Monte Carlo Simulation - Power Analysis

Adam Garber

Norwegian University of Science and Technology - A Course in `MplusAutomation`

June 01, 2021

A replication of analyses presented in Muthén & Muthén (2002)

This tutorial closely follows the concepts and model syntax presented here:

Muthén, L. K., & Muthén, B. O. (2002). How to use a Monte Carlo study to decide on sample size and determine power. Structural equation modeling, 9(4), 599-620.

The associated Mplus syntax can be found here: http://www.statmodel.com/examples/penn.shtml

---

This project contains the following 4 sub-folders:

a. "figures"
b. "mplus_files"
c. "mplus_bias"
d. "mplus_tune"

Loading packages

```
library(tidyverse)
library(MplusAutomation)
library(here)
library(glue)
library(psych)
library(gt)
```

Practicing Monte Carlo Simulation

Factors that influence minimum sample size requirements:

- size of the model (number of parameters)
- distribution of the variables (skew, kurtosis, multi-modal..)
- amount of missing data & pattern of missingness
- reliability of the variables
- strength of the relations among the variables

Simulation purpose: "This article focuses on parameter estimates, standard errors, coverage, and power assuming correctly specified models. Misspecified models can also be studied in the Mplus Monte Carlo framework but are not included here." - Muthen & Muthen (2002)

CFA Model example:

- 2 factors
- 10 indicators (5 per factor)
- 31 free parameters & 24 *df*
- factor loadings = 0.8 (freely estimated in the model)
- residual variances = 0.36 (error)
- factor variances = 1 (fixed)
- reliability of factor = 0.64 = .8ˆ2(1) /.8ˆ2(1 + 0.36)

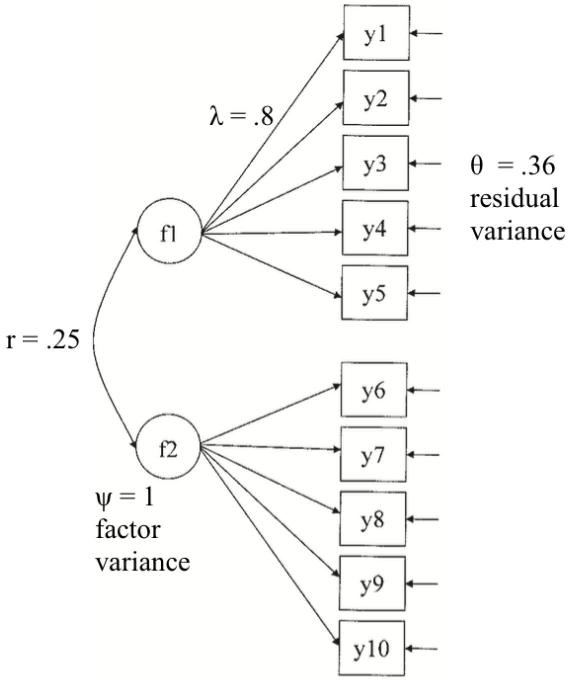


Figure 1. CFA model example. Picture adapted from, Muthen & Muthen 2002

Monte Carlo conditions:

1. normally distributed continuous factor indicators without missing data
2. normally distributed continuous factor indicators with missing data
3. non-normal continuous factor indicators without missing data
4. non-normal continuous factor indicators with missing data

Missing data:

"... all participants have data on y1, y2, y3, y4, and y5, and 50% of the participants have data on y6, y7, y8, y9, and y10" (Muthen & Muthen, 2002).

Non-normal data (how to create variables with skew & kurtosis):

Muthen & Muthen (2002):

"non-normal data are created using a mixture of two normal sub-populations or classes of individuals. Normal data are generated for two classes that have different means and variances for the factor indicators. The combined data are analyzed as though they come from a single population."

[...]

"The first step is to generate data for two classes such that the combination of the data from the two classes **has the desired skewness and kurtosis**."

[...]

"For the CFA model with non-normal data, Class 1, the outlier class, contains 12% of the participants and Class 2 contains the remaining 88%. Only the factor indicators for the second factor are non-normal. Therefore, the Class 1 mean for the second factor is chosen to be 15 and the variance 5, as compared to the Class 2 mean and variance of zero and 1. The resulting population univariate skewness for variables y6 through y10 is 1.2. The resulting population univariate kurtosis for variables y6 through y10 ranges from 1.5 to 1.6."

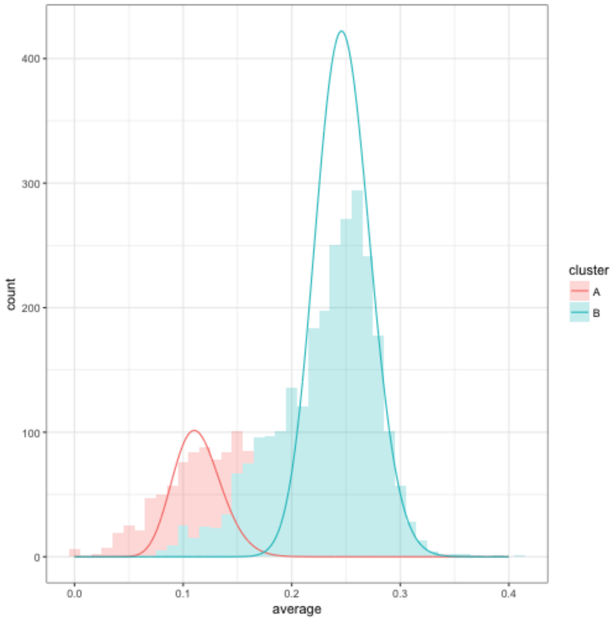


Figure 2. Two class model. What are the data generating processes that might result in skew?

Simulation 0 - tuning skew exericse

- "The second step is to run the analysis with one replication and a large sample to obtain approximate population values for the one class model (e.g., achieve factor indicator reliabilities of 0.64)." (Muthen & Muthen, 2002).
- We will just do 1 replication with an N-size of 100,000 (exploit law of large numbers!)

```
cfa_tune  <- lapply(1:5, function(k) {
  cfa_0  <- mplusObject(

  TITLE = "CFA 1 - non-normal, no missing",

  MONTECARLO =
    sprintf("NAMES ARE y1-y10;
     NOBSERVATIONS = 100000;
     NREPS = 1;
     SEED = 53487;
     CLASSES = C(1);
     GENCLASSES = C(2);
     SAVE = cfa0_%d.sav;",k),

  ANALYSIS =
    "TYPE = MIXTURE;
     ESTIMATOR = MLR;",

  MODELPOPULATION =
    glue("%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*.8;
     f1@1 f2@1;
     y1-y5*.36 y6-y10*9;
     f1 WITH f2*.95;
     [C#1@-{k*.5}];   ! parameter we will tune to adjust the size of the outliers

     %C#1%            ! outlier class

     [f1@0 f2@15];    ! means (factor 2 set to 15 to tune skewness & kurtosis)
     f1@1 f2@5;       ! variances (factor 2 set to 5  to tune skewness & kurtosis)

     %C#2%            ! majority class

     [f1@0 f2@0];
     f1@1 f2@1;"),

  MODEL =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*4;
     f1@1 f2@1;
     y1-y5*.36 y6-y10*9;
     f1 WITH f2*.20;

   [y6-y10*1.42];" ,

  OUTPUT = " SAMPSTAT TECH9;")

cfa_0_fit <- mplusModeler(cfa_0,
             dataout=here("24-simulation", "mplus_tune", "cfa0_demo_sim.dat"),
             modelout=sprintf(here("24-simulation", "mplus_tune", "%d_cfa0_demo_sim.inp"), k),
             check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

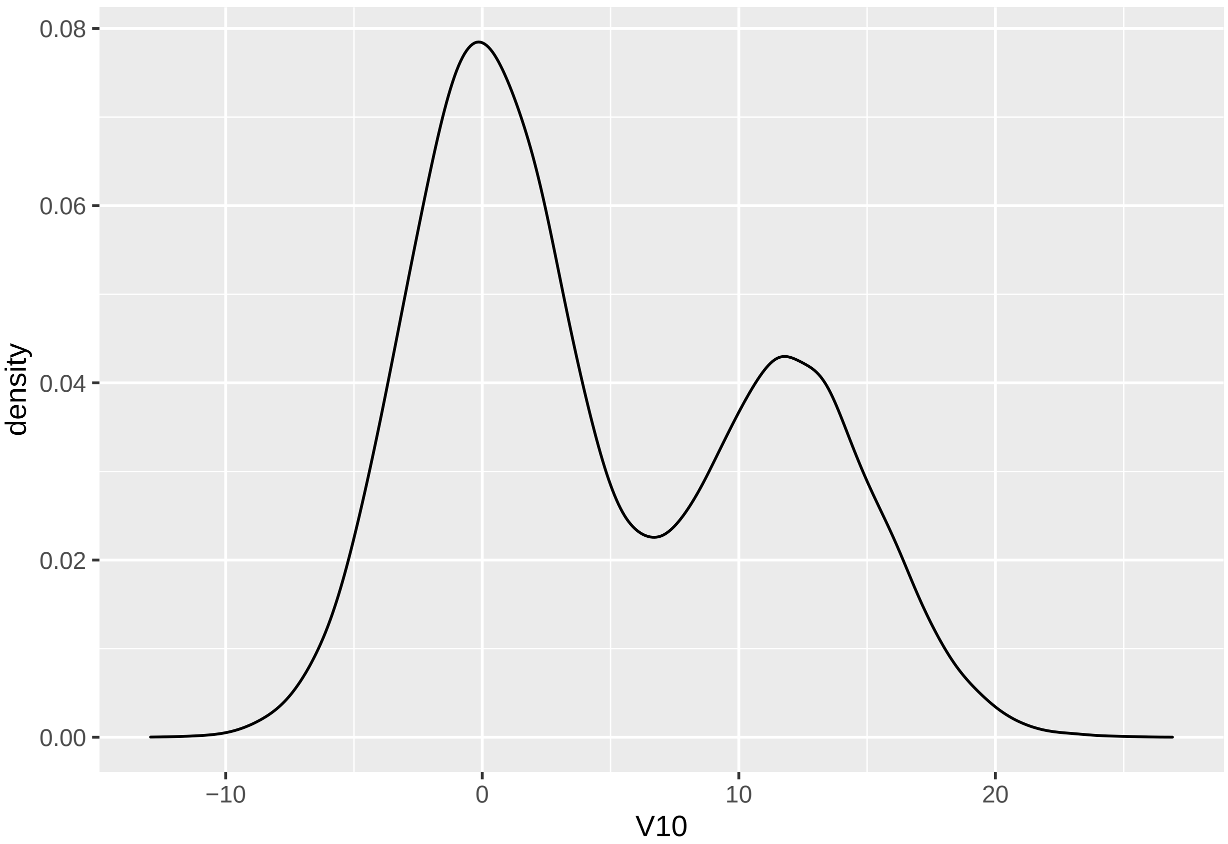
Read in simulated data, tabulate, plot distribution shape

```
data0_1 <- read.delim(here("24-simulation", "mplus_tune", "cfa0_1.sav"), sep = "",
                   header = FALSE,
                   na.strings = "999.000000")

data0_2 <- read.delim(here("24-simulation", "mplus_tune", "cfa0_5.sav"), sep = "",
                   header = FALSE,
                   na.strings = "999.000000")

describe(data0_1) %>%
  gt() %>%
  fmt_number(c(3:13), decimals = 3)
```

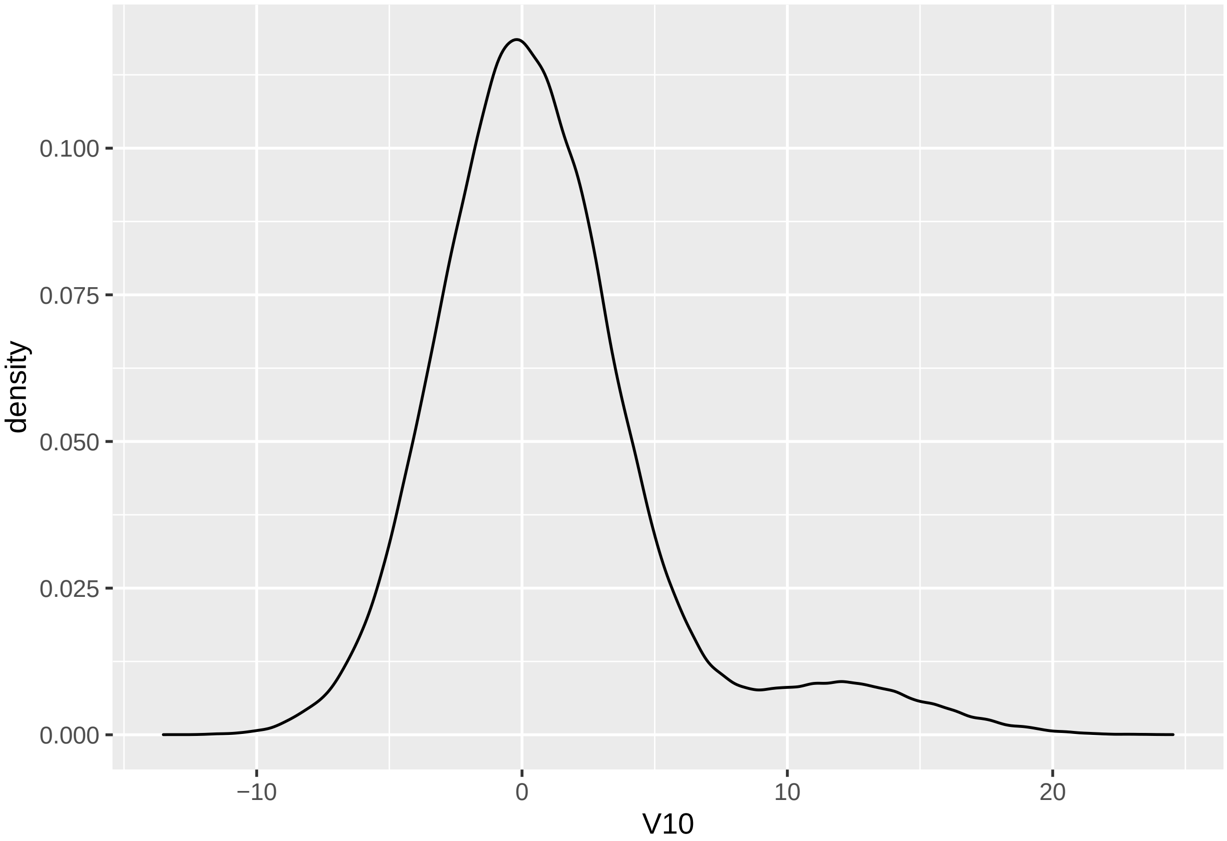
vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
1	1e+05	0.001	1.001	0.002	0.001	1.001	−4.462	4.490	8.952	−0.004	−0.010	0.003
2	1e+05	0.003	1.001	0.004	0.003	1.002	−4.018	4.568	8.586	0.001	−0.007	0.003
3	1e+05	0.003	1.001	−0.000	0.002	1.004	−4.067	4.292	8.360	0.015	−0.033	0.003
4	1e+05	−0.001	1.002	−0.005	−0.003	1.006	−4.556	4.314	8.869	0.019	−0.002	0.003

5	1e+05	0.003	0.999	−0.000	0.003	1.000	−4.656	4.829	9.485	0.005	−0.009	0.003
6	1e+05	4.540	6.652	2.656	4.219	7.048	−13.657	27.605	41.262	0.412	−0.934	0.021
7	1e+05	4.546	6.675	2.661	4.234	7.146	−12.919	26.903	39.822	0.401	−0.946	0.021
8	1e+05	4.535	6.667	2.636	4.214	7.074	−12.434	27.136	39.570	0.410	−0.946	0.021
9	1e+05	4.569	6.683	2.648	4.251	7.122	−13.387	28.178	41.565	0.408	−0.954	0.021
10	1e+05	4.543	6.670	2.647	4.225	7.085	−12.929	26.902	39.831	0.409	−0.938	0.021
11	1e+05	1.621	0.485	2.000	1.652	0.000	1.000	2.000	1.000	−0.500	−1.750	0.002

```
data0_1 %>%  ggplot(aes(x=V10)) +
  geom_density()
```



```
data0_2 %>%  ggplot(aes(x=V10)) +
  geom_density()
```



Criteria to assess level of bias

- "The first criterion is that parameter and standard error biases do not exceed 10% for any parameter in the model."
- "The second criterion is that the standard error bias for the parameter for which power is being assessed does not exceed 5%"
- "The third criterion is that coverage remains between 0.91 and 0.98."
- "Once these three conditions are satisfied, the sample size is chosen to keep power close to 0.80."

Simulation 1

- Normally distributed
- No missing
- Sample size (n) = 150
- Number of replecations = 10,000

```
cfa_1  <- mplusObject(

  TITLE = "CFA 1 - normal, no missing",
  MONTECARLO =
    "NAMES ARE y1-y10;
     NOBSERVATIONS = 150;
     NREPS = 10000;
     SEED = 53487;
     CLASSES = C(1);
     GENCLASSES = C(1);
     SAVE = cfa1.sav;",

  ANALYSIS =
    "TYPE = MIXTURE;
     ESTIMATOR = ML; ! when normal MLR simplifies to ML",

  MODELPOPULATION =
    "%OVERALL%           !
     f1 BY y1-y5*.8;     !  factor loadings = .8 (average?)
     f2 BY y6-y10*.8;    !
     f1@1 f2@1;          !  factor variances = 1 (fixed)
     y1-y10*.36;         !  residual variances = .36
     f1 WITH f2*.25;     !  factor correlation = .25
    ",

  MODEL =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*.8;
     f1@1 f2@1;
     y1-y10*.36;
     f1 WITH f2*.25;" ,

  OUTPUT = "TECH9;")

cfa_1_fit <- mplusModeler(cfa_1,
             dataout=here("24-simulation", "mplus_files", "cfa1_demo_sim.dat"),
             modelout=here("24-simulation", "mplus_files", "cfa1_demo_sim.inp"),
             check=TRUE, run = TRUE, hashfilename = FALSE)
```

Example of some results from simulation 1 (Y1):

- Parameter bias (columns 1 & 2): $(.8 - .7979)/.8 = .0026$
- Standard error bias (columns 2 & 3): $(.0706 - .0699)/.0706 = 0.0099$
- Coverage (column 6): .947
- Power(column 7): 1.0

Simulation 2

- Normally distributed
- **missing set to 50% for y6 - y10**

```
cfa_2  <- mplusObject(

  TITLE = "CFA 2 - normal, missing (50%)",

  MONTECARLO =
    "NAMES ARE y1-y10;
     NOBSERVATIONS = 175;
     NREPS = 10000;
     SEED = 53487;
     CLASSES = C(1);
     GENCLASSES = C(1);
     PATMISS = y6 (.5) y7 (.5) y8 (.5) y9 (.5) y10 (.5);
     PATPROB = 1;
     SAVE = cfa2.sav;",

  ANALYSIS =
    "TYPE = MIXTURE MISSING;
     ESTIMATOR = ML; ! when normal MLR simplifies to ML",

  MODELPOPULATION =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*.8;
     f1@1 f2@1;
     y1-y10*.36;
     f1 WITH f2*.25;",

  MODEL =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*.8;
     f1@1 f2@1;
     y1-y10*.36;
     f1 WITH f2*.25;" ,

  OUTPUT = "PATTERNS TECH9;")

cfa_2_fit <- mplusModeler(cfa_2,
             dataout=here("24-simulation", "mplus_files", "cfa2_demo_sim.dat"),
             modelout=here("24-simulation", "mplus_files", "cfa2_demo_sim.inp"),
             check=TRUE, run = TRUE, hashfilename = FALSE)
```

Simulation 3

- **Non-normally distributed**
- No missing

```
cfa_3  <- mplusObject(

  TITLE = "CFA 3 - non-normal, no missing",
  MONTECARLO =
    "NAMES ARE y1-y10;
     NOBSERVATIONS = 265;
     NREPS = 10000;
     SEED = 53487;
     CLASSES = C(1);
     GENCLASSES = C(2);
     SAVE = cfa3.sav;",

  ANALYSIS =
    "TYPE = MIXTURE;
     ESTIMATOR = MLR;",

  MODELPOPULATION =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*.8;
     f1@1 f2@1;
     y1-y5*.36 y6-y10*9;
     f1 WITH f2*.95;
     [C#1@-2];

     %C#1%         ! outlier class (size = 12%)

     [f1@0 f2@15]; ! means (facotr 2 set to 15 to tune skewness & kurtosis)
     f1@1 f2@5;    ! variances (facotr 2 set to 5  to tune skewness & kurtosis)

     %C#2%         ! majority class (size = 88%)

     [f1@0 f2@0];
     f1@1 f2@1;",

  MODEL =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*4;
     f1@1 f2@1;
     y1-y5*.36 y6-y10*9;
     f1 WITH f2*.20;

   [y6-y10*1.42];" ,

  OUTPUT = "TECH9;")

cfa_3_fit <- mplusModeler(cfa_3,
             dataout=here("24-simulation", "mplus_files", "cfa3_demo_sim.dat"),
             modelout=here("24-simulation", "mplus_files", "cfa3_demo_sim.inp"),
             check=TRUE, run = TRUE, hashfilename = FALSE)
```

Simulation 4a

- **Non-normally distributed**
- **missing set to 50% for y6 - y10**

```
cfa_4  <- mplusObject(

  TITLE = "CFA 4 - non-normal, missing (50%)",

  MONTECARLO =
    "NAMES ARE y1-y10;
     NOBSERVATIONS = 315;
     NREPS = 10000;
     SEED = 53487;
     CLASSES = C(1);
     GENCLASSES = C(2);
     PATMISS = y6 (.5) y7 (.5) y8 (.5) y9(.5) y10 (.5);
     PATPROB = 1;
     SAVE = cfa4.sav;",

  ANALYSIS =
    "TYPE = MIXTURE;
     ESTIMATOR = MLR;",

  MODELPOPULATION =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*.8;
     f1@1 f2@1;
     y1-y5*.36 y6-y10*9;
     f1 WITH f2*.95;
     [C#1@-2];
     %C#1%
     [f1@0 f2@15];
     f1@1 f2@5;
     %C#2%
     [f1@0 f2@0];
     f1@1 f2@1;",

  MODEL =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*4;
     f1@1 f2@1;
     y1-y5*.36 y6-y10*9;
     f1 WITH f2*.20;
     [y6-y10*1.42];" ,

  OUTPUT = "PATTERNS TECH9;")

cfa_4_fit <- mplusModeler(cfa_4,
             dataout=here("24-simulation", "mplus_files", "cfa4_demo_sim.dat"),
             modelout=here("24-simulation", "mplus_files", "cfa4_demo_sim.inp"),
             check=TRUE, run = TRUE, hashfilename = FALSE)
```

```
cfa4 <- read.delim(here("24-simulation", "mplus_files", "cfa4.sav"), sep = "",
                   header = FALSE,
                   na.strings = "999.000000")

describe(cfa4) %>%
  gt() %>%
  fmt_number(c(3:13), decimals = 3)
```

vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
1	315	−0.012	1.028	−0.095	0.003	1.019	−3.268	2.970	6.238	−0.128	−0.004	0.058
2	315	−0.017	0.997	0.003	0.008	0.966	−3.006	2.648	5.653	−0.182	0.072	0.056
3	315	−0.018	1.002	0.025	−0.024	1.113	−2.333	2.793	5.126	0.065	−0.470	0.056
4	315	−0.057	0.996	−0.026	−0.055	1.026	−2.846	2.867	5.713	−0.013	−0.222	0.056
5	315	−0.067	0.970	−0.038	−0.052	0.956	−3.246	2.823	6.069	−0.151	0.077	0.055
6	168	0.635	4.220	0.277	0.200	2.896	−7.538	16.745	24.283	1.282	2.526	0.326
7	155	1.459	5.028	0.849	1.008	4.523	−11.340	18.046	29.386	0.761	0.797	0.404
8	150	1.376	4.639	0.639	0.898	3.477	−7.371	18.302	25.673	1.045	1.226	0.379
9	153	1.319	4.639	0.550	0.806	3.376	−7.774	20.451	28.226	1.199	2.071	0.375
10	153	1.403	5.019	0.621	0.752	3.962	−9.297	18.321	27.618	1.195	1.578	0.406
11	315	1.898	0.303	2.000	1.996	0.000	1.000	2.000	1.000	−2.625	4.906	0.017
12	315	1.000	0.000	1.000	1.000	0.000	1.000	1.000	0.000	NaN	NaN	0.000

view characteristics of simulated data

Simulation 4b

- explore biased outputs
- vary sample size to see changes in bias

```
# testing & tuning

cfa_bias  <- lapply(1:5, function(k) {
  cfa_004  <- mplusObject(

  TITLE = "CFA 1 - non-normal, no missing",

  MONTECARLO =
    glue("NAMES ARE y1-y10;
     NOBSERVATIONS = {315-k*25}; ! vary sample size
     NREPS = 10000;
     SEED = 53487;
     CLASSES = C(1);
     GENCLASSES = C(2);
     SAVE = cfa004_{k}.sav;"),

  ANALYSIS =
    "TYPE = MIXTURE;
```

```
     ESTIMATOR = MLR;",

  MODELPOPULATION =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*.8;
     f1@1 f2@1;
     y1-y5*.36 y6-y10*9;
     f1 WITH f2*.95;
     [C#1@-2];

     %C#1%         ! outlier class

     [f1@0 f2@15]; ! means (factor 2 set to 15 to tune skewness & kurtosis)
     f1@1 f2@5;    ! variances (factor 2 set to 5  to tune skewness & kurtosis)

     %C#2%         ! majority class

     [f1@0 f2@0];
     f1@1 f2@1;",

  MODEL =
    "%OVERALL%
     f1 BY y1-y5*.8;
     f2 BY y6-y10*4;
     f1@1 f2@1;
     y1-y5*.36 y6-y10*9;
     f1 WITH f2*.20;

   [y6-y10*1.42];" ,

  OUTPUT = " SAMPSTAT TECH9;")

cfa_004_fit <- mplusModeler(cfa_004,
               dataout=here("24-simulation", "mplus_bias", "cfa004_demo_sim.dat"),
               modelout=sprintf(here("24-simulation", "mplus_bias", "%d_cfa004_demo_sim.inp"), k),
               check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

End of simulation practice

References

Hallquist, M. N., & Wiley, J. F. (2018). MplusAutomation: An R Package for Facilitating Large-Scale Latent Variable Analyses in Mplus. Structural equation modeling: a multidisciplinary journal, 25(4), 621-638.

Muthén, L. K., & Muthén, B. O. (2002). How to use a Monte Carlo study to decide on sample size and determine power. Structural equation modeling, 9(4), 599-620.

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686