bernoulli(),
             iter = 3000,
             warmup = 1000,
             cores = 4, chains = 4,
             prior = c(
               prior(normal(-1, 2.5), class = b, coef = Intercept),
               prior(normal(0, 2.5), class = b)
             ),
             control = list(adapt_delta = 0.9),
             seed = 42,
             file = "models/mod_tx_bl",
             file_refit = "on_change"
  )
```

Blocked Generalization

```
mod_gx_bl <- brm(
  response ~ 0 + Intercept + baseline_slope + level_change + slope_change +
             (1 + baseline_slope + level_change + slope_change | participant) +
             (1| item),
                 data = df_itts_group %>% filter(condition == "blocked",
                                                 itemType == "gx"),
                 family = bernoulli(),
                 iter = 3000,
                 warmup = 1000,
                 cores = 4, chains = 4,
                 prior = c(
                   prior(normal(-1, 2.5), class = b, coef = Intercept),
                   prior(normal(0, 2.5), class = b)
                 ),
                 control = list(adapt_delta = 0.85),
                 seed = 42,
                 file = "models/mod_gx_bl",
                 file_refit = "on_change"
)
```

Random Treated

```
mod_tx_ra <- brm(
  response ~ 0 + Intercept + baseline_slope + level_change + slope_change +
             (1 + baseline_slope + level_change + slope_change | participant) +
             (1| item),
                 data = df_itts_group %>% filter(condition == "random",
                                                 itemType == "tx"),
                 family = bernoulli(),
                 iter = 3000,
                 warmup = 1000,
                 cores = 4, chains = 4,
                 prior = c(
                   prior(normal(-1, 2.5), class = b, coef = Intercept),
                   prior(normal(0, 2.5), class = b)
                 ),
                 seed = 42,
                 control = list(adapt_delta = 0.85),
                 file = "models/mod_tx_ra",
                 file_refit = "on_change"
)
```

Random Generalization

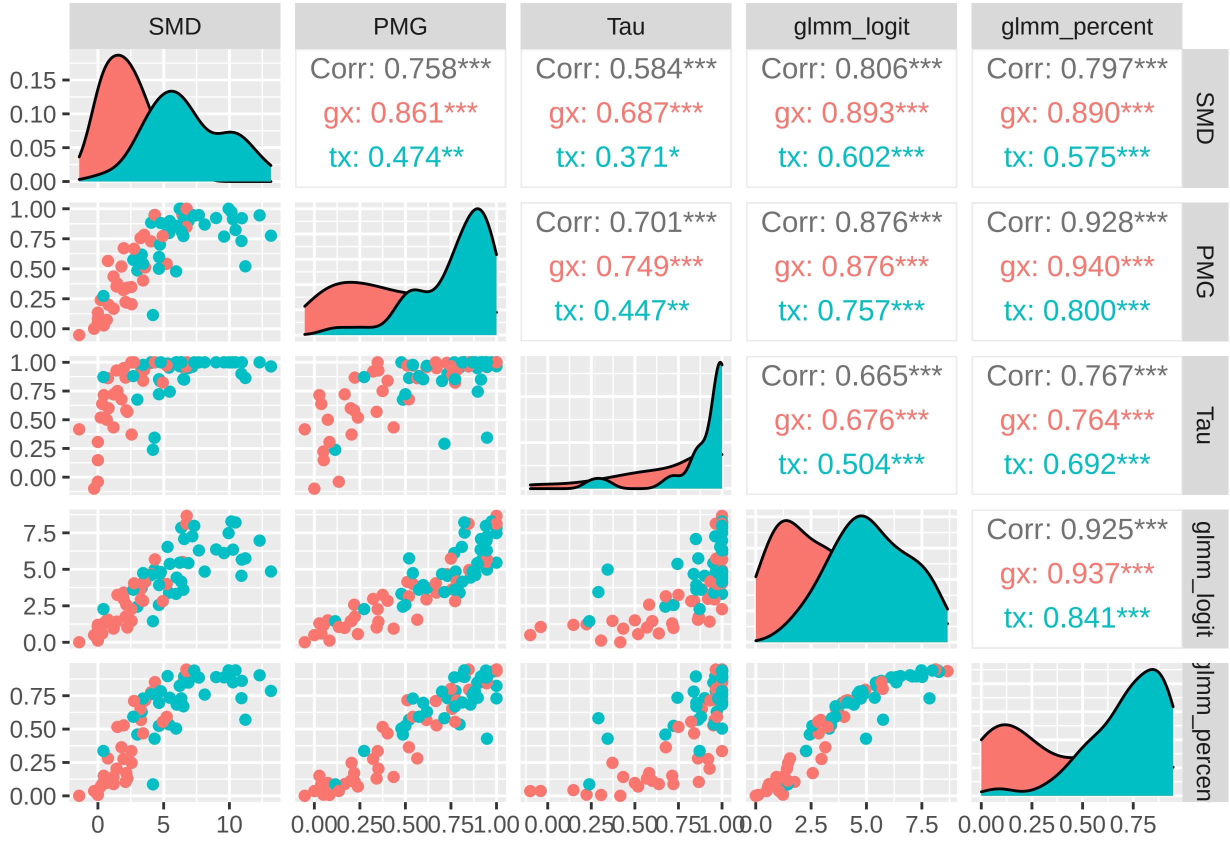
```
  mod_gx_ra <- brm(
    response ~ 0 + Intercept + baseline_slope + level_change + slope_change +
               (1 + baseline_slope + level_change + slope_change | participant) +
               (1| item),
                   data = df_itts_group %>% filter(condition == "random",
                                                   itemType == "gx"),
                   family = bernoulli(),
                   iter = 3000,
                   warmup = 1000,
                   cores = 4, chains = 4,
                   control = list(adapt_delta = 0.9),
                   prior = c(
                     prior(normal(-1, 2), class = b, coef = Intercept),
                     prior(normal(0, 2), class = b)
                   ),
                   seed = 42,
                   file = "models/mod_gx_ra",
                   file_refit = "on_change"
  )
```

Pull the effect sizes together and plot

```
es_tx_bl = getES(mod_tx_bl, "tx", "blocked")
es_tx_ra = getES(mod_tx_ra, "tx", "random")
es_gx_bl = getES(mod_gx_bl, "gx", "blocked")
es_gx_ra = getES(mod_gx_ra, "gx", "random")

smd =
  df_smd %>%
  select(participant, condition, itemType, SMD)

pmg =
  df_pmg %>%
  select(participant, condition, itemType, PMG, raw_change = raw_change_exit)

tau =
  df_tau %>%
  select(participant, condition, itemType, Tau = Est)

bglmm =
  bind_rows(es_tx_bl, es_tx_ra, es_gx_bl, es_gx_ra) %>%
  select(participant, ES, unit, itemType, condition) %>%
  pivot_wider(names_from = unit, values_from = ES) %>%
  rename(glmm_logit = logit, glmm_percent = pred)

es = smd %>%
  left_join(pmg, by = c("participant", "itemType", "condition")) %>%
  left_join(tau, by = c("participant", "itemType", "condition")) %>%
  left_join(bglmm, by = c("participant", "itemType", "condition")) %>%
  select(-raw_change)
```

Plot comparisons

```
GGally::ggpairs(es,
                columns = 4:8,
                mapping = aes(color = itemType))
```


SMD
PMG
Tau
glmm_logit
glmm_percent
Corr: 0.758***
gx: 0.861***
tx: 0.474**
Corr: 0.584***
gx: 0.687***
tx: 0.371*
Corr: 0.806***
gx: 0.893***
tx: 0.602***
Corr: 0.797***
gx: 0.890***
tx: 0.575***
Corr: 0.701***
gx: 0.749***
tx: 0.447**
Corr: 0.876***
gx: 0.876***
tx: 0.757***
Corr: 0.928***
gx: 0.940***
tx: 0.800***
Corr: 0.665***
gx: 0.676***
tx: 0.504***
Corr: 0.767***
gx: 0.764***
tx: 0.692***
Corr: 0.925***
gx: 0.937***
tx: 0.841***
SMD
PMG
Tau
glmm_logit
glmm_percen