Data Preparation With SPSS

This chapter reviews the general issues involving data analysis and introduces SPSS, the Statistical Package for the Social Sciences, one of the most commonly used software programs now used for data analysis.

History of Data Analysis in Psychology

The chapter Computers in Psychology introduced the history of psychology's relationship with computers. In that chapter, we discussed the early use of the term "computer" as a person who performed computations using paper and pencil or electromechanical devices such as adding machines. Statistical analyses were initially performed by these biological computers for psychological research. Using slide rules and a variety of progressively complex machines, they performed large statistical analyses, laboriously and slowly. The legacy of this hand calculation phase could be found in statistics texts up to as late as the 1980s. Older stats books presented formulae for statistical procedures in a form primarily suitable for calculation, and much effort was spent trying to come up with calculation-friendly formulae for the most complicated analyses. The problem with this approach to formulae writing was that the computational formulae were not very effective in communicating the concepts underlying the calculations. Most modern texts use conceptual or theoretical formulae to teach the concepts and software to perform the calculations.

The invention of practical computers by the 1960s brought about a huge change in statistics and in how we deal with data in psychology, and these changes have not stopped yet. Early on, programmers began writing small programs in Assembly language (can you spell p-a-i-n?) and in FORTRAN to compute individual statistical tests. When the author was an undergraduate he discovered a library of FORTRAN procedures and was very impressed by the possibilities, compared to his slide rule. The way this worked was: a set of punch cards described the statistical program to be run (e.g., a t-test), and a set of cards contained the data on which to run it. The cards were fed into a card reader, and a while later a printout appeared on the computer center's high-speed line-printer with the results (unless you punched a card incorrectly).

Eventually, the individual FORTRAN programs were collected together in "packages." The package–a single, large piece of software–put everything together: a way to read the data, a way to transform variables (e.g., average several to make a scale score), and a standard way to tell the program which analyses to perform on which variables. Several major packages appeared: SPSS, the Statistical Package for the Social Sciences; SAS, the Statistical Analysis System; BMDP, Biomedical Data

Processing system; and others. Universities placed one or more of these packages on their mainframe computers, and everyone hiked to the computer center to use the packages for their research projects. These computer centers were open all night, and the most dedicated students would walk several miles at night in the snow in order to take advantage of the faster processing times between midnight and 8am.

When desktop computers became readily available in the 1980s, the large packages were ported to Macs and PCs. These ports were poorly done, opening the market to new programs designed specifically for Macs (and later for PCs) that took advantage of the friendly GUI (graphical user interface) operating systems such as MacOS. The most popular of these new programs was StatView. StatView was the standard data analysis program used by the Florida Tech undergraduate program from the late 1980s to about 2000, and remains the standard program in the Science Education graduate program. By the late 1990s, however, the bad ports of the old big packages became nearly as friendly as, and far more powerful than, the newer GUI programs, so SPSS is now most psychologists' program of choice for desktop data analysis. Unfortunately, this development has given SPSS Inc. and the SAS Institute nearly monopoly control of the data analysis market, hence their prices are artificially high (actually, exorbitant) and their attention to quality control is wanting (in a manner similar to the other major software monopoly).

Describing SPSS

SPSS is a huge program that includes functionality that no one I know of has ever completing used. Down deep, it is still a package of FORTRAN programs, but to the average user this is not very important. SPSS does four things:

1. It creates and maintains a dataset.
2. It analyzes your data.
3. It facilitates saving and printing the results of your analyses
4. It graphs your results

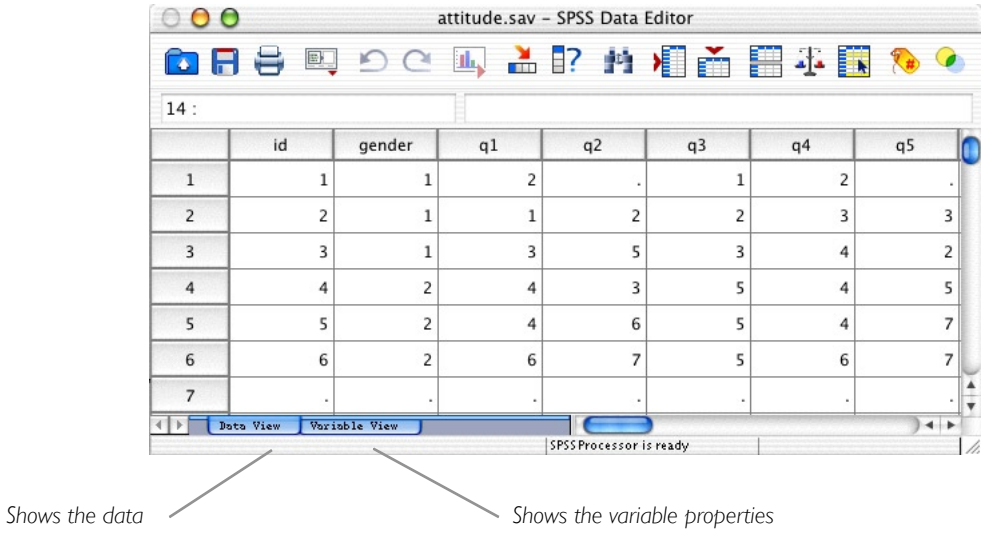
Datasets

SPSS, like all other modern data analysis packages, uses a spreadsheet device for data entry and transformation. A dataset is a file that includes the data, variable names, and other attributes of the data such as labels. In SPSS, dataset files end in the file extension .sav, for example, mythesis.sav. Datasets normally are organized so that columns are variables and rows are cases. A case is a "subject" or "participant" in the parlance of psychology, but could be a corn plant or star in another science.

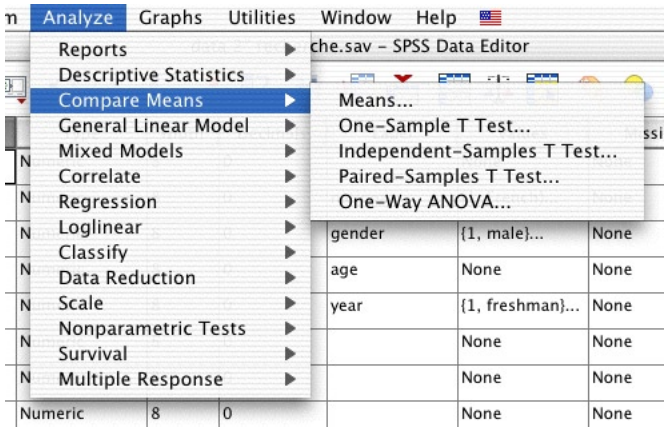
The dataset window in SPSS has two "panes," accessed via tabs at the bottom of the window. The data pane shows the data and the variables pane shows the properties of the variables: name, type, labels, missing values, etc. (See figures)

Analyzing

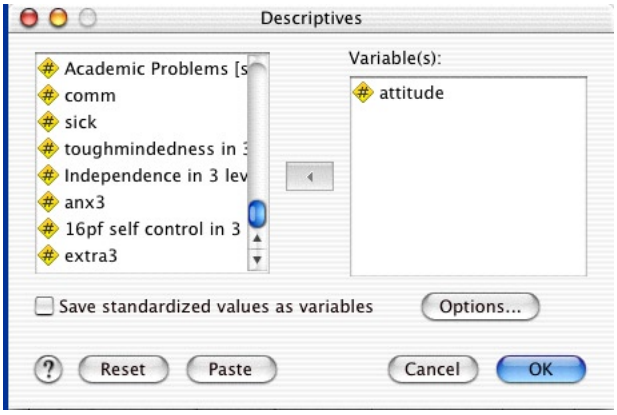
SPSS contains a bewildering set of statistical analyses. The program is sold in



"modules," each of which contains a set of related analyses. For example, the Base module can do all kinds of descriptive statistics, t-tests, correlations, linear regression, etc. The Advanced Statistics module adds some very exotic analyses of variance (ANOVAs). As more modules are added (purchased), the list of items in the Analyze menu grows. This modularity allows SPSS Inc. to add additional components to the program as new types of statistics are invented. It also helps the company harvest a lot of money since each module is quite expensive.



Analyses are performed by choosing an item or sub-item in the Analysis menu then filling in all the details required for the analysis. All analyses present additional options that allow you to customize the analysis in various ways, but these are optional. For example, let's say that you want to calculate the grand mean of attitude toward tuition hikes for your sample. A grand mean is the average of all the people in the sample on a variable. Your analysis screen will look like this:



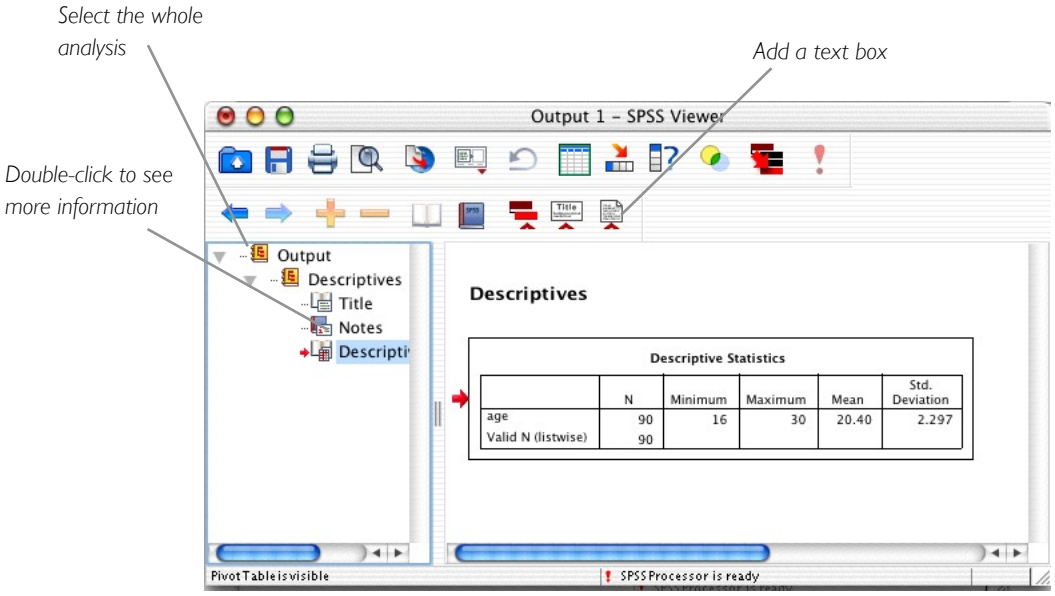
Just as in the 1970s, when you run a SPSS analysis you are actually sending the program a set of arcane textual commands. These commands in a very strict syntax that communicates with the relevant underlying components of the program. For example, the grand mean analysis described above is actually performed by sending this command to the Descriptives component of the program:

```
DESCRIPTIVES
  VARIABLES=attitude
  /STATISTICS=MEAN STDDEV MIN MAX .
```

The GUI window that you used to perform this simple analysis created the command for you, and sent it to the Descriptives component of SPSS. If you click on the Paste button in an analysis window, the command syntax is saved for future use (as described in a later section). In other words, the friendly GUI is helping you construct this command syntax. The great strength of SPSS for serious data analysis is its ability to save the command syntax for reuse.

Output

The results of every analysis you perform, and every message that the program sends you, is sent to an output window. Results and messages accumulate (concatenate) down the window, forming a record of your actions. On the left, a window pane similar to a side-frame on a web page allows you to browse through the output items. Clicking on the name of an analysis in this pane allows you to delete it or print just that one item.



When psychologist perform analyses of complicated datasets,

they need to keep a log or running account of what they are doing. This log allows them to go back to their work later and recall why they performed a certain analysis, what it seems to mean, how angry they were to the results, etc. This kind of log can be kept in two ways. The common way is to copy and paste the analyses that you want to use into a word processor document and add explanatory text. Another way is to type comments into the SPSS output window (see figure).

Creating Datasets

General Guidelines

Datasets hold as much of your quantitative data as possible, and the mathematical operations that you perform on the data, in a single location. The more data you can bring together in one place, the more convenient your data analysis will be. Some guidelines for coding data and creating datasets follow.

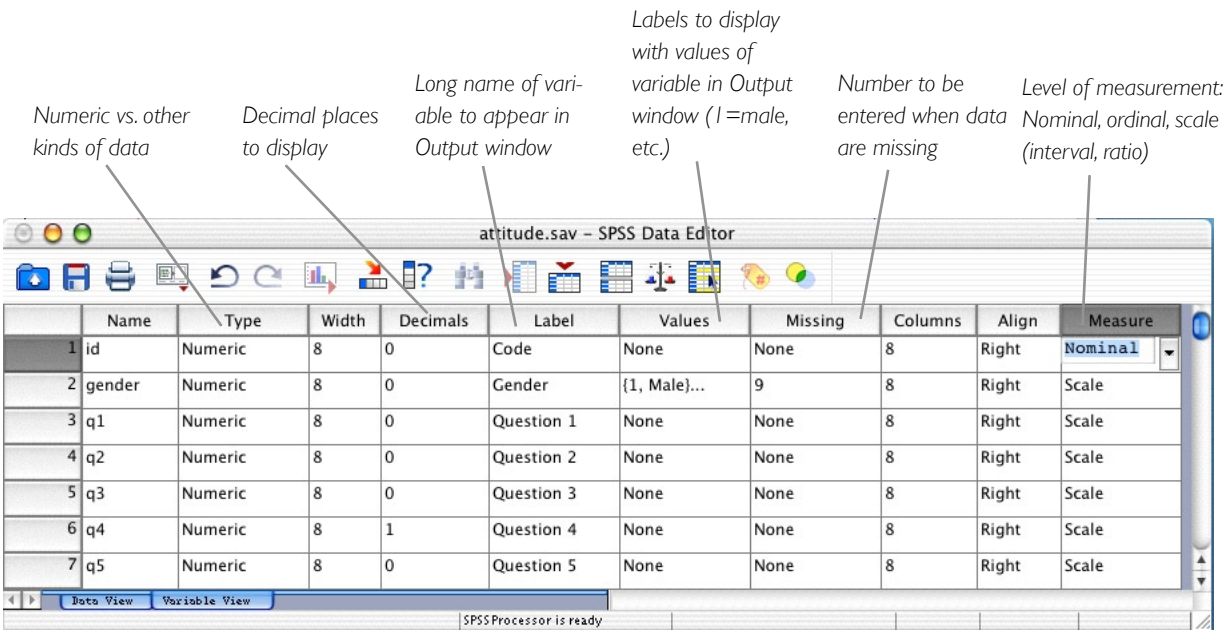
1. Give your data source IDs. Data source means wherever the data come from before they are coded in the dataset: questionnaires, observation sheets, lab device records, and so on. Give each subject's data source material an ID (any unique number), and include this ID as a variable in your dataset. The IDs will help you go back and forth between the data source and the data.

2. Code everything. Don't try to decide which variables you think you'll want to analyze and which you won't prior to performing the analysis. Instead, assuming you have the time, code everything you have into the dataset. Otherwise, you'll find it highly frustrating and distracting to code additional variables in the middle of your analysis.

3. Code missing values. Use the Missing Values property (see later section) to code missing values. Missing values are values of a variable that are...well...missing, for some reason. The most common reason is that you were not able to obtain the data: a missing questionnaire answers, the data gathering device broke down, you looked away for a minute. It is also possible that you will obtain data that are clearly wrong and should not be used: text answers on questionnaires that require numeric answers; insults written in by angry participants; people who just stopped responding/answering/talking. Find some impossible answers and use them as codes for various types of missing values.

4. Do some pilot coding. Don't assume that you'll be able to figure out exactly how to code the data before you start. Set up a dataset, code some cases, revise the dataset, code some more, until you have it right.

5. Clean the data before you analyze it. Cleaning refers to finding coding errors. The best way to clean the data is to have someone else check your coding against the data source. The second best way is to do some analyses that will identify impossible values. The latter method involves performing FREQUENCIES analyses on each variable to see if any have values outside the possible range. For example, the variable Age (of a person) cannot be 225 at the present time. Chances are the data coder accidently typed a '5' after typing '22'; or maybe she typed too many '2s'. Which is it? You can use the ID info to check the data source or, if the data source is gone, you must change the value to a missing variable value.



Creating Variables

Use the variables pane to create variables and set their properties (see figure). The process is intuitively simple and is best learned by hands-on experimentation. Some issues to consider:

1. Variable names can only be 8 characters long, and you cannot control upper vs. lower case. Use variable names that make some sense, then use Variables Labels to add more information that will appear in the Output window.

2. Everything (almost) is numeric in SPSS, including variables that you might normally think of as text (alpha). For example, gender is coded with numbers rather than "male" and "female" or "M" and "F". Use the Value Labels column to indicate what the numbers refer to, e.g., 1 indicates Male and 2 indicates Female. It is very important to set up these Value Labels so you don't commit an error interpreting your results later on.

3. Set missing values to impossible values (e.g., 9 for gender). You might need to use one missing value code for "just plain missing," one for "couldn't figure out the answer," and still another for "subject said a nasty thing."

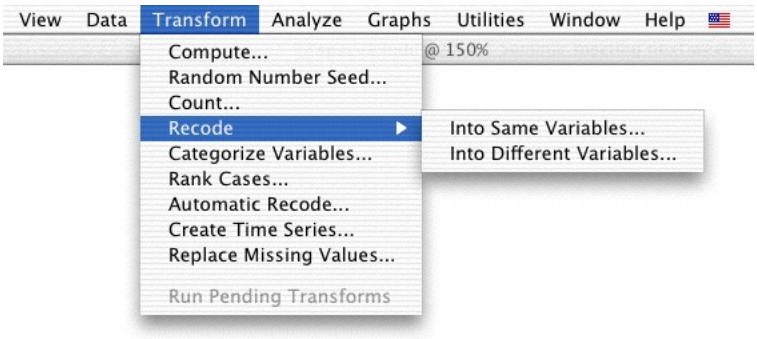
4. Decimal places are set for display on the data pane only and don't affect the underlying values. For integers, set this column to 0 decimal places.

Computing New Variables

The dataset component of SPSS does a lot of things besides hold on to your data. Most importantly, it allows you to transform your data in many way. For example, if you performed a survey study in which you measured "attitudes toward tuition hikes" on a five item scale, you might want to average the five items together to produce a single attitude score. In SPSS, this sort of math is done using a COMPUTE utility in the dataset component of the program. This and other data transformation procedures in SPSS are similar in style and capability to Microsoft Excel.

Like Excel, new variables are added as new columns in the dataset window.

Use the Transform menu to create new variables based on existing ones. The two most commonly used items in this menu are Compute and Recode. Compute performs Excel-like calculations using basic math (Q1 + Q2) or built-in functions. Recode allows you to change the values of a variable.



Using Compute

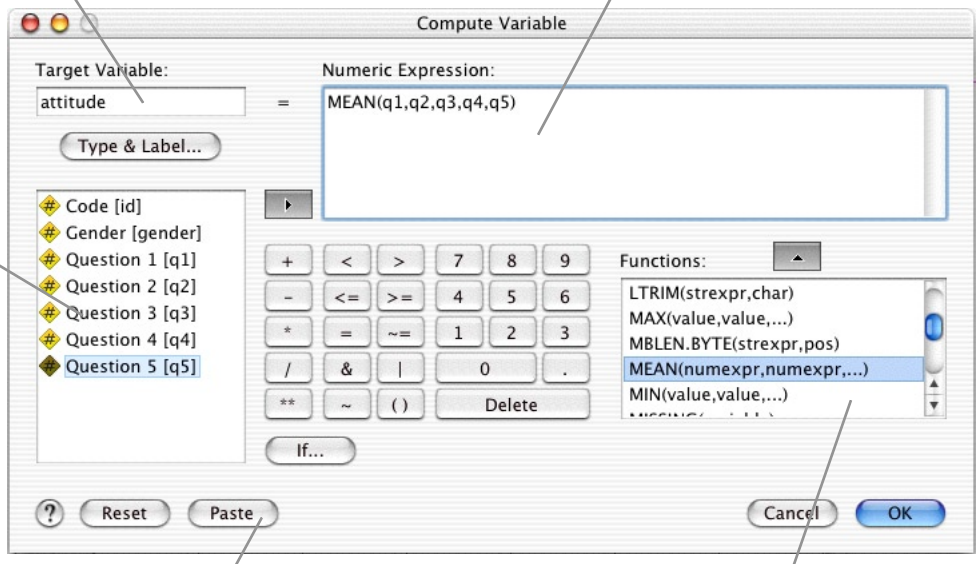
Like Excel, the Compute function allows you to create new variables from existing ones either manually, by typing in a formula, or by using built-in functions. The figure shows the computation of the mean of questions 1 - 5 to produce the new variable "attitude." When "OK" is clicked, the new variable is added to the dataset.

```
Syntax: COMPUTE attitude = MEAN(q1,q2,q3,q4,q5) .
```

Type the name of the variable you want to create here

Variables that go into functions are separated by commas

Double-click on a variable to enter it in the formula



Click on Paste to place the finished formula into the Syntax window for future use.

Double-click on a function to place it in the formula field

Using Recode

Recode is often used to create a new variable that "rearranged" the values of an existing variable by assigned them new values. The most common use of the function is to reduce the number of response categories of a continuous variable. For example, if Question 1 in the attitudes toward tuition hike survey where "Higher tuition is good because it allows the faculty to drive Beamers" and the response scale were a 5-point Likert scale:

___ Strongly agree
___ Agree
___ Not sure
___ Disagree
___ Strongly disagree

Perhaps you would like to combine "Strongly agree" and "Somewhat agree" so you can make the statement, "nn% of students agreed or strongly agreed that raising tuition is good idea so that their teachers can drive BMWs." A new variable in which "Strongly agree" and "Somewhat agree" were combined would allow you to directly obtain the value nn%.

Another common use is to "dichotomize" a variable. Dichotomize means break the variable into two values, usually "high" and "low" by splitting it up, usually at the median (midpoint). For example, the average of the five questions on this survey, "attitude," could be dichotomized at the median to form a new variable on which all the respondents would be either "favorable" or "unfavorable." Why would you do this? Perhaps you also assessed parental income in your survey, and you expected a gender difference in attitudes. You might then want to know the mean income of student broken down by gender and by attitude toward the tuition hike. By dichotomizing attitude, you have a 2x2 design (gender x attitude), and it would be easy to compute the four means in this design.

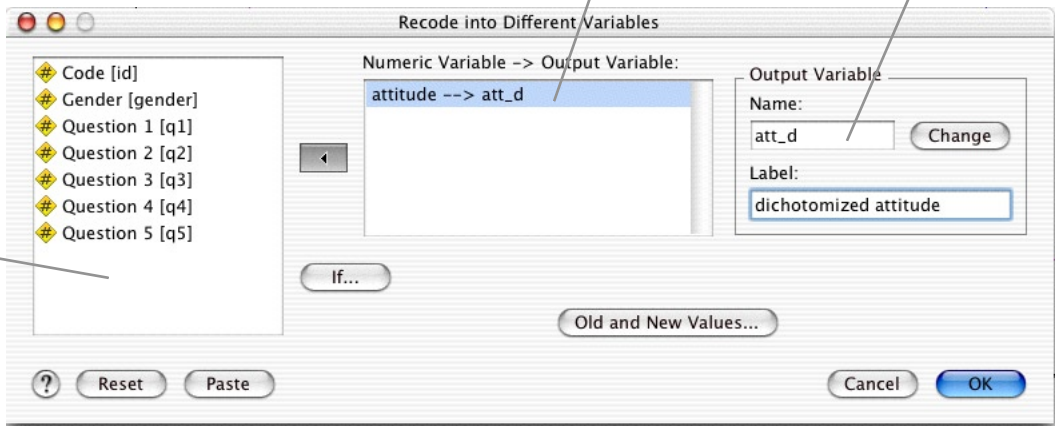
The Recode procedure is tricky and you must follow a series of steps very carefully to get it right. (See figures.)

1. Choose Transform -> Recode -> Into Different Variable
2. Double click on the existing variable that will form the basis for the new one
3. Type the new variable name in the appropriate field
4. Click on "Old and New Values"
5. Create the recode scheme
    a. Choose the range of values of the existing variable to be replaced by a value in the new variable
    (several possible ways to do this)
    b. Enter the value of the new variable to which this range should be recoded
    c. Click on Add. A new line appears in the Old -> New field
    d. Repeat (a) and (b) as needed
    e. Click on Continue to return to the original Recode screen
6. Click on Change in original Recode screen
7. Click on OK

6. This appears after you complete (2) and (3) and click on Change

3. Type new variable name here.

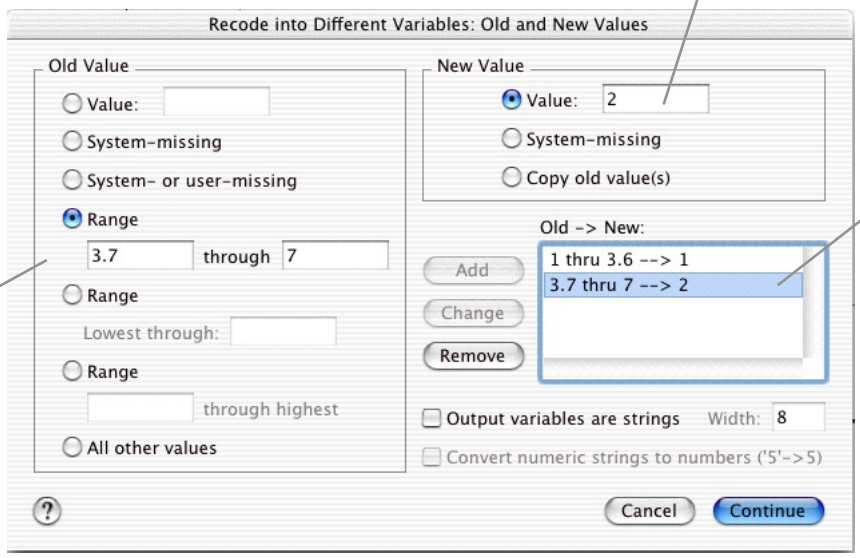
2. Double-click on the existing variable from which you will create the new one



5b. Value of new variable goes here

5a. There are several ways to indicate the values of the old variable that you want to recode into the values of the new one.

5c. Clicking on Add adds a new line in this field



After going through all of this, you may appreciate why some people prefer the old-fashioned, syntax-based way of doing things:

```
RECODE
  attitude
  (1 thru 3.6=1) (3.7 thru 7=2) INTO  att_d .
VARIABLE LABELS att_d 'dichotomized attitude'.
```