Sample size estimation for an effect size of interest in a GLMM design

Pratik Bhandari

Psycholinguistic experiment analysis & design (PLEAD), vd-mit

16/03/2022

Section 1

Data and setup

Setup

Set seed and load packages.

```
set.seed(2022)

library(tidyr);
library(dplyr);
library(ggplot2);
library(simr)
```

Create a dataset

- Data from a [hypothetical] pilot experiment

The dataset

- Within group design, with n=26

The dataset

- Within group design, with n=26
- 2 conditions:

The dataset

- Within group design, with n=26
- 2 conditions:
    - Noise: Mild and Severe

The dataset

- Within group design, with n=26
- 2 conditions:
  - Noise: Mild and Severe
  - Cloze (=predictability): High and Low

The dataset

- Within group design, with n=26
- 2 conditions:
  - Noise: Mild and Severe
  - Cloze (=predictability): High and Low
- 10 items in a repeated measures design

The dataset

- Within group design, with n=26
- 2 conditions:
  - Noise: Mild and Severe
  - Cloze (=predictability): High and Low
- 10 items in a repeated measures design
- High and Low cloze sentences are embedded in Mild and Severe background noise.

The dataset

- Within group design, with n=26
- 2 conditions:
  - Noise: Mild and Severe
  - Cloze (=predictability): High and Low
- 10 items in a repeated measures design
- High and Low cloze sentences are embedded in Mild and Severe background noise.
- 10items * 4conditions = 40 trials for each of the 26 participants.

The dataset

```
dat <- expand_grid(
  participants = paste(letters, 1:26, sep = '_'),
  noise = c('mild', 'severe'),
  cloze = c('high', 'low'),
  item = paste('item', 1:10, sep = '_'))
dat$acc <- sample(c(0,1), nrow(dat), replace = T)

# Assign class to the variables of interest:
dat$noise <- as.factor(dat$noise)
dat$cloze <- as.factor(dat$cloze)
dat$acc <- as.integer(dat$acc)
```

The dataset

```
## # A tibble: 1,040 x 5
##    participants noise cloze item      acc
##    <chr>        <fct> <fct> <chr>   <int>
##  1 a_1          mild  high  item_1      1
##  2 a_1          mild  high  item_2      0
##  3 a_1          mild  high  item_3      1
##  4 a_1          mild  high  item_4      0
##  5 a_1          mild  high  item_5      0
##  6 a_1          mild  high  item_6      1
##  7 a_1          mild  high  item_7      1
##  8 a_1          mild  high  item_8      0
##  9 a_1          mild  high  item_9      1
## 10 a_1          mild  high  item_10     1
## # ... with 1,030 more rows
```

The dataset

- Note that the experimental conditions are just labels.

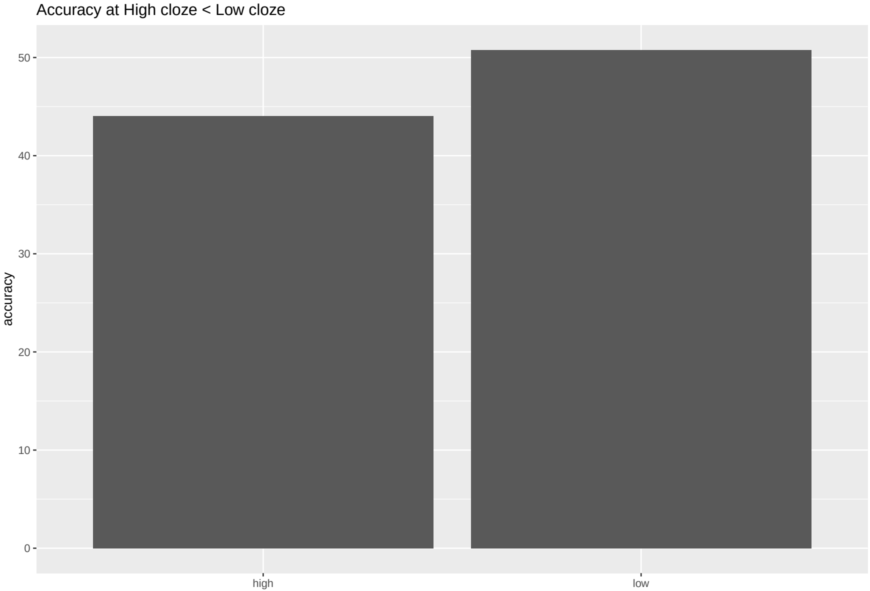
The dataset

- Note that the experimental conditions are just labels.
- Participants' response accuracy in High cloze sentences does not necessarily have to be higher than Low cloze sentences.

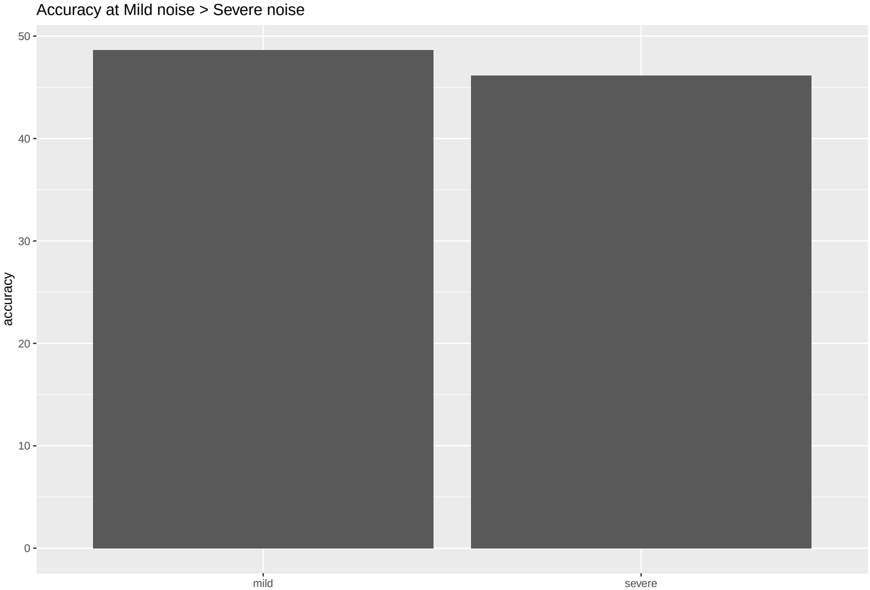
The dataset

- Note that the experimental conditions are just labels.
- Participants' response accuracy in High cloze sentences does not necessarily have to be higher than Low cloze sentences.
  - Same thing for the noise condition (Severe vs. Mild)

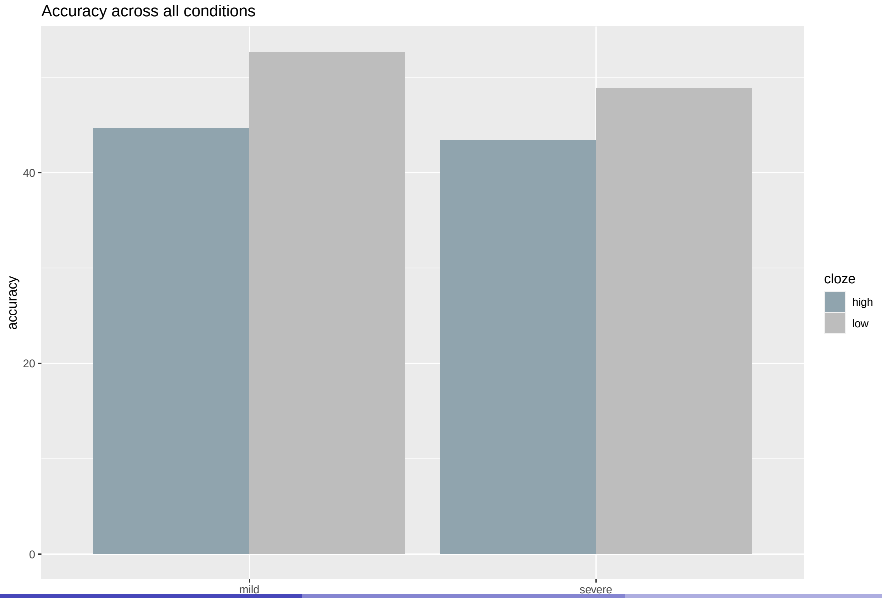
Data summary



Data summary



Data summary



Section 2

Model

Assign contrast

- Treatment contrasts for both *Cloze* and *Noise*

```
contrasts(dat$cloze) <- contr.treatment(2) # High cloze is the
contrasts(dat$noise) <- contr.treatment(2) # Mild noise is the
```

Assign contrast

- Treatment contrasts for both *Cloze* and *Noise*

```
contrasts(dat$cloze) <- contr.treatment(2) # High cloze is the
contrasts(dat$noise) <- contr.treatment(2) # Mild noise is the
```

- High cloze, Mild noise conditions in the Intercept

Run a GLMM

- An intercept-only model, with all the main effects and interaction in the fixed effects term.

```
m0 <- glmer(acc ~ 1 + (cloze + noise + cloze:noise) +
                   (1 | participants) + (1 | item),
                  data=dat, family = "binomial",
                  control = glmerControl(calc.derivs = FALSE,
                                         optimizer="bobyqa",
                                         optCtrl = list(maxfun=
                  nAGQ = 0, na.action = na.exclude)
```

- Is this an appropriate model for a repeated measures design?

Model summary

```
summary(m0)$coef
```

```
##                  Estimate Std. Error    z value   Pr(>|z|)
## (Intercept)   -0.21710136  0.1384937 -1.5675898 0.11697690
## cloze2         0.32538206  0.1764229  1.8443298 0.06513509
## noise2        -0.04701666  0.1770520 -0.2655528 0.79058366
## cloze2:noise2 -0.10759011  0.2495983 -0.4310531 0.66642975
```

- Remember that High cloze and Mild noise conditions are the reference levels.

Observed effect size

- Interest: Noise condition, i.e., `noise2`

```
fixef(m0)[3] #fixef(m0)['noise2']
```

```
##      noise2
## -0.04701666
```

- The accuracy at Severe noise is lower than at Mild noise, although statistically not significant.

Expected effect size

- Say, we expect the main effect of noise to be 0.05.
- Then, we assign this **effect size of interest** to the model.

```
fixef(m0)['noise2'] <- 0.05 #changed from -0.04701666 to 0.05

# If I assume the effect size will not change in the "real dat
# `fixef(m0)['noise2'] <- fixef(m0)['noise2']` instead.
```

[Questionable: Is it theoretically driven — changing the observed effect size?]

Section 3

Power analyses

Increase sample size

- We want to find out what the power is when the sample size is increased up to a certain number, e.g., n=160.

```
N_tar_grid     <- seq(32, 160, by = 32) # 32, 64, 96, 128, 160
# A vector of five which we'll use later

fit_ext_simr0  <- simr::extend(m0, #increases sample size for
                               along = "participants",
                               n = max(N_tar_grid)
                               )
```

- `nrow(getData(fit_ext_simr0))` = 6400
- `nrow(getData(m0))` = 1040

Increase sample size

To the best of our understanding, adding the rows in extend() is done via upsampling and is performed once before the power estimation procedure starts. Then, in the powerCurve() function, the rows removing is done by subsetting the "extended" model data frame to the target sample size deterministic (by keeping the first observations in the sample). Therefore, the same set of observations will be used in the SIMR power estimation procedure across all iterations when simulating new values of the response.

— Karas and Crainiceanu (2021)

Power analysis

- To visualize the power at different sample sizes
- Refit the model to a simulated dataset:

```
B_boot <- 100 # number of boot repetitions within one experime
# recommended: at least 1000 simulations

noise_power0 <- simr::powerSim(fit_ext_simr0,
                               nsim = B_boot,
                               progress = TRUE,
                               test = fixed('noise2', 'z'))
```

- Apply tests to the simulated model fit

Power analysis

The test will either correctly detect the effect, or make a Type II error in failing to detect the effect.
The power of the test is then estimated based on the proportion of successful tests.

— *Green and MacLoed (2016)*

Power analysis

```
noise_power0
```

```
## Power for predictor 'noise2', (95% confidence interval):
##        5.00% ( 1.64, 11.28)
##
## Test: z-test
##       Effect size for noise2 is 0.050
##
## Based on 100 simulations, (0 warnings, 0 errors)
## alpha = 0.05, nrow = 6400
##
## Time elapsed: 0 h 1 m 8 s
```

- The power to reject the null hypothesis is about 5%.

Power analysis

```
noise_power0
```

```
## Power for predictor 'noise2', (95% confidence interval):
##        5.00% ( 1.64, 11.28)
##
## Test: z-test
##       Effect size for noise2 is 0.050
##
## Based on 100 simulations, (0 warnings, 0 errors)
## alpha = 0.05, nrow = 6400
##
## Time elapsed: 0 h 1 m 8 s
```

- The power to reject the null hypothesis is about 5%.
- Note that this power analysis was performed at the sample size of 160.

Power analysis and power curve

- Estimate power across a pre-specified range of sample sizes.

```
noise_powercurve0 <- simr::powerCurve(fit_ext_simr0,
                            along = 'participants',
                            nsim = B_boot,
                            breaks = N_tar_grid, #seq(32, 160,
                            progress = TRUE,
                            test = fixed('noise2', 'z'))
```

Power analysis and power curve

- Estimate power across a pre-specified range of sample sizes.

```
noise_powercurve0 <- simr::powerCurve(fit_ext_simr0,
                        along = 'participants',
                        nsim = B_boot,
                        breaks = N_tar_grid, #seq(32, 160
                        progress = TRUE,
                        test = fixed('noise2', 'z'))
```

- Check if any of the pre-specified sample sizes provides 80% power for the effect size of interest.

Power curve

- The power for `noise` at different pre-specified sample sizes:

```
noise_powercurve0
```

```
## Power for predictor 'noise2', (95% confidence interval),
## by largest value of participants:
##      127:  3.00% ( 0.62,  8.52) - 1280 rows
##      156:  5.00% ( 1.64, 11.28) - 2560 rows
##       40:  7.00% ( 2.86, 13.89) - 3840 rows
##        7:  8.00% ( 3.52, 15.16) - 5120 rows
##       99: 13.00% ( 7.11, 21.20) - 6400 rows
##
## Time elapsed: 0 h 2 m 36 s
```

Power curve

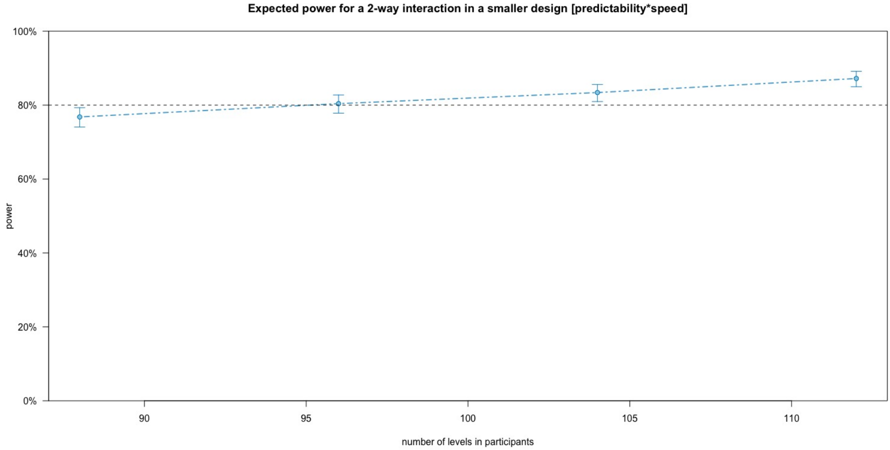


Figure 1: A typical powercurve.

Problems with the analysis

- Problem 1: None of the pre-specified sample sizes have >80% power
- Problem 2: Power for n=99 > Power for n=127

Sources of the problems

- Not enough simulations: Some simulated datasets might be poor fit
- Repeated measures design: Can we ignore other variables, only consider `noise2` and extend along `participants`?

Section 4

Afterthoughts

Discussion

- What do you consider an effect size of interest?

Discussion

- What do you consider an effect size of interest?
  - The units of effect size, like η_p^2, ω^2, etc. are also crucial in determining the power and sample size (Albers & Lakens, 2018).

Discussion

- What do you consider an effect size of interest?
  - The units of effect size, like η_p^2, ω^2, etc. are also crucial in determining the power and sample size (Albers & Lakens, 2018).
- How do you calculate power wrt one (or more) effects of interest?

Resources

- https://dpaape.shinyapps.io/ipower/ (posted on *Workplace* on Sep 07, 2019)
- `simr` by Green and MacLeod, 2015
- Upstrap in complex models by Karas and Crainiceanu, 2021
- Follow-up bias by Albers and Lakens, 2018

Test slide

- Coffee is overrated.
- Drink water, or tea.
- Just kidding ;)