DATA Step Information 4 – LABEL Statement

In an INPUT statement variables are specified. Variable names can be chosen to be lengthy, but SAS programmers often prefer shorter variable names. Rather than using a lengthy variable name to identify or define the variable, a LABEL statement can be used. A LABEL statement is used after the INPUT statement or after the SET or MERGE commands. The LABEL statement can be used in conjunction with an INFILE statement or when a DATALINES (or CARDS) statement is used.

```
DATA  one;
INPUT    x  $  y    z ;
LABEL    x  =  ' variable definition '   y  =  ' variable definition '
         z  =  ' variable definition '  ;
DATALINES;
```

In a LABEL statement, the variable definition can be up to 40 characters long including blanks. Enclose the text of the label in single or double quotes. Several variables can be specified in a single LABEL statement. Use a semicolon only after the last variable definition given. You may choose to label only some of the variables in a data table.

The LABEL statement only defines the variable name. It does not define the values that the variable can take. For that you need PROC FORMAT which will be covered later.

Labels allow you to use a simple naming convention for the variable names in the program, and have variables well defined in the output. If the labels are included in the DATA step, SAS will use them in the output for most procedures. One procedure that does not automatically use the labels is the PRINT procedure. If you'd like the PRINT procedure to use the labels, you must use the LABEL option on the PROC PRINT statement. (See The PRINT Procedure information.)

***Objective 3*:** Run the following program. Note the appearance of the labels on the output of the second and third procedures.

```
DATA one;
INPUT x y z;
LABEL x = 'height'  y = 'weight'  z = 'cost' ;
DATALINES;
65 215 132
32 44 87
;
PROC PRINT DATA=one;
TITLE 'Objective 3';
PROC PRINT DATA=one LABEL ;
PROC MEANS DATA=one;
RUN;
QUIT;
```