Measurement Invariance: Syntax and Software Package Information

Syntax for Mplus

The Mplus syntax examples are presented using the formatting of Mplus input files. Blue font and black font are used for commands, while green font in rows headed by exclamation marks is used for comments.

For each approach to measurement invariance/differential item functioning and response shift testing, syntax for one or more steps is presented in full, followed by tables describing how the models are adapted for subsequent steps.

Measurement invariance/Differential item functioning

Multi-group Confirmatory Factor Analysis (MG-CFA)

Step 1 - Configural invariance

```
!==Start of input syntax==
DATA:
  File is !add file name and location  ;

VARIABLE:
  Names are
  GROUP Y1_B Y2_B Y3_B Y4_B Y5_B Y1_F Y2_F Y3_F Y4_F Y5_F;
  Missing are .;
!Items used in analysis
usevariables are Y1_B-Y5_B;
!Indicate that these variables are categorical (ordered categorical)
categorical are Y1_B-Y5_B;
!Identify the two groups you want to compare
GROUPING IS GROUP (1=GROUP1 2=GROUP2);

ANALYSIS:
!Theta parameterization is needed when including item residual variances in model with
!categorical variables and WLSMV estimator
PARAMETERIZATION = THETA;

!Overall model
!These parameters will be held equal across any groups that do not
!include them in a group-specific model
MODEL:
F BY Y1_B-Y5_B;
!========================Model for group=GROUP1=====================================
!These parameters will be estimated for group 1
Model GROUP1:
!Factor loadings will be estimated for group 1
!Note that we have to specify @1 for a loading for these group models
F BY Y1_B@1 Y2_B-Y5_B;
```

```
!By default, the factor mean for group 1 is zero
!But, we may want to change this later, so we will add the command here now
[F@0];
!By default, the variance of the factor for group 1 is estimated
!But, we may want to change this later, so we will add the command here now
F;
!By default, the residual variances of the items for group 1 are 1.00
!But, we may want to change this later, so we will add the command here now
Y1_B@1;
Y2_B@1;
Y3_B@1;
Y4_B@1;
Y5_B@1;
!The thresholds for group 1 are estimated
!For model identification purposes, we need to force some thresholds to be the same across groups
!Specifically, 1 threshold per item, plus 1 extra threshold per factor
!For this model, this means 1 threshold for items 1-5, plus 1 extra threshold for the F factor
!Equality constraints are set by adding (#) after the parameter - every parameter with the same
!number will be made equal
[Y1_B$1] (6);
[Y1_B$2] (7);
[Y1_B$3];
[Y1_B$4];
[Y2_B$1] (8);
[Y2_B$2];
[Y2_B$3];
[Y2_B$4];
[Y3_B$1] (9);
[Y3_B$2];
[Y3_B$3];
[Y3_B$4];
[Y4_B$1] (10);
[Y4_B$2];
[Y4_B$3];
[Y4_B$4];
[Y5_B$1] (11);
[Y5_B$2];
[Y5_B$3];
[Y5_B$4];

!======================Model for group=GROUP2==================================
!These parameters will be estimated specifically for group 2
Model GROUP2:
!Factor loadings will be estimated for group 2
```

```
!Note that we have to specify @1 for a loading for these group models
F BY Y1_B@1 Y2_B-Y5_B;
!By default, the factor mean for group 2 is estimated
!But, we may want to change this later, so we will add the command here now
[F];
!By default, the variance of the factor for group 2 is estimated
!But, we may want to change this later, so we will add the command here now
F;
!By default, the residual variances of the items for group 2 are estimated
!But, we may want to change this later, so we will add the command here now
Y1_B;
Y2_B;
Y3_B;
Y4_B;
Y5_B;
!The thresholds for group 2 are estimated
!For model identification purposes, we add the equality constraints
[Y1_B$1] (6);
[Y1_B$2] (7);
[Y1_B$3];
[Y1_B$4];
[Y2_B$1] (8);
[Y2_B$2];
[Y2_B$3];
[Y2_B$4];
[Y3_B$1] (9);
[Y3_B$2];
[Y3_B$3];
[Y3_B$4];
[Y4_B$1] (10);
[Y4_B$2];
[Y4_B$3];
[Y4_B$4];
[Y5_B$1] (11);
[Y5_B$2];
[Y5_B$3];
[Y5_B$4];

!Save a new file with information for conducting a test of the difference
!between this model and the next model
SAVE: DIFFTEST = F_CONFIG.DIF;
!==End of input syntax==
```

Step 2 - Metric invariance

Change the following in the Configural Invariance to obtain the Metric Invariance model
Add to ANALYSIS: !Request invariance testing DIFFTEST = GH_CONFIG.DIF;
Change in Model GROUP1: !Because the factor loadings are in "Model" above, by default they will !be held equal across groups *if* the are not !specified for each group. But, if the first factor loading = 1.00 in both groups, they will be equivalent by default and their !equivalence when they are *estimated* will not be tested. So we will free the factor loading of the first item and set the !factor variance in the first group to 1.00 instead (so that we still have a scale for the factor) F BY Y1_B*(1) Y2_B (2) Y3_B (3) Y4_B (4) Y5_B (5);
Change in Model GROUP 1: !By default, the variance of the factor for group 1 is estimated. But since we have freed the first factor loadings, the factor !variance is set to 1.00 to set the factor scale GH@1;
Change in Model GROUP 2: !Factor loadings are equal to group 1 GH BY GH1_B(1) GH2_B (2) GH3_B (3) GH4_B (4) GH5_B (5);
Change in SAVE: SAVE: DIFFTEST = F_METRIC.DIF;

Step 3 - Scalar invariance

Change the following in the Metric Invariance to obtain the Scalar Invariance model
Change in ANALYSIS: !Request invariance testing DIFFTEST = F_METRIC.DIF;
Add to MODEL: !By adding the thresholds here and removing them from the group-specific models, !the thresholds will be held equal across groups and no longer estimated separately [Y1_B$1]; [Y1_B$2]; [Y1_B$3]; [Y1_B$4]; [Y2_B$1]; [Y2_B$2]; [Y2_B$3];

[Y2_B$4];
[Y3_B$1];
[Y3_B$2];
[Y3_B$3];
[Y3_B$4];
[Y4_B$1];
[Y4_B$2];
[Y4_B$3];
[Y4_B$4];
[Y5_B$1];
[Y5_B$2];
[Y5_B$3];
[Y5_B$4];

Remove thresholds from Model GROUP1:

!By removing the thresholds from this model, they are no longer estimated separately for this group. Instead, they are estimated in the overall model and held equal across the groups. Note that we could also hold them equal across groups by creating a label (#) for each threshold and matching label for group 2

Remove thresholds from Model GROUP2:

!By removing the thresholds from this model, they are no longer estimated separately for this group. Instead, they are estimated in the overall model and held equal across the groups. Note that we could also hold them equal across groups by creating a label (#) for each threshold and matching label for group 1

Change in SAVE:

SAVE: DIFFTEST = GH_SCALAR.DIF;

Step 4 – Strict/Full invariance

Change the following in the Scalar Invariance to obtain the Strict/Full Invariance model

Change in ANALYSIS:

!Request invariance testing
DIFFTEST = F_SCALAR.DIF;

Add to MODEL:

!By adding the item error variances here, they will be fixed to 1 in both groups.
!We could also do this by adding this to the group 2 model
GH1_B@1;
GH2_B@1;
GH3_B@1;
GH4_B@1;
GH5_B@1;

Remove the item error variances from Model GROUP1:

!By removing the error from this model, they are no longer estimated separately for this group. Instead, they are estimated in the overall model and held equal across the groups. Note that we could also hold them equal across groups by creating a label (#) for each error variance and matching label for group 2

Remove the item error variances from Model GROUP2:

!By removing the error from this model, they are no longer estimated separately for this group. Instead, they are estimated in the overall model and held equal across the groups. Note that we could

also hold them equal across groups by creating a label (#) for each error variance and matching label for group 1
Remove SAVE:

Multi-group Item Response Theory (MG-IRT)

Step 1 – Testing item Y1_B

```
!==Start of input syntax==
DATA:
  File is !add file name and location ;

VARIABLE:
  Names are
  GROUP Y1_B Y2_B Y3_B Y4_B Y5_B Y1_F Y2_F Y3_F Y4_F Y5_F;
  Missing are .;
!Items used in analysis
usevariables are GROUP Y1_B-Y5_B;
!Indicate that these variables are categorical (ordered categorical)
categorical are Y1_B-Y5_B;
!Use "knownclass" to identify the grouping variable, (2)=# of groups, adjust as needed
CLASSES = CGROUP (2);
KNOWNCLASS IS CGROUP (GROUP = 1 GROUP = 2);

!Change estimator to MLR (from from WLSMV default for categorical). Use "type=mixture" and
!"algorithm=integration". Mixture models are latent class models, but using "knownclass"
! 'tricks' Mplus into using identified groups in a mixture model

ANALYSIS:
estimator = MLR;
TYPE = MIXTURE;
ALGORITHM = INTEGRATION

!First specify the model in the overall population
MODEL:
!Set variance of factor to 1 and mean to 0. Placing a * after the first item
!frees the factor loading (otherwise automatically fixed to 1.00)
%overall%
F BY Y1_B* Y2_B Y3_B Y4_B Y5_B;
F@1;
[F@0];

!Then identify those parameters you want to estimate separately within each group
!(known class). The codes in brackets designate a label for each parameter, which
!can be used in the MODEL TEST portion of the input. In this case, the loading and
```

```
!thresholds for Y1_B are being estimated in each class
%cgroup#1%
F BY Y1_B* (p1);
[Y1_B$1-Y1_B$4] (p2-p5);

%cgroup#2%
F BY Y1_B* (p11);
[Y1_B$1-Y1_B$4] (p12-p15);

!The model test command can be used to test restrictions on the model. Test whether the
!parameters are significantly different.
MODEL TEST:
p1=p11;
p2=p12;
p3=p13;
p4=p14;
p5=p15;
!==End of input syntax==
```

Steps 2-5 – Test items Y2_B to Y5_B

Change the following in order to test the remaining items
Change in %cgroup#1% !In sequential models, change Y1_B to Y2_B, Y3_B, Y4_B, Y5_B. Keep the (labels)
Change in %cgroup#2% !In sequential models, change Y1_B to Y2_B, Y3_B, Y4_B, Y5_B. Keep the (labels)

MIMIC Model

Step 1 – Baseline model, no DIF

```
!==Start of input syntax==
DATA: File is !add file name and location;

VARIABLE:
  Names are
  GROUP Y1_B Y2_B Y3_B Y4_B Y5_B Y1_F Y2_F Y3_F Y4_F Y5_F;
  Missing are .;
!Items used in analysis
usevariables are Y1_B-Y5_B GROUP;
!Indicate that these variables are categorical (ordered categorical)
categorical are Y1_B-Y5_B;

ANALYSIS:
estimator = ml;

MODEL:
```

```
!Specify the factor model for F
F BY Y1_B* Y2_B-Y5_B;
F@1;
!Regress the factor on the grouping variable
F ON GROUP;
!The grouping variable does not directly affect the items. That is, there is no DIF,
!the items are invariant across groups
Y1_B-Y5_B ON GROUP@0;
!==End of input syntax==
```

Step 2 – Test uniform DIF for item Y1_B

Change the following in the syntax to test for uniform DIF for item Y1_B
Add to MODEL: !Estimate the regression for the item being tested for uniform DIF !Does the grouping variable have a significant effect? Y1_B on GROUP;
Change in MODEL: !The grouping variable does not affect the remaining items Y2_B-Y5_B ON gender@0;

Step 3 – Test non-uniform DIF for item Y1_B

Change the following in the syntax to test for non-uniform DIF
Add to ANALYSIS: !Needed to include interaction between factor and covariate (below) type = random;
Add to MODEL: !Create an interaction of the factor with the covariate. XWITH is the latent variable version of WITH FGROUP | F XWITH GROUP;
Change in MODEL: !Estimate the regression for the item being tested for non-uniform DIF. Do the grouping variable and !interaction have a significant effect? Y1_B ON GROUP FGROUP;
Change in MODEL: !The grouping variable and interaction do not affect the remaining items Y2_B-Y5_B ON GROUP @0; Y2_B-Y5_B ON FGROUP@0;

Steps 4-11 – Test DIF for items Y2_B – Y5_B

Change the following in order to test the other items
!In sequential models, change Y1_B to Y2_B, Y3_B, Y4_B, Y5_B as the item being regressed on GROUP and on FGROUP. Adjust the variables being regressed @0 on GROUP and FGROUP accordingly

Response Shift

Step 1 – Full Response Shift model

```
!==Start of input syntax==
```

```
DATA:
  File is !add file name and location ;

VARIABLE:
  Names are
  GROUP Y1_B Y2_B Y3_B Y4_B Y5_B Y1_F Y2_F Y3_F Y4_F Y5_F;
  Missing are .;
!Items used in analysis
usevariables are
Y1_B-Y5_B !Variables at baseline
Y1_F-Y5_F; !Variables at follow up
!Indicate that these variables are categorical (ordered categorical)
categorical are Y1_B-Y5_B Y1_F-Y5_F;

ANALYSIS:
ESTIMATOR = WLSMV;
!Theta parameterization is needed when including item residual variances in model with
!categorical variables and WLSMV estimator
PARAMETERIZATION = THETA;

MODEL:
!----------MODEL AT BASELINE-----------------------------------------
!Latent factor loadings
! *override default of fixing the first loading at 1.00
F_B BY Y1_B* GH2_B-Y5_B;
!Fix latent factor mean to zero and variance to 1.00
[F_B@0];
F_B@1;
!Item thresholds
[Y1_B$1-Y1_B$4];
[Y2_B$1-Y2_B$4];
[Y3_B$1-Y3_B$4];
[Y4_B$1-Y4_B$4];
[Y5_B$1-Y5_B$4];

!----------MODEL AT FOLLOWUP-----------------------------------------
!Latent factor loadings
! *override default of fixing the first loading at 1.00
F_F BY Y1_F* Y2_F-Y5_F;
!Fix latent factor mean to zero and variance to 1.00
[F_F@0];
F_F@1;
!Item thresholds
[Y1_F$1-Y1_F$4];
```

```
[Y2_F$1-Y2_F$4];
[Y3_F$1-Y3_F$4];
[Y4_F$1-Y4_F$4];
[Y5_F$1-Y5_F$4];

!----------Correlations across measurement occasions----------------------------
!Correlate the latent factors;
F_B WITH F_F;
Correlate the same indicators across measurement occasions
Y1_B WITH Y1_F;
Y2_B WITH Y2_F;
Y3_B WITH Y3_F;
Y4_B WITH Y4_F;
Y5_B WITH Y5_F;

!Save a new file with information for conducting a test of the difference
!between this model and the next model
!Note that because the difftest requires nested models, the step 1 model must be the 'Full response
!shift model', not the 'No response shift' model'
SAVEDATA: difftest is F_response shift_step 1.dat;
```

Step 2- No Response Shift model

Change the following in order to obtain the full Response Shift model
Add to ANALYSIS: !Test the difference between the full RS and no RS models difftest is F_response shift_step 1.dat;
Add in MODEL: !Add labels for the factor loadings for model at baseline F_B BY Y1_B* Y2_B-Y5_B(1-5); !Add labels for the factor loadings for model at follow up. Loadings held equal to baseline F_F BY Y1_F* Y2_F-Y5_F(1-5);
Add in MODEL: !add labels for item thresholds at baseline [Y1_B$1-Y1_B$4] (6-9); [Y2_B$1-Y2_B$4] (10-13); [Y3_B$1-Y3_B$4] (14-17); [Y4_B$1-Y4_B$4] (18-21); [Y5_B$1-Y5_B$4] (22-25);
Add in MODEL: !Add labels for the item thresholds for model at follow up. Loadings held equal to baseline [Y1_F$1-Y1_F$4] (6-9); [Y2_F$1-Y2_F$4] (10-13); [Y3_F$1-Y3_F$4] (14-17); [Y4_F$1-Y4_F$4] (18-21); [Y5_F$1-Y5_F$4] (22-25);

Next Steps- Remove individual parameter constraints (examples)

Change the following in the "No Response Shift" model (Step 2) in order to remove the constraint of "no response shift" for the factor loading of item Y1
Change in MODEL: !Remove label for the factor loading for Y1 for model at baseline F_B BY Y1_B* Y2_B (2) Y3_B (3) Y4_B (4) Y5_B (5); !Remove label for the factor loading for Y1 for model at follow up. Now no longer held equal to baseline F_F FY Y1_F* Y2_F (2) Y3_F (3) Y4_F (4) Y5_F (5);
Change the following in the "No Response Shift" model (Step 2) in order to remove the constraint of "no response shift" for the thresholds of item Y1
Change in MODEL: !Remove labels for the factor loading for Y1 for model at baseline F_B BY Y1_B* Y2_B (2) Y3_B (3) Y4_B (4) Y5_B (5); !Remove label for the factor loading for Y1 for model at follow up. Now no longer held equal to baseline F_F FY Y1_F* Y2_F (2) Y3_F (3) Y4_F (4) Y5_F (5);

Syntax and Procedures for SAS

Differential Item Functioning

LOGISTIC and GENMOD Procedures for dichotomous and polytomous items

LOGISTIC procedure to detect DIF for dichotomous items

```
PROC LOGISTIC DATA = XYZ;

MODEL /LINK=logit;

RUN;
```

GENMOD procedure to detect DIF and polytomous items. PROC GENMOD fits generalized linear logistic regression models using the Newton-Raphson method to maximize the likelihood

```
PROC GENMOD DATA= XYZ_poly;

MODEL  /LINK=clogit DIST=mutl;

RUN;
```

PROC NLMIXED and PROC GLIMMIX for IRT models

Item response theory (IRT) models such as; 1-parameter (1PL), 2-parameter (2PL), and 3-parameter (3PL) logistic regression can be used to detect DIF in SAS using PROC NLMIXED and PROC GLIMMIX

```
PROC GLIMMIX DATA= XYZ METHOD= quadrature GRADIENT NOCLPRINT ORDER=data;

CLASS person;

MODEL  /DIST=binary LINK=logit SOLUTION;

RANDOM intercept/SUBJECT =person TYPE=UN;

RUN;
```

```
PROC NLMIXED DATA=XYZ NOAD METHOD=gauss QPOINT;

PARMS;

MODEL;

RANDOM SUBJECT=person;

RUN;
```

Response Shift

PROC CALIS procedure (Covariance Analysis of Linear Structural Equations)

PROC IRT procedure (Response shift at the item level)

R Packages

Differential Item Functioning

lordif

The R package is available from the Comprehensive R Archive Network at [http://CRAN.R-project.org/package=lordif.](http://CRAN.R-project.org/package=lordif)

difR

The difR package contains several traditional methods to detect DIF in dichotomously scored items. The package is available at https://cran.r-project.org/web/packages/difR

raschtree

The *psychotree* package contains the function *raschtree*, that can be used to detect DIF in the Rasch model. The package is available at https://cran.r-project.org/web/packages/psychotree

DIFlasso uses a penalty approach to DIF in Rasch models. It can be used with multiple (metric) covariates.

GPCMlasso provides a function to detect DIF in generalized partial credit models (GPCM).

DIFtree performs recursive partitioning for simultaneous selection of items and variables that induce DIF in dichotomous or polytomous items.

DIFboost can be used for DIF detection in Rasch models by boosting techniques

lavaan

The lavaan package free open-source for latent variable modeling

OpenMx package allows for the estimation of structural equation model (SEM) estimate parameters given observed data

sem package fits SEMs by FIML, and structural equations in observed-variable models by 2SLS. Categorical variables in SEMs can be accommodated via the *polycor* package.