How do I create and modify string (character) variables?

Authored by

stats writer

June 30, 2024

Creating and modifying string (character) variables involves declaring a variable with a string data type and assigning a value to it. This value can be a single character or a sequence of characters, enclosed in quotation marks. To modify a string variable, you can use various built-in functions and methods to manipulate the content of the string. These include concatenation, slicing, and replacing specific characters or substrings. Additionally, you can also use formatting techniques to customize the output of a string variable. Overall, understanding the syntax and available functions for string variables is crucial in creating and modifying them efficiently.

How do I create and modify string (character) variables? | SPSS FAQ

There are at least two ways to create a string variable in SPSS. In our first example, we show how to input string variables into a new data set. In the next example, we show how to create a string variable in an existing data set. In the last example, we will show how to removed unwanted characters from a string variable.

Example 1: Inputting string variables into a new data set

In this example, we will enter an id number, the first and last name, age and weight for nine folks. All of the variables will be numeric, except of course, the names. We will also save the file.

```
data list list / id * fname (A5) lname (A10) age wt.
begin data
1 "Beth" "Jones" 20 .
2 "Bob" "Jensen" 23 210
3 "Barb" "Andersen" 25 125
4 "Andy" "Smith" 26 160
5 "Al" "Peterson" 21 190
6 "Ann" "Glenn" 22 115
7 "Pete" "." 29 175
8 "Pam" "Wright" 21 145
9 "Phil" "Brown" 29 200
end data.
save outfile 'c:names.sav'.
```

The (A_) after fname and
lname tells SPSS that the variable(s) before that option are string variables, and they have a
length of five and ten, respectively. If you are listing only one string variable and there is one or more numeric variables listed before the string variable, you need to put an asterisk
before the name of the string variable to tell SPSS that the variables listed before the
asterisk are numeric variables. Hence, the asterisks (*)

after id is necessary because SPSS assumes that all variables
listed before (A8) option are string variables. The asterisk tells SPSS that all prior variables are numeric.

You may also notice that SPSS produced
an error message, shown below, while reading in the data. It was caused by the missing data
value for wt in case 1. Despite this error message, the data were read in
correctly, as we can see by using the list command. An error message was not
generated for the missing value in lname in case 7 because
"." is a valid value in a string variable. In other words, SPSS does
not consider it a missing value. We will return to this issue shortly.

>Warning # 1111
>A numeric field contained no digits. The result has been set to the
>system-missing value.

>Command line: 978 Current case: 1 Current splitfile

group: 1
>Field contents: '.'
>Record number: 1 Starting column: 21 Record length: 21

list.

ID FNAME LNAME AGE WT

1.00 Beth Jones 20.00 .
2.00 Bob Jensen 23.00 210.00
3.00 Barb Andersen 25.00 125.00
4.00 Andy Smith 26.00 160.00
5.00 Al Peterson 21.00 190.00
6.00 Ann Glenn 22.00 115.00
7.00 Pete . 29.00 175.00
8.00 Pam Wright 21.00 145.00
9.00 Phil Brown 29.00 200.00

Number of cases read: 9 Number of cases listed: 9

Example 2: Adding a string variable to an existing data set

Suppose that we would like to add a string variable

called
gender. First, we need to create the new variable using the string command. Then we will assign values to the variable.

string gender (A6).
execute.

Let's look at the frequency of a few variables to see how gender is different from the variables that we entered with the data list command.

freq var=lname wt gender
/format=notable.

	LNAME	WT	GENDER	
N	Valid	9	8	9
Missing	0	1	0	

Notice that although there are no values for
gender, there are also no missing values. (This is why you can not use the
nmiss function in aggregate.) In other words, SPSS considers a
blank to be a valid value for a string variable.

Now let's assign values to gender. We will use the compute and the if commands to do this. Remember that while you can modify a string variable with compute and if, you cannot create a string variable with these commands. (However, you can create a numeric variable with the compute or the if command.) Note that the value of a string variable must always be enclosed in quote marks.

```
compute gender = 'female'.
execute.
```

Of course, not everyone in our data set is female, so we need to change some of the values of gender. If we want to make the values of gender contingent on the value of another variable, we use the if command. In this example, we will use the vertical bars to indicate or.

```
if id = 2 | id = 4 | id = 5 | id = 7 | id = 9 gender = 'male'.
```

execute.

We can also use numeric values in string variables. Remember that
even if numeric values are used, SPSS still considers those values to be strings.

We can assign variable labels and value labels to string variables in the same way that we can assign them to numeric variables.

```
variable label gender 'This is the gender of the subject'.
value label gender 'male' 'm'
'female' 'f'.
execute.
```

Example 3: Combining string variables

In our current data set, the first name (called
fname) and the last name (called lname) are two different variables. Suppose
that we wanted to combine them into a single variable. To do this, we will create a new variable called
name1 with a length of 10. Next, we will use the concat function (short for

"concatenate") to combine the first and last name into a single variable.

```
string name1 (A10).
execute.

compute name1 = concat(fname, lname).
execute.list name1.
```

```
NAME1

Beth Jones
Bob Jense
Barb Ander
Andy Smith
Al Peter
Ann Glenn
Pete .
Pam Wrigh
Phil Brown

Number of cases read: 9 Number of cases listed: 9
```

As you can see, the length of name1 is too short. Although you

can use the alter type command (available in SPSS versions 16 and higher)
to make the variable name1 longer, we have already lost the information
at the end of some of the cases (in other words, some of the letters at the end
have already been cut off). Hence, simply making name1 longer isn't
helpful. Rather, we will need to create a new string variable (which we will call
fn) with a longer length and copy name1 into fn.

```
string fn (A15).
compute fn = concat(fname, lname).
execute.

list fn.
```

```
FN

Beth Jones
Bob Jensen
Barb Andersen
Andy Smith
Al Peterson
```

Ann Glenn
Pete .
Pam Wright
Phil Brown

Number of cases read: 9 Number of cases listed: 9

While this worked, it does not look exactly as we would like. (The unequal number of spaces between the first and last name
does not look good.) Therefore, let's create another string variable and call it
fullname. We will use the rtrim function, which will trim off any extra blanks
on the right of fname, and use the concat function to combine fname, a space, and
lname.

```
string fullname (A15).
compute fullname = concat(rtrim(fname), " ", lname).
execute.

list fullname.
```

FULLNAME

Beth Jones
Bob Jensen
Barb Andersen
Andy Smith
Al Peterson
Ann Glenn
Pete .
Pam Wright
Phil Brown

Number of cases read: 9 Number of cases listed: 9

Example 4: Deleting unwanted characters from a string variable

Sometimes you need to remove unwanted characters from
a string variable. For example, social security numbers are often given
with hyphens in them. The code below can be used to remove the hyphens.
First, we input a small data set. We use the list command to

ensure that the data were read in properly. Next, we create a string variable called strvar, which has a length of nine (a9). We use the compute command, the concat function (short for "concatenation") and the subst function (short for "substring") to assign the values to strvar. Finally, we use the list command again to see the results. The substring function is used to break apart each value of ssn. The first number (a.k.a. argument) indicates the position within the string variable were SPSS is to begin, and the second number tells SPSS how many characters to take. Hence, subst(ssn, 1, 3) tells SPSS to use the variable ssn, start at the first position in the variable and take three characters. For the row of data, that would be 123.

```
data list list / ssn(a11).
begin data.
```

123-45-6789
987-65-4321
132-54-9687
798-65-4213
end data.

list.

SSN

123-45-6789
987-65-4321
132-54-9687
798-65-4213

Number of cases read: 4 Number of cases listed: 4

string strvar (a9).
compute strvar = concat(substr(ssn, 1, 3), substr(ssn, 5, 2), substr(ssn, 8, 4)).

list.

SSN STRVAR

123-45-6789 123456789

987-65-4321 987654321

132-54-9687 132549687

798-65-4213 798654213

Number of cases read: 4 Number of cases listed: 4

We gratefully acknowledge Mr. Mark Casazza for writing the code used in this example and Jose Benuzillo for sending it to us.