Harmonicity in early auditory processing - power analysis

Krzysztof Basiński (based on a script by David Quiroga)

2022-03-16

First, load the *lme4* library and define a function to conduct power estimates.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
pwr_calc <- function(b0,b1,b0_sd,res_sd,nsims,ssizes) {
  # intialize data frame to store the output
  pwr <- data.frame()

  for (s in 1:length(ssizes)) {
    for (n in 1:nsims) {
      # df index
      idx <- (s-1) * nsims + n

      # current sample size
      ssize <- ssizes[s]

      # conditions factor
      conds <- rep(c(0,1,2),ssize)

      # subject id's
      subs <- rep(1:ssize, each = length(unique(conds)))

      # add intercept
      intercept <- rep(rnorm(ssize,b0,b0_sd), each = length(unique(conds)))

      # add condition effect
      conds_effect <- rep(c(0,0,1),ssize)
      beta1 <- rep(b1,each= length(unique(conds)))*conds_effect

      # add residual noise
      residuals <- rnorm(length(subs),0,res_sd)

      # collect in a dataframe and calculate simulated measured outcome (y)
      d <- data.frame('cond' = as.character(conds),
                      'sub' = subs,
                      'b0' = intercept,
                      'b1' = beta1,
                      'res' = residuals,
                      'y' = intercept + beta1 + residuals)

      # fit models
      m0 <- lmer(y~1 + (1|sub), data = d, REML = FALSE)
```

```
      m1 <- lmer(y~cond + (1|sub), data = d, REML = FALSE)

      # perform likelihood ratio test
      test <- anova(m0,m1)

      #store output of simulation
      pwr[idx,'sim'] <- n
      pwr[idx, 'ssize'] <- ssize
      pwr[idx, 'b0'] <- summary(m1)$coefficients[1]
      pwr[idx, 'b1'] <- summary(m1)$coefficients[2]
      pwr[idx, 'sd_int'] <- attr(summary(m1)$varcor$sub,"stddev")
      pwr[idx, 'sd_res'] <- summary(m1)$sigma
      pwr[idx, 'x2'] <- test$Chisq[2]
      pwr[idx, 'p'] <- test$`Pr(>Chisq)`[2]
    }
  }
  return(pwr)
}
```

This function uses a mixed effects modelling to compare a null model in the form:

`y ~ 1 + (1 | participant)`

to the model:

`y ~ harmonicity + (1 | participant)`

Where `y` is the dependent variable (EEG results), harmonicity is a fixed effect and participant is a random effect. Harmonicity is specified as a three-level factor (harmonic, inharmonic, inharmonic changing).

Now specify the parameters to simulate EEG data. Let's assume a minimum difference of -1uV (micro volts) between the harmonic and inharmonic conditions and a residual SD of 1.5uV. Let's also assume a 'worst case' scenario, where there are no detectable differences between inharmonic and inharmonic-changing conditions. This script runs 10 000 simulations for each possible sample size from N = 25 to N = 40.

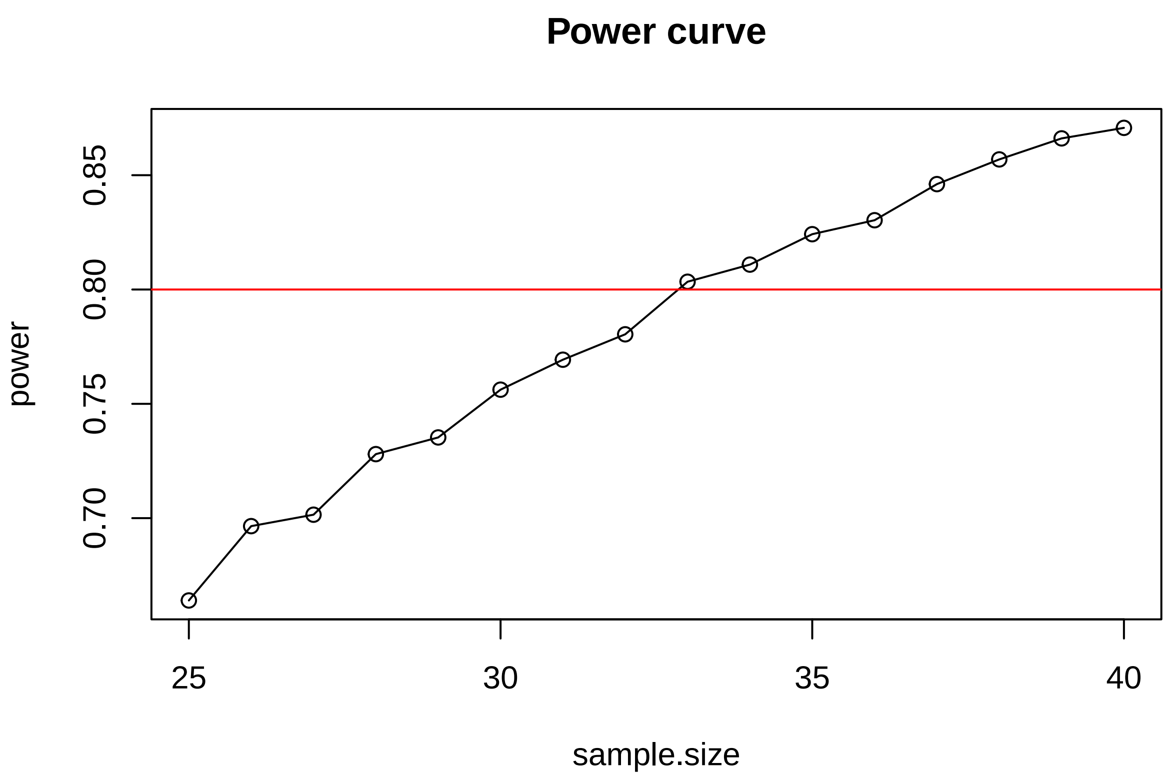
```
e0 <- 3 # uV # uV # intercept (in fT or micro Volts)
e1 <- -1 # -1 uv # uV # minimum difference between conditions
e0_sd <- 0.95 # 0.95 uV # standard deviation of the intercept
eres_sd <- 1.5 # 1.5 uV # residual standard deviation
nsims <- 10000 # number of simulations per sample size
ssizes <- seq(from = 25, to = 40, by = 1) # sample sizes

set.seed(777)
pwr2 <- pwr_calc(e0,e1,e0_sd,eres_sd,nsims,ssizes)



summary2 <- aggregate(pwr2$p,by = list(pwr2$ssize), FUN = function(x) sum(x < 0.05)/length(x))
colnames(summary2) <- c('sample.size','power')
print(summary2)
```

```
##    sample.size  power
## 1           25 0.6640
## 2           26 0.6965
## 3           27 0.7015
## 4           28 0.7280
## 5           29 0.7353
```

```
## 6            30 0.7562
## 7            31 0.7693
## 8            32 0.7804
## 9            33 0.8034
## 10           34 0.8109
## 11           35 0.8242
## 12           36 0.8303
## 13           37 0.8461
## 14           38 0.8569
## 15           39 0.8661
## 16           40 0.8707
```

Now plot the power curve as a function of sample size:

```r
with(summary2, plot(sample.size, power, type = 'ol'))
abline(h=.8, col='red')
title('Power curve')
```



This analysis shows that a power level of 0.8 is achieved for a sample size of N = 33.