COM8005 Stats Workshop

7. Using SPSS Syntax to Manipulate Data Files

Jonathan Zhu

Oct 22, 2014

Most Frequent Tasks/Commands

- Read and save data files
- Create a random sample
  - input program …
- Combine 2+ data files vertically
  - add files …
- Combine 2+ data files horizontally
  - match files …
- Aggregate data files from individual-level to collective-level
  - aggregate outfile …

Read and save data files

- Read from SPSS data

get file ...

- Read from other data

get data ...
get translate ...

- Save to SPSS data

save outfile ...

- Save to other data

save translate ...
write outfile ... /*txt file

- Save part of SPSS data

temp.
select if ...
save out file ...

Create a random sample (n = 400)

```
input program.          /* Start of ”input program” cycle
loop #i=1 to 400.       /* Start of “loop ” cycle
compute ID=#i.          /* “#i” is a temporary variable
end case.               /* End of cases (no start statement)
end loop.               /* End of “loop” cycle
end file.               /* End of file (no start statement)
end input program.      /* End of “input program
exe.                    /* To “execute” the above statements
descr ID.               /* To “describe” ID
```

Create a random sample of hierarchically nested structure

*The example involves 2 levels, with 600 (=30 x 20) individuals at level 1 and 20 groups at level 2.

```
input program.
loop #j=1 to 20.
loop #i=1 to 30.
compute GROUP.ID=#j.
compute INDIVIDUAL.ID=#i.
end case.
end loop.
end loop.
end file.
end input program.
exe.
descr GROUP.ID INDIVIDUAL.ID.
```

Combine 2+ data files vertically

- Use “add files” to merge cases from 2+ files

```
add files
    file 'C:\FILE1.SAV'
   /file 'C:\FILE2.SAV'
  [/file ...]              /* Optional, up to 48 files
   /map.                   /* Show the resulting structure
exe.                       /* Execute the above statements.
```

Combine 2+ data files horizontally (1): Prepare data files for the merging

Use “match files” to merge variables from 2+ files.

- 1+ unique (non-duplicated) ID variable(s) is/are required for each file
- Each file must be sorted ascendingly based on the ID variable(s) before the merge.

ID1	X	Y
1		
2		
…		
n		

+

ID1	Z	W
1		
2		
…		
n		

=

ID1	X	Y	Z	W
1				
2				
…				
n				

Sort each file:

*Sort file 1.

```
get file 'C:\FILE1.SAV'.
sort cases by ID1 [ID2
...].
save out 'C:\FILE1.SAV'.
```

*Sort file 2.

```
get file 'C:\FILE2.SAV'.
sort cases by ID1 [ID2
...].
save out 'C:\FILE2.SAV'.
```

*Optional: repeat the above for more files.

Combine 2+ data files horizontally (2): Merge files of the same level

```
match files                          /* Merge variables of 2+ files
   file 'C:\FILE1.SAV'               /* File 1 to be merged
  /file 'C:\FILE2.SAV'               /* File 2 to be merged
[/file ...]                          /* Optional, up to 48 files
  /by ID1 [ID2 ...]                  /* In the same order as in each file
   /map.                             /* Show the new file structure
exe.                                 /* Execute the above statements.
```

Combine 2+ data files horizontally (3): Merge files of different levels

- If the 2+ files involve different units of analysis (e.g., file 1 contains individuals (level 1) and file 2 organizations (level 2), the ID variable(s) could (and should) be duplicated in level 1 but must be non-duplicated in level 2.

ID1	ID2	X	Y
1	1		
2	1		
3	2		
4	2		
…	…		
n	…		

+

ID2	Z	W
1	9	5
2	3	1
…		
m		

=

ID1	ID2	X	Y	Z	W
1	1			9	5
2	1			9	5
3	2			3	1
4	2			3	1
…	…				
n	…				

Combine 2+ data files horizontally (4): Merge files of different levels (cont'd)

Sort each file:

*Sort file 1 (individuals).

```
get file 'C:\FILE1.SAV'.
sort cases by ID2.
save out 'C:\FILE1.SAV'.
```

*Sort file 2 (organizations).

```
get file 'C:\FILE2.SAV'.
sort cases by ID2.
save out 'C:\FILE2.SAV'.
```

*Optional: repeat the above for more files at level 1.

Match files (note the use of "table"):

```
match files
   file 'C:\FILE1.SAV'
  /table 'C:\FILE2.SAV'
  /by ID2
  /map.
exe.
```

Aggregate a data file from individual-level to collective-level (1)

File 1. Individual-level data

ID1	ID2	X	Y	…
1	1			
2	1			
1	2			
2	2			
…	…			
n	c			

File 2. Collective-level data

ID2	C.n	X.m	X.sd	Y.m	Y.sd	…
1						
2						
…						
m						

ID1 is a unique ID at level 1 (with a non-duplicated value for each individual); ID2 is a unique ID at level 2 (non-duplicated for each collective unit); X, Y, etc. are usual variables at level 1;

C.n is the number of cases (i.e., individuals) for collective unit C; X.m, X.sd, Y.m, Y.sd are the mean and standard deviation of X and Y, respectively, for unit C. All the varaibles are created based on file 1 during the aggregation.

Aggregate a data file (2)

```
get file 'C:\FILE1.SAV'.
sort cases by ID2. /*the file must be sorted before aggregated.
aggregate out 'C:\FILE2.SAV'
 /break=ID2
 /C.N=n(ID2)
 /X.M Y.M …=mean(X, Y [, …])
 /X.SD Y.SD …=sd(X, Y [, …])
 [/…].
exe.
```