

How do I define variables in SAS tutorials?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How do I define variables in SAS tutorials?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=150155>

SAS tutorials provide step-by-step instructions on how to define variables in SAS programming. Defining variables in SAS involves assigning a name and a data type to a specific value or set of values. This process is crucial in organizing and manipulating data in SAS programs. By following the instructions provided in the SAS tutorials, users can effectively define variables and perform data analysis with confidence.

Defining Variables

A SAS dataset consists of columns of variables and rows of observations. Variables correspond to characteristics of the data that are documented for all (or most) of the observations. For example, the rows (observations) in your data set might represent people, and the columns in your data set would contain characteristics about the people (like gender, age, height, etc).

In a SAS dataset, variables themselves have five important properties: name, type, length, format, and label.

Variable name

Variable names are just that: they are a name used to refer to a variable in a dataset. When naming a variable in SAS, there are a few rules you must follow:

The name cannot contain more than 32 characters. The name can start with a letter or an underscore (_), but cannot start with a number. Numbers can be used after the first character. Blanks are not recognized in names. For example, you cannot name a variable "First Name" because SAS will not recognize the blank. Instead, use "FirstName" or "first_name".

It is good practice to give your variables descriptive yet succinct names. When possible, avoid generic variable names like x1 that don't provide any information about what the variable represents. Additionally, it is also good practice to assign a *label* to each of your variables. Variable labels provide information about a variable that might not fit into the variable name itself.

Variable types

In SAS, there are two types of variables: **numeric** and **character**.

Numeric variables are variables that store numbers. Typically, these are variables that you'll want to perform arithmetic calculations on, like addition and subtraction. However, numeric variables can also be used as *indicator variables* to represent categorical data, especially if the categories are ordinal. Additionally, date-time variables are also considered numeric in SAS. Missing values for numeric variables appear as a period (.).

Character variables (also known as *string variables*) contain information that the system recognizes as text. This can include letters, special characters (such as parentheses or pound signs), and even numbers. (Numeric values can be treated as characters if the numbers are used as labels and would not be used for meaningful mathematical calculations. For example, zip codes cannot be added or multiplied even though they are made up of numbers, but they are useful when treated as categories.) Missing values for character variables appear as a blank ("").

Variable length

The "length" of a variable in SAS corresponds to the number of bytes for storing variables (source: [SAS 9.2 Language Reference: LENGTH Statement](#)). The default number of bytes for both numeric and character variables is 8.

Variable format

A variable's format describes how it should be displayed in the SAS output. In SAS, this encompasses:

Number of decimal places to show for numeric variables (e.g., 1 versus 1.00)
Date format for dates (e.g. 06/19/2014 versus 2014JUN06 versus 2014-06-19)
Inclusion of commas, dollar signs, or other "prettifying" marks for numeric variables
Value labels for numerically-coded categorical variables (e.g., 1 = Male, 2 = Female)

Formats are crucial for helping readers understand your data and your output. Variable formats are detailed enough that we've split them into their own tutorials.

[Formats and Informats: How to apply SAS's built-in formats](#) (covers most quantitative numeric variables and character variables)
[User-defined Formats \(adding value labels\)](#): How to define your own formats (covers value labels for numerically-coded categorical variables)

Variable label

Besides formatting a variable, adding a variable label is another way to make your dataset and output easier to read and interpret. A label can provide information about a variable that you might not be able to incorporate into the variable name. For example, in our sample data of students we have a variable for the student's birth date and a variable for the student's enrollment date. If you want to create a variable for their current age and also a variable for the age they were when they started at the college, you might name the variables "age" and "age_start" respectively. At the time, it might be perfectly clear to you what the distinction is between the two variables. However, if someone else wants to look at your data, the variable names aren't descriptive enough to impart their full meaning. In this case, the variable "age" could be labeled "Student's current age", and the

variable "age_start" could be labeled "Student's age when they started attending the college". This way, anyone reading your data can tell exactly what is being measured.

Just like variable formats, variable labels can be assigned in a data step or a proc step. When a label is assigned in a proc step, it is temporary; it will only be associated with that variable during the execution of the proc step it is contained within. (That is, the label will only appear in the output of the procedure it's included in.) When a label is assigned in a data step, it becomes a permanent part of the dataset.

A variable label statement SAS looks like this:

```
LABEL variable_name = "Variable label";
```

Example

Let's add a label to the variable dob in our sample dataset. Before adding a label, note that when you view the dataset the name at the top of the column that represents the date of birth values is "bday."

	ids	bday	Gender	Rank
1	23643	22NOV1990	0	.
2	30953	23AUG1995	1	1
3	20531	29DEC1994	0	1
4	22416	.	0	1
5	41227	19APR1994	1	2
6	37301	06JUN1993	1	2
7	39181	17MAY1992	0	3
8	22652	04DEC1989	1	3
9	35684	29JUN1991	0	4
10	43344	26MAR1993	0	.

Let's temporarily add a label to the "bday" variable in a proc step. We'll demonstrate this using `PROC PRINT`. We will cover this and other proc steps later on, but for now the important thing to note is that you can put a label statement in a proc step to give the variable a label for the output you produce in the proc step. This will not change the label of the variable in the dataset.

```
PROC PRINT DATA=sample LABEL;  
VAR bday;  
LABEL bday = "Date of Birth";  
RUN;
```

Note in the image below that although the "bday" variable prints in the output with the label "Date

of Birth", the "bday" variable itself is unchanged in the dataset (lower panel).

The SAS System

Obs	Date of Birth
1	22NOV1990
2	23AUG1995
3	29DEC1994
4	.
5	19APR1994
6	06JUN1993
7	17MAY1992
8	04DEC1989
9	29JUN1991
10	26MAR1993

	ids	bday	Gender	Rank
1	23643	22NOV1990	0	.
2	30953	23AUG1995	1	1
3	20531	29DEC1994	0	1
4	22416	.	0	1
5	41227	19APR1994	1	2
6	37301	06JUN1993	1	2
7	39181	17MAY1992	0	3
8	22652	04DEC1989	1	3
9	35684	29JUN1991	0	4
10	43344	26MAR1993	0	.

To permanently add a label to the "bday" variable, put the label statement in a data step.

```
DATA students_formatted;
SET sample;
LABEL bday = "Date of Birth";
RUN;
```

	ids	Date of Birth	Gender	Rank
1	23643	22NOV1990	0	.
2	30953	23AUG1995	1	1
3	20531	29DEC1994	0	1
4	22416	.	0	1
5	41227	19APR1994	1	2
6	37301	06JUN1993	1	2
7	39181	17MAY1992	0	3
8	22652	04DEC1989	1	3
9	35684	29JUN1991	0	4
10	43344	26MAR1993	0	.

This will permanently assign the label "Date of Birth" to variable "bday" in the dataset "students_formatted", without making any changes to the dataset "sample".

For More Information

[SAS 9.2 Language Reference: SAS Variable Attributes](#)