# Online Appendix: Instructions for Using Automated Moderated Nonlinear Factor Analysis (aMNLFA v. 1.0.0)

*Last updated: July 27, 2021*

*These instructions will be kept up-to-date on* http://nishagottfredson.web.unc.edu/amnlfa

Please note that the aMNLFA package is provided as a convenient way to generate templates for pieces of code which should be edited, run, and interpreted manually in Mplus. While you can use this package to facilitate the process, all model output must be inspected manually. There are a number of vital pieces of information which must be gleaned from actually looking at the output. For instance, the aMNLFA package does not read in warnings from Mplus about negative standard errors, untrustworthy parameter estimates, and the like. **The user must inspect their Mplus inputs and outputs themselves and alter them according to empirical judgment and substantive theory.** While we put this package into the scientific community with the aim of making it easier and more convenient for people to do high-quality measurement work, the reality is that the code it generates is not likely to be perfect. Each and every M*plus* input file is meant to be checked, and potentially altered, by the user.

## Preparing the data

- Column headers must be in all-caps with no periods. Keep names short (8 characters max)
- Nominal predictors should be hard coded in two formats:
  1. A single variable with all nominal levels included
  2. Contrast-coded variables for all levels except the reference level.
- Hard-code interaction terms. Use the first three characters from each main effect with an underscore in between. Example: interaction between ABCDEFG and ZYXWV should be named ABC_ZYX.
- Hard-code quadratic terms. Use the first three characters of the linear term followed by _2. Example: GRADE and GRA_2.
- If data are longitudinal, the file should be in long format (one record per response occasion)
- Read the data in as an R dataframe. This may be done using the **read.table** function, e.g.,:

  df <- read.table("C:/Users/Dropbox/aMNLFA/data.dat", header=TRUE)

## R object definitions (User input)

The main function which defines the automated MNLFA is an **aMNLFA object.** In the first step of the analysis (described below), the user creates an aMNLFA object which is passed to all subsequent functions.

- **dir:** In quotations, indicate the location of the data. Use forward-slashes instead of back-slashes in the path. Do not include a slash at the end of the path.
- **mrdata: Multiple record data file for the user to read in. This must be an R dataframe (i.e., not a reference to an external file), as described above.**
- **indicators**: Within parentheses, list a set of indicators of a single factor in quotations with a comma separating each indicator.
- **catindicators**: Within parentheses, list the subset of indicators from **indicators** that are binary or ordinal.
- **countindicaors:** Noninvariance testing of this type is not supported by Mplus for count indicators at this time. It is best to leave this blank and use MLR estimation for any model testing measurement noninvariance of count items.
- **meanimpact**: List all variables that should be tested in the mean impact models. Use contrast-coded versions of nominal variables (not one-item factors). Include hard-coded interactions and quadratic terms formatted as described in the ‘Prepare the Data’ section. All variables should be in quotes with commas separating them. Leave blank or write **measinvar = NULL** if you do not wish to test for mean impact.

- **varimpact**: List variables to be included in the variance impact model. Use contrast-coded versions of nominal variables (not one-item factors). Include hard-coded interactions and quadratic terms formatted as described in the ‘Prepare the Data’ section. All variables should be in quotes with commas separating them. We strongly suggest limiting this list to main effects unless absolutely necessary. Leave blank or write **measinvar = NULL** if you do not wish to test for variance impact.
- **measinvar**: List variables to be included in tests for measurement non-invariance. Use effect-coded versions of nominal variables. Include hard-coded interactions and quadratic terms formatted as described in the ‘Prepare the Data’ section. All variables should be in quotes with commas separating them. Leave blank or write **measinvar = NULL** if you do not wish to test for measurement invariance.
- **ID**: Define the person ID variable in quotation marks.
- **auxiliary**: List all variables that should be included to identify each case in long file. Typically, this will include the ID variable and a time metric (e.g., wave) that differs from **time** (below). Leave blank or write **auxiliary = NULL** if there are no auxiliary variables.
- **time**: If applicable, list the variable that represents the metric of time that will be used for plots and for data analysis. This variable will be rounded to the nearest integer for plots, but it will be left in its raw form for data analysis. Leave blank or write **time = NULL** if data are cross-sectional.
- **factors**: List the full set of factors for which impact and measurement invariance is to be tested. Use dummy coding for nominal variables. Enclose each variable in quotations and separate with commas. These are the variables that will be used to generate plots. Leave blank or write **factors = NULL** if there are no factors.
- **thresholds**: TRUE or FALSE: indicate whether you would like to test measurement invariance of thresholds for ordinal indicators.

## *Analysis Step*

### 1. *Define aMNLFA object (aMNLFA.object)*

- This step defines an aMNLFA object, which converts all of the variables above to a set of instructions for all of the subsequent steps.
  - **Example:**

```
obj <- aMNLFA.object(dir =
"C:/Users/AddictionResearcher/Dropbox/aMNLFA/data",
indicators =
c("AU2","AU3","AU4","AU5","AU6”,”AU7”,”AU8”,”AC1”,”AC2”,”AC3”,”AC4”,”AC
5”),
catindicators =
c("AU2","AU3","AU4","AU5","AU6”,”AU7”,”AU8”,”AC1”,”AC2”,”AC3”,”AC4”,”AC
5”),
countindicators = NULL,
id = “ID”,
auxiliary = “ID”,
time = “CNGRADE”,
factors = c(“X2”,”X3”),
meanimpact = c("X2", "X3", "X4"),
varimpact = c("X2", "X3", "X4"),
measinvar = c("X2", "X3", "X4"),
thresholds = FALSE)
```

- **Please check the specification of the aMNLFA object, as misspecifications at this step will affect all following steps. In the previous example, submitting the command `obj` will display all of the arguments above.**

### 2. *Plot items over time and/or as a function of predictors (aMNLFA.itemplots)*

- If data are longitudinal, running this code outputs PNG files containing plots of each item over time (rounded to the nearest integer; **time**) as a function of each factor being considered (**factors**)
- If data are cross-sectional, running this code outputs PNG files containing boxplots of each item by factors in **factors**.
- Occasions with less than 1% of the sample responding are omitted from the plots.
- **Example:**

```
aMNLFA.itemplots(myObject)
```

### 3. *Draw a calibration sample create Mplus input files for mean impact, variance impact, and item-by-item measurement non-invariance (aMNLFA.sample)*

- Running this code will output a data file with one record per ID (**myID**) chosen at random ("sample.dat"). This calibration sample will be used for obtaining parameter estimates to be used in scoring models.
- If data are cross-sectional, the calibration sample will be identical to the original file.
- **Example:**

```
        aMNLFA.sample(obj)
```

-

### 4. *Create Mplus input files for mean impact, variance impact, and item-by-item measurement non-invariance (aMNLFA.initial)*

- This code generates the following separate M*plus* input scripts: one for mean impact (including predictors in **meanimpact; filename = meanimpactscript.inp)**, one for variance impact (from **varimpact; filename = varimpactscript.inp**), and one for testing measurement noninvariance for each latent variable indicator (from **measinvar; filename = measinvarscript_<item name>.inp**).
- To avoid errors, the variance impact model includes all predictors tested in the variance model as predictors of the latent variable mean.
- Run these scripts manually. They may take several hours. **One time-saving technique is to run all of the Mplus files in a batch. There are two ways to do this. First, you may use the R function runModels, which is a part of the MplusAutomation package. Second, you may group-select and open your Mplus files all at once to run them using the Mplus GUI. This may require first directing your computer to always open *.inp files using Mplus.**
- Due to the complexity of these models, it may be necessary for you to manually adjust the input scripts to remove problematic parameters. Do not change the order of parameter labels for model constraint parameters.
- Proceed to the next step after all of these models have converged.
- **Example:**

```
        aMNLFA.initial(obj)
```

### 5. *Incorporate all 'marginally significant' terms into a simultaneous Mplus input file (aMNLFA.simultaneous*

- ***Running this code results in a single Mplus script file (round2calibration.inp)***
- ***All mean and variance impact terms with p<.05 are included***

**Check to make sure that all loading DIF effects with p<.05 have been included in round2calibration.inp.**

- All noninvariance terms are included if either the loading or the intercept have *p*<.05
- Run the resulting script manually. This model make take several hours.
- Due to the complexity of these models, it may be necessary for you to manually adjust the input scripts to remove problematic parameters. Do not change the order of parameter labels for model constraint parameters.
- Proceed to the next step after this model has converged.

- **Example:**

```
aMNLFA.simultaneous(obj)
```

- *Explore which DIF effects should be retained using aMNLFA.prune()*

- aMNLFA.prune() applies three different alpha criteria to DIF effects: no correction (i.e., alpha=.05), Bonferroni (i.e., alpha=.05/# tests), or Benjamini-Hochberg (a compromise between the two extremes; see Thissen, Steinberg, & Kuang, 2002, *JEBS*).
- The function applies these corrections with # tests = the number of *possible* tests (i.e., number of items times number of covariates) or the number of *actual tests conducted* (i.e., the number of effects included in the simultaneous model).
- **Example:**

```
prune.object<-aMNLFA.prune(some.object)
```

6. *Visualize DIF effects as a function of trimming criteria (no correction, Bonferroni, or Benjamini-Hochberg) – iterate with step 6.*

- Create an object using aMNLFA.prune.
- Feed the object into aMNLFA.DIFplot.
- When p-values are small, it can be helpful to log the y-axis. For this option, choose log=TRUE.
- **Example:**

```
aMNLFA.DIFplot(prune.object,”loading”,log=FALSE)
aMNLFA.DIFplot(prune.object,”intercept”,log=FALSE)
```

7. *Generate simultaneous model for non-invariance terms using the preferred method for Type I error correction and generate final calibration model for longitudinal data; generate factor score estimates for cross-sectional data (aMNLFA.final)*

- *All mean and variance impact terms with p<.10 are included (impact models should be inclusive but parsimonious).*
- Noninvariance terms for factor loadings are trimmed using either Bonferonni (**method=“bonferroni”**) or the Benjamini Hochberg procedure with a 5% false detection rate (**method=”bh”**).
- The number of tests is set to **mchoice=”ibc”** to equal to the number of items times the number of predictors, or **mchoice=”actual”** to equal the number of tests that were actually included in the simultaneous model.
- If you are testing threshold DIF, you can choose highest.category=TRUE to trim nonsignificant effects based on the category with the highest test statistic for each item or FALSE to trim based on all categories’ thresholds for each item.
  - This is best explained with an example. Suppose that item *j* has 4 levels, and threshold DIF is being estimated (i.e., threshold = TRUE). The round 2 calibration model tests for the effect of covariate *q* on each of item *j*’s thresholds. Note that there are 3 thresholds corresponding to the 4 levels. The resulting test statistics (not the value of the parameters themselves!) and *p* values are as follows:
    - Threshold 1: $z = 0.57$, $p = .569$
    - Threshold 2: $z = 2.45$, $p = .014$
    - Threshold 3: $z = 1.97$, $p = .049$
  - Only Thresholds 2 and 3 are significant, even given an unadjusted *p* value. But we would still estimate DIF effects for Threshold 1 – we estimate effects for all thresholds for a given item, even if only one is significant. So, when deciding on the number of tests, should item *j* count for

3, since there are 3 parameters? If **highest.category = FALSE**, this will be the case; it will test 3 parameters per item. Alternatively, should item *j* only count for 1, since we would include all thresholds effects even if only one is significant? If **highest.category = TRUE**, this will be the case; only Threshold 2 will be tested, because if it is significant we'll include Thresholds 1 and 3 anyway.

- Note that noninvariance terms for intercepts are tested if the corresponding factor loading is invariant.
- Running this code produces a M*plus* input script containing only the effects that meet these criteria (round2calibration.inp). This is the final calibration model for longitudinal data.
- This script produces a file containing factor score estimates if data are cross-sectional.
- Run the resulting script manually. This model make take several hours.
- If your data are longitudinal, proceed to the next step after this model has converged.
- **Example:**

```
aMNLFA.final(obj)
```

### 8. *(Only for longitudinal data) Use parameter values generated from the last calibration model to fix parameter values in the scoring model using the full, longitudinal dataset (aMNLFA.scoring)*

- Running this code creates an M*plus* script (scoring.inp) that fixes model parameter values to the estimates that were obtained in the final calibration model.
- The resulting M*plus* script uses the long (mr.dat) data file and outputs factor score estimates for each observation. Run the resulting script manually.

**Example:**

```
aMNLFA.scoring(obj)
```

### 9. ***Describe and visualize factor score estimates and generate empirical item characteristic curves (aMNLFA.scoreplots)***

- This segment of code reads in factor score estimates and merges them with mr.dat.
- When applicable, factor scores are visualized over time (**time**) rounded to the nearest integer and as a function of the predictive factors (**factors**). Plots are output as PNGs.
- Empirical item characteristic curves are plotted and output as PNGs. These represent the average item value as a function of the factor score estimates.
- The merged file with the raw data and factor score estimates is saved as mr_with_scores.dat.
- **Example:**

```
aMNLFA.scoreplots(obj)
```