

## Using Aggregation in SPSS

Some constructs require multiple observations per subject in order to provide accurate measurement. We have discussed two such constructs in class. First, Professor Pierce discussed how Factor Analysis can detect a cluster of related items that represent a single dimension. Importantly, a single item does not have the explanatory power that the cluster of items possesses. Therefore, multiple items (or observations) are needed to best represent the construct. Second, reaction time (RT) experiments often make use of multiple observations per subject to protect against measurement error. Specifically, RT experiments are designed to measure fractions of a second, where the length of an eye blink (300 ms) is considered a dramatic effect. If a single observation were used, distraction, even in its most fleeting form, could lead to a spurious conclusion. Therefore, multiple observations are used to minimize the effect of momentary lapses of attention by using the mean of multiple observations (usually 20-60).

For illustrative purposes, let's imagine that we are interested in measuring the facilitation (in ms) that a prime of a particular valence (say, positive) provides when categorizing a target word that is of the same valence (positive). For example, viewing the word diamond, a positive word, immediately prior to the word sunshine typically facilitates the categorization of sunshine as positive. That is, sunshine is categorized as belonging to the category "positive" more quickly than if sunshine were preceded by a negative word or no word at all. The same is true of negative words. Therefore, an experiment can be performed in which the valence of the prime and target match or differ. Multiple observations of the four possible prime+target combinations (positive+positive, positive+negative, negative+positive, and negative+negative) will be needed to examine the prime+target facilitation effect.

The difficulty with multiple observations of a single construct (per subject) is that the data file contains several rows for each subject. In most cases, data analysis requires a single row of data per subject. Otherwise, data analysis can become cumbersome. In the worst case, experimental power is erroneously inflated due to the statistical program considering each row an individual subject rather than a subset of a particular subject or subjects. The data set in this example includes columns for data, time, subject, trialcode, trialnumber, trialdata1, trialdata2, trialdata3, latency, response and correct (see Figure 1). For the aggregate analysis, we need to be concerned only about the subject, trialcode, and latency (and in some cases, the correct) variables. Below I provide a step-by-step outline of how to use the aggregate command in SPSS to summarize multiple observations into a single line of data. (Please keep in mind that this data set is a repeated-measures design and some steps are particular to this design. However, the logic can easily be applied to a between-subject design.) The following analysis is specific to SPSS, version 14 (available on <http://citrixweb.hamilton.edu/citrix>):

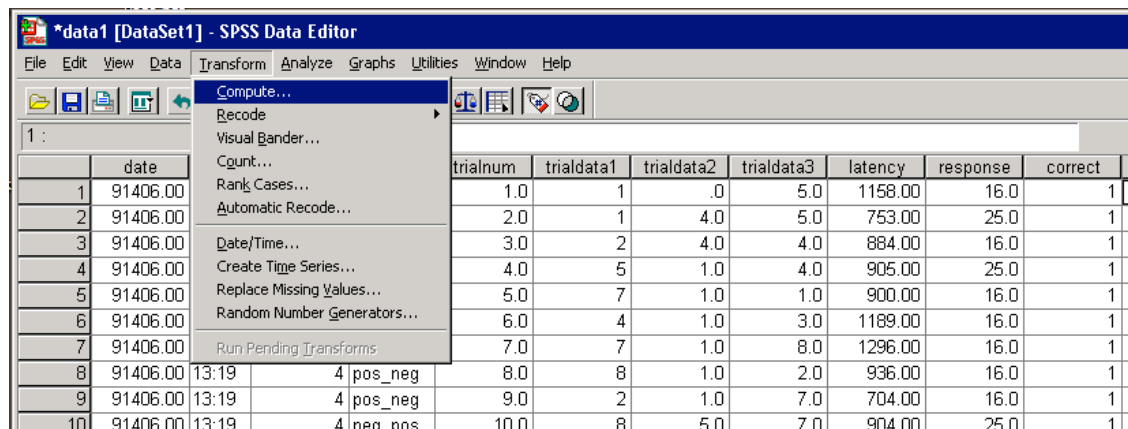
Figure 1. (Example data set)

|   | date     | time  | subject | trialcode | trialnum | trialdata1 | trialdata2 | trialdata3 | latency | response | correct |
|---|----------|-------|---------|-----------|----------|------------|------------|------------|---------|----------|---------|
| 1 | 91406.00 | 13:19 | 4       | neg_neg   | 1.0      | 1          | .0         | 5.0        | 1158.00 | 16.0     | 1       |
| 2 | 91406.00 | 13:19 | 4       | pos_pos   | 2.0      | 1          | 4.0        | 5.0        | 753.00  | 25.0     | 1       |

If your data set does not include repeated-measures IVs, please skip the following step and go directly to step 2.

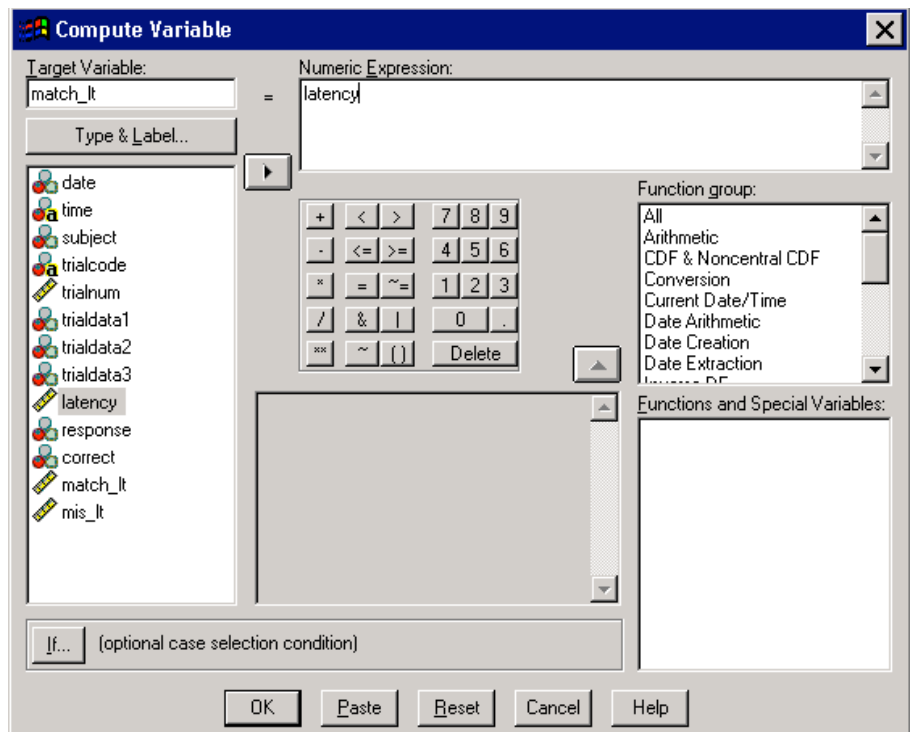
1) If you have any repeated-measures variables, you will need to compute separate dependent variables (i.e., columns) for each level of the independent variable. In our current example we have a single repeated-measures variable: prime+target combination (match vs. mismatch). In order to assess the RT for each IV level, we must use the “compute” command. Let’s examine the RT for the prime+target combination in which the valence matches. Using the windows menu, select the *Transform* tab and select *Compute* (see Figure 2).

Figure 2. (Selecting Compute Function)



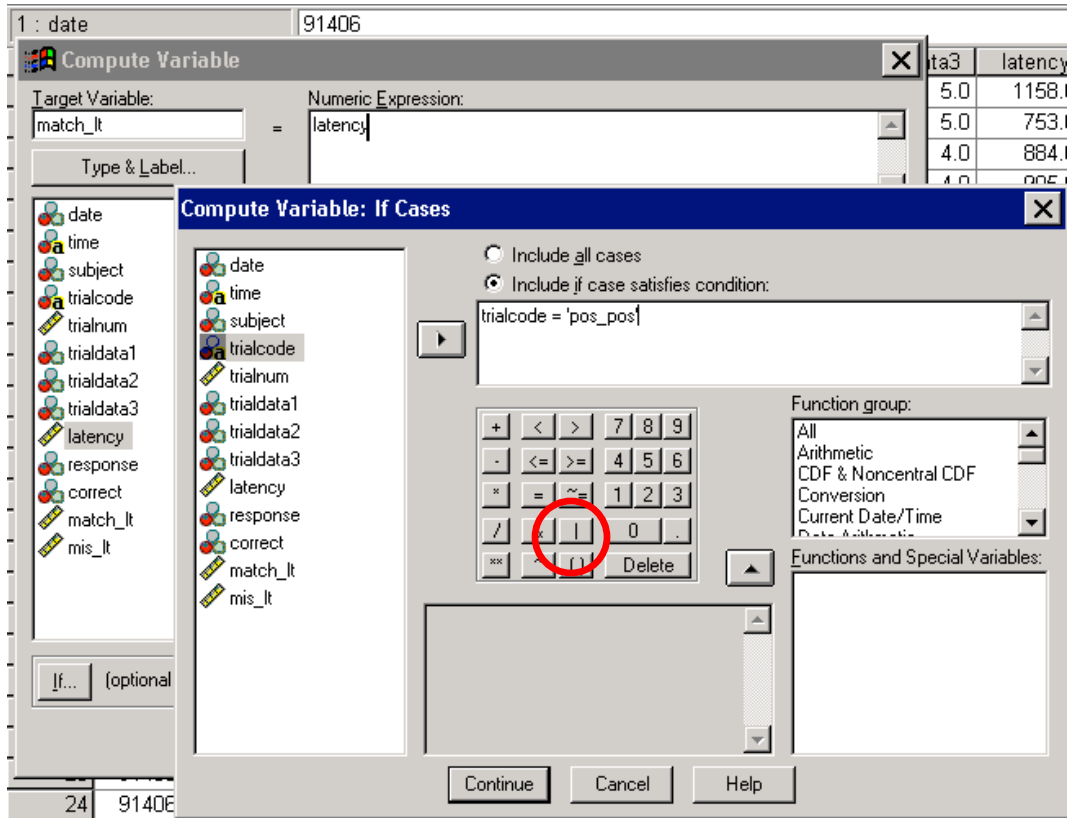
Once the compute window (see Figure 3) is opened, enter the name of the variable you want to compute in the *Target Variable* field. Let’s call it *match\_lt* for matching latency. In the *Type & Label* field, select (by double-clicking) the *latency* variable. You will notice that once the variable has been selected, it will appear in the *Numeric Expression* field.

Figure 3. (Compute Window)



In order to examine ONLY those RTs in which the prime+target valence match, we must use the *If* button (located at the bottom left-hand corner; see Figure 4). Once you have selected the *If* button, select the radial button that reads “include if case satisfies condition.” In the variable field to the left, select (by double-clicking) trialcode. Trialcode should appear in the upper right-hand field. The reason for selecting this variable is that it uniquely identifies trials in the data set in which the prime+target match. After the trialcode, write the following: = ‘pos\_pos’. Press the *Continue* button and then the *OK* button to execute the compute command.

Figure 4. (If command in the Compute Function)



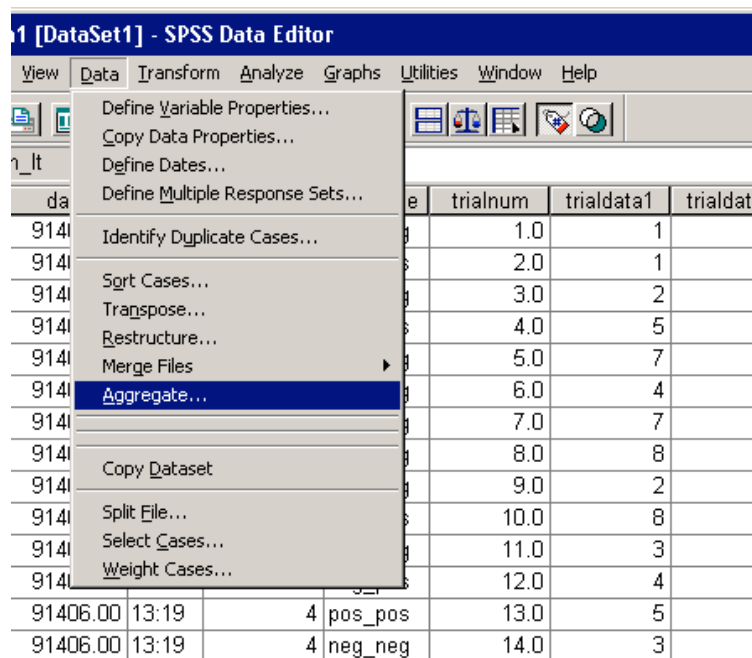
Your data file should now include another variable called `match_lt` that includes RTs for all trials in which the prime+target were both a positive valence. Please repeat the same procedure for trials labeled `neg_neg` using the same variable name. You will be asked at the end of the process whether it is ok to change an existing variable; select yes. (Alternatively, you can conduct the analysis in a single step by writing the following statement during the first *If* window: `trialcode = 'pos_pos' | trialcode = 'neg_neg'`; see red circle in Figure 4 for the | command.) The next step is to create another variable called `mis_lt` for the prime+target pairings that are mismatched. Use the same procedure as above, but now compute the `mis_lt` using the mismatched prime+targets labeled `pos_neg` and `neg_pos`.

Alternatively, you can open a syntax window and write the following command to accomplish the identical process.

```
IF (trialcod = 'pos_pos') match_lt = latency .
IF (trialcod = 'neg_neg') match_lt = latency .
IF (trialcod = 'neg_pos') mis_lt = latency .
IF (trialcod = 'pos_neg') mis_lt = latency .
```

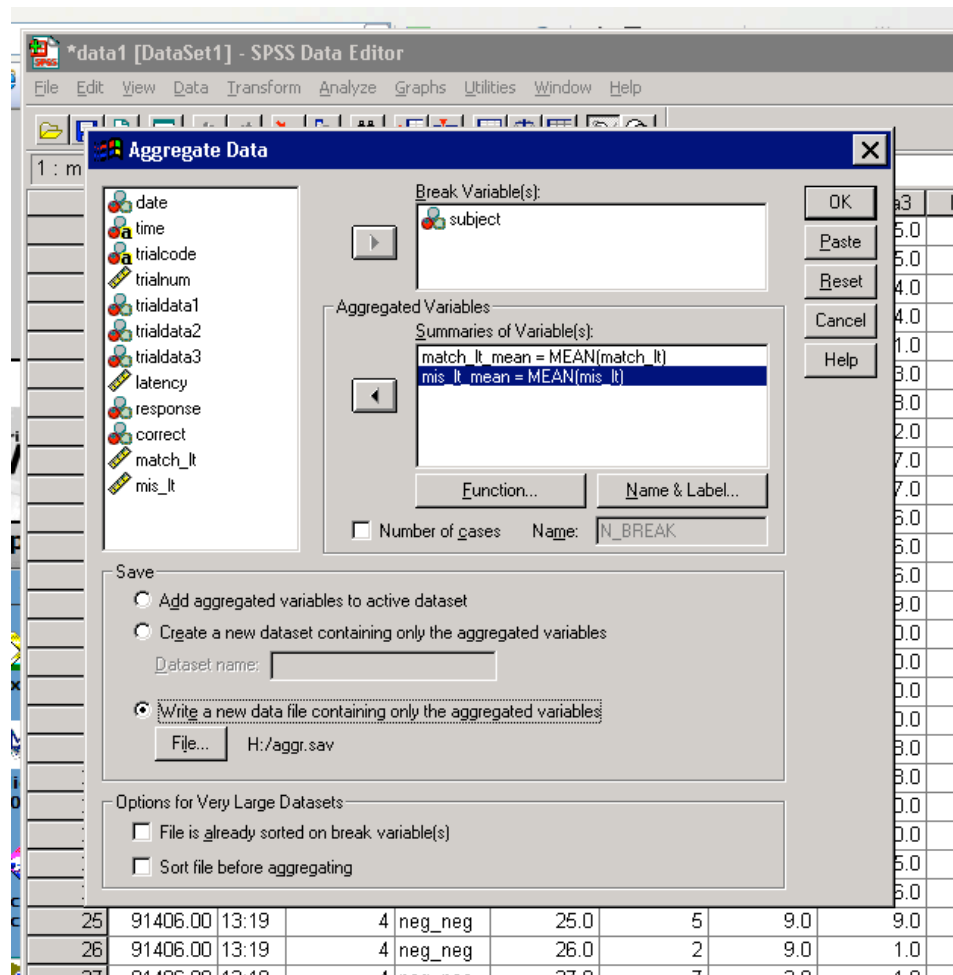
2) You are now ready to summarize the data. In the windows interface, select the *Data* tab, and then *Aggregate* (See Figure 5).

Figure 5. (Selecting Aggregate command)



Once the aggregate window appears (see Figure 6), in the *Summaries of Variable* field, place the variable(s) you would like to summarize. In our example, that would be the variables labeled *match\_lt* and *mis\_lt*. The default operation is an average (i.e., mean). However, you may change this summary operation by selecting the *Function* button. You also may change the Name and Label of the aggregated variable by selecting the *Name & Label* variable. The most important aspect of the *Aggregate* command is the *Break Variable*. The break variable(s) tell SPSS how you would like to break up the data (i.e., what chunks of data will be summarized together). When you enter a variable in the break field, SPSS will summarize the data that shares the same value for that break variable. For example, if you enter *subject* in the *Break Variable* field, SPSS will summarize all data with a subject value of 1, all the data of subject value 2, etc (see Figure 7).

Figure 6. (Aggregate Window)



You may enter more than one break variable, such as subject & correct (where 1 = correct, and 0 = error). SPSS will create a nested summary of these variable values in the order they are entered. In this example, SPSS would summarize data for subject value 4 into two rows, one representing errors (correct = 0) and one representing correct responses (correct = 1; see Figure 8). The more break variables that you enter, the more rows per subject SPSS is going to create. That being said, if you have between-subject variables that you want to keep in your file, enter them into the *Break Variable* field. (Note: Because a between-subject variable has only a single value per subject, this will not affect the number of rows per subject.) Under the *Save* field, select the radial button labeled “Write a new data file containing only the aggregated variables.” SPSS will automatically assign the new data file a name and location (Note: the default name for an aggregate file is aggr.sav). To change this information, select the *File* button. To run the aggregate command, select *OK*. Do not worry if your data file does not appear to change. You saved the aggregated data in another file and now you will need to locate the file and open it by

selecting the *File* tab, *Open, Data*. Select the aggregated data file and press *Open*. SPSS will open another data file that includes only the aggregated data files (see Figure 7).

Figure 7. (Aggregated data set with subject as break variable)

|   | subject | match lt mean | mis lt mean | var |
|---|---------|---------------|-------------|-----|
| 1 | 4       | 697.80        | 714.77      |     |
| 2 |         |               |             |     |
| 3 |         |               |             |     |
| 4 |         |               |             |     |
| 5 |         |               |             |     |
| 6 |         |               |             |     |

Figure 8. (Aggregated data set with subject & correct as break variables)

|   | subject | correct | match lt mean | mis lt mean | v |
|---|---------|---------|---------------|-------------|---|
| 1 | 4       | 0       | 582.00        | 655.50      |   |
| 2 | 4       | 1       | 701.79        | 719.00      |   |
| 3 |         |         |               |             |   |
| 4 |         |         |               |             |   |

3) Conduct the appropriate statistical test!