Appendix-C.I: Sample-Size Power Simulation for Hypothesized Adaptive Flexibility Construct

Adam Garber

Updated: October 02, 2024

```
library(tidyverse)
library(MplusAutomation)
library(rhdf5)
library(here)
library(glue)
library(gt)
library(janitor)
library(reshape2)
```

Power Simulation: Hypothesized Adaptive Flexibility Construct

- Estimate LCA model (K=5, U=17)
- Sample size conditions (9): N=500,1000,1500,2000,2500,3000,4000,5000,6000

```
# looping over conditions 'N'
N = c(500,1000,1500,2000,2500,3000,4000,5000,6000)

lca_sim_func  <- function(N) {
  lca_sim  <- mplusObject(

  TITLE = glue("LCA Power Simulation: Condition N = {N}"),

    MONTECARLO = glue(
     "NAMES = primary change interupt initiat engage approach response expect
              new same relative objects sequence trans avoid control touch;
        GENERATE = primary change interupt initiat engage approach response expect
              new same relative objects sequence trans avoid control touch(1);
        CATEGORICAL = primary change interupt initiat engage approach response expect
              new same relative objects sequence trans avoid control touch;
        GENCLASSES = c(5);
        CLASSES = c(5);
        NOBSERVATIONS = {N};
        SEED = 08202024; !!! USE SEED TO REPLICATE STUDY RESULTS !!!
        NREPS = 1000;
      repsave = all;
      save = n_{N}_lca_rep*.dat;
```

```
      RESULTS = lcaresults{N}.csv;"),

  ANALYSIS =
      "TYPE = MIXTURE;
       !!!STARTS = ; !!! No Starts !!!
         processors = 8;",

  MODELPOPULATION =

    "%OVERALL%

       [c#1*-0.466]; !!! c1 = 6.1%  !!!
       [c#2* 1.279]; !!! c2 = 35.4% !!!
       [c#3* 1.041]; !!! c3 = 27.9% !!!
       [c#4* 0.739]; !!! c4 = 20.6% !!!
                     !!! c5 = 9.8%  !!!

!!! Class 1: High Flexibility !!!

%c#1%
[primary$1*-2.325907 change$1*-1.815290 interupt$1*-1.945910 initiat$1*-2.090741
 engage$1*-1.840434 approach$1*-2.231012 response$1*-2.556366 expect$1*-2.231012
 new$1*-2.987364 same$1*- 2.428837 relative$1*-2.254058 objects$1*-3.555
 sequence$1*-2.132267 trans$1*-3.152319 avoid$1*-2.699549 control$1*-2.050519
 touch$1*-1.774368];

!!! Class 2: Social and Interest Flexibility !!!
%c#2%
[primary$1*-1.0773913 change$1*-1.1039528 interupt$1*-1.5299568 initiat$1*-1.9550845
 engage$1*-0.9643905 approach$1*-1.3370233 response$1*-1.5299568 expect$1*-1.9187592
 new$1*1.1914483 same$1*1.2891306 relative$1*0.8905323 objects$1*1.9367985
 sequence$1*2.4838238 trans$1*1.9643226 avoid$1*1.2540491 control$1*3.555
 touch$1*2.1972246];

!!! Class 3: Environmetal Flexibility !!!

%c#3%
[primary$1*3.7489924 change$1*1.2139664 interupt$1*2.4423470 initiat$1*1.1200608
 engage$1*4.4987991 approach$1*1.4500102 response$1*2.0211513 expect$1*2.7339416
 new$1*-2.3891996 same$1*-1.0355993 relative$1*-1.6434218 objects$1*-0.9197934
 sequence$1*-0.9247059 trans$1*-0.8616248 avoid$1*-1.2196389 control$1*-1.9736255
 touch$1*-0.8760355];

!!! Class 4: Location and Sensory Flexibilty !!!

%c#4%
[primary$1*-0.2736958 change$1*-0.4853059 interupt$1*-0.7217852 initiat$1*-0.5451086
 engage$1*-0.4641576 approach$1*-0.4305291 response$1*-0.5666941 expect$1*-0.7861311
 new$1*1.1914483 same$1*2.6021528 relative$1*1.8070090 objects$1*-2.1751973
 sequence$1*-1.5645131 trans$1*-1.3925561 avoid$1*1.0511726 control$1*1.7905933
 touch$1*1.2891306];

!!! Class 5: Low Flexibility !!!
```

```
%c#5%
[primary$1*3.287572 change$1*1.909830 interupt$1*2.186163 initiat$1*3.623315
 engage$1*2.843852 approach$1*3.009467 response$1*3.555 expect$1*2.338303
 new$1*2.153550 same$1*2.571429 relative$1*2.242481 objects$1*3.511031
 sequence$1*2.363483 trans$1*1.614246 avoid$1*1.665687 control$1*2.338303
 touch$1*3.409496];",

  MODEL =

  "%OVERALL%

       [c#1*-0.466]; !!! c1 = 6.1%  !!!
       [c#2* 1.279]; !!! c2 = 35.4% !!!
       [c#3* 1.041]; !!! c3 = 27.9% !!!
       [c#4* 0.739]; !!! c4 = 20.6% !!!
                     !!! c5 = 9.8%  !!!


!!! Class 1: High Flexibility !!!

%c#1%
[primary$1*-2.325907 change$1*-1.815290 interupt$1*-1.945910 initiat$1*-2.090741
 engage$1*-1.840434 approach$1*-2.231012 response$1*-2.556366 expect$1*-2.231012
 new$1*-2.987364 same$1*- 2.428837 relative$1*-2.254058 objects$1*-3.555
 sequence$1*-2.132267 trans$1*-3.152319 avoid$1*-2.699549 control$1*-2.050519
 touch$1*-1.774368];

!!! Class 2: Social and Interest Flexibility !!!
%c#2%
[primary$1*-1.0773913 change$1*-1.1039528 interupt$1*-1.5299568 initiat$1*-1.9550845
 engage$1*-0.9643905 approach$1*-1.3370233 response$1*-1.5299568 expect$1*-1.9187592
 new$1*1.1914483 same$1*1.2891306 relative$1*0.8905323 objects$1*1.9367985
 sequence$1*2.4838238 trans$1*1.9643226 avoid$1*1.2540491 control$1*3.555
 touch$1*2.1972246];

!!! Class 3: Environmetal Flexibility !!!

%c#3%
[primary$1*3.7489924 change$1*1.2139664 interupt$1*2.4423470 initiat$1*1.1200608
 engage$1*4.4987991 approach$1*1.4500102 response$1*2.0211513 expect$1*2.7339416
 new$1*-2.3891996 same$1*-1.0355993 relative$1*-1.6434218 objects$1*-0.9197934
 sequence$1*-0.9247059 trans$1*-0.8616248 avoid$1*-1.2196389 control$1*-1.9736255
 touch$1*-0.8760355];

!!! Class 4: Location and Sensory Flexibilty !!!

%c#4%
[primary$1*-0.2736958 change$1*-0.4853059 interupt$1*-0.7217852 initiat$1*-0.5451086
 engage$1*-0.4641576 approach$1*-0.4305291 response$1*-0.5666941 expect$1*-0.7861311
 new$1*1.1914483 same$1*2.6021528 relative$1*1.8070090 objects$1*-2.1751973
 sequence$1*-1.5645131 trans$1*-1.3925561 avoid$1*1.0511726 control$1*1.7905933
 touch$1*1.2891306];
```

```
!!! Class 5: Low Flexibility !!!

%c#5%
[primary$1*3.287572 change$1*1.909830 interupt$1*2.186163 initiat$1*3.623315
 engage$1*2.843852 approach$1*3.009467 response$1*3.555 expect$1*2.338303
 new$1*2.153550 same$1*2.571429 relative$1*2.242481 objects$1*3.511031
 sequence$1*2.363483 trans$1*1.614246 avoid$1*1.665687 control$1*2.338303
 touch$1*3.409496];"
)

lca_sim.fit <- mplusModeler(lca_sim,
                dataout=glue(here("C1-Simulation", "{N}_lca_nsize_sim.dat")),
                modelout=glue(here("C1-Simulation", "{N}_lca_nsize_sim.inp")),
                check=TRUE, run = TRUE, hashfilename = FALSE)

return(lca_sim.fit)
}

result_list <- lapply(1:N, function(i) {
  lca_sim_func(N[i])
               })
```

Read simulation summary output files for each sample size condition

NOTE: The 9 output files were copied to folder `C1_Outputs` for accessibility

```
lca_sim <- readModels(here("C1_Outputs"), what = "parameters")

lcaNames <- names(lca_sim)
unstandardizedlca <- sapply(sapply(lca_sim, "[", "parameters"), "[", "unstandardized")
names(unstandardizedlca) <- lcaNames
```

Combine simulation output results

```
apply_table_setup <- function(data) {
    table_setup <- data %>%
        rename(Power = pct_sig_coef, Parameter = param, Coverage = cover_95) %>%
        select(Parameter:LatentClass)

    return(table_setup)
}

resultslca <- lapply(unstandardizedlca, apply_table_setup)

lca_combined <- do.call("rbind", resultslca) %>%
    as.data.frame() %>%
    mutate(Parameter = factor(Parameter)) %>%
    purrr::modify_if(is.character, as.numeric) %>%
    mutate(Condition = as.array(rep(1:9, each = 89))) %>%
    mutate(Condition = factor(Condition, labels = c(`1` = "N=500", `2` = "N=1000",
        `3` = "N=1500", `4` = "N=2000", `5` = "N=2500", `6` = "N=3000", `7` = "N=4000",
        `8` = "N=5000", `9` = "N=6000")))
```

Calculate results: Group by condition (NOTE: Values are averaged across model parameters)

```
fit_by_cond <- lca_combined %>%
    group_by(Condition) %>%
    summarize(avg_Bias = mean(abs(average - population)/abs(population)) * 100, max_Bias = max(abs(avera
        population)/abs(population)) * 100, avg_SE_Bias = mean(abs(average_se - population_sd)/populati
        100, max_SE_Bias = max(abs(average_se - population_sd)/population_sd) * 100,
        avg_Coverage = mean(Coverage), avg_Power = mean(Power)) %>%
    adorn_rounding(digits = 2)
```

Create table

```
sim_table <- fit_by_cond %>%
    gt()

sim_table
```

Condition	avg_Bias	max_Bias	avg_SE_Bias	max_SE_Bias	avg_Coverage	avg_Power
N=500	18.80	127.23	32.42	89.13	0.91	0.96
N=1000	5.40	58.39	18.47	80.97	0.95	0.98
N=1500	2.59	29.92	12.10	76.34	0.95	0.99
N=2000	1.46	12.88	7.98	70.83	0.95	1.00
N=2500	1.09	8.41	6.31	64.01	0.95	1.00
N=3000	0.83	4.54	5.34	53.19	0.95	1.00
N=4000	0.63	2.46	3.23	36.86	0.95	1.00
N=5000	0.50	1.70	1.75	5.59	0.95	1.00
N=6000	0.42	1.45	1.94	6.30	0.95	1.00

Save table

```
gtsave(sim_table, here("figures", "C1_sim_results.png"))
```
