LI Analysis Training Series

Merging Files: Adding Variables or Cases

(Last Revised: 1/10/2007)
Melissa Ives, Rodney Funk & Michael Dennis
Chestnut Health Systems
Bloomington IL 61701
309-827-6026
www.chestnut.org

Acknowledgement: This document was developed under contract #270-2003-00006 from the Substance Abuse and Mental Health Services Administration (SAMHSA's) Center for Substance Abuse Treatment (CSAT). Any opinions about this data are those of the authors and do not represent official positions of the government or individual grantees.

Purpose: To show by example how to add new variables or new records to an existing datafile and check the results.

Data Requirements: If adding variables, data in both files must be sorted in the same order. If adding records, identical variables in both files must have the same format.

Procedure: Match Files/Add Files. The procedures listed here (and additional variables) can be found on pages 84-90 and 509-516 in the *SPSS Syntax Reference Guide Release 7.5* or in the online help under "Add Cases" for Add Files or "Add Variables" for Match files. The procedures listed here are in many ways easier to understand and shorter if typed directly into a syntax file. This is especially true if you wish to drop or keep only a few variables from the data set. To use the dialog box for either one choose from the menus:

Data
  Merge Files >
Then choose one of the following:
        Add Variables
        Add Cases

Adding Variables: (Match Files) Use this choice when two or more files contain different variables for the same record. Each file must contain a variable that uniquely identifies each record. Each file must be sorted by the identifying variable. If a data file does not contain an identifying variable, it will be assumed that the two files are sorted in the same order, and the first record of the first file will be matched with the first record of the second file etc. Open the first file you wish to include, then select the dialog choices listed above ending with Add Variables. Please note that the "Data" option is only available from the Data Editor screen. This will bring up a new dialog box "Add Variable: Read File". Select the file you wish to match with the first file and click

"Open". In this example, data from Example1.sav will be matched with data from Example2.sav.

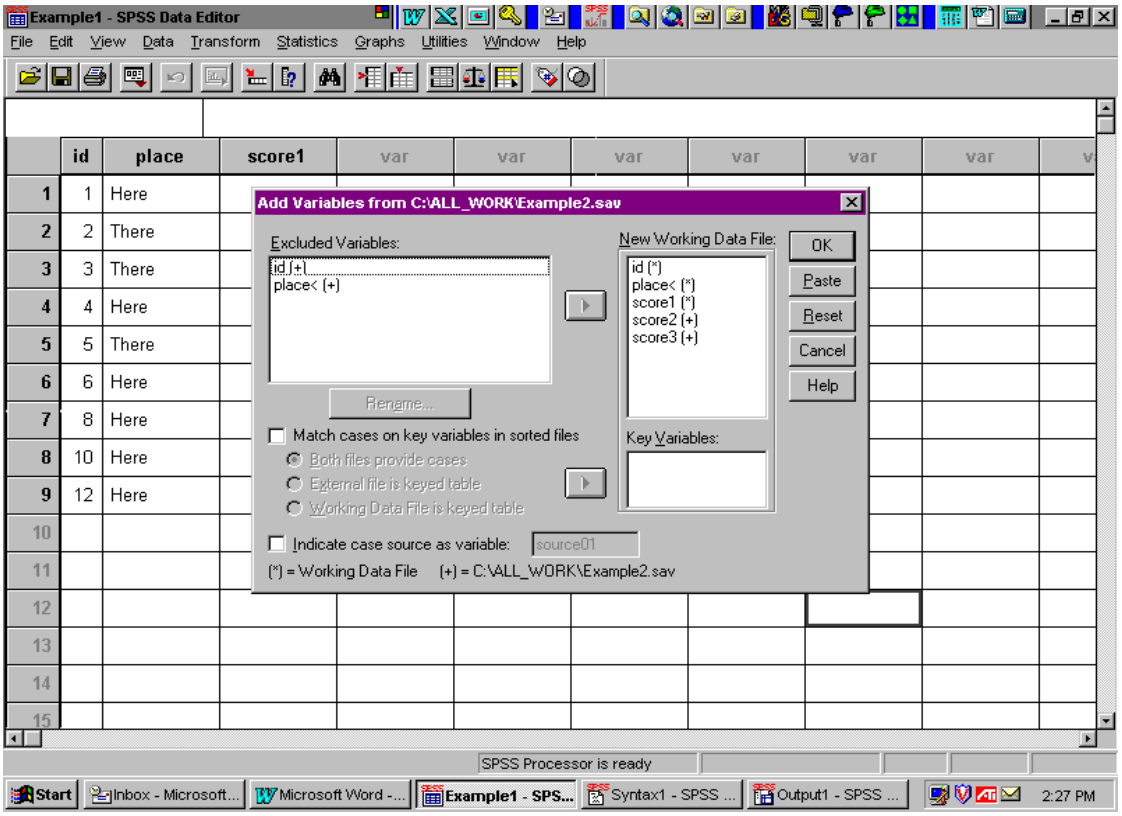


Figure 1.

A new dialog box appears (Figure 1) with an unduplicated list of variables in the box labeled "New Working Data File". Variables from either file may be removed from the list or added to the list. This is done by clicking on the variable name in either the "New Working Data File" box or in the "Excluded Variable" box and then on the arrow between the two boxes. Variables from the working data file (the first file opened) are indicated with and asterisk (*). Variables from the second file are indicated with a plus sign (+).

To assure that each record in the first file is matched with its corresponding record in subsequent files, click the check box labeled "Match cases on keyed variables in sorted files". Then select the identifying variable and move it to the "Key Variables" box by clicking on the arrow next to this box. It is possible to have several key variables. Key variables must uniquely identify a record, and the data file must be sorted by each key variable in the order it is included.

To create a variable that indicates whether the second file contributed to the matched file, check the box "Indicate case source as variable" and type in a logical variable name. If the file contributed values to the match, the value of this variable will be 1, if not, the value of this variable will be 0.

It is possible to choose OK here and run the MATCH FILES command. However, should you later discover that you need to re-run the same match, it will be easier if you have the syntax saved. In addition, there are some extra features you can add or change if you use a syntax file. Click "Paste" to place the corresponding syntax into file. SPSS will warn you that the match will fail if data are not sorted in ascending order of the keyed variable. Click "OK".

Here is the pasted syntax from this example. (The file 'C:\ALL_WORK\Example1.sav' was already opened and is represented by the * in the first line.)

```
MATCH FILES /FILE=*
 /FILE='C:\ALL_WORK\Example2.sav'
 /RENAME (place = d0)
 /IN=inEx2
 /BY id
 /DROP= d0.
VARIABLE LABELS inEx2
 'Case source is C:\ALL_WORK\Example2.sav'.
EXECUTE.
```

Additional notes:

You may wish to add a source variable for the first file. To do this, add a line before the second /FILE subcommand that reads as follows (replace varname with a logical variable name.)

```
 /IN=varname.
```

It is not absolutely necessary to include the /RENAME= and /DROP= subcommands for this example. All this does is change the name of an excluded variable to a numbered variable, and then tell SPSS not to include this variable in the new working file (the matched file). It does however explicitly account for every variable in each file and avoids the possibility of keeping or dropping the wrong variable.

Use the /RENAME= subcommand if variables in each file are named the same, but represent different things, or if two variables are not named the same but do represent the same thing. The /RENAME subcommand follows the /FILE= subcommand for the file that contains the variable(s) that need to be renamed. In the example above, the /RENAME= subcommand changes the name of the place variable to d0. If renaming multiple variables, list the current names in order to the left of the equals sign and the corresponding new names IN ORDER to the right of the equals sign. For example in the statement:

```
/RENAME (place time=where when)
```

The variable 'place' will be renamed to 'where', while the variable 'time' will be renamed to 'when'.

If you wish to re-order the variables in the matched file, do not use the /DROP= subcommand, instead, replace /DROP=varlist with /KEEP=varlist (replacing varlist with the list of variable in the order you desire in the matched file. /KEEP= may be used to keep only a few selected variables. Determining whether to use /DROP= or /KEEP= depends primarily upon how many variables you wish to drop or keep, or whether you also need to re-order the variables.

If the values of the same variable in the two files are different, SPSS retains the value from the first file listed in the MATCH FILES command in which the variable is not missing. (In this example, that would be the file named Example1.sav)

It is possible to match several files at one time. Simply add additional /FILE= subcommands (and any desired /IN= or /RENAME= subcommands.) for each file. Each file will be added in the order in which it appears. Each file should be sorted by the key variable or if a key variable is not used, in the same order as the other files.

The variable label created by the dialog box for the /IN= variable is often overly long. Change it to something shorter or clearer if you desire. If you added an /IN=variable command for additional files, you may wish to add variable label(s) for the new variable(s) here. Use the menu choices Data>Definitions>Labels or type the syntax command:

```
VARIABLE LABELS varname  'type label here'
                varname2 'type second label here'.
```

SPSS will replace the current working file with an untitled new working file. Match files must be executed by using the EXECUTE command or by using another procedure command (frequencies, crosstabs, means etc.) that causes a data pass (the computer looks at each record). Be sure to save this file with a new name if you will need to use it again.

If a record does not exist in one of the files, (see records 7, 9 and 11 below) the values for variables included in that file will be set to system missing, and any /IN variable will have a value of 0. Note that for these records, the value of 'Place' exists because it was not missing in Example2.sav. However, Score1 is missing because that value is unique to Example1.sav.

Here is the syntax as edited:

```
MATCH FILES /FILE=*
 /In=inEx1
 /FILE='C:\ALL_WORK\Example2.sav'
 /IN=inEx2
 /BY id.
VARIABLE LABELS
 inEx1 'In Example1'
```

```
 inEx2 'In Example2'.
EXECUTE.
```

Results of matching Example1.sav and Example2.sav (Saved as Examp1_2.sav)

ID	Place	Score1	Score2	Score3	inEx1	inEx2
1	Here	5.90	2.30	1.50	1	1
2	There	2.60	3.33	2.40	1	1
3	There	4.98	1.40	1.42	1	1
4	Here	.	.	2.50	1	1
5	There	7.21	5.12	.	1	1
6	Here	8.45	4.10	4.20	1	1
7	Here	.	2.25	3.00	0	1
8	Here	3.51	1.23	2.00	1	1
9	There	.	5.30	3.50	0	1
10	Here	9.10	.	4.12	1	1
11	There	.	3.27	2.70	0	1
12	Here	6.00	3.81	3.22	1	1

Example 2: TABLE

It is possible to match cases where one file contains one record per case and the other file (or files) contains multiple records per case. To do this, the file with one record per case is indicated as a TABLE rather than a file. (In the syntax, this would be /TABLE='filename'. Matching in this way assigns each variable from the TABLE file to each record in the multiple record file. If a record does not exist in the TABLE file, records from the multiple records file will contain system-missing values for the variables from the TABLE file. If a record exists in the TABLE file, but not in the multiple record file, the cases from the TABLE file are ignored and not retained in the merged file. For example, to add type of treatment site to client records:

Treatment TABLE

Tx site	TxType
1	A
2	A
3	B
4	C

Client Data

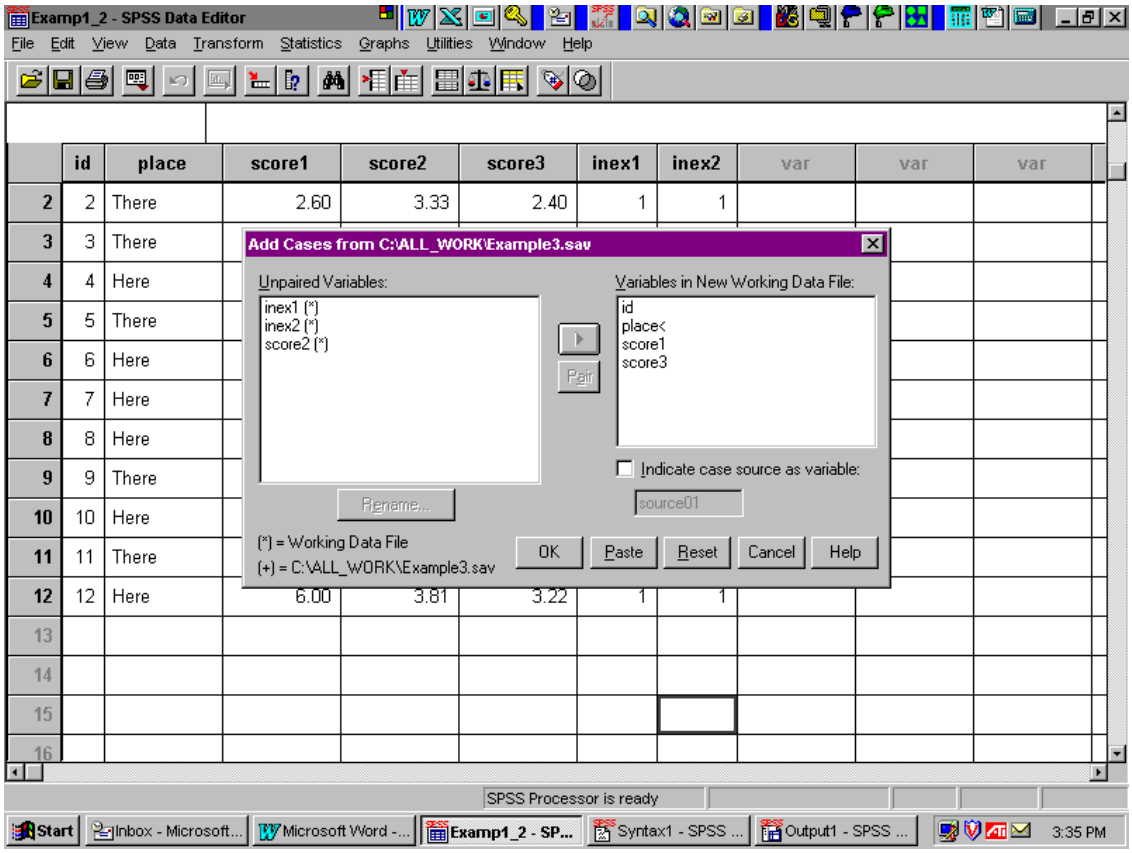
Client	Tx Site	Age	Gender
100	1	15	M
101	1	14	M
102	1	17	M
201	2	13	F
202	2	15	F
401	4	16	M
402	4	15	F
501	5	14	M

Merged file with Client data first

Client	TxSite	Age	Gender	TxType
100	1	15	M	A
101	1	14	M	A
102	1	17	M	A
202	2	15	F	A
201	2	13	F	A
401	4	16	M	B
402	4	15	F	B
501	5	14	M	.

Adding Records (Cases): (Add Files). Use this choice when two or more files contain different records with the same variables. The example used here will match the new file created above (Examp1_2.sav) with Example3.sav. For the ADD FILES command, it is not necessary to sort the data. This command begins in a similar fashion to the MATCH FILES command. Open the first file you wish to include, then select the dialog choices listed above ending with Add Cases. Please note that the "Data" option is only available from the Data Editor screen. This will bring up a new dialog box "Add Cases: Read File". Select the file you wish to match with the first file and click "Open".

Figure 2.



This will bring up a dialog box that is similar but not identical to the MATCH FILES dialog box. Variables from either file may be removed from the list or added to the list. This is done by clicking on the variable name in either the "New Working Data File" box or in the "Excluded Variable" box and then on the arrow between the two boxes. Variables from the working data file (the first file opened) are indicated with and asterisk (*). Variables from the second file are indicated with a plus sign (+).

In this case, we wish to retain the scores in numerical order. To do this, click on score3 under "Variables in the New Working Data File". Then click the left arrow to move it back to Unpaired variables. Note that in doing this, score3 is listed as 2 variables: one from the working file, and one from the file to be added. Click on Score2 and move it to "Variables in the New Working Data File". Score2 only exists in Exam1_2.sav, and not in Example3.sav, so it is listed only one time.

To move score3 back to the New Working Data File, it will be necessary to re-pair the variables. This procedure also works if the same variable has a different in each file. Just select the two that match and click on 'Pair'. This will associate these variables with each other and move the variable to the new working file. If the names are different, the name in the working file will be retained. (If you wish to change this name, you may do so using the syntax. See below.)

To create a variable that indicates whether the second file contributed to the matched file, check the box "Indicate case source as variable" and type in a logical variable name. If the file contributed values to the match, the value of this variable will be 1, if not, the value of this variable will be 0.

In this example, I also included the /IN= variables created by the match file by moving them from 'Unpaired Variables" to "Variables in the New Working File". The resulting pasted syntax is as follows: (The * in this example refers to the currently open working file: Examp1_2.sav)

```
ADD FILES /FILE=*
 /FILE='C:\ALL_WORK\Example3.sav'
 /IN=inex3.
VARIABLE LABELS inex3
 'Case source is C:\ALL_WORK\Example3.sav'.
EXECUTE.
```

Additional Notes:

The /RENAME=, /DROP= and /KEEP= subcommands may also be used with ADD FILES. The /RENAME= subcommand should follow the appropriate /FILE= (or associated /IN=) subcommand. The /DROP= and /KEEP= subcommands should follow the last /FILE= (or associated /IN= or /RENAME= subcommands).

It might be desirable to change the variable label for the /IN= variable.
VARIABLE LABELS inex3 'In Example3'.

It might be desirable to sort data by a unique identifier. This is particularly true if the first file skipped an ordered identifier (for example: 1,2,3,5,6,8) and the new file contains the skipped records (for example:
4,7,9,10). To do this, be sure that both files are sorted by the identifier variable(s), and add (after the last /FILE= listing and before any /DROP= or /KEEP= subcommands):
/BY=varlist

Variables must have the same format and size in both files in order for the files to be merged correctly.

Unpaired variables will be missing for records from the that did not contain the variables (see score2, inEx1 and inEx2 for Ids 13-16 below).

Results of adding Example3.sav to Examp1_2.

Id	Place	Score1	Score2	Score3	inEx1	inEx2	inEx3
1	Here	5.90	2.30	1.50	1	1	0
2	There	2.60	3.33	2.40	1	1	0
3	There	4.98	1.40	1.42	1	1	0
4	Here	.	.	2.50	1	1	0
5	There	7.21	5.12	.	1	1	0
6	Here	8.45	4.10	4.20	1	1	0
7	Here	.	2.25	3.00	0	1	0
8	Here	3.51	1.23	2.00	1	1	0
9	There	.	5.30	3.50	0	1	0
10	Here	9.10	.	4.12	1	1	0
11	There	.	3.27	2.70	0	1	0
12	Here	6.00	3.81	3.22	1	1	0
13	Here	6.40	.	1.50	.	.	1
14	There	5.23	.	2.39	.	.	1
15	There	7.12	.	1.42	.	.	1
16	Here	8.44	.	2.50	.	.	1

Data from Example1.sav

ID	place	score1
1	Here	5.90
2	There	2.60
3	There	4.98
4	Here	.
5	There	7.21
6	Here	8.45
8	Here	3.51
10	Here	9.10
12	Here	6.00

Data from Example2.sav

ID	place	score2	score3
1	Here	2.30	1.50
2	There	3.33	2.40
3	There	1.40	1.42
4	Here	.	2.50
5	There	5.12	.
6	Here	4.10	4.20
7	Here	2.25	3.00
8	Here	1.23	2.00
9	There	5.30	3.50
10	Here	.	4.12
11	There	3.27	2.70
12	Here	3.81	3.22

Data from Example3.sav

ID	place	score1	score3
13	Here	6.40	1.50
14	There	5.23	2.39
15	There	7.12	1.42
16	Here	8.44	2.50

Comments

Note that SPSS 10 does not handle out of order files (for adding variables) in the same way as previous versions. If one of the files is out of order, prior versions would produce an error message and return the original working file. In SPSS 10 (including 10.0.5), the error message will be produced, but the match will be performed up to the point where the file is out of order. No cases beyond that point will be matched. Thus, it is quite important, regardless of version, to check the output log for any error messages immediately following the execution of the matching files by a key variable and before saving the data. Also SPSS 10 maintains the original file name (instead of renaming to "Untitled"). Care is needed to avoid saving over the original working file.