CHAPTER 13
EXAMPLES: SPECIAL FEATURES

In this chapter, special features not illustrated in the previous example chapters are discussed. A cross-reference to the original example is given when appropriate.

Following is the set of special feature examples included in this chapter:

- 13.1: A covariance matrix as data
- 13.2: Means and a covariance matrix as data
- 13.3: Reading data with a fixed format
- 13.4: Non-numeric missing value flags
- 13.5: Numeric missing value flags
- 13.6: Selecting observations and variables
- 13.7: Transforming variables using the DEFINE command
- 13.8: Freeing and fixing parameters and giving starting values
- 13.9: Equalities in a single group analysis
- 13.10: Equalities in a multiple group analysis
- 13.11: Using PWITH to estimate adjacent residual covariances
- 13.12: Chi-square difference testing for WLSMV and MLMV
- 13.13: Analyzing multiple imputation data sets
- 13.14: Saving data
- 13.15: Saving factor scores
- 13.16: Using the PLOT command
- 13.17: Merging data sets
- 13.18: Using replicate weights
- 13.19: Generating, using, and saving replicate weights

EXAMPLE 13.1: A COVARIANCE MATRIX AS DATA

```
TITLE:     this is an example of a CFA with
           continuous factor indicators using a
           covariance matrix as data
DATA:      FILE IS ex5.1.dat;
           TYPE = COVARIANCE;
           NOBSERVATIONS = 1000;
VARIABLE:  NAMES ARE y1-y6;
MODEL:     f1 BY y1-y3;
           f2 BY y4-y6;
```

The example above is based on Example 5.1 in which individual data are analyzed. In this example, a covariance matrix is analyzed. The TYPE option is used to specify that the input data set is a covariance matrix. The NOBSERVATIONS option is required for summary data and is used to indicate how many observations are in the data set used to create the covariance matrix. Summary data are required to be in an external data file in free format. Following is an example of the data:

```
1.0
.86  1.0
.56  .76  1.0
.78  .34  .48  1.0
.65  .87  .32  .56  1.0
.66  .78  .43  .45  .33  1.0
```

EXAMPLE 13.2: MEANS AND A COVARIANCE MATRIX AS DATA

```
TITLE:     this is an example of a mean structure CFA
           with continuous factor indicators using
           means and a covariance matrix as data
DATA:      FILE IS ex5.9.dat;
           TYPE IS MEANS COVARIANCE;
           NOBSERVATIONS = 1000;
VARIABLE:  NAMES ARE y1a-y1c y2a-y2c;
MODEL:     f1 BY y1a y1b@1 y1c@1;
           f2 BY y2a y2b@1 y2c@1;
           [y1a y1b y1c] (1);
           [y2a y2b y2c] (2);
```

The example above is based on Example 5.9 in which individual data are analyzed. In this example, means and a covariance matrix are analyzed. The TYPE option is used to specify that the input data set contains means and a covariance matrix. The NOBSERVATIONS option is required for summary data and is used to indicate how many observations are in the data set used to create the means and covariance matrix. Summary data are required to be in an external data file in free format. Following is an example of the data. The means come first followed by the covariances. The covariances must start on a new record.

```
.4 .6 .3 .5
1.0
.86  1.0
.56  .76  1.0
.78  .34  .48  1.0
```

EXAMPLE 13.3: READING DATA WITH A FIXED FORMAT

```
TITLE:     this is an example of a CFA with
           covariates (MIMIC) with continuous factor
           indicators using data in a fixed format
DATA:      FILE IS ex5.8.dat;
           FORMAT IS 3f4.2 3f2 f1 2f2;
VARIABLE:  NAMES ARE y1-y6 x1-x3;
MODEL:     f1 BY y1-y3;
           f2 BY y4-y6;
           f1 f2 ON x1-x3;
```

The example above is based on Example 5.8 in which individual data with a free format are analyzed. Because the data are in free format, a FORMAT statement is not required. In this example, the data have a fixed format. The inclusion of a FORMAT statement is required in this situation. The FORMAT statement describes the position of the nine variables in the data set. In this example, the first three variables take up four columns each and are read such that two digits follow the decimal point (3f4.2). The next three variables take three columns with no digits after the decimal point (3f2). The seventh variable takes one column with no digits following the decimal point (f1), and the eighth and ninth variables each take two columns with no digits following the decimal point (2f2).

EXAMPLE 13.4: NON-NUMERIC MISSING VALUE FLAGS

```
TITLE:     this is an example of a SEM with
           continuous factor indicators using data
           with non-numeric missing value flags
DATA:      FILE IS ex5.11.dat;
VARIABLE:  NAMES ARE y1-y12;
           MISSING = *;
MODEL:     f1 BY y1-y3;
           f2 BY y4-y6;
           f3 BY y7-y9;
           f4 BY y10-y12;
           f4 ON f3;
           f3 ON f1 f2;
```

The example above is based on Example 5.11 in which the data contain no missing values. In this example, there are missing values and the asterisk (*) is used as a missing value flag. The MISSING option is used to identify the values or symbol in the analysis data set that will be treated as missing or invalid. Non-numeric missing value flags are applied to all variables in the data set.

EXAMPLE 13.5: NUMERIC MISSING VALUE FLAGS

```
TITLE:     this is an example of a SEM with
           continuous factor indicators using data
           with numeric missing value flags
DATA:      FILE IS ex5.11.dat;
VARIABLE:  NAMES ARE y1-y12;
           MISSING = y1-y3(9) y4(9 99) y5-y12(9-12);
MODEL:     f1 BY y1-y3;
           f2 BY y4-y6;
           f3 BY y7-y9;
           f4 BY y10-y12;
           f4 ON f3;
           f3 ON f1 f2;
```

The example above is based on Example 5.11 in which the data contain no missing values. In this example, there are missing values and numeric missing value flags are used. The MISSING option is used to identify the values or symbol in the analysis data set that will be treated as missing or invalid. Numeric missing value flags can be applied to a single variable, to groups of variables, or to all of the variables in a data set. In the example above, y1, y2, and y3 have a missing value flag of 9; y4 has missing value flags of 9 and 99; and y5 through y12 have missing value flags of 9, 10, 11, and 12. If all variables in a data set have the same missing value flags, the keyword ALL can be used as follows:

MISSING = ALL (9);

to indicate that all variables have the missing value flag of 9.

EXAMPLE 13.6: SELECTING OBSERVATIONS AND VARIABLES

```
TITLE:     this is an example of a path analysis
           with continuous dependent variables using
           a subset of the data
DATA:      FILE IS ex3.11.dat;
VARIABLE:  NAMES ARE y1-y6 x1-x4;
           USEVARIABLES ARE y1-y3 x1-x3;
           USEOBSERVATION ARE (x4 EQ 2);
MODEL:     y1 y2 ON x1 x2 x3;
           y3 ON y1 y2 x2;
```

The example above is based on Example 3.11 in which the entire data set is analyzed. In this example, a subset of variables and a subset of observations are analyzed. The USEVARIABLES option is used to select variables for an analysis. In the example above, y1, y2, y3, x1, x2, and x3 are selected. The USEOBSERVATIONS option is used to select observations for an analysis by specifying a conditional statement. In the example above, individuals with the value of 2 on variable x4 are included in the analysis.

EXAMPLE 13.7: TRANSFORMING VARIABLES USING THE DEFINE COMMAND

```
TITLE:     this is an example of a path analysis
           with continuous dependent variables where
           two variables are transformed
DATA:      FILE IS ex3.11.dat;
DEFINE:    y1 = y1/100;
           x3 = SQRT(x3);
VARIABLE:  NAMES ARE y1-y6 x1-x4;
           USEVARIABLES = y1-y3 x1-x3;
MODEL:     y1 y2 ON x1 x2 x3;
           y3 ON y1 y2 x2;
```

The example above is based on Example 3.11 where the variables are not transformed. In this example, two variables are transformed using the DEFINE command. The variable y1 is transformed by dividing it by 100. The variable x3 is transformed by taking the square root of it. The transformed variables are used in the estimation of the model. The DEFINE command can also be used to create new variables.

EXAMPLE 13.8: FREEING AND FIXING PARAMETERS AND GIVING STARTING VALUES

```
TITLE:     this is an example of a CFA with
           continuous factor indicators where
           parameters are freed, fixed, and starting
           values are given
DATA:      FILE IS ex5.1.dat;
VARIABLE:  NAMES ARE y1-y6;
MODEL:     f1 BY y1* y2*.5 y3;
           f2 BY y4* y5 y6*.8;
           f1-f2@1;
```

The example above is based on Example 5.1 where default starting values are used. In this example, parameters are freed, assigned starting values, and fixed. In the two BY statements, the factor loadings for y1 and y4 are fixed at one as the default because they are the first variable following the BY statement. This is done to set the metric of the factors. To free these parameters, an asterisk (*) is placed after y1 and y4. The factor loadings for variables y2, y3, y5, and y6 are free as the default with starting values of one. To assign starting values to y2 and y6, an asterisk (*) followed by a number is placed after y2 and y6. The starting value of .5 is assigned to y2, and the starting value of .8 is assigned to y6. The variances of f1 and f2 are free to be estimated as the default. To fix these variances to one, an @ symbol followed by 1 is placed after f1 and f2 in a list statement. This is another way to set the metric of the factors.

EXAMPLE 13.9: EQUALITIES IN A SINGLE GROUP ANALYSIS

```
TITLE:     this is an example of a CFA with
           continuous factor indicators with
           equalities
DATA:      FILE IS ex5.1.dat;
VARIABLE:  NAMES ARE y1-y6;
MODEL:     f1 BY y1
                 y2-y3 (1-2);
           f2 BY y4
                 y5-y6 (1-2);
           y1-y3 (3);
           y4-y6 (4);
```

This example is based on the model in Example 5.1 where there are no equality constraints on model parameters. In the example above, several model parameters are constrained to be equal. Equality constraints are specified by placing the same number in parentheses following the parameters that are to be held equal. The label (1-2) following the factor loadings uses the list function to assign equality labels to these parameters. The label 1 is assigned to the factor loadings of y2 and y5 which holds these factor loadings equal. The label 2 is assigned to the factor loadings of y3 and y6 which holds these factor loadings equal. The third equality statement holds the residual variances of y1, y2, and y3 equal using the label (3), and the fourth equality statement holds the residual variances of y4, y5, and y6 equal using the label (4).

EXAMPLE 13.10: EQUALITIES IN A MULTIPLE GROUP ANALYSIS

```
TITLE:    this is an example of a multiple group CFA
          with covariates (MIMIC) with continuous
          factor indicators and a mean structure
          with between and within group equalities
DATA:     FILE IS ex5.15.dat;
VARIABLE: NAMES ARE y1-y6 x1-x3 g;
          GROUPING IS g (1=g1 2=g2 3=g3);
MODEL:    f1 BY y1-y3;
          f2 BY y4-y6;
          f1 f2 ON x1-x3;
          f1 (1);
          y1-y3 (2);
          y4-y6 (3-5);
MODEL g1: f1 BY y3*;
          [y3*];
          f2 (6);
MODEL g3: f2 (6);
```

This example is based on Example 5.15 in which the model has two groups. In this example, the model has three groups. Parameters are constrained to be equal by placing the same number in parentheses following the parameters that will be held equal. In multiple group analysis, the overall MODEL command is used to set equalities across groups. The group-specific MODEL commands are used to specify equalities for specific groups or to relax equalities specified in the overall MODEL command. In the example above, the first equality statement holds the variance of f1 equal across the three groups in the analysis using the equality label 1. The second equality statement holds the residual variances of y1, y2, and y3 equal to each other and equal across groups using the equality label 2. The third equality statement uses the list function to hold the residual variance of y4, y5, and y6 equal across groups by assigning the equality label 3 to the residual variance of y4, the label 4 to the residual variance of y5, and the label 5 to the residual variance of y6. The fourth and fifth equality statements hold the variance of f2 equal across groups g1 and g3 using the equality label 6.

EXAMPLE 13.11: USING PWITH TO ESTIMATE ADJACENT RESIDUAL COVARIANCES

```
TITLE:     this is an example of a linear growth
           model for a continuous outcome with
           adjacent residual covariances
DATA:      FILE IS ex6.1.dat;
VARIABLE:  NAMES ARE y11-y14 x1 x2 x31-x34;
           USEVARIABLES ARE y11-y14;
MODEL:     i s | y11@0 y12@1 y13@2 y14@3;
           y11-y13 PWITH y12-y14;
```

The example above is based on Example 6.1 in which a linear growth model with no residual covariances for the outcome is estimated. In this example, the PWITH option is used to specify adjacent residual covariances. The PWITH option pairs the variables on the left-hand side of the PWITH statement with the variables on the right-hand side of the PWITH statement. Residual covariances are estimated for the pairs of variables. In the example above, residual covariances are estimated for y11 with y12, y12 with y13, and y13 with y14.

EXAMPLE 13.12: CHI-SQUARE DIFFERENCE TESTING FOR WLSMV AND MLMV

This example shows the two steps needed to do a chi-square difference test using the WLSMV and MLMV estimators. For these estimators, the conventional approach of taking the difference between the chi-square values and the difference in the degrees of freedom is not appropriate because the chi-square difference is not distributed as chi-square. This example is based on Example 5.3.

```
TITLE:     this is an example of the first step
           needed for a chi-square difference test
           for the WLSMV or the MLMV estimator
DATA:      FILE IS ex5.3.dat;
VARIABLE:  NAMES ARE u1-u3 y4-y9;
           CATEGORICAL ARE u1 u2 u3;
MODEL:     f1 BY u1-u3;
           f2 BY y4-y6;
           f3 BY y7-y9;
SAVEDATA:  DIFFTEST IS deriv.dat;
```

The input setup above shows the first step needed to do a chi-square difference test for the WLSMV and MLMV estimators. In this analysis, the less restrictive H1 model is estimated. The DIFFTEST option of the SAVEDATA command is used to save the derivatives of the H1 model for use in the second step of the analysis. The DIFFTEST option is used to specify the name of the file in which the derivatives from the H1 model will be saved. In the example above, the file name is deriv.dat.

```
TITLE:     this is an example of the second step
           needed for a chi-square difference test
           for the WLSMV or the MLMV estimator
DATA:      FILE IS ex5.3.dat;
VARIABLE:  NAMES ARE u1-u3 y4-y9;
           CATEGORICAL ARE u1 u2 u3;
ANALYSIS:  DIFFTEST IS deriv.dat;
MODEL:     f1 BY u1-u3;
           f2 BY y4-y6;
           f3 BY y7-y9;
           f1 WITH f2-f3@0;
           f2 WITH f3@0;
```

The input setup above shows the second step needed to do a chi-square difference test for the WLSMV and MLMV estimators. In this analysis, the more restrictive H0 model is estimated. The restriction is that the covariances among the factors are fixed at zero in this model. The DIFFTEST option of the ANALYSIS command is used to specify the name of the file that contains the derivatives of the H1 model that was estimated in the first step of the analysis. This file is deriv.dat.

EXAMPLE 13.13: ANALYZING MULTIPLE IMPUTATION DATA SETS

```
TITLE:     this is an example of a CFA with
           continuous factor indicators using
           multiple imputation data sets
DATA:      FILE IS implist.dat;
           TYPE = IMPUTATION;
VARIABLE:  NAMES ARE y1-y6;
MODEL:     f1 BY y1-y3;
           f2 BY y4-y6;
```

The example above is based on Example 5.1 in which a single data set is analyzed. In this example, data sets generated using multiple imputation are analyzed. The FILE option of the DATA command is used to give the name of the file that contains the names of the multiple imputation data sets to be analyzed. The file named using the FILE option of the DATA command must contain a list of the names of the multiple imputation data sets to be analyzed. This file must be created by the user unless the data are imputed using the DATA IMPUTATION command in which case the file is created as part of the multiple imputation. Each record of the file must contain one data set name. For example, if five data sets are being analyzed, the contents of implist.dat would be:

```
imp1.dat
imp2.dat
imp3.dat
imp4.dat
imp5.dat
```

where imp1.dat, imp2.dat, imp3.dat, imp4.dat, and imp5.dat are the names of the five data sets created using multiple imputation.

When TYPE=IMPUTATION is specified, an analysis is carried out for each data set in the file named using the FILE option. Parameter estimates are averaged over the set of analyses, and standard errors are computed using the average of the standard errors over the set of analyses and the between analysis parameter estimate variation (Schafer, 1997).

EXAMPLE 13.14: SAVING DATA

```
TITLE:     this is an example of a path analysis
           with continuous dependent variables using
           a subset of the data which is saved for
           future analysis
DATA:      FILE IS ex3.11.dat;
VARIABLE:  NAMES ARE y1-y6 x1-x4;
           USEOBSERVATION ARE (x4 EQ 2);
           USEVARIABLES ARE y1-y3 x1-x3;
MODEL:     y1 y2 ON x1 x2 x3;
           y3 ON y1 y2 x2;
SAVEDATA:  FILE IS regress.sav;
```

The example above is based on Example 3.11 in which the analysis data are not saved. In this example, the SAVEDATA command is used to save the analysis data set. The FILE option is used to specify the name of the ASCII file in which the individual data used in the analysis will be saved. In this example, the data will be saved in the file regress.sav. The data are saved in fixed format as the default unless the FORMAT option of the SAVEDATA command is used.

EXAMPLE 13.15: SAVING FACTOR SCORES

```
TITLE:     this is an example of a CFA with
           covariates (MIMIC) with continuous factor
           indicators where factor scores are
           estimated and saved
DATA:      FILE IS ex5.8.dat;
VARIABLE:  NAMES ARE y1-y6 x1-x3;
MODEL:     f1 BY y1-y3;
           f2 BY y4-y6;
           f1 f2 ON x1-x3;
SAVEDATA:  FILE IS mimic.sav; SAVE = FSCORES;
```

The example above is based on Example 5.8 in which factor scores are not saved. In this example, the SAVEDATA command is used to save the analysis data set and factor scores. The FILE option is used to specify the name of the ASCII file in which the individual data used in the analysis will be saved. In this example, the data will be saved in the file mimic.sav. The SAVE option is used to specify that factor scores will be saved along with the analysis data. The data are saved in fixed format as the default unless the FORMAT option of the SAVEDATA command is used.

EXAMPLE 13.16: USING THE PLOT COMMAND

```
TITLE:     this is an example of a linear growth
           model for a continuous outcome
DATA:      FILE IS ex6.1.dat;
VARIABLE:  NAMES ARE y11-y14 x1 x2 x31-x34;
           USEVARIABLES ARE y11-y14;
MODEL:     i s | y11@0 y12@1 y13@2 y14@3;
PLOT:      SERIES = y11-y14 (s);
           TYPE = PLOT3;
```

The example above is based on Example 6.1 in which no graphical displays of observed data or analysis results are requested. In this example, the PLOT command is used to request graphical displays of observed data and analysis results. These graphical outputs can be viewed after the Mplus analysis is completed using a post-processing graphics module. The SERIES option is used to list the names of a set of variables along with information about the x-axis values to be used in the graphs. For growth models, the set of variables is the repeated measures of the outcome over time, and the x-axis values are the time scores in the growth model. In the example above, the s in parentheses after the variables listed in the SERIES statement is the name of the slope growth factor. This specifies that the x-axis values are the time scores values specified in the growth model. In this example, they are 0, 1, 2, and 3. Other ways to specify x-axis values are described in Chapter 18. The TYPE option is used to request specific plots. The TYPE option of the PLOT command is described in Chapter 18.

EXAMPLE 13.17: MERGING DATA SETS

```
TITLE:     this is an example of merging two data
           sets
DATA:      FILE IS data1.dat;
VARIABLE:  NAMES ARE id y1-y4;
           IDVARIABLE IS id;
           USEVARIABLES = y1 y2;
           MISSING IS *;
ANALYSIS:  TYPE = BASIC;
SAVEDATA:  MFILE = data2.dat;
           MNAMES ARE id y5-y8;
           MFORMAT IS F6 4F2;
           MSELECT ARE y5 y8;
           MMISSING = y5-y8 (99);
           FILE IS data12.sav;
           FORMAT IS FREE;
           MISSFLAG = 999;
```

This example shows how to merge two data sets using TYPE=BASIC. Merging can be done with any analysis type. The first data set data1.dat is named using the FILE option of the DATA command. The second data set data2.dat is named using the MFILE option of the SAVEDATA command. The NAMES option of the VARIABLE command gives the names of the variables in data1.dat. The MNAMES option of the SAVEDATA command gives the names of the variables in data2.dat. The IDVARIABLE option of the VARIABLE command gives the name of the variable to be used for merging. This variable must appear on both the NAMES and MNAMES statements. The merged data set data12.dat is saved in the file named using the FILE option of the SAVEDATA command. The default format for this file is free and the default missing value flag is the asterisk (*). These defaults can be changed using the FORMAT and MISSFLAG options as shown above. In the merged data set data12.dat, the missing value flags of asterisk (*) in data1.dat and 99 in data2.dat are replaced by 999.

For data1.dat, the USEVARIABLES option of the VARIABLE command is used to select a subset of the variables to be in the analysis and for merging. The MISSING option of the VARIABLE command is used to identify the values or symbol in the data set that are treated as missing or invalid. In data1.dat, the asterisk (*) is the missing value flag. If the data are not in free format, the FORMAT statement can be used to specify a fixed format.

For data2.dat, the MFORMAT option is used to specify a format if the data are not in the default free format. The MSELECT option is used to select a subset of the variables to be used for merging. The MMISSING option is used to identify the values or symbol in the data set that are treated as missing or invalid.

EXAMPLE 13.18: USING REPLICATE WEIGHTS

```
TITLE:     this is an example of using replicate
           weights
DATA:      FILE IS rweights.dat;
VARIABLE:  NAMES ARE y1-y4 weight r1-r80;
           WEIGHT = weight;
           REPWEIGHTS = r1-r80;
ANALYSIS:  TYPE = COMPLEX;
           REPSE = JACKKNIFE1;
MODEL:     f BY y1-y4;
```

This example shows how to use replicate weights in a factor analysis. Replicate weights summarize information about a complex sampling design. The WEIGHT option must be used when the REPWEIGHTS option is used. The WEIGHT option is used to identify the variable that contains sampling weight information. In this example, the sampling weight variable is weight. The REPWEIGHTS option is used to identify the replicate weight variables. These variables are used in the estimation of standard errors of parameter estimates (Asparouhov & Muthén, 2009b). The data set in this example contains 80 replicate weights variables, r1 through r80. The STRATIFICATION and CLUSTER options may not be used in conjunction with the REPWEIGHTS option. Analysis using replicate weights is available only with TYPE=COMPLEX. The REPSE option is used to specify the resampling method that was used to create the replicate weights. The setting JACKKNIFE1 specifies that Jackknife draws were used.

EXAMPLE 13.19: GENERATING, USING, AND SAVING REPLICATE WEIGHTS

```
TITLE:     this is an example of generating, using,
           and saving replicate weights
DATA:      FILE IS ex13.19.dat;
VARIABLE:  NAMES ARE y1-y4 weight strat psu;
           WEIGHT = weight;
           STRATIFICATION = strat;
           CLUSTER = psu;
ANALYSIS:  TYPE = COMPLEX;
           REPSE = BOOTSTRAP;
           BOOTSTRAP = 100;
MODEL:     f BY y1-y4;
SAVEDATA:  FILE IS rweights.sav;
           SAVE = REPWEIGHTS;
```

This example shows how to generate, use, and save replicate weights in a factor analysis. Replicate weights summarize information about a complex sampling design (Korn & Graubard, 1999; Lohr, 1999; Asparouhov & Muthén, 2009b). When replicate weights are generated, the REPSE option of the ANALYSIS command and the WEIGHT option of the VARIABLE command along with the STRATIFICATION and/or CLUSTER options of the VARIABLE command are used. The WEIGHT option is used to identify the variable that contains sampling weight information. In this example, the sampling weight variable is weight. The STRATIFICATION option is used to identify the variable in the data set that contains information about the subpopulations from which independent probability samples are drawn. In this example, the variable is strat. The CLUSTER option is used to identify the variable in the data set that contains clustering information. In this example, the variable is psu. Replicate weights can be generated and analyzed only with TYPE=COMPLEX. The REPSE option is used to specify the resampling method that will be used to create the replicate weights. The setting BOOTSTRAP specifies that bootstrap draws will be used. The BOOTSTRAP option specifies that 100 bootstrap draws will be carried out. When replicate weights are generated, they can be saved for further analysis using the FILE and SAVE options of the SAVEDATA command. Replicate weights will be saved along with the other analysis variables in the file named rweights.sav.