Statistical modelling of affective valence (`affval`) and perceived exertion (`perexe`) outcomes

01 Feb 2023

Table of contents

Set up

Packages

```r
suppressPackageStartupMessages(suppressWarnings({

  library("readr")      # read data funcs
  library("dplyr")      # data manipulation
  library("tibble")     # improved data frames
  library("ggplot2")    # plotting
  library("purrr")      # apply functions over lists/vectors
  library("forcats")    # handling categorical variables
  library("tidyr")      # manipulate data to long/short representations

  library("mice")       # missing value utilities
  library("car")        # regression utilities

  library("lme4")       # linear mixed effects modelling
  library("merTools")   # alows prediction intervals using merMod objects
  library("lmerTest")   # step-wise lmer model selection

  library("knitr")      # pretty printing of tables: kable()
  library("gtsummary") # print summary tables of regression mods
  library("lattice")    # diagnostic plots

}))
```

Constants

```r
### plotting characters
# steep x long: "|"
# steep x short:  "/"
# less steep x long: "-"
# less steep x short: "\"
plchs <-
  c(
    "yes x long" = "|",
    "yes x short" = "/",
    "no x long" = "-",
    "no x short" = "\\"
  )

# plotting colour scheme
col_lohi <-
  c(
    "Lower IAcc" = "darkorange",
    "Higher IAcc" = "purple"
  )
```

Functions

The below functions make the calculation of Cohen's f^2 effect size statistic on `lme4::lmer()` (`merMod` class objects) possible. These functions are used later after models are fitted.

```r
# effect size (f^2), R^2 and residual variance functions

get_res_var_lmer <- function(lmer_obj) {
  return(sigma(lmer_obj) ^ 2)
}

get_lmer_r2 <- function(lmer_obj) {

  # residual variance of input model
  v_mod <- get_res_var_lmer(lmer_obj)

  # get formula for null model (intercept and REs only)
  null_form <- formula(lmer_obj, random.only = TRUE)
  # create null model
  lmer_obj_null <- update(lmer_obj, null_form)

  # res var of null mod
  v_null <- get_res_var_lmer(lmer_obj_null)

  r2 <- (v_null - v_mod) / v_null

  attr(r2, "v_null") <- v_null
  attr(r2, "v_mod") <- v_mod

  return(r2)

}

rm_terms_lmer <- function(lmer_obj, terms) {

  update_form <- as.formula(paste0("~ . -", paste(terms, collapse = " - ")))
  print(update_form)

  lmer_obj_less_term <- update(lmer_obj, update_form)

  return(lmer_obj_less_term)
}


eff_size_f2 <- function(lmer_obj, terms) {

  r2_full <- get_lmer_r2(lmer_obj)[1]
  r2_less_term <- get_lmer_r2(rm_terms_lmer(lmer_obj, terms))[1]

  f2 <- (r2_full - r2_less_term) / (1 - r2_full)

  return(f2)

}
```

Convenience function for tidy printing of data.

```
# function that replaces repeated values in a vector with empty strings
# as the verbose redundancy is too busy in some cases
```

```r
rm_rpts <- function(x) {
  x <- as.character(x)
  nx <- length(x)
  rm_ii <- rep(FALSE, nx)
  for (i in 2:nx) {
    if (x[i - 1] == x[i])
      rm_ii[i] <- TRUE
  }
  x[rm_ii] <- ""
  return(x)
}
```