Appendix: Power Analysis

2021-11-03

Design

For this power analysis we will simulate 20 labs contributing 20 infants (400 participants) from 5 to 12 months of age.

Notes: MB1 overall effect size was 0.29 for the single-screen central fixation (CF) method, with additional effect of 0.21 for HPP, and eye-tracking (ET) yielding a slight (non-significant) decrease in effect of -0.06. We expect to have 5-6 labs running HPP (100-120 infants), and the other 15-20 labs running CF/ET.

Factors:

- *familiarized_rule*: indicates the sequence to which infants were exposed during familiarization (ABA or ABB). Infants were exposed to only one sequence, with the sequence determined by random assignment. (GK: not counterbalanced per lab?)
- *trial_type*: indicates whether each test sequence followed the same rule to which the infant was familiarized or a different rule. For example, if an infant heard an ABA rule during familiarization, ABA trials would be the *same* trial type and ABB trials would be the *different* trial type (each infant get 6 of same and 6 different)
- *trial_num*: indicates the sequential order in which test trials were presented. Trial number thus ranges from 1 to 12.
- *age_mos*: the infants' age in months (5.0-12.0), centered in *age* column.
- *procedure*: indicates the experimental method that was used to record infants' responses to the stimuli: headturn preference procedure (HPP), central fixation (CF), or eye tracking (ET).
- *test_order*: indicates which of the four pseudorandom test orders (from our provided scripts) were used to present test trials to the infant.
- *multilingual_exposure*: indicates the infants' exposure to the secondary/primary language, ranging from 0% (no exposure to a secondary language) to 49% (i.e., baby hears 51% of their primary language and 49% of the secondary language).

To do our power analysis, we will generate 1,000 datasets of this structure with a given effect size (e.g., .3), run the mixed-effects regression for each simulated dataset, and count the number of times that the effect is significant. Note that we generate normally-distributed looking times, assuming that they have already been log-transformed.

Simulate Datasets

```r
set.seed(123) # reproducible sampling

generate_dataset <- function(n_labs=20, n_per_lab=20, effect_sizes=list(type = .3, age = 0, "age*type"=0
  # rewrite to use expand.grid ?
  labID = rep(LETTERS[1:n_labs], each=n_per_lab)
  subjID = 1:(n_labs*n_per_lab)

  # assume each lab uses one procedure
  lab_procedure = sample(c("HPP","CF","EF"), n_labs, replace=T, prob=c(.5,.3,.2))
  procedure = rep(lab_procedure, each=n_per_lab)

  test_order = rep(1:4, 5*n_labs)

  # familiarized rule (ms says randomly assigned: we don't want counterbalanced per lab?)
  familiarized_rule = sample(c("ABB","ABA"), length(subjID), replace=T)

  simd <- tibble(subjID, labID, procedure, test_order, familiarized_rule)

  # uniform random vars
  simd$age_mos = runif(nrow(simd), min=5.0, max=12.0)
  simd$age = scale(simd$age_mos, center=T, scale=T)[,1]

  # should actually be bimodal (use MB1 distro?)
  simd$multilingual_exposure = runif(nrow(simd), min=0, max=.5) # 0=monolingual, .5=50% secondary langu

  # now generate looking times for 12 trials per subject
  for(t in 1:12) {
    simd[,paste0("trial.",t)] = rnorm(n = nrow(simd), mean=0, sd=1) # = .05
  }

  siml <- simd %>% pivot_longer(cols=starts_with("trial."),
                      names_to="trial_num",
                      names_prefix="trial.",
                      values_to="looking_time")

  siml$trial_num = as.numeric(siml$trial_num)
  siml$trial_num_sc = scale(siml$trial_num, center=T, scale=T)

  # 6 same / 6 different per child; should be according to 1 of 4 pseudorandom orders, but we're not  a
  siml$trial_type = rep_len(c(rep("same", 6), rep("different", 6)), nrow(siml)) # each

  per_subj_trial_type = c(rep("same", 6), rep("different", 6))
  # add subject random intercept
  siml$subjInt = 0.0
  for(s in 1:length(unique(siml$subjID))) {
    subjInd = which(siml$subjID==s)
    siml[subjInd,]$trial_type = sample(per_subj_trial_type, 12, replace = F)
    siml[subjInd,]$subjInt = rnorm(1, mean=0, sd=1)
  }

  # add lab random intercept
  siml$labInt = 0.0
  for(lab in labID) {
```

```
    labInd = which(siml$labID==lab)
    siml[labInd,]$labInt = rnorm(1, mean=0, sd=1) # could increase per-lab variability ..
  }

  trial_type = with(siml, ifelse(trial_type=="same", 0, 1))
  error_term = rnorm(nrow(siml), 0, sd=1) + siml$labInt + siml$subjInt
  siml$looking_time = trial_type * effect_sizes$type + siml$age * effect_sizes$age + trial_type * siml$a

  siml$subjID = as.factor(siml$subjID)
  # switch from dummy-code to effects code
  siml$familiarized_rule = as.factor(siml$familiarized_rule)
  siml$trial_type = as.factor(siml$trial_type)
  contrasts(siml$familiarized_rule) = contr.sum(2)
  contrasts(siml$trial_type) = contr.sum(2)
  return(siml)
}
```

Plot Example Dataset

We generate and plot an example dataset with trial_type main effect size of .3, age main effect size of -.2, and an age*trial_type interaction effect size of .3.

```
#siml = generate_dataset(effect_sizes=list(type = .3, age = 0, "age*type"=0))
siml = generate_dataset(effect_sizes=list(type = .3, age = -.2, "age*type"=.3))

dag <- siml %>% group_by(subjID, trial_type, age_mos) %>%
  summarise(looking_time = mean(looking_time)) %>%
  group_by(trial_type, age_mos) %>%
  tidyboot::tidyboot_mean(looking_time) # quite slow..
```

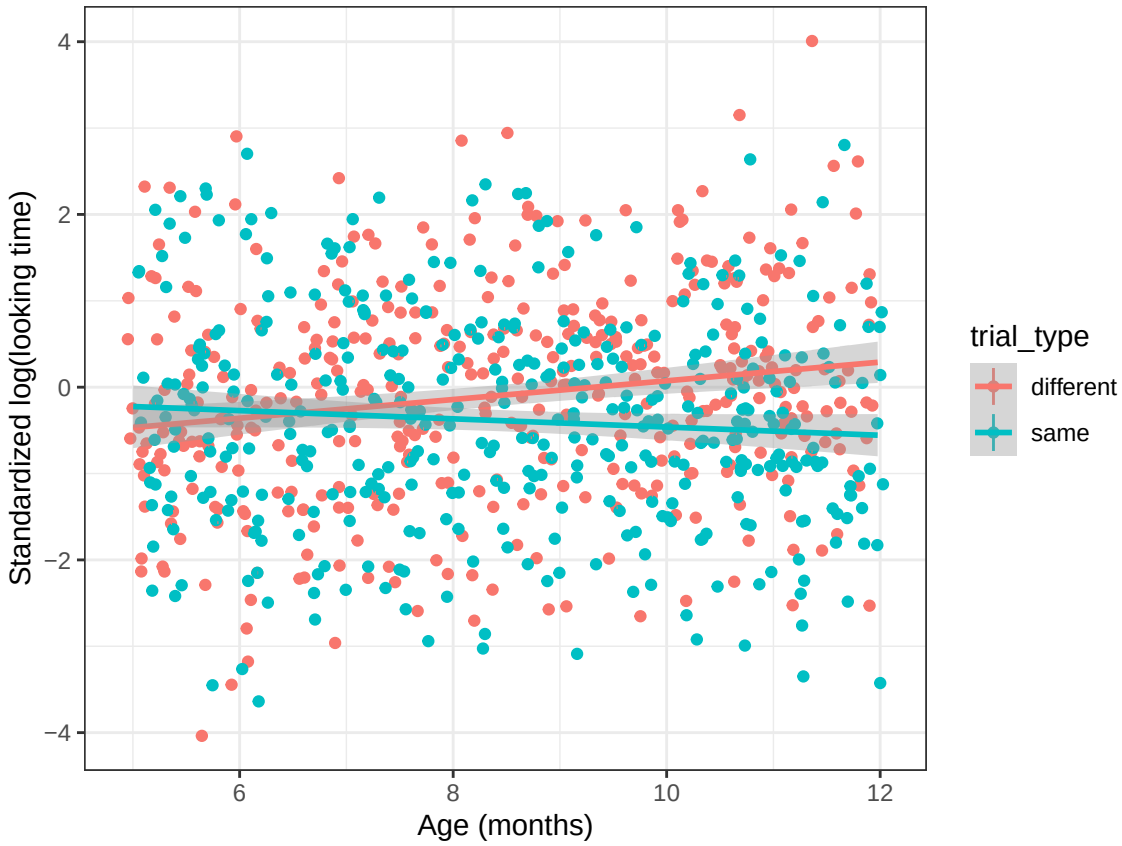
```
## `summarise()` has grouped output by 'subjID', 'trial_type'. You can override using the `.groups` argu
```

```
pos = position_dodge(width=.2)
ggplot(dag, aes(x=age_mos, y=mean, group=trial_type, color=trial_type)) +
  geom_point(aes(y=mean, x=age_mos), pos=pos) +
  ylab("Standardized log(looking time)") + xlab("Age (months)") +
  geom_linerange(aes(ymin=ci_lower, ymax=ci_upper), pos=pos) +
  theme_bw() + geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals
```



Model Structure

Infants' log(looking time) (DV) ~ 1 + familiarization order (ABB vs ABA) * trial_type + age * trial_type (same rule vs different rule at test) + experimental_method (HPP vs central fixation vs eye-tracking) * trial_type + multilingual_exposure * trial_type + trial_num * trial_type + (trial_num*trial_type | subject) + (test_order | lab)

```
# m1 <- lmer(looking_time ~ 1 + trial_type *
#              (familiarized_rule + age + procedure + multilingual_exposure + trial_num) +
#              (trial_num * trial_type | subjID) + (test_order | labID), data=siml)

# model without age
fit_simple_model <- function(siml) {
  m1 <- lmer(looking_time ~ 1 + trial_type * trial_num_sc + (1 | subjID), data=siml)
  return(summary(m1)$coefficients["trial_type1","Pr(>|t|)"]) # "Estimate","t value",
} # trial_type1 = different

# check both
fit_model <- function(siml) {
  m1 <- lmer(looking_time ~ 1 + trial_type * trial_num_sc + trial_type * age + (1 | subjID) + (1 | labI
  sig =c(summary(m1)$coefficients["trial_type1","Pr(>|t|)"],
       summary(m1)$coefficients["trial_type1:age","Pr(>|t|)"])
  return(sig) # "Estimate","t value",
}

# need to update fit_model to return significance of all desired effects (e.g., if effect_size$age!=0)
```

Power Analysis

We use this simplified model for the power analysis: y ~ 1 + trial_type * trial_num + trial_type * age + (1 | subjID) + (1 | labID)

To do the power analysis, we simply generate 1000 datasets with main effect sizes of 0.1, 0.2, and 0.3 for trial type, age, and their interaction, run the above linear mixed-effects model, and report how many times 1) the trial type main effect and 2) the trial type * age interaction is significant.

```
# repeatedly generate data and  significance of trial_typesame
get_power <- function(effect_sizes, N=100, alpha=.05, verbose=F) {
  p = data.frame(type=numeric(), "age*type"=numeric())
  colnames(p) = c("type","age*type")
  for(i in 1:N) {
    p[i,] = fit_model(generate_dataset(effect_sizes=effect_sizes))
  }
  if(verbose) {
    print(paste(length(which(p$type<alpha)), "of",N, "simulations had p <",alpha, "for trial type"))
    print(paste(length(which(p[,"age*type"]<alpha)), "of",N, "simulations had p <",alpha, "for age*trial
  }
  return(p)
}

N = 1000
pvalues_pt1 = get_power(effect_sizes=list(type = .1, age = .1, "age*type"=.1), N=N)

pvalues_pt2 = get_power(effect_sizes=list(type = .2, age = .2, "age*type"=.2), N=N)

pvalues_pt3 = get_power(effect_sizes=list(type = .3, age = .3, "age*type"=.3), N=N)
```

Effect sizes = .1

928 of 1000 simulations had $p < 0.05$ for trial type. 916 of 1000 simulations had $p < 0.05$ for age*trial type.

Effect sizes = .2

1000 of 1000 simulations had $p < 0.05$ for trial type. 1000 of 1000 simulations had $p < 0.05$ for age*trial type.

Effect sizes = .3

1000 of 1000 simulations had $p < 0.05$ for trial type. 1000 of 1000 simulations had $p < 0.05$ for age*trial type.

For context, .25 is the average effect size from the meta-analysis of rule learning, and .3 is the average effect size across all published developmental experiments. Thus, the latter two power simulations probably pertain in our case.