Power Analysis for AR(1) Multilevel Models in Intensive Longitudinal Designs

Ginette Lafit

2023-06-07

Table of contents

This file needs attention!

Settings things up

Before we proceed, we need to ensure we have several packages installed and loaded into our R session. For the scripts below, we will use the following packages:

- htmltools
- DT
- nlme
- ggplot2
- gridExtra
- data.table
- plyr
- dplyr
- formattable
- tidyr

- MASS
- kableExtra
- future.apply
- PowerAnalysisIL
- tictoc
- viridis
- ggpubr

Which we can install in one go as follows:

```
# Prepare the package list.
packages = c(
    "htmltools", "DT", "nlme", "ggplot2", "gridExtra",
    "data.table", "plyr", "dplyr", "formattable", "tidyr",
    "MASS", "kableExtra", "future.apply", "tictoc",
    "viridis", "ggpubr", "devtools"
)

# Install packages.
install.packages(packages)
```

> **Tip**
>
> You may consider first checking if the packages are installed before actually installing them. Nevertheless, the code above will not reinstall packages that are already installed and up-to-date.

At last, we can load the packages into our `R` session:

```
library(htmltools)
library(DT)
library(nlme)
library(ggplot2)
library(gridExtra)
library(data.table)
library(plyr)
library(dplyr)
library(formattable)
library(tidyr)
library(MASS)
library(future.apply)
library(tictoc)
```

```
library(kableExtra)
library(viridis)
library(ggpubr)
```

Finally, we will also install the `Shiny` application package `PowerAnalysisIL` because it contains the functions we will use to conduct the power analysis.

```
# Install the shiny application package.
devtools::install_github("ginettelafit/PowerAnalysisIL")

## Load package `PowerAnalysisIL`.
library(PowerAnalysisIL)
```

Description

We now focus on conducting a **sensitivity** power analysis to investigate different factors that influence statistical power when testing group differences in the autoregressive effect.

The goal of the new study is to determine the sample size (i.e., N_0 the number of control participants and N_1 the number of participants diagnosed with depression), when assuming the new study will include 70 repeated measurement occasions.

To estimate this research question we will use a multilevel AR(1) model with a cross-level interaction effect between a level 1 continuous predictor (lagged NA) and level 2 binary predictor (diagnosis):

$$\begin{aligned} \text{NA}_{it} = \beta_{00} + \beta_{01}\text{Diagnosis}_i + \beta_{10}\text{NA}_{it-1} + \\ \beta_{01}\text{Diagnosis}_i\text{NA}_{it-1} + \nu_{0i} + \nu_{1i}\text{NA}_{it-1} + \epsilon_{it} \end{aligned} \quad (1)$$

where β_{00} is the fixed intercept, β_{01} represents the main effect of Diagnosis, β_{10} is the fixed autoregressive effect, and β_{11} represents the differences in the autoregressive effect between persons diagnosed with depression and control participants. The level 1 predictor (lagged NA) is centered within-persons and within-days.

In the model above, we assume the level 1 errors are independent and normally distributed with mean zero and variance σ^2_ϵ.

Between-person differences in the autoregressive effect are captured by including a random intercept ν_{0i} and random slope ν_{1i}. These random effects are multivariate normal distributed with mean zero and covariance matrix Σ_ν:

$$\Sigma_\nu = \begin{bmatrix} \sigma^2_{\nu_0} & \sigma_{\nu_{01}} \\ \sigma_{\nu_{01}} & \sigma^2_{\nu_1} \end{bmatrix}$$

In this model, it is also assumed that the level 2 random effects and the level 1 errors are independent.

To investigate group differences in the autoregressive effect of NA between persons diagnosed with depression and control participants we are going to conduct the following hypothesis test:

$$H_0 : \beta_{11} = 0$$

$$H_1 : \beta_{11} \neq 0$$

The goal of this exercise is to conduct a sensitivity power analysis to investigate:

- differences in statistical power when varying the number of measurement occasions T due to different levels of compliance (i.e., 60% and 80%)
- differences in statistical power when varying the value of β_{11}: we assume $\beta_1 1$ is 10% lower/higher than the one obtained using the Leuven clinical data set

Determining model parameter values

To obtain the values of the model parameters we will use data from the Leuven clinical study. The code to estimate the multilevel model is included in the exercise `Multilevel model estimation using the Leuven Clinical Dataset`. The output of the fitted model is:

Thus, the values of the model parameter that will be used to conduct the power analysis are:

$$\begin{aligned}
\beta_{00} &= 6.82 \quad \text{fixed intercept} \\
\beta_{01} &= 16.33 \quad \text{main effect of diagnosis} \\
\beta_{10} &= 0.31 \quad \text{fixed autoregressive effect} \\
\beta_{10} &= 0.12 \quad \text{cross-level interaction effect} \\
\sigma_\epsilon &= 8.75 \quad \text{std. deviation level 1 errors} \\
\sigma_{\nu_0} &= 5.79 \quad \text{std. deviation random intercept} \\
\sigma_{\nu_1} &= 0.14 \quad \text{std. deviation random slope} \\
\rho_{\nu_{01}} &= -0.199 \quad \text{correlation between the random effects}
\end{aligned}$$

Estimation output

```
## Random effects:
##  Formula: ~1 + NA.lag | PID
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 5.7874498 (Intr)
## NA.lag      0.1402727 -0.199
## Residual    8.7540300
##
## Fixed effects:  NA. ~ 1 + MDD + NA.lag + MDD * NA.lag
##                 Value Std.Error   DF   t-value p-value
## (Intercept)  6.824841 0.9800413 3911  6.963830  0.0000
## MDD         16.326600 1.5896204   76 10.270754  0.0000
## NA.lag       0.313887 0.0366665 3911  8.560574  0.0000
## MDD:NA.lag   0.116184 0.0472239 3911  2.460275  0.0139
```

Figure 1: Estimated model parameter values

Set the model parameters values

For our sensitivity power analysis, we can use the estimated parameter values using the Leuven clinical data set:

```
# Number of persons in reference group (Control participants).
N.0 = c(20, 40, 60, 100)

# Number of participants diagnosed with MDD.
N.1 = c(20, 40, 60, 100)

# Number of measurement occasions.
T.obs = 70

# If Ylag.center = TRUE, person-mean centered the lagged outcome, otherwise the lagged pre
Ylag.center = FALSE

# Fixed intercept for participants in the reference group.
b00 = 6.82

# Differences in the fixed intercept between the two groups.
b01.Z = 16.33
```

```r
# Fixed autoregressive effect for participants in the reference group.
b10 = 0.31

# Differences in the fixed autoregressive effect between the two groups.
b11.Z = 0.12

# Std. deviation of the level 1 errors.
sigma = 8.75

# Std. deviation of the random intercept.
sigma.v0 = 5.79

# Std. deviation of the random autoregressive effect.
sigma.v1 = 0.14

# Correlation between the random intercept and random autoregressive effect.
rho.v = -0.199

# Significant level.
alpha = 0.05

# Select the side of the test as follows.

# One-side test H1: b01.Z > 0
# side.test = 1

# One-side test H1: b01.Z < 0
# side.test = 2

# One-side test H1: b01.Z is different from 0
side.test = 3

# Set the optimization method for `lme4`.
Opt.Method = "REML"

# Number of Monte Carlo replicates.
R = 10

# Set seed or the random number generator for reproducibility.
set.seed(123)
```

> **!** Important
>
> In the code above we set the number of replications `R` to 10. This is just so we can compile the document quickly. In practice, you should set `R` to a much higher number. We recommend at least 1000 replications.

Conduct the sensitivity power analysis

We can now conduct a sensitivity analysis to assess how power varies under different conditions. First, we focus on the scenario in which we varied the value of the main effect of interest.

Varying the value of the main effect of interest

```
# We investigate what happens with power when the main effect of interest is 10%
# larger/smaller than the expected value based on previous studies.

# Differences in the fixed autoregressive effect between the two groups.
b11.Z.list <- b11.Z * c(0.9, 1, 1.10)

# Prepare the lists where we will store the results.
Power.Simulation.list.REML.list <- list()
R.Converge.Simulation.list.REML.list <- list()

# Function for computing power using simulation-based approach using REML.
for (i in 1:length(b11.Z.list)) {
    Power.Simulation.list.REML.list[[i]] <- lapply(1:length(N.0), function(n) {
        Power.Curve.Simulation.ML.AR.Group.Diff(
            N.0[n],
            N.1[n],
            T.obs,
            Ylag.center,
            b00,
            b01.Z,
            b10,
            b11.Z.list[i],
            sigma,
            sigma.v0,
            sigma.v1,
            rho.v,
```

```
            alpha,
            side.test,
            Opt.Method = "REML",
            R
        )
    })

    # Number of replicates that converge.
    R.Converge.Simulation.list.REML.list[[i]] <- unlist(
        lapply(1:length(N.0), function(n) {
            Power.Simulation.list.REML.list[[i]][[n]]$n.R
        })
    )
}

# Print the number of replicates that converged.
R.Converge.Simulation.list.REML.list
```

We plot the power curves for the sensitivity analysis for the value of the difference in NA between participants diagnosed with MDD and participants in the control group.

```
# Construct a matrix with the results of the sensitivity power analysis.
Power.b11.power <- matrix(0, length(N.0), length(b11.Z.list))

# Get the values of power.
for (i in 1:length(b11.Z.list)) {
    Power.b11.power[, i] = unlist(
        lapply(
            1:length(N.0), function(n) {
                Power.Simulation.list.REML.list[[i]][[n]]$power.hat.lme["b11.Z"]
            }
        )
    )
}

# Convert to long format.
Power.b11.power = melt(Power.b11.power, id.vars = "Power")

# Add column names.
colnames(Power.b11.power) <- c("N.0", "Value", "Power")

# Store the values.
```

```r
Power.b11.power$N.0 <- rep(N.0, length(b11.Z.list))
Power.b11.power$N.1 <- rep(N.1, length(b11.Z.list))
Power.b11.power$Value <- rep(b11.Z.list, each = length(N.0))
Power.b11.power <- Power.b11.power[, c("N.0", "N.1", "Value", "Power")]

# Convert to data frame.
Power.b11.power <- data.frame(Power.b11.power)

# Create curve with the results of the sensitivity power analysis.
ggplot(Power.b11.power, aes(
        x = N.0,
        y = Power,
        colour = as.factor(Value),
        group = as.factor(Value)
    )) +
    geom_line(
        size = 1
    ) +
    geom_point(
        size = 1
    ) +
    labs(
        title = "Power curves for the difference in
                the inertia of NA between participants \n
                diagnosed with MDD and participants in the
                control group",
        x = "Number of Persons",
        color = "Difference in \n
                inertia of NA"
    ) +
    scale_x_continuous(
        name = "Participants",
        breaks = N.0,
        labels = c(paste(N.0, N.1, sep = ";"))
    ) +
    theme(
        axis.text.x = element_text(angle = 90)
    ) +
    scale_colour_manual(
        values = c("#CC79A7", "#56B4E9", "#009E73")
    ) +
```

```
    geom_hline(
        yintercept = 0.9, linetype = "dashed"
    ) +
    theme(
        legend.position = "bottom",
        legend.title = element_text(size = 10)
    ) +
    theme_minimal()
```

We now conduct a sensitivity analysis to assess how power varies under different conditions. This time, we focus on the scenario in which we varied the average compliance rate.

Varying the average compliance rate

```
# Average compliance rate.
T.obs.list <- T.obs * c(0.6, 0.8, 1)

# Prepare the lists where we will store the results.
Power.Simulation.list.REML.Tpoints.list <- list()
R.Converge.Simulation.list.REML.Tpoints.list <- list()

# Function for computing power using simulation-based approach using REML.
for (i in 1:length(T.obs.list)) {
    Power.Simulation.list.REML.Tpoints.list[[i]] <- lapply(1:length(N.0), function(n) {
        Power.Curve.Simulation.ML.AR.Group.Diff(
            N.0[n],
            N.1[n],
            T.obs.list[i],
            Ylag.center,
            b00,
            b01.Z,
            b10,
            b11.Z,
            sigma,
            sigma.v0,
            sigma.v1,
            rho.v,
            alpha,
            side.test,
            Opt.Method = "REML",
            R
```

```
        )
    })

    # Number of replicates that converge.
    R.Converge.Simulation.list.REML.Tpoints.list[[i]] <- unlist(
        lapply(
            1:length(N.0), function(n) {
                Power.Simulation.list.REML.Tpoints.list[[i]][[n]]$n.R
            }
        )
    )
}

# Number of replicates that converge.
R.Converge.Simulation.list.REML.Tpoints.list
```

We plot the power curves for the sensitivity analysis for the value of the difference in NA between participants diagnosed with MDD and participants in the control group.

```
# Construct a matrix with the results of the sensitivity power analysis.
Power.b11.power.Tpoints <- matrix(0, length(N.0), length(T.obs.list))

# Get the values of power.
for (i in 1:length(T.obs.list)) {
    Power.b11.power.Tpoints[, i] <- unlist(
        lapply(
            1:length(N.0), function(n) {
                Power.Simulation.list.REML.Tpoints.list[[i]][[n]]$power.hat.lme["b11.Z"]
            }
        )
    )
}

# Transform to format for the plot.
Power.b11.power.Tpoints <- melt(Power.b11.power.Tpoints, id.vars = "Power")

# Add column names.
colnames(Power.b11.power.Tpoints) <- c("N.0", "Value", "Power")

# Store the values.
Power.b11.power.Tpoints$N.0 <- rep(N.0, length(T.obs.list))
```

```r
Power.b11.power.Tpoints$N.1 <- rep(N.1, length(T.obs.list))
Power.b11.power.Tpoints$Value <- rep(T.obs.list, each = length(N.0))
Power.b11.power.Tpoints <- Power.b11.power.Tpoints[, c("N.0", "N.1", "Value", "Power")]

# Create curve with the results of the sensitivity power analysis.
Power.b11.power.Tpoints <- data.frame(Power.b11.power.Tpoints)

# Create curve with the results of the sensitivity power analysis.
ggplot(Power.b11.power.Tpoints, aes(
        x = N.0,
        y = Power,
        colour = as.factor(Value),
        group = as.factor(Value)
    )) +
    geom_line(
        size = 1
    ) +
    geom_point(
        size = 1
    ) +
    labs(
        title = "Power curves for the difference in
                the inertia of NA between participants \n
                diagnosed with MDD  and participants in
                the control group",
        x = "Number of Persons",
        color = "Number of \n time points"
    ) +
    scale_x_continuous(
        name = "Participants",
        breaks = N.0,
        labels = c(paste(N.0, N.1, sep = ";"))
    ) +
    theme(
        axis.text.x = element_text(angle = 90)
    ) +
    scale_colour_manual(
        values = c("#CC79A7", "#56B4E9", "#009E73")
    ) +
    geom_hline(
        yintercept = 0.9,
```

```
        linetype = "dashed"
    ) +
    theme(
        legend.position = "bottom",
        legend.title = element_text(size = 10)
    ) +
    theme_minimal()
```

Session information

Using the command below, we can print the `session` information (i.e., operating system, details about the `R` installation, and so on) for reproducibility purposes.

```
# Session information.
sessionInfo()
```

```
R version 4.3.0 (2023-04-21)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.4

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  l

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: Europe/Amsterdam
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.3.0  fastmap_1.1.1   cli_3.6.1       tools_4.3.0
 [5] htmltools_0.5.5 rstudioapi_0.14 yaml_2.3.7      rmarkdown_2.22
 [9] knitr_1.43      jsonlite_1.8.5  xfun_0.39       digest_0.6.31
[13] rlang_1.1.1     evaluate_0.21
```