Power Simulation for Reason-Giving Studies

Hanna Schleihauf

1/15/2023

Load Packages

```
library("tidyverse")
library("cowplot")
library("gghalves")
library("lme4")
library("kyotil")
```

Generate Data

```
options(scipen = 9999) # prevent scientific notation
set.seed(1) # set seed to make result replicable
n.dyads <- 80 * 4 # number of dyads
min.age <- 3
max.age <- 11 # range of age

n.n.per.dyad <- 1 # observations per dyad
n.condition <- 2 # number of conditions
n.per.condition <- 1 # observations per dyad and condition (between-dyad design)
dyad.id <- as.factor(paste("dyad", 1:n.dyads, sep = ".")) # creating a dyad ids

# expected performance levels
condition_1.usa.3 <- 0.10 # <- reference level
condition_1.china.3 <- 0.10
condition_1.kenya.3 <- 0.05
condition_1.netherlands.3 <- 0.10
condition_1.usa.11 <- 0.30
condition_1.china.11 <- 0.30
condition_1.kenya.11 <- 0.30
condition_1.netherlands.11 <- 0.30

condition_2.usa.3 <- 0.20
condition_2.china.3 <- 0.20
condition_2.kenya.3 <- 0.15
condition_2.netherlands.3 <- 0.10
condition_2.usa.11 <- 0.70
condition_2.china.11 <- 0.70
condition_2.kenya.11 <- 0.65
```

```r
condition_2.netherlands.11 <- 0.75

start.data <- data.frame(dyad.id)

# duplicate rows according to the number obs. per dyad:
start.data <- start.data[rep(x = 1:nrow(start.data), times = n.n.per.dyad), ]
start.data <- as.data.frame(start.data)
names(start.data) <- "dyad.id"

# create between-dyad predictors
# predictor culture
start.data$condition <-
  as.factor(rep(x = c(".condition_1", ".condition_2"), each = n.dyads /
    2))[as.numeric(start.data$dyad.id)]
start.data$culture <-
  as.factor(rep(x = c(".usa", ".china", ".kenya", ".netherlands"), times = n.dyads /
    3))[as.numeric(start.data$dyad.id)]
start.data$culture <- relevel(start.data$culture, ref = ".usa")
# predictor sex
# start.data <- start.data[order(start.data$culture, start.data$dyad.id),]
start.data$gender <-
  as.factor(rep(
    x = c(".male", ".female", ".male", ".female"),
    each = n.dyads / 4
  ))[as.numeric(start.data$dyad.id)]
# predictor age
start.data$age <-
  rep(x = runif(n = n.dyads, min = 3, max = 11))[as.numeric(start.data$dyad.id)]

# z-transformation of covariates
start.data$z.age <- as.vector(scale(start.data$age))

# dummy code factors and center them for random slopes
start.data$condition.condition_2 <-
  as.numeric(start.data$condition == levels(start.data$condition)[2])
start.data$condition.condition_2.c <-
  as.numeric(start.data$condition.condition_2) -
  mean(as.numeric(start.data$condition.condition_2)) # centering
start.data$culture.china <-
  as.numeric(start.data$culture == levels(start.data$culture)[2])
start.data$culture.china.c <-
  start.data$culture.china - mean(start.data$culture.china) # centering
start.data$culture.kenya <-
  as.numeric(start.data$culture == levels(start.data$culture)[3])
start.data$culture.kenya.c <-
  start.data$culture.kenya - mean(start.data$culture.kenya) # centering
start.data$culture.netherlands <-
  as.numeric(start.data$culture == levels(start.data$culture)[3])
start.data$culture.netherlands.c <-
  start.data$culture.netherlands -
  mean(start.data$culture.netherlands) # centering

# dummy code and center gender to make the estimates unconditional of
```

```
# the reference category
start.data$gender.male <-
  as.numeric(start.data$gender == levels(start.data$gender)[2])
start.data$gender.male.c <-
  start.data$gender.male - mean(start.data$gender.male)
```

Calculate estimates/slopes based on our hypotheses

```
# to calculate slope between two point one need to (y2-y2)/(x2-x1)
# reference levels
intercept <-
  mean(c(qlogis(condition_1.usa.3), qlogis(condition_1.usa.11)))

# slope of age is the effect of reason-giving with the other factors being on
# their reference levels (culture = usa, condition = condition_1)
s.age <-
  (qlogis(condition_1.usa.11) - qlogis(condition_1.usa.3)) /
    (max(start.data$z.age) - min(start.data$z.age))

# slope of condition is the effect of reason-giving at age being at its
# average (0) or the average of the slopes of reason-giving against
# condition at age being at its minimum and maximum, respectively
s.condition.condition_2 <-
  mean(c(
    (qlogis(condition_2.usa.11) - qlogis(condition_1.usa.11)),
    (qlogis(condition_2.usa.3) - qlogis(condition_1.usa.3))
  ))

# slope of culture.china is zero because we expect the reason-giving rate for
# the US (reference level) and china to be the same
s.culture.china <-
  mean(c(
    (qlogis(condition_1.china.11) - qlogis(condition_1.usa.11)),
    (qlogis(condition_1.china.3) - qlogis(condition_1.usa.3))
  ))

# slope of culture.kenya
s.culture.kenya <-
  mean(c(
    (qlogis(condition_1.kenya.11) - qlogis(condition_1.usa.11)),
    (qlogis(condition_1.kenya.3) - qlogis(condition_1.usa.3))
  ))

# slope of culture.netherlands
s.culture.netherlands <-
  mean(c(
    (qlogis(condition_1.netherlands.11) - qlogis(condition_1.usa.11)),
    (qlogis(condition_1.netherlands.3) - qlogis(condition_1.usa.3))
  ))

# two-way interactions
```

```r
# slope of age in the condition_2 condition with culture at the reference level
# the slope for condition.condition_2:culture.age determines the effect of
# the interaction, i.e.,
# how much the effect of condition on reason-giving changes when age increases
# by one (or how much much the effect of age on reason-giving changes when
# condition increases by one, which is the same).
# We need to calculate:
# ((slope of age with condition being at its maximum(condition_2)  -
# slope of age with condition being at its minimum(condition_1)) /
# (maximum of condition(condition_2) - min of condition(condition_1))
s.age.condition.condition_2 <-
  ((qlogis(condition_2.usa.11) - qlogis(condition_2.usa.3)) /
    (max(start.data$z.age) - min(start.data$z.age))) -
  # slope of age at condition being at its maximum (condition_2)
  ((qlogis(condition_1.usa.11) - qlogis(condition_1.usa.3)) /
    (max(start.data$z.age) - min(start.data$z.age)))
# slope of age with #condition being at its minimum (condition_1)

# since it also needs to work the other way around
s.condition.condition_2.age <-
  ((qlogis(condition_2.usa.11) - qlogis(condition_1.usa.11)) -
    (qlogis(condition_2.usa.3) - qlogis(condition_1.usa.3))) /
    (max(start.data$z.age) - min(start.data$z.age))
# test whether both versions lead to the same result
round(s.age.condition.condition_2, 5) == round(s.condition.condition_2.age, 5)
```

```
## [1] TRUE
```

```r
# ((slope of age with culture being at its maximum(china)  -
# slope of age with culture being at its minimum(usa)) /
# (maximum of culture(china) - min of culture(usa))
s.age.culture.china <-
  (((qlogis(condition_1.china.11) - qlogis(condition_1.china.3)) /
    (max(start.data$z.age) - min(start.data$z.age))) -
    ((qlogis(condition_1.usa.11) - qlogis(condition_1.usa.3)) /
      (max(start.data$z.age) - min(start.data$z.age))))
# since it also needs to work the other way around
s.culture.china.age <-
  ((qlogis(condition_1.china.11) - qlogis(condition_1.usa.11)) -
    (qlogis(condition_1.china.3) - qlogis(condition_1.usa.3))) /
    (max(start.data$z.age) - min(start.data$z.age))
# test whether both versions lead to the same result
round(s.age.culture.china, 5) == round(s.culture.china.age, 5)
```

```
## [1] TRUE
```

```r
# ((slope of age with culture being at its maximum(kenya)  -
# slope of age with culture being at its minimum(usa)) /
# (maximum of culture(kenya) - min of culture(usa))
s.age.culture.kenya <-
  ((qlogis(condition_1.kenya.11) - qlogis(condition_1.kenya.3)) /
    (max(start.data$z.age) - min(start.data$z.age))) -
```

```r
  ((qlogis(condition_1.usa.11) - qlogis(condition_1.usa.3)) /
    (max(start.data$z.age) - min(start.data$z.age)))
# since it also needs to work the other way around
s.culture.kenya.age <-
  ((qlogis(condition_1.kenya.11) - qlogis(condition_1.usa.11)) -
    (qlogis(condition_1.kenya.3) - qlogis(condition_1.usa.3))) /
    (max(start.data$z.age) - min(start.data$z.age))
# test whether both versions lead to the same result
round(s.culture.kenya.age, 5) == round(s.age.culture.kenya, 5)
```

```
## [1] TRUE
```

```r
# ((slope of age with culture being at its maximum(kenya)  -
# slope of age with culture being at its minimum(usa)) /
# (maximum of culture(kenya) - min of culture(usa))
s.age.culture.netherlands <-
  ((qlogis(condition_1.netherlands.11) - qlogis(condition_1.netherlands.3)) /
    (max(start.data$z.age) - min(start.data$z.age))) -
  ((qlogis(condition_1.netherlands.11) - qlogis(condition_1.netherlands.3)) /
    (max(start.data$z.age) - min(start.data$z.age)))
# since it also needs to work the other way around
s.culture.netherlands.age <-
  ((qlogis(condition_1.netherlands.11) - qlogis(condition_1.netherlands.11)) -
    (qlogis(condition_1.netherlands.3) - qlogis(condition_1.netherlands.3))) /
    (max(start.data$z.age) - min(start.data$z.age))
# test whether both versions lead to the same result
round(s.culture.netherlands.age, 5) == round(s.age.culture.netherlands, 5)
```

```
## [1] TRUE
```

```r
# ((slope of condition with culture being at its maximum(china)  -
# slope of condition with culture being at its minimum(usa)) /
# (maximum of culture(china) - min of culture(usa))
s.culture.china.condition.condition_2 <-
  mean(c(
    ((qlogis(condition_2.china.3) - qlogis(condition_1.china.3)) -
      (qlogis(condition_2.usa.3) - qlogis(condition_1.usa.3))),
    ((qlogis(condition_2.china.11) - qlogis(condition_1.china.11)) -
      (qlogis(condition_2.usa.11) - qlogis(condition_1.usa.11)))
  ))
# since it also needs to work the other way around
s.condition.condition_2.culture.china <-
  mean(c(
    ((qlogis(condition_2.china.3) - qlogis(condition_2.usa.3)) -
      (qlogis(condition_1.china.3) - qlogis(condition_1.usa.3))),
    ((qlogis(condition_2.china.11) - qlogis(condition_2.usa.11)) -
      (qlogis(condition_1.china.11) - qlogis(condition_1.usa.11)))
  ))
# test whether both versions lead to the same result
round(s.culture.china.condition.condition_2, 5) ==
  round(s.condition.condition_2.culture.china, 5)
```

```
## [1] TRUE
```

```r
# ((slope of condition with culture being at its maximum(kenya)  -
# slope of condition with culture being at its minimum(usa)) /
# (maximum of culture(kenya) - min of culture(usa))
s.culture.kenya.condition.condition_2 <-
  mean(c(
    ((qlogis(condition_2.kenya.3) - qlogis(condition_1.kenya.3)) -
      (qlogis(condition_2.usa.3) - qlogis(condition_1.usa.3))),
    ((qlogis(condition_2.kenya.11) - qlogis(condition_1.kenya.11)) -
      (qlogis(condition_2.usa.11) - qlogis(condition_1.usa.11)))
  ))
# since it also needs to work the other way around
s.condition.condition_2.culture.kenya <-
  mean(c(
    ((qlogis(condition_2.kenya.3) - qlogis(condition_2.usa.3)) -
      (qlogis(condition_1.kenya.3) - qlogis(condition_1.usa.3))),
    ((qlogis(condition_2.kenya.11) - qlogis(condition_2.usa.11)) -
      (qlogis(condition_1.kenya.11) - qlogis(condition_1.usa.11)))
  ))
# test whether both versions lead to the same result
round(s.culture.kenya.condition.condition_2, 5) ==
  round(s.condition.condition_2.culture.kenya, 5)
```

```
## [1] TRUE
```

```r
# ((slope of condition with culture being at its maximum(netherlands)  -
# slope of condition with culture being at its minimum(usa)) /
# (maximum of culture(kenya) - min of culture(usa))
s.culture.netherlands.condition.condition_2 <-
  mean(c(
    ((qlogis(condition_2.netherlands.3) - qlogis(condition_1.netherlands.3)) -
      (qlogis(condition_2.usa.3) - qlogis(condition_1.usa.3))),
    ((qlogis(condition_2.netherlands.11) - qlogis(condition_1.netherlands.11)) -
      (qlogis(condition_2.usa.11) - qlogis(condition_1.usa.11)))
  ))
# since it also needs to work the other way around
s.condition.condition_2.culture.netherlands <-
  mean(c(
    ((qlogis(condition_2.netherlands.3) - qlogis(condition_2.usa.3)) -
      (qlogis(condition_1.netherlands.3) - qlogis(condition_1.usa.3))),
    ((qlogis(condition_2.netherlands.11) - qlogis(condition_2.usa.11)) -
      (qlogis(condition_1.netherlands.11) - qlogis(condition_1.usa.11)))
  ))
# test whether both versions lead to the same result
round(s.culture.netherlands.condition.condition_2, 5) ==
  round(s.condition.condition_2.culture.netherlands, 5)
```

```
## [1] TRUE
```

```r
# three way-interactions
s.condition.condition_2.culture.china.age <-
```

```
  # slope for condition at max age for china
  ((((qlogis(condition_2.china.11) - qlogis(condition_1.china.11)) -
    # slope for condition at min age for china
    (qlogis(condition_2.china.3) - qlogis(condition_1.china.3))) /
    (max(start.data$z.age) - min(start.data$z.age))) -
    # slope for condition at max age for us
    (((qlogis(condition_2.usa.11) - qlogis(condition_1.usa.11)) -
      # slope for condition at min age for us
      (qlogis(condition_2.usa.3) - qlogis(condition_1.usa.3))) /
      (max(start.data$z.age) - min(start.data$z.age))))

s.condition.condition_2.culture.kenya.age <-
  # slope for condition at max age for kenya
  ((((qlogis(condition_2.kenya.11) - qlogis(condition_1.kenya.11)) -
    # slope for condition at min age for kenya
    (qlogis(condition_2.kenya.3) - qlogis(condition_1.kenya.3))) /
    (max(start.data$z.age) - min(start.data$z.age))) -
    # slope for condition at max age for us
    (((qlogis(condition_2.usa.11) - qlogis(condition_1.usa.11)) -
      # slope for condition at min age for us
      (qlogis(condition_2.usa.3) - qlogis(condition_1.usa.3))) /
      (max(start.data$z.age) - min(start.data$z.age))))

s.condition.condition_2.culture.netherlands.age <-
  # slope for condition at max age for netherlands
  ((((qlogis(condition_2.netherlands.11) - qlogis(condition_1.netherlands.11)) -
    # slope for condition at min age for netherlands
    (qlogis(condition_2.netherlands.3) - qlogis(condition_1.netherlands.3))) /
    (max(start.data$z.age) - min(start.data$z.age))) -
    # slope for condition at max age for us
    (((qlogis(condition_2.usa.11) - qlogis(condition_1.usa.11)) -
      # slope for condition at min age for us
      (qlogis(condition_2.usa.3) - qlogis(condition_1.usa.3))) /
      (max(start.data$z.age) - min(start.data$z.age))))
```

Define number of simulations

```
n.simus <- 500 # small number for testing
```

Prepare simulation

```
# create object to store the simulation parameters and results:
all.res <-
  data.frame(expand.grid(
    n.simus = 1:n.simus
  ))

# add columns for estimates
all.res$icpt <- NA
```

```
all.res$z.age <- NA
all.res$condition.condition_2 <- NA
all.res$culture.china <- NA
all.res$culture.kenya <- NA
all.res$culture.netherlands <- NA
all.res$z.trial <- NA
all.res$z.age.condition.condition_2 <- NA
all.res$z.age.culture.china <- NA
all.res$z.age.culture.kenya <- NA
all.res$z.age.culture.netherlands <- NA
all.res$condition.condition_2.culture.china <- NA
all.res$condition.condition_2.culture.kenya <- NA
all.res$condition.condition_2.culture.netherlands <- NA
all.res$z.age.condition.condition_2.culture.china <- NA
all.res$z.age.condition.condition_2.culture.kenya <- NA
all.res$z.age.condition.condition_2.culture.netherlands <- NA
# add columns for re.sd and warnings for full model and null model
all.res$re.sd <- NA
all.res$warns.full <- NA
all.res$warns.null <- NA
# add columns for likelihood ratio test results (p-values)
all.res$full.null.p <- NA
all.res$lrt.p.condition <- NA
all.res$lrt.p.z.age <- NA
all.res$lrt.p.culture <- NA
all.res$lrt.p.z.trial <- NA
all.res$lrt.p.z.age.condition <- NA
all.res$lrt.p.condition.culture <- NA
all.res$lrt.p.z.age.culture <- NA
all.res$lrt.p.z.age.condition.culture <- NA

# create vector with coefficients
coefs <- c(
  "(Intercept)" = intercept,
  "z.age" = s.age,
  "condition.condition_2" = s.condition.condition_2,
  "culture.china" = s.culture.china,
  "culture.kenya" = s.culture.kenya,
  "culture.netherlands" = s.culture.netherlands,
  "z.age:condition.condition_2" = s.condition.condition_2.age,
  "z.age:culture.china" = s.culture.china.age,
  "z.age:culture.kenya" = s.culture.kenya.age,
  "z.age:culture.netherlands" = s.culture.netherlands.age,
  "condition.condition_2:culture.china" =
    s.condition.condition_2.culture.china,
  "condition.condition_2:culture.kenya" =
    s.condition.condition_2.culture.kenya,
  "condition.condition_2:culture.netherlands" =
    s.condition.condition_2.culture.netherlands,
  "z.age:condition.condition_2:culture.china" =
    s.condition.condition_2.culture.china.age,
  "z.age:condition.condition_2:culture.kenya" =
    s.condition.condition_2.culture.kenya.age,
```

```r
  "z.age:condition.condition_2:culture.netherlands" =
    s.condition.condition_2.culture.netherlands.age
)
```

#start simulation

```r
xdata <- start.data # change start.data to xdata

# create model matrix
m.mat <- model.matrix(object = ~ z.age * condition * culture, data = xdata)
# create LP wrt fixed effects
LP <- m.mat[, names(coefs)] %*% coefs

# define control structure to make convergence more likely:
# contr <- glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 10000))

# run simulation
for (i in 1:nrow(all.res)) {
  set.seed(i)
  # generate response:
  xdata$reasons <- rbinom(
    n = nrow(xdata), size = 1,
    prob = exp(LP) / (1 + exp(LP))
  )

  # fit full model:
  full <- keepWarnings(glm(reasons ~
    (z.age + condition + culture)^3,
  data = xdata, family = binomial
  ))

  # fit null model:
  null <- keepWarnings(glm(reasons ~ 1,
    data = xdata, family = binomial
  ))
  # fit reduced model with two-way interactions:
  red1 <- keepWarnings(glm(reasons ~
    (z.age + condition + culture)^2,
  data = xdata, family = binomial
  ))
  # fit reduced model with only main effects:
  red2 <- keepWarnings(glm(reasons ~
    (z.age + condition + culture),
  data = xdata, family = binomial
  ))

  # store results:
  all.res[i, c(
    "icpt", "z.age", "condition.condition_2", "culture.china",
    "culture.kenya", "culture.netherlands",
    "z.age.condition.condition_2", "z.age.culture.china",
    "z.age.culture.kenya", "z.age.culture.netherlands",
    "condition.condition_2.culture.china",
    "condition.condition_2.culture.kenya",
```

```
    "condition.condition_2.culture.netherlands",
    "z.age.condition.condition_2.culture.china",
    "z.age.condition.condition_2.culture.kenya",
    "z.age.condition.condition_2.culture.netherlands"
  )] <-
    summary(full$value)$coefficients[, "Estimate"]

  all.res[i, "warns.full"] <- nchar(paste(full$warnings, collapse = ""))
  all.res[i, "warns.null"] <- nchar(paste(null$warnings, collapse = ""))
  all.res[i, "full.null.p"] <-
    as.data.frame(anova(null$value, full$value, test = "Chisq"))[2, "Pr(>Chi)"]
  all.res[i, "lrt.p.z.age.condition.culture"] <-
    as.data.frame(drop1(full$value,
      test = "Chisq"
    ))["z.age:condition:culture", "Pr(>Chi)"]

  xx <- drop1(red1$value, test = "Chisq")
  all.res[i, "lrt.p.z.age.culture"] <-
    as.data.frame(xx)["z.age:culture", "Pr(>Chi)"]
  all.res[i, "lrt.p.condition.culture"] <-
    as.data.frame(xx)["condition:culture", "Pr(>Chi)"]
  all.res[i, "lrt.p.z.age.condition"] <-
    as.data.frame(xx)["z.age:condition", "Pr(>Chi)"]

  xx <- drop1(red2$value, test = "Chisq")
  all.res[i, "lrt.p.condition"] <- as.data.frame(xx)["condition", "Pr(>Chi)"]
  all.res[i, "lrt.p.culture"] <- as.data.frame(xx)["culture", "Pr(>Chi)"]
  all.res[i, "lrt.p.z.age"] <- as.data.frame(xx)["z.age", "Pr(>Chi)"]
  all.res[i, "lrt.p.z.trial"] <- as.data.frame(xx)["z.trial", "Pr(>Chi)"]
}

save.image("ReasonGiving_Studies_Simulation_glm.RData")
```

Evaluation of results

```
# number of full models that converged without warnings
sum(all.res[, "warns.full"] == 0)
```

```
## [1] 499
```

```
# number of null models that converged without warnings
sum(all.res[, "warns.null"] == 0)
```

```
## [1] 500
```

```
#from here on we continue with only that models without warnings
all.res1 <- subset(all.res, warns.full == 0)
```

Power for all effects

```
# Power for condition effect
sum(all.res1$lrt.p.condition < 0.05) / n.simus
```

```
## [1] 0.998
```

```
# Power for interaction effect condition * age
sum(all.res1$lrt.p.z.age.condition < 0.05) / n.simus
```

```
## [1] 0.158
```

```
# Power for interaction effect condition * culture
sum(all.res1$lrt.p.condition.culture < 0.05) / n.simus
```

```
## [1] 0.064
```

```
# Power for interaction effect age * condition * culture
sum(all.res1$lrt.p.z.age.condition.culture < 0.05) / n.simus
```

```
## [1] 0.122
```