

Working With Your Data

Outline:

- I. Creating labels for your variables
- II. Creating and redefining variables -- Part I
- III. Creating and redefining variables -- Part II
- IV. IF/THEN Statements
- V. Converting Character Variables to Numeric Variables and Vice-Versa
- VI. Working with SAS dates
- VII. Stacking and merging SAS datasets

I. Creating labels for your variables

- A. Many times, variable names will be less than clear. The LABEL statement allows you to create a label for the variable that is up to 256 characters long.
 1. The labels will appear in PROC CONTENTS, as well as in other procedures such as PROC FREQ.

B. Example

```
LABEL exam_1 = 'Grade on First Midterm'  
exam_2 = 'Grade on Second Midterm'  
exam_f = 'Grade on Final Exam';
```

II. Creating and redefining variables -- Part I

A. Creating new variables

1. *exam_1 = 91;* creates a numeric variable from scratch
2. *exam_1 = 'A-';* creates a character variable from scratch

B. Redefining existing variables

1. *exam1 = exam_1;* creates a new variable from an old variable
2. *exam_1 = exam_1 + 10;* uses simple arithmetic to redefine an existing variable and overwrite it

C. Things to remember

1. If the *oldvar* is numeric, *newvar* will also be numeric (and vice-versa for character variables)
2. When using arithmetic operators, the basic rules of arithmetic apply (i.e., PEMDAS). You can use +, -, *, /, ^, etc.
3. Be careful when overwriting variables that you don't erase needed data
4. The arithmetic operators will not work with missing observations. They will return missing observations.

III. Creating and redefining variables -- Part II

- A. Sometimes it is easier to use the SAS functions to do many of the tasks mentioned above

B. Examples

1. *totalcredits = SUM(sem1, sem2, sem3, sem4);*
2. *finalgrade = MEAN(exam_1, exam_2, exam_3);*

C. Key to these functions

1. The commands will still work, even if some of the values are missing. The missing values are ignored and they won't tank the whole equation.
2. See Rob's photocopied handout for a list of many SAS functions (taken from pp. 70-71 of *The Little SAS Book*).

IV. IF/THEN Statements

A. Conditional Logic

1. Dictates that the first part of the expression needs to be true for the second part of the expression to take place.

B. Examples

1. *IF exam_1 > 10 THEN pass = 1;*
2. *IF exam_1 =< 10 THEN pass = 0;*

C. If-Then/Else Statements

1. Similar to if-then statements, but used if you want to group a variable in a series of different ways. More efficient than doing a series of IF/THEN statements. Additionally, it allows you to make sure that your groups are mutually exclusive.
2. Example

```
IF age > 30 THEN agecat = 3;  
ELSE IF age > 25 and age <= 30 THEN agecat = 2;  
ELSE IF age <= 25 THEN agecat = 1;
```

- D. Important to note – Missing values are considered low in SAS (even though they are considered high in STATA). When using < or <=, note that the missing values will be included unless you specify a lower bound.

V. Converting Character Variables to Numeric Variables and Vice-Versa

A. Character Variables to Numeric Variables

1. *newvar = INPUT(oldvar, 1.)*
2. The informat must be the type you are converting to. In this case, you would be converting '5' to 5.

B. Numeric Variables to Character Variables

1. *newvar = PUT(oldvar, 2.);*
2. The informat here must be the type you are converting from. In this case, you'd be changing 98 to '98.'

VI. Working with SAS Dates

A. SAS counts dates as the number of days since January 1, 1960

1. As a result, January 1, 2001 is seen by SAS as 14976

- B. If you need to work with dates, best to consult documentation or get a friend who can help you!

VII. Combining Datasets

A. Using the MERGE statement

1. Used when you've got two datasets with data on the same observation, but different variables (e.g., state by state data on health care data and welfare data) that you would want to combine.

2. Example:

```
DATA stateinfo;  
MERGE health welfare;  
BY statename;
```

B. Using the SET statement

1. Used when you've got two datasets with data on the same variables, but different observations (e.g., state by state from different years) that you would want to combine.

2. Example:

```
DATA statehealthdata;  
SET health1998 health1999;
```