Power analysis for quantitative analysis

Nele Albers

23 June, 2022

Contents

Introduction

This file is meant to reproduce our Bayesian power analysis results reported in the "Participants"-section of our paper. For the power analysis, we follow the Monte Carlo approach by (Kruschke (2014)). We compute the power for sample sizes of 71, 148 and 300, which are the smallest, median, and largest number of samples we obtained for an interaction scenario group used in our quantitative analysis.

Authored by Nele Albers, Mark A. Neerincx, Kristell M. Penfornis, and Willem-Paul Brinkman.

Setup

First, we load the packages that we need.

```
library(BayesianFirstAid)  # to compute correlations
library(dplyr)  # for the %>% function
library(mvtnorm)
library(tidyverse)
```

Function to generate data

Now we create a function to generate sample data with a moderate correlation of 0.3 (Cohen (1992)) using a certain random seed and number of samples. This function is based on code by A. Solomon Kurz: https://solomonkurz.netlify.app/post/bayesian-power-analysis-part-i/.

```
sim_d <- function(seed, n){
```

```
  m <- c(0, 0)  # the means
  s <- c(1, 1)  # the sigmas
  r <- c(0.3)  # the correlation, this is a medium effect size

  # Here's the variance/covariance matrix
  v <-
    matrix(c((s[1] * s[1]),        (s[2] * s[1] * r[1]),
             (s[2] * s[1] * r[1]), (s[2] * s[2])),
           nrow = 2, ncol = 2)

  # After setting our seed, we're ready to simulate with `rmvnorm()`
  set.seed(seed)
  d <-
    rmvnorm(n = n, mean = m, sigma = v) %>%
    as_tibble() %>%
    set_names("x", "y")
}
```

Power analysis with 71 samples

Now we will repeatedly generate data sets with 71 samples and compute Bayesian Spearman correlations for each of them. For each correlation, we keep track of the lower bound of the resulting 95% HDI.

```
num_samples = 71  # number of data points per generated data set
num_iter = 1000
hdi_lower_list = vector( ,num_iter)

for (iter in 1:num_iter){
  d <- sim_d(iter, num_samples)

  # To get the Spearman correlation coefficient, we need to convert the data to ranks.
  rank_x = rank(d$x, ties.method = 'average')
  rank_y = rank(d$y, ties.method = 'average')

  # Compute Pearson correlation
  fit <- bayes.cor.test(rank_x, rank_y)
  results <- summary(fit)
  hdi_lower <- results[1, 5]  # lower-bound of 95% HDI for correlation
  hdi_lower_list[iter] = hdi_lower
}
```

Now we check how many of the lower bounds are greater than 0. This allows us to compute the power.

```
power = length(which(hdi_lower_list > 0))/num_iter

print(paste0("Power for 71 samples: ", round(power, 2)))
```

```
## [1] "Power for 71 samples: 0.68"
```

Power analysis with 148 samples

Next, we repeatedly generate data sets with 148 samples and compute Bayesian Spearman correlations for each of them. For each correlation, we keep track of the lower bound of the resulting 95% HDI.

```r
num_samples = 148  # number of data points per generated data set
num_iter = 1000
hdi_lower_list = vector( ,num_iter)

for (iter in 1:num_iter){
  d <- sim_d(iter, num_samples)

  # To get the Spearman correlation coefficient, we need to convert the data to ranks.
  rank_x = rank(d$x, ties.method = 'average')
  rank_y = rank(d$y, ties.method = 'average')

  # Compute Pearson correlation
  fit <- bayes.cor.test(rank_x, rank_y)
  results <- summary(fit)
  hdi_lower <- results[1, 5]  # lower-bound of 95% HDI for correlation
  hdi_lower_list[iter] = hdi_lower
}
```

Now we check how many of the lower bounds are greater than 0. This allows us to compute the power for a sample size of 148.

```r
power = length(which(hdi_lower_list > 0))/num_iter

print(paste0("Power for 148 samples: ", round(power, 2)))
```

```
## [1] "Power for 148 samples: 0.95"
```

Power analysis with 300 samples

Next, we repeatedly generate data sets with 300 samples and compute Bayesian Spearman correlations for each of them. For each correlation, we keep track of the lower bound of the resulting 95% HDI.

```r
num_samples = 300  # number of data points per generated data set
num_iter = 1000
hdi_lower_list = vector( ,num_iter)

for (iter in 1:num_iter){
  d <- sim_d(iter, num_samples)

  # To get the Spearman correlation coefficient, we need to convert the data to ranks.
  rank_x = rank(d$x, ties.method = 'average')
  rank_y = rank(d$y, ties.method = 'average')

  # Compute Pearson correlation
  fit <- bayes.cor.test(rank_x, rank_y)
  results <- summary(fit)
  hdi_lower <- results[1, 5]  # lower-bound of 95% HDI for correlation
  hdi_lower_list[iter] = hdi_lower
}
```

Now we check how many of the lower bounds are greater than 0. This allows us to compute the power for a sample size of 300.

```r
power = length(which(hdi_lower_list > 0))/num_iter
```

```r
print(paste0("Power for 300 samples: ", round(power, 2)))
```

```
## [1] "Power for 300 samples: 1"
```

References

Cohen, Jacob. 1992. “A Power Primer.” *Psychological Bulletin* 112 (1): 155.
Kruschke, John. 2014. “Doing Bayesian Data Analysis: A Tutorial with r, JAGS, and Stan.”