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
      beta1 <- rep(b1,each= length(unique(conds)))*conds

      # 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 between conditions of -1uV (micro volts) and a residual SD of 1.5uV. This script runs 2000 simulations for each possible sample size from N = 10 to N = 30.

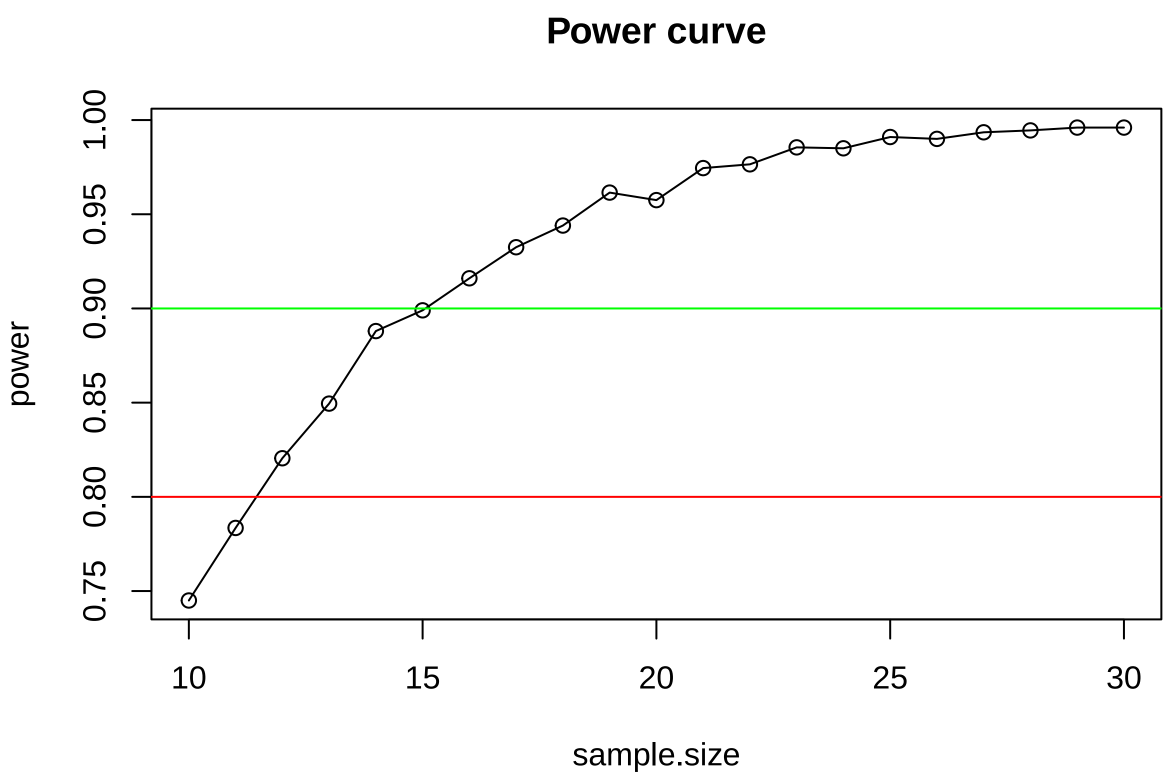
```
e0 <- 3 # uV # uV # intercept (in fT or micro Volts)
e1 <- -1 # -1 uv # uV # minimum difference between conditions
e0_sd <- 0.95 # 0.95 uV # standard deviation of the intercept
eres_sd <- 1.5 # 1.5 uV # residual standard deviation
nsims <- 2000 # number of simulations per sample size
ssizes <- 10:30 # 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           10 0.7450
## 2           11 0.7835
## 3           12 0.8205
## 4           13 0.8495
## 5           14 0.8880
## 6           15 0.8990
## 7           16 0.9160
## 8           17 0.9325
```

```
## 9            18 0.9440
## 10           19 0.9615
## 11           20 0.9575
## 12           21 0.9745
## 13           22 0.9765
## 14           23 0.9855
## 15           24 0.9850
## 16           25 0.9910
## 17           26 0.9900
## 18           27 0.9935
## 19           28 0.9945
## 20           29 0.9960
## 21           30 0.9960
```

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

```
with(summary2, plot(sample.size, power, type = 'ol'))
abline(h=.8, col='red')
abline(h=.9, col='green')
title('Power curve')
```



This analysis shows that a power level of 0.8 is achieved for a sample size of N = 12 and power level of 0.9 for a sample size of N = 16.

Now repeat the process for a weaker effect of -.8uV:

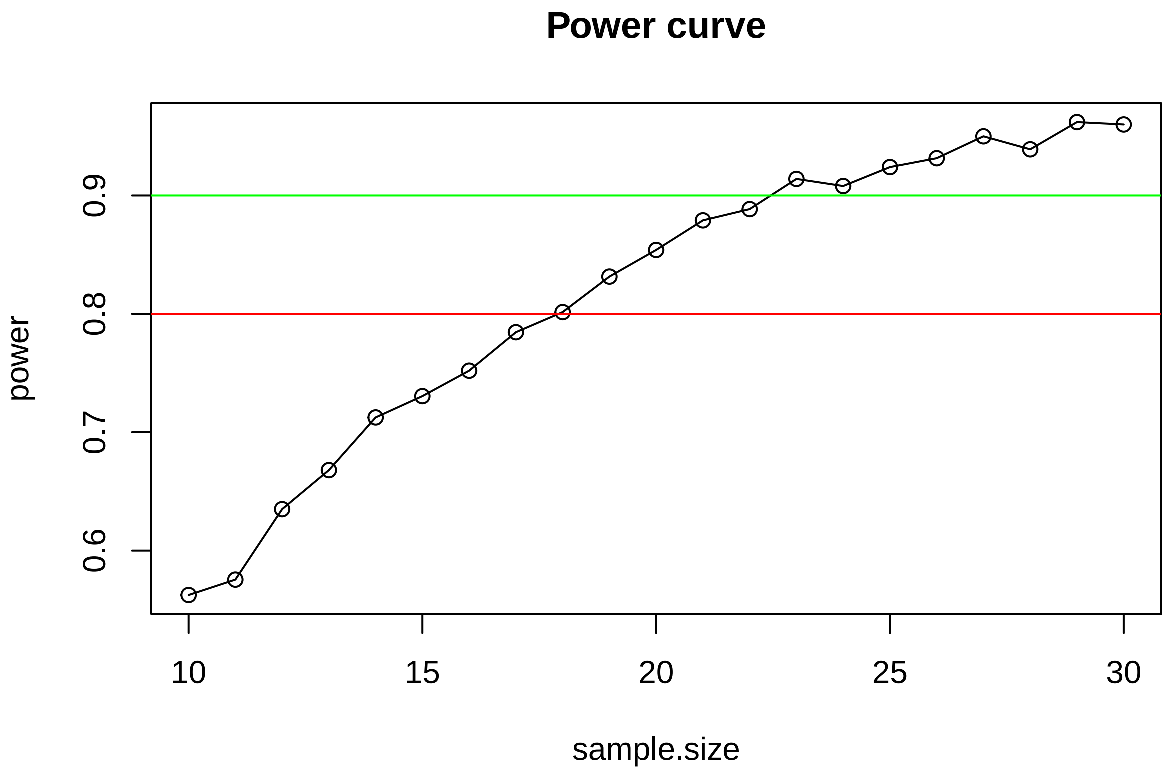
```
e3 <- -.8 # -1 uv # uV # minimum difference between conditions

set.seed(777)
pwr2 <- pwr_calc(e0,e3,e0_sd,eres_sd,nsims,ssizes)

summary3 <- aggregate(pwr2$p,by = list(pwr2$ssize), FUN = function(x) sum(x < 0.05)/length(x))
```

```r
colnames(summary3) <- c('sample.size','power')
print(summary3)
```

```
##    sample.size  power
## 1           10 0.5625
## 2           11 0.5755
## 3           12 0.6350
## 4           13 0.6680
## 5           14 0.7125
## 6           15 0.7305
## 7           16 0.7520
## 8           17 0.7845
## 9           18 0.8015
## 10          19 0.8315
## 11          20 0.8540
## 12          21 0.8790
## 13          22 0.8885
## 14          23 0.9140
## 15          24 0.9080
## 16          25 0.9240
## 17          26 0.9315
## 18          27 0.9500
## 19          28 0.9390
## 20          29 0.9620
## 21          30 0.9600
```

And plot the power curve:

```r
with(summary3, plot(sample.size, power, type = 'ol'))
abline(h=.8, col='red')
abline(h=.9, col='green')
title('Power curve')
```



Here, satisfactory statistical power of 0.8 is achieved above N = 18, while N = 23 provides power above 0.9.

This analysis indicates that the planned sample size of N = 30 will provide enough statistical power to reliably detect a difference between conditions of 0.8uV.