Data Set Preparation for Mplus

Mplus reads only ASCII (text) data. This is inconvenient, but I'm going to give you some good tips and examples on how to prepare data sets for Mplus. There are several points that are helpful to keep in mind to avoid problems reading data out of your statistical package and into Mplus. It is possible to create a data set that is being incorrectly read and never know it! So, you need to be extra careful.

1. Remember the exact location of the data file, including the full path specification. I always keep folder names to a minimum, so that minor errors typing complex file paths don't trip me up on the FILE statement in Mplus. If you store your input (syntax) files in the same folder as the data file (.dat), no file path specification is needed in the Mplus input statements, you just need to refer to the file name.

2. Make sure that the list of the variables that you read out of your statistical package (e.g., SPSS) exactly matches the list of variables in the same order you give Mplus in the DATA section. Always double check that the lists of variables you read out and in are both complete and have the variables appearing in the same order.

3. If using free format for input, which I recommend because it is the simplest, you need numerical values (asterisk or period are also acceptable) to designate missing data for all variables. This means that system missing values must be given a discrete missing value code in your statistical package before saving out the data. Unless you recode missing values to some discrete number (e.g., -99) of symbol (e.g., *), SPSS prints spaces, which will pose problems for free format reading in Mplus.

4. It is good practice to open your data set to look at it to make sure it looks ok. You can open the file in any text editor, but I often just open it in Mplus (by changing the extension drop down box to *.*).

5. Finally, double check that the descriptive statistics in SPSS or another source program match those in Mplus ***exactly***. To correctly compare them, you need to use listwise deletion[1] in the source package (e.g., SPSS) to get the means for all of the variables and then use `type=basic` on the analysis command in Mplus together with `listwise=on` under the data command.[2] Make sure the N and the mean values match to at least 3 decimals.

SPSS

In SPSS, there are two ways to create the raw data files—through the menus and with syntax. My preferred method is with syntax, because a) I like to keep a record of the data files I created, 2) I can double check the variable list to verify the variables in my model are the ones I intended to use, and 3) I can copy the variable names into the Mplus program file to save time and avoid typos, incorrectly ordered variable lists, or omitted variables.

SPSS Using Menus.

1. file -> save as (specify location and filename and uncheck the "write variable names to spreadsheet" checkbox). Make sure that under "Save as type," you choose "Tab-delimited (*.dat)".

[1] If you want to create a data set in SPSS no missing values, you can use two lines of code for listwise deletion: `count nummiss=x1 x2 x3 (MISSING).` and `SELECT IF nummiss EQ 0`. Otherwise, give your missing values discrete codes (e.g., -99) and use the subcommand `/MISSING = LISTWISE.` subcommand on your `DESCRIPTIVES` procedure.

[2] SPSS now annoyingly chooses the number of decimals in the output for you. For some statistics such as the mean, you may only get two decimals, which makes it difficult to match against the three decimal places in Mplus. There is no simple way to change this for all output jobs apparently. To fix the problem for a particular descriptives output pivot table, however, select the cell values in the Mean column, right click, and choose Cell Properties. At the bottom of the dialogue box, you can increase the number of decimals to 3.

2. Click the “variables” button and check the boxes next to the variables you wish to save out. Note that it is often convenient to first click the “drop all” button and then check the subset of variables that you desire, especially when working with a large data set.

3. Click “continue”.

4. Click “save”.

SPSS Using Syntax. The following syntax lines can be used to save the data as tab-delimited text. I use just a simple example with four variables here. The DESCRIPTIVES command issued afterwards is helpful for double checking that the N is the same as that used in Mplus. The MISSING=LISTWISE command is used to check the N in Mplus when listwise deletion is used (discussed later). If a DESCIPTIVES or other command is not used, an EXECUTE statement is needed following the SAVE command.

```
RECODE program TO b3p_conf (SYSMIS=-99).

MISSING VALUES program T0 b3p_conf (-99,-6 thru -1).

SAVE TRANSLATE OUTFILE='c:\jason\mplus\consult\ehs\temp.dat'
 /TYPE=TAB /MAP /REPLACE
 /KEEP=b1p_cesd b1v3pdet b1v3pint b1v3pneg .

DESCRIPTIVES VARS=b1p_cesd b1v3pdet b1v3pint b1v3pneg
  /MISSING=LISTWISE.
```

SAS Using Syntax. The code below illustrates one method of saving a raw data file in SAS. Note that the RETAIN statement between the DATA and SET statements is needed to keep the order of the variables (the KEEP command is not sufficient to retain the order). I recode missing values here to -99, but the default period is acceptable in Mplus as long as you declare the period symbol as a missing value).

```
DATA two (KEEP=female norms subj pdesc pinj sdesc sinj intention champs);
retain female norms subj pdesc pinj sdesc sinj intention champs;
SET one;
ARRAY miss{*} female norms subj pdesc pinj sdesc sinj intention champs;
DO i =1 TO DIM(miss);
IF miss{i} = . THEN miss{i} = -99;
END; DROP i;
RUN;

PROC EXPORT DATA=two
OUTFILE='c:\jason\mplus\shaw\pretest\norms3.dat\'
 DBMS=DLM REPLACE;
 putnames=no;
run;

*listwise deletion routine for checking means and SDs against Mplus;
data three; set two;
array miss{*} female norms subj pdesc pinj sdesc sinj intention champs;
do i = 1 to dim(miss);
if miss{i} = -99 then miss{i} = .;
if miss{i} ne .;
end; drop i;

PROC MEANS;
VAR female norms subj pdesc pinj sdesc sinj intention champs;
RUN;
```

R Code

If you are using R, you will likely use the `lavaan` package instead of Mplus. But if you need to save a raw data file to do so, use the `write.table` function as tab delimited, as in

```
write.table(mydata, file = "c:/jason/R/semclass/nphs 1994.dat", sep = "\t", row.names=FALSE,
col.names=FALSE)
```

Example 1: Reading Data into Mplus

In Example 1 below, I read the data I created with the above examples into Mplus.

If you have user-defined missing values, you can identify those in Mplus with the MISSING statement in the VARIABLE section. The following are acceptable: MISSING = *; MISSING = . ; MISSING = BLANK; MISSING = varname(#); In the last example, “varname” is any variable name and # is the value in the data set that indicates missing where you can specify multiple discrete values or a range of values.

If using free format for input, as I illustrate below, you cannot use blanks to represent missing data in SPSS. An asterisk, period, or numerical value must be used.

The TYPE=BASIC command is not required, but generates some descriptive data useful for verifying that you have read the data correctly.

```
TITLE: Example 1, Reading in raw data;

DATA: FILE=ex1.dat;
    FORMAT=FREE;

!You can specify a list of missing values separated by commas or use a dash to separate
values from low to high, here below, -99 and -6 through -1;

VARIABLE: NAMES = b1p_cesd b1v3pdet b1v3pint b1v3pneg;
       MISSING = b1p_cesd-b1v3pneg(-99,-6--1);

! The following TYPE=BASIC command gives descriptive data
!  and is a good idea for checking to make sure the data
!  are read in correctly.

ANALYSIS: TYPE=BASIC;
```

The path on the FILE statement can be abbreviated (e.g., FILE=ex1.dat) as long as the data file resides in the same folder as the input file. Otherwise, the full path is required (e.g., c:\jason\mplus\ehs\ex2.dat).

CAUTION: Always, always, double check that you are reading the data correctly into Mplus by comparing descriptive statistics for all variables in the original statistical program to the values obtained in Mplus (using the TYPE=BASIC). It is possible (and I have done this) to proceed without ever noticing that there is an error. Make certain that the N is the same in both packages to make the values match. The values should match exactly to as many decimals as are printed in the output. If there are missing data, the best way to compare the values is to use listwise deletion in both programs so that you have precisely the same N.