

Using SPSS to Restructure Overtime Dyadic Data

Rob Ackerman & Debby Kashy

Starting with a **Dyad Overtime** data set

Dyadid	MProg1	MProg2	MProg3	FProg1	FProg2	FProg3	Manxiety	Fanxiety
1	75	85	85	80	80	85	3.20	2.80
2	70	70	65	60	65	60	5.90	3.50
3	50	70	70	75	80	75	2.60	4.00

There are three things that we should notice about the above data set. First, each dyad (denoted by “Dyadid”) has scores for the man (e.g., MProg1) and the woman (e.g., FProg1) on the same line. Second, we have a time-varying variable (i.e., prognosis) that for the purposes of data entry is split up into three variables (e.g., Men’s Prognosis at Times 1, 2, and 3, is designated by “MProg1”, “MProg2”, and “MProg3”, respectively). Third, we have one variable that is the same for each time period (i.e., anxiety).

To be able to conduct an Overtime Actor-Partner Interdependence Model (APIM) using Multilevel Modeling (MLM), we need to convert this data file into a **Person-Period-Pairwise** structure. Doing so involves a series of three steps. First, we must transform the current **Dyad Overtime** data set into a **Dyad-Period Overtime** data set. Second, we must use the **Dyad-Period Overtime** data set to create two more data files: (a) **Person-Period Overtime Actor File**; and (b) **Person-Period Overtime Partner File**. Third, we merge these two files together to create the **Person-Period-Pairwise** data set.

Before beginning, we must make sure to save the current **Dyad Overtime** data set into a different file so that we do not accidentally write over it. Let us call it “workingfile.” Syntax is below:

```
save outfile='C:\workingfile.sav'.  
execute.
```

Creating the Dyad-Period Overtime Data Set

After opening our working file, we need to restructure the data set so that the different variables for Men’s Prognosis and Women’s Prognosis are considered cases of the same general variables (i.e., Men’s Prognosis and Women’s Prognosis), though collected at three different times. Below is the syntax:

```
VARSTOCASES  
  /MAKE mprog FROM mProg1 mProg2 mProg3  
  /MAKE fprog FROM fProg1 fProg2 fProg3
```

```

/INDEX=Time(3)
/KEEP=dyadid mAnxiety fAnxiety
/NULL=KEEP.

```

Notice how although the three different prognosis variables are collapsed into the same variable, the time at which they were collected is still indexed by another variable (“Time”). Also, because the variable “Anxiety” does not vary across time, we place it in the line that states “KEEP.” After our data set is created, we save it as “Dyad Period Overtime.sav” Syntax is below:

```

save outfile='C:\ Dyad Period Overtime.sav'.
EXECUTE .

```

This is what the **Dyad-Period Overtime** data set looks like:

Dyadid	Manxiety	Fanxiety	Time	MProg	FProg
1	3.20	2.80	1	75	80
1	3.20	2.80	2	85	80
1	3.20	2.80	3	85	85
2	5.90	3.50	1	70	60
2	5.90	3.50	2	70	65
2	5.90	3.50	3	65	60
3	2.60	4.00	1	50	75
3	2.60	4.00	2	70	80
3	2.60	4.00	3	70	75

Creating the Person-Period Overtime Data Sets

Note that we now have a data set in which period of prognosis is denoted by its own variable (i.e., Time). To create a person-period overtime pairwise data structure, however, we also need a variable that denotes which person the data within the row refers to (i.e., person), as well as data from both partners within the same row (i.e., actor and partner variables). This involves creating two different Person-Period Overtime data sets—one in which Person 1 within the dyad is the man and Person 2 is the woman (i.e., Person-Period Overtime Actor), and another in which Person 1 within the dyad is the woman and Person 2 is the man (i.e., Person-Period Overtime Partner).

Person-Period Overtime Actor

After opening up the Dyad-Period Overtime data set, we need to restructure the data set such that we have one variable to denote Prognosis (i.e., Aprognosis) and another variable to denote Anxiety (i.e., Anxiety). We are therefore collapsing data from men and women on the same variables into a single variable. Although we are combining this data, we still want denote where the data came from. Because the variable names for the men's variables come first in the syntax lines below, these scores will be shown as data from Person 1. Like before, because "dyadid" and "time" do not vary according to person, we will place these variables after the word "keep=" in the syntax.

```
varstocases
```

```
    /make Aprognosis from mprog fprog  
    /make Aanxiety from manxiety fanxiety  
    /index=person (2) /keep=dyadid time.
```

```
execute.
```

To ensure that we are able to merge this data set with the next one we create, let us sort the data by dyadid, person, and time in ascending order. Syntax for this is below:

```
SORT CASES BY
```

```
dyadid (A) person (A) time (A).
```

Finally, let us save the current file as "Person Period Overtime A.sav" Syntax is below.

```
save outfile='C:\ Person Period Overtime A.sav'.
```

```
execute.
```

This is what the **Person-Period Overtime Actor** data set looks like:

Dyadid	Time	person	Aprognosis	Aanxiety
1	1	1	75	3.20
1	2	1	85	3.20
1	3	1	85	3.20
1	1	2	80	2.80
1	2	2	80	2.80
1	3	2	85	2.80
2	1	1	70	5.90
2	2	1	70	5.90
2	3	1	65	5.90
2	1	2	60	3.50
2	2	2	65	3.50
2	3	2	60	3.50
3	1	1	50	2.60
3	2	1	70	2.60
3	3	1	70	2.60
3	1	2	75	4.00
3	2	2	80	4.00
3	3	2	75	4.00

Person-Period Overtime Partner

Before anything else, we must reopen the Dyad Period Overtime data. Syntax is below:

```
get file='c:\Dyad Period Overtime.sav'.
```

This next step is very similar to creating the previous Person-Period Overtime Actor file. As before, we need to collapse data from men and women on the same variables into a single variable. Moreover, because “dyadid” and “time” do not vary according to person, we will place these variables after the word “keep=” in the syntax. However, because we are creating the Partner file, we now would like Person 1 to be the women and Person 2 to be the man. To do this, we simply put the variable names for the women’s variables before the variable names for the men’s variables in the syntax lines (see below).

```
varstocases  
  /make Pprognosis from fprog mprog  
  /make Panxiety from fanxiety manxiety  
  /index=person (2) /keep=dyadid time.  
EXECUTE .
```

To ensure that we are able to merge this data set with the previous one we created, let us sort the data by dyadid, person, and time in ascending order. Syntax for this is below:

```
SORT CASES BY  
  dyadid (A) person (A) time (A).
```

Finally, let us save the current file as “Person Period Overtime P.sav” Syntax is below.

```
save outfile='C:\ Person Period Overtime P.sav'.  
execute.
```

This is what the **Person-Period Overtime Partner** data set looks like:

Dyadid	Time	person	Pprognosis	Panxiety
1	1	1	80	2.80
1	2	1	80	2.80
1	3	1	85	2.80
1	1	2	75	3.20
1	2	2	85	3.20
1	3	2	85	3.20
2	1	1	60	3.50
2	2	1	65	3.50
2	3	1	60	3.50
2	1	2	70	5.90
2	2	2	70	5.90
2	3	2	65	5.90
3	1	1	75	4.00
3	2	1	80	4.00
3	3	1	75	4.00
3	1	2	50	2.60
3	2	2	70	2.60
3	3	2	70	2.60

Creating the Person-Period Pairwise Overtime Data Set

To create the Person-Period-Pairwise Overtime data set, all we need to do is merge the Person-Period Overtime data sets for Actor and Partner that we just created. To do this in SPSS, we simply open the Person-Period Overtime Partner data set and add the actor variables (i.e., Aprognosis and Aanxiety) from the Person-Period Overtime Actor data set to it. Importantly, we match these actor variables to the partner variables by the dyadid, person, and time. Syntax for this procedure is below:

```
match files file='c:\Person Period Overtime A.sav'  
/file=* /by dyadid person time.  
execute.
```

Let us now save this file as "Person Period Pairwise Overtime.sav" Syntax for this is below:

```
save outfile='C:\Person Period Pairwise Overtime.sav'.  
execute.
```

This is what the **Person-Period Pairwise Overtime** data set looks like:

Dyadid	Time	person	Aprognosis	Pprognosis	Aanxiety	Panxiety
1	1	1	75	80	3.20	2.80
1	2	1	85	80	3.20	2.80
1	3	1	85	85	3.20	2.80
1	1	2	80	75	2.80	3.20
1	2	2	80	85	2.80	3.20
1	3	2	85	85	2.80	3.20
2	1	1	70	60	5.90	3.50
2	2	1	70	65	5.90	3.50
2	3	1	65	60	5.90	3.50
2	1	2	60	70	3.50	5.90
2	2	2	65	70	3.50	5.90
2	3	2	60	65	3.50	5.90
3	1	1	50	75	2.60	4.00
3	2	1	70	80	2.60	4.00
3	3	1	70	75	2.60	4.00
3	1	2	75	50	4.00	2.60
3	2	2	80	70	4.00	2.60
3	3	2	75	70	4.00	2.60