stat > mplus > seminars > intromplus_part2 > additional.htm

Mplus Class Notes
Freeing, Constraining, and Testing Parameters

There are at least two reason a researcher might want to fix or free parameters. The first is that while the default settings in Mplus do a fairly good job of deciding which parameters should be freed and which should be fixed, researchers often want to fit models that differ from these defaults in some way. Second, in addition fitting an overall model, researchers often wish to test hypotheses about specific parameters. Freeing and/or fixing parameters is often necessary in order to perform these tests. These commands are general in the sense that the same basic commands are used with all of the models estimated by Mplus.

Mplus version 5.2 was used for these examples.

1.0 Constraining a Parameter to a Given Value

You may want to fix paths to a given value for a variety of reasons, for example, to identify a model (although Mplus will often do this by default), or test a nested model. In the example below we fix the path from adjust to achieve to zero. We do this using the **@** symbol followed by the number we wish to fix the parameter to (**adjust@0**).

```
Title: Fixing a parameter to zero
Data:
  File is worland_data.dat ;
Variable:
  Names are ppsych ses verbal vissp mem read arith spell motiv extra harm stabi;
Model:
  family by ppsych ses;
  cog by verbal vissp  mem;
  achieve by read arith spell;
  adjust by motiv extra harm stabi;

  achieve on cog adjust@0 ;
  adjust on family;
```

The MODEL RESULTS section of the output is shown below.

```
MODEL RESULTS

                                                    Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value

 FAMILY   BY
    PPSYCH             1.000      0.000    999.000    999.000
    SES               -1.095      0.125     -8.790      0.000

 COG      BY
    VERBAL             1.000      0.000    999.000    999.000
    VISSP              0.833      0.045     18.358      0.000
    MEM                0.980      0.044     22.335      0.000

 ACHIEVE  BY
    READ               1.000      0.000    999.000    999.000
    ARITH              0.826      0.034     24.548      0.000
    SPELL              0.942      0.026     35.902      0.000

 ADJUST   BY
    MOTIV              1.000      0.000    999.000    999.000
    EXTRA              0.231      0.049      4.676      0.000
    HARM               0.880      0.043     20.294      0.000
    STABI              0.677      0.046     14.824      0.000

 ACHIEVE  ON
    COG                1.008      0.041     24.605      0.000
    ADJUST             0.000      0.000    999.000    999.000

 ADJUST   ON
    FAMILY            -1.267      0.141     -9.015      0.000

 COG      WITH
    FAMILY            -0.403      0.045     -8.897      0.000

 Intercepts
    PPSYCH             0.000      0.045      0.000      1.000
    SES                0.000      0.045      0.000      1.000
    VERBAL             0.000      0.045      0.000      1.000
    VISSP              0.000      0.045      0.000      1.000
    MEM                0.000      0.045      0.000      1.000
    READ               0.000      0.045      0.000      1.000
    ARITH              0.000      0.045      0.000      1.000
    SPELL              0.000      0.045      0.000      1.000
    MOTIV              0.000      0.045      0.000      1.000
    EXTRA              0.000      0.045      0.000      1.000
    HARM               0.000      0.045      0.000      1.000
    STABI              0.000      0.045      0.000      1.000

 Variances
    FAMILY             0.235      0.047      5.017      0.000
    COG                0.736      0.063     11.616      0.000

 Residual Variances
    PPSYCH             0.763      0.054     14.247      0.000
    SES                0.716      0.051     13.947      0.000
    VERBAL             0.262      0.024     10.909      0.000
    VISSP              0.487      0.035     13.966      0.000
    MEM                0.292      0.025     11.538      0.000
    READ               0.086      0.014      6.347      0.000
    ARITH              0.376      0.027     13.764      0.000
    SPELL              0.188      0.016     11.782      0.000
    MOTIV              0.119      0.032      3.728      0.000
    EXTRA              0.951      0.061     15.699      0.000
    HARM               0.318      0.033      9.771      0.000
    STABI              0.596      0.042     14.251      0.000
    ACHIEVE            0.164      0.022      7.379      0.000
    ADJUST             0.501      0.050     10.097      0.000


QUALITY OF NUMERICAL RESULTS

     Condition Number for the Information Matrix              0.233E-02
       (ratio of smallest to largest eigenvalue)
```

Note that the coefficient for **achieve** regressed on **adjust** is equal to zero, as specified in the input file. Also note that the standard errors of parameters that were fixed rather than estimated are listed as 0. When setting constraints, it is worth checking both the parameter and its standard error, as well as the model degrees of freedom to be sure the constraints were implemented as desired.

2.0 Freeing a Parameter

Although parameters can be fixed manually, to simplify model input, Mplus often fixes parameters to a given value by default. By convention, these values are typically zero or one. For example, by default Mplus fixes the path loading for the first observed variable to 1 in order to identify the latent variable. To free a parameter that is otherwise fixed by default, an asterisk (*) is placed immediately after the parameter. In the example below, an alternative method of identifying a latent variable in a confirmatory factor analysis is used. Instead of using the default method of identifying the latent variable by fixing the first path coefficient (**motiv**) to 1, we free this parameter using **motiv*** in the line **adjust by motiv* extra harm stabi;**. Nothing else in that line of syntax is changed. In order to identify this model, the mean of the latent variable (**adjust**) is fixed to 0 and it's variance to 1. Note that listing the name of a variable in brackets refers to its mean, intercept, or threshold, while listing the variable name without brackets refers to its variance or residual variance.

```
Title: Freeing a parameter
Data: worland_data.dat ;
Variable:
  Names are
     ppsych ses verbal vissp mem read arith spell motiv extra harm stabi;
  usevariables are  motiv extra harm stabi;
Model:
adjust by motiv* extra harm stabi;
[adjust@0];
adjust@1;
```

Below the MODEL RESULTS section of the output is shown.

```
<output omitted>
MODEL RESULTS

                                                    Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value

 ADJUST   BY
    MOTIV              0.901      0.041     22.031      0.000
    EXTRA              0.190      0.048      3.916      0.000
    HARM               0.860      0.041     20.823      0.000
    STABI              0.650      0.042     15.407      0.000

 Means
    ADJUST             0.000      0.000    999.000    999.000
 Intercepts
    MOTIV              0.000      0.045      0.000      1.000
    EXTRA              0.000      0.045      0.000      1.000
    HARM               0.000      0.045      0.000      1.000
    STABI              0.000      0.045      0.000      1.000

 Variances
    ADJUST             1.000      0.000    999.000    999.000

 Residual Variances
    MOTIV              0.187      0.041      4.506      0.000
    EXTRA              0.962      0.061     15.693      0.000
    HARM               0.259      0.040      6.499      0.000
    STABI              0.575      0.041     14.055      0.000


QUALITY OF NUMERICAL RESULTS

     Condition Number for the Information Matrix              0.729E-01
       (ratio of smallest to largest eigenvalue)
```

Looking at the above output, we see that the factor loading for **motiv** is equal to .901 rather than 1, and that it has a non-zero standard error indicating that this parameter has been estimated rather than fixed. Looking further down, the mean of the latent variable **adjust** is exactly zero, and a variance of exactly 1, the values specified in the input file. Note that both parameters have a standard error of 0, which is consistant with a parameter that has been fixed rather than freed.

3.0 Constraining Parameters to Equality

In some cases, rather than specifying that a parameter take on a given value, you may want to constrain two parameters to be equal. This is particularly common in multiple group models where a parameter may be fixed to equality across groups. Below we constrain the path coefficients for **vissp** and **mem** to be the same. We do this by placing the same number within parentheses following the specification of the parameter (e.g. **vissp (1)**) in the **Model:** command. There are two things important things to remember. First, the number in parentheses must be the last thing on a given line (but not necessarily the last part of the command) , this is why **mem (1);** appears on a separate line, note that the semicolon (;) to end the command does not appear until after **mem (1)**. Second, when a single number is listed at the end of a line, it constrains all the (previously free) parameters on that line to equality. This means that if I only want to constrain one parameter in a command, that parameter needs to be on a line by itself. The exception to this is parameters fixed to some value in order to identify the model (e.g. fixing the factor loading of the first variable listed after **by** to one), these parameters will continue to be fixed at their typical value unless they are specifically freed using an asterisks (*). Additional parameters could be fixed to the same value as **vissp** and **mem** by adding the **(1)** in the appropriate place. If we wanted to fix another pair of variables to the same value (but not the same value as **vissp** and **mem**) we could do this by placing a **(2)** (or other number) in the appropriate places. Note that while we don't cover it here, list operators can be used to assign constraints to lists of variables.

```
Title: Constraining Parameters to Equality
Data:
  File is worland_data.dat ;
Variable:
  Names are ppsych ses verbal vissp mem read arith spell motiv extra harm stabi;
Model:
  family by ppsych ses;
  cog by verbal
     vissp (1)
     mem (1);
  achieve by read arith spell;
  adjust by motiv extra harm stabi;

  achieve on cog adjust;
  adjust on family;
```

Because the a constraint applies to all parameters on the line, the following produces the same constraint.

```
  cog by verbal
     vissp mem (1);
```

The abridged output generated by the input file above is shown below.

```
<output omitted>

MODEL RESULTS

                                                    Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value

 FAMILY   BY
    PPSYCH             1.000      0.000    999.000    999.000
    SES               -1.093      0.119     -9.167      0.000

 COG      BY
    VERBAL             1.000      0.000    999.000    999.000
    VISSP              0.910      0.037     24.352      0.000
    MEM                0.910      0.037     24.352      0.000

 ACHIEVE  BY
```

```
    READ               1.000      0.000    999.000    999.000
    ARITH              0.835      0.034     24.670      0.000
    SPELL              0.950      0.027     35.676      0.000

 ADJUST   BY
    MOTIV              1.000      0.000    999.000    999.000
    EXTRA              0.232      0.049      4.744      0.000
    HARM               0.868      0.043     20.207      0.000
    STABI              0.669      0.046     14.683      0.000

<output omitted>
```

Note that the path coefficients and standard errors for **vissp** and **mem** are identical, indicating that they have been fixed to equality.

4.0 Assigning Names to Parameters

Assigning names to parameters allows you to refer them in either the **model test:** or **model constraint:** commands, allowing the user to perform custom hypothesis tests or impose complex constraints. Names are assigned to parameters by placing a name in parentheses after the parameter is listed (similar to the way in which one assigns constraints). Parameter names must begin with a letter; can contain only letters, numbers, and the underscore symbol (i.e. _); and must be 8 or fewer characters. Similar to the assignment of constraints discussed above, the label must be the last item on a line, and will be applied to all parameters listed on that line. Putting a single label at the end of a line with multiple parameters listed applies that label to all of the parameters and in the process constrains the parameters to equality. Below we assign the label **a1** to the estimate of the covariance between **family** and **cog**, the label **a2** to the covariance between **family** and **achieve**, and **a3** to the covariance between **family** and **adjust**.

```
Title: Assigning parameter labels.
Data:
  File is worland_data.dat ;
Variable:
  Names are
     ppsych ses verbal vissp mem read arith spell motiv extra harm stabi;
Model:
  family by ppsych ses;
  cog by verbal vissp mem;
  achieve by read arith spell ;
  adjust by motiv extra harm stabi ;

  family with cog (a1)
    achieve (a2)
    adjust (a3);
  cog with achieve adjust;
  achieve with adjust;
```

Assigning parameter labels without specifying any tests or constraints results in output that is identical to the output for this model without specifying the parameter labels. Next we will use these labels to test hypotheses about these parameters.

5.0 Testing Hypotheses About Parameter Estimates Using Wald Tests

Mplus will allow you to test hypotheses about individual parameters (e.g. b=2), hypotheses about groups of parameters (e.g. b1=0 and b2=0), and tests about the equality of parameters (e.g. b1=b2). The **model test:** command tells Mplus that we would like to test a series of parameter constraints. The lines following the **model test:** command give the specific constraints to be tested. In the input file below three parameters are assigned names (a1, a2, and a3), then the **model test:** command is used to test the null hypothesis that **a1 = a2;** and **a1 = a3;**. This corresponds to the null hypothesis a1 = a2 = a3, that is, that the covariances of **family** with **cog**, **family** with **achieve**, and **family** with **adjust** are all equal. It is important to note that all of the constraints listed in the model test command are tested simultaneously, so if the null hypothesis was a1 = a2 (without worrying about a3) this hypothesis would need to be tested separately.

```
Title:
Data:
  File is worland_data.dat ;
Variable:
  Names are
     ppsych ses verbal vissp mem read arith spell motiv extra harm stabi;
Model:
  family by ppsych ses;
  cog by verbal vissp mem;
  achieve by read arith spell ;
  adjust by motiv extra harm stabi ;

  family with cog (a1)
    achieve (a2)
    adjust (a3);
  cog with achieve adjust;
  achieve with adjust;
model test:
  a1 = a2;
  a1 = a3;
```

The abridged output associated with this model is shown below.

```
<output omitted>

THE MODEL ESTIMATION TERMINATED NORMALLY


TESTS OF MODEL FIT

Chi-Square Test of Model Fit

          Value                            600.106
          Degrees of Freedom                    48
          P-Value                           0.0000

Chi-Square Test of Model Fit for the Baseline Model

          Value                           4124.707
          Degrees of Freedom                    66
          P-Value                           0.0000

CFI/TLI

          CFI                                0.864
          TLI                                0.813

Wald Test of Parameter Constraints

          Value                             21.395
          Degrees of Freedom                     2
          P-Value                           0.0000

Loglikelihood

          H0 Value                       -6745.325
          H1 Value                       -6445.272

Information Criteria

          Number of Free Parameters             42
          Akaike (AIC)                   13574.649
          Bayesian (BIC)                 13751.663
          Sample-Size Adjusted BIC       13618.352
            (n* = (n + 2) / 24)

RMSEA (Root Mean Square Error Of Approximation)

          Estimate                           0.152
          90 Percent C.I.                    0.141  0.163
          Probability RMSEA <= .05           0.000

SRMR (Standardized Root Mean Square Residual)

          Value                              0.063



MODEL RESULTS

                                                    Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value

 FAMILY   BY
    PPSYCH             1.000      0.000    999.000    999.000
    SES               -1.107      0.115     -9.657      0.000

 COG      BY
    VERBAL             1.000      0.000    999.000    999.000
    VISSP              0.833      0.045     18.393      0.000
    MEM                0.972      0.044     22.326      0.000

 ACHIEVE  BY
    READ               1.000      0.000    999.000    999.000
    ARITH              0.842      0.034     24.840      0.000
    SPELL              0.954      0.027     35.622      0.000

 ADJUST   BY
    MOTIV              1.000      0.000    999.000    999.000
    EXTRA              0.233      0.048      4.813      0.000
    HARM               0.857      0.042     20.295      0.000
    STABI              0.662      0.045     14.615      0.000
 FAMILY   WITH
    COG               -0.411      0.046     -8.852      0.000
    ACHIEVE           -0.363      0.044     -8.151      0.000
    ADJUST            -0.245      0.040     -6.099      0.000

<output omitted>
```

The output associated with the **model test:** command appears under TESTS OF MODEL FIT with the heading Wald Test of Parameter Constraints. The test statistic (labeled Value), degrees of freedom, and p-value for the test are given. Note that under MODEL RESULTS the estimates for the parameters we have tested are identical to those from an input file that did not include the **model test:** command. This is because the Wald test is performed after the model has been estimated, and hence does not influence the parameter estimates.

As discussed above, a variety of specific null hypotheses can be tested using the **model test:** command. Below are a few examples of different hypotheses and how to test them. One potential hypothesis is that all three of the covariances (i.e. **family** with **cog**, **family** with **achieve**, and **family** with **adjust** are all) are equal to zero. The **model test:** command shown below tests the hypothesis that all three parameters (a1, a2, and a3) are simultaneously equal to zero. (Note that the tests of the coefficients under MODEL RESULTS test the hypothesis that the coefficients are equal to zero individually.)

```
model test:
a1=0;
a2=0;
a3=0;
```

The **model test:** command can also be used to test whether a parameter (or multiple parameters if one wants to test the parameters simultaneously) is equal to some non-zero value. The **model test:** command shown below tests the hypothesis that a1 = 1.

```
model test:
a1=1;
```

How to cite this page Report an error on this page or leave a comment

The content of this web site should not be construed as an endorsement of any particular web site, book, or software product by the University of California.

IDRE RESEARCH TECHNOLOGY GROUP

High Performance Computing

Statistical Computing

GIS and Visualization

High Performance Computing
Hoffman2 Cluster
Hoffman2 Account Application
Hoffman2 Usage Statistics
UC Grid Portal

GIS
Mapshare
Visualization
3D Modeling
Technology Sandbox

Statistical Computing
Classes
Conferences
Reading Materials
IDRE Listserv

About IDRE

ABOUT CONTACT NEWS EVENTS OUR EXPERTS