HOW TO CREATE AND MERGE DATASETS IN SPSS

If the original datasets to be merged are large, the process may be slow and unwieldy. Therefore the preferred method for working on multiple sweeps of data is to create bespoke datasets with the necessary variables (using "DROP" or "KEEP" commands) and then merge these datasets together. For this workshop it is the number of cases in the Sweep 1 sample which has been reduced (to approximately 30% of the original sample), and those cases only have been selected (when applicable) in the subsequent datasets for cross-sweep comparison.

CREATING BESPOKE DATASETS USING THE 'DROP' AND 'KEEP' COMMANDS

- The **KEEP** command allows you to open a large data file specifying which of the variables from that file you wish to ***INCLUDE*** in your working data file.
- The KEEP command can be appended to either the GET FILE or SAVE OUTFILE commands
- Both individual variables and ranges of variables can be specified
- The case unique identifier – Idnumber - will usually have to be included (to permit later merging with other datasets)

Examples:

```
GET FILE='C:\temp\GUSSW3B_30.sav'
 /Keep = idnumber, dcwinc01, dchgmag2 to ddmedu02.

SAVE OUTFILE='C:\temp\Keep Save As Test.sav'
 /Keep = idnumber, dcwinc01, dchgmag2 .
```

- The **DROP** command allows you to open a large data file specifying which of the variables from that file you wish to ***REMOVE*** from your working data file.
- The DROP command can be appended to either the GET FILE or SAVE OUTFILE commands
- Both individual variables and ranges of variables can be specified
- Again, the case unique identifier – Idnumber - will usually have to be included (to permit later merging with other datasets

Examples:

```
GET FILE='C:\temp\GUSSW3B_30.sav'
 /Drop = samptype to dcwtchd2.

SAVE OUTFILE='C:\temp\Drop Save As Test.sav'
 /Drop = dcurind1, dcurind2 .
```

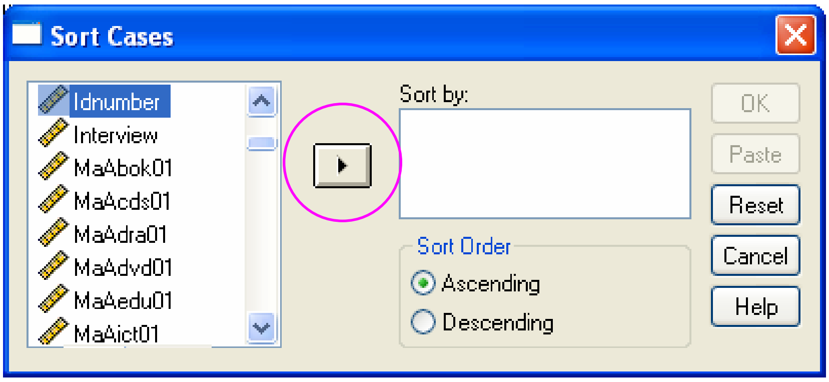
MERGING DATASETS

Datasets can be merged using the unique case ID stored in the variable ‘IDnumber’. Whenever you are first merging files, it is easier to use the SPSS menus and then paste the syntax (automatically generated and recorded in the output) rather than using the syntax from scratch as it can be quite tricky depending on how large each of your datasets are and how many identical variables are in each already. The datasets to be merged must always be **sorted on the same variable before merging** otherwise the matching will not proceed.

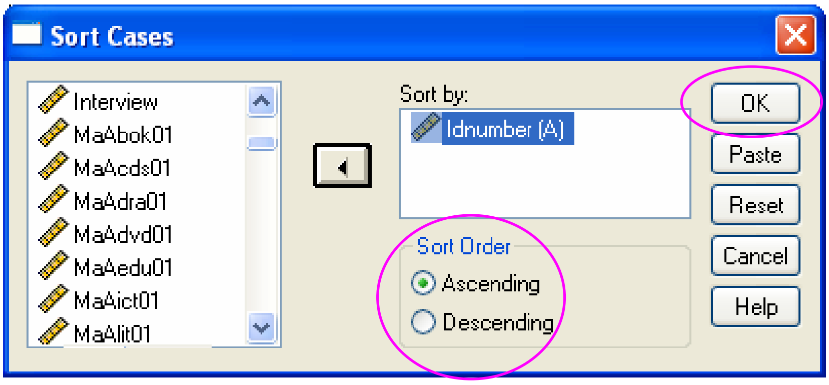
1) Open the dataset you want to merge data into: in the example below it is the Sweep 1 birth cohort dataset

2) Sort this dataset on the key variable ‘IDnumber’ in ascending order via the menu: go to Data\Sort Cases:

   ○ Select the variable ‘IDnumber’ on the left part of the screen



   ○ And move it to the right part of the screen using the arrow – the default option is ‘Ascending’ order

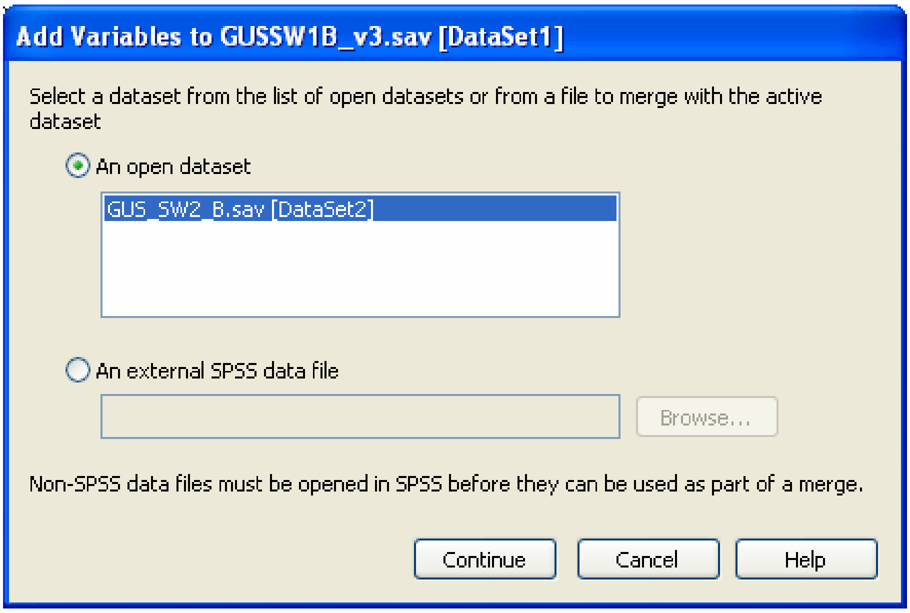


   ○ Click ‘OK’

3) Repeat the same process 1) and 2) above with the dataset you want to extract the data from: the Sweep 2 birth cohort in the example below, to be added to the 1st dataset = Sweep 1 birth cohort

4) On the menu of the **1st** dataset go to: Data\Merge files\Add variables

5) In the dialogue box, unless the dataset from which you want to merge is already open, select the button for ‘An external SPSS data file’ and click ‘Browse’. If the dataset is open then select it in the ‘open dataset’ box (as below).



Add Variables to GUSSW1B_v3.sav [DataSet1]

Select a dataset from the list of open datasets or from a file to merge with the active dataset

● An open dataset

GUS_SW2_B.sav [DataSet2]

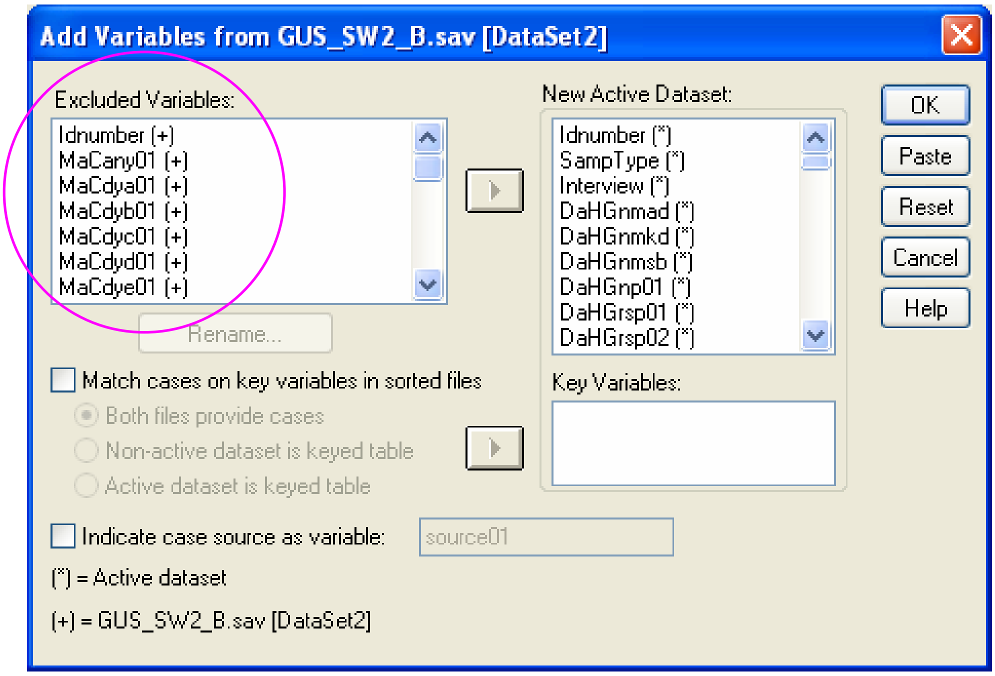
○ An external SPSS data file

Non-SPSS data files must be opened in SPSS before they can be used as part of a merge.

6) If not already open, browse to the dataset of interest and double-click on it

7) Click ‘Continue’

8) The following dialog box will come up; in this example you can see that there is a big list of ‘Excluded Variables’ on the left, which are the variables shared by both datasets, instead of just the expected variable ‘IDnumber’. This is due to the feed forward process: the archived datasets from Sweep 2 include some of the previous sweep variables since original information is only updated when applicable and we want the full information for all cases at each sweep, including those with no changes. To get the full information for this type of variable you need to incorporate the successive sweeps variables.



Add Variables from GUS_SW2_B.sav [DataSet2]

Excluded Variables:

Idnumber (+)
MaCany01 (+)
MaCdya01 (+)
MaCdyb01 (+)
MaCdyc01 (+)
MaCdyd01 (+)
MaCdye01 (+)

New Active Dataset:

Idnumber (*)
SampType (*)
Interview (*)
DaHGnmad (*)
DaHGnmkd (*)
DaHGnmsb (*)
DaHGnp01 (*)
DaHGrsp01 (*)
DaHGrsp02 (*)

□ Match cases on key variables in sorted files
● Both files provide cases
○ Non-active dataset is keyed table
○ Active dataset is keyed table

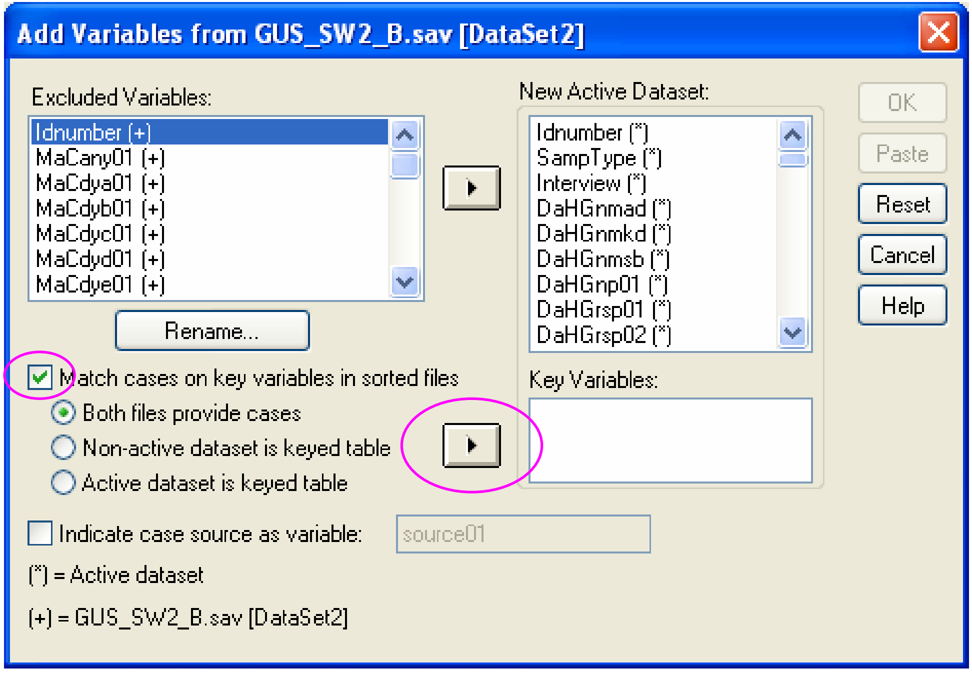
Key Variables:

□ Indicate case source as variable: source01

(*) = Active dataset

(+) = GUS_SW2_B.sav [DataSet2]

9) In this ‘Add variables’ dialogue box, click the box ‘Match cases on key variables in sorted files’, and browse to and highlight the variable ‘IDnumber’:



Add Variables from GUS_SW2_B.sav [DataSet2]

Excluded Variables:

Idnumber (+)
MaCany01 (+)
MaCdya01 (+)
MaCdyb01 (+)
MaCdyc01 (+)
MaCdyd01 (+)
MaCdye01 (+)

New Active Dataset:

Idnumber (*)
SampType (*)
Interview (*)
DaHGnmad (*)
DaHGnmkd (*)
DaHGnmsb (*)
DaHGnp01 (*)
DaHGrsp01 (*)
DaHGrsp02 (*)

☑ Match cases on key variables in sorted files
● Both files provide cases
○ Non-active dataset is keyed table
○ Active dataset is keyed table

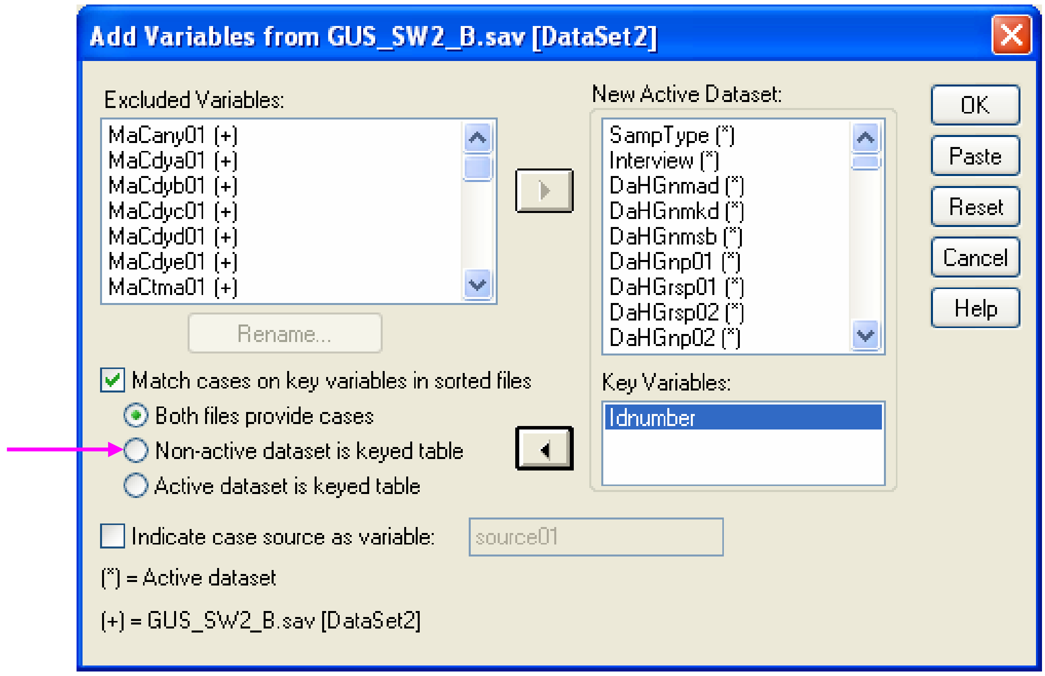
Key Variables:

□ Indicate case source as variable: source01

(*) = Active dataset

(+) = GUS_SW2_B.sav [DataSet2]

10) Click on the arrow next to the ‘Key variables’ box. ‘IDnumber’ should now appear in the ‘Key variables’ box.



Add Variables from GUS_SW2_B.sav [DataSet2]

Excluded Variables:
MaCany01 (+)
MaCdya01 (+)
MaCdyb01 (+)
MaCdyc01 (+)
MaCdyd01 (+)
MaCdye01 (+)
MaCtma01 (+)

New Active Dataset:
SampType (*)
Interview (*)
DaHGnmad (*)
DaHGnmkd (*)
DaHGnmsb (*)
DaHGnp01 (*)
DaHGrsp01 (*)
DaHGrsp02 (*)
DaHGnp02 (*)

☑ Match cases on key variables in sorted files
● Both files provide cases
○ Non-active dataset is keyed table
○ Active dataset is keyed table

Key Variables:
Idnumber

□ Indicate case source as variable: source01

(*) = Active dataset

(+) = GUS_SW2_B.sav [DataSet2]

The steps you take next will depend on what dataset you're already working on:

11) Under 'Match cases on key…" if you select…
   - 'Both files provide cases' (default option): All cases from the merged dataset will be transferred into the working dataset. **If you are working on a later dataset and merging in data from an earlier dataset, choosing this option means that additional <u>cases</u> from the earlier dataset will be merged along with the variables. These cases will have 'missing' data for the variables at the later sweep because they were not achieved at that sweep.**
   - 'Non-active dataset is keyed table': Only merged data for those cases already in the working dataset will be transferred. **This avoids the above issue if you are working on a later dataset and merging in a variable from an earlier sweep. Only information from those cases in the working (later) dataset will be merged so you won't generate entire cases with 'missing' data which would need to be deleted or filtered out later on.**
   - 'Active dataset is keyed table': All cases from the merged dataset will be transferred into the working dataset. This produces the same result as the first scenario.

12) In this example you need to **select Option 2 'Non-active dataset is keyed table'**
13) Click 'OK' and again 'OK' in the warning message re cases needing to be sorted before merging