

# How can I easily convert a string variable to a categorical numeric variable in SPSS?

Authored by  
**stats writer**

June 30, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I easily convert a string variable to a categorical numeric variable in SPSS?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=162053>

The process of converting a string variable to a categorical numeric variable in SPSS can be easily achieved by using the "Recode into Different Variables" function. This allows for the string variable to be recoded into a new variable with numerical values representing each category. This conversion is useful for data analysis and can be done efficiently in SPSS.

## How can I easily convert a string variable to a categorical numeric variable? | SPSS FAQ

Let's suppose that you received the following data set and were asked to analyze the data. You quickly notice that independent variable, group, is a string variable, but you want to try running an ANOVA anyway.

```
data list list / id * group (A8) score *.
```

```
begin data
```

```
1 "group 1" 57
```

```
2 "group 1" 65
```

```
3 "group 1" 70
```

```
4 "group 2" 45
```

```
5 "group 2" 80
```

```
6 "group 2" 81
```

```
7 "group 3" 66
```

```
8 "group 3" 60
```

```
9 "group 3" 70
```

```
10 "group 3" 80
```

**end data.**

**oneway score by group.**

Text: GROUP

A string variable was used in a variable list where only numeric variables are allowed.  
This command not executed.

**Unhappily, you get an error message indicating that the ANOVA cannot be run with group because group is a string variable (but please see the note at the end of this page). You can use the autorecode command to change group from a string variable, such as group, into a numeric variable with values corresponding to the groups. In other words, all values of "group 1" will be coded as one in the new variable, which we will call rcdgrp. The into subcommand tells SPSS to put the recoded values into a new variable. If this subcommand is omitted, the new values will overwrite the values in the original variable.**

**autorecode variables = group**

/into rcdgrp.

Let's run a crosstab with the old variable, group and the new variable, rcdgrp, to ensure that the recoding went as expected.

crosstabs tables = group by rcdgrp.

	Cases					
Valid	Missing	Total				
N	Percent	N	Percent	N	Percent	
GROUP * RCDGRP	10	100.0%	0	.0%	10	100.0%

	RCDGRP		Total			
group 1	group 2	group 3				
GROUP	group 1	3				3
group 2		3			3	
group 3				4	4	
Total	3	3		4	10	

We can see that the recoding was done correctly, so now we can conduct the ANOVA.

oneway score by rcdgrp.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	49.733	2	24.867	.153	.861

<b>Within Groups</b>	1138.667	7	162.667		
<b>Total</b>	1188.400	9			

**Let's take a moment to briefly describe how the autorecode command works. The autorecode command sorts the values of the variable to be recoded and then assigns them numeric values. By default, the values are assigned in ascending order. User-defined missing values are recoded into values higher than any nonmissing values. System-missing values remain system-missing.**

**(There are no system missing values in string variables; however, you can use autorecode with numeric variables that do have system missing values.)**

**In SPSS version 13, the blank subcommand was added. This allows you to specify how you would like missing values in a string variable to be handled**

**when autorecoded. (Remember that in a string variable, a missing value is**

**a null, or empty, string; it is simply a blank or empty cell in the Data**

**Editor.) The default option is valid, which means that**

**the empty**

**string will be coded as a valid value in the new numeric variable. You can**

**also use the missing option, causing blank string values to be autorecoded into a user-missing value higher than the highest nonmissing value.**

**You may be wondering about the differences between the**

**autorecode and the recode commands. The two commands are very similar. The main difference is that autorecode automatically assigns a numeric value to each unique string**

**value. It also creates value labels for the new numeric values that are**

**the original string values. With recode, you need to specify**

**the values for the new variable. If you want to add value labels to the**

**numeric values, you need to do that in a separate step using the value labels**

**command. Hence, autorecode is particularly useful when you have**

**numerous values that need to be converted.**

The examples below illustrate how you can autorecode multiple variables at once. Perhaps the most important thing to remember is that the variables must be positionally consecutive in the data set. However, they do not need to be numbered consecutively. If the variables that you want to autorecode are not positionally consecutive, you can make them positionally consecutive by using the save command with the keep subcommand, listing the variables in the necessary order.

```
data list list / a1 (A1) a2 (A1) a3 (A1) a4 (A1) a5 (A1).  
begin data.  
a b c d e  
f g h i j  
end data.
```

We will use the print subcommand to print out the old values, the new values and the labels of the new values.

```
autorecode a1 a2 a3 a4 a5  
/into newa newb newc newd newe
```

**/print.**

**A1 NEWA**

**Old Value New Value Value Label**

**a 1 a**

**f 2 f**

**A2 NEWB**

**Old Value New Value Value Label**

**b 1 b**

**g 2 g**

**A3 NEWC**

**Old Value New Value Value Label**

**c 1 c**

**h 2 h**

**A4 NEWD**

**Old Value New Value Value Label**

**d 1 d**

**i 2 i**

## A5 NEWE

Old Value New Value Value Label

e 1 e

j 2 j

The syntax above is equivalent to the syntax below. Note that you can use the keyword `to` when specifying the variables to be autorecoded.

```
autorecode a1 a2 a3 a4 a5 /into new1 to new5.
```

The following examples illustrate what happens when the variables to be autorecoded are not in positionally consecutive order. Notice the addition of the variable `newvar`.

```
data list list / a1 (A1) a2 (A1) a3 (A1) newvar * a4 (A1) a5 (A1).
```

```
begin data.
```

```
a b c 9 d e
```

```
f g h 6 i j
```

```
end data.
```

**With newvar located between a1 and a5, the following syntax will not work and an error message will be issued.**

**autorecode a1 to a5 /into b1 to b5.**

**>Error # 17008**

**>The number of new variable names must equal the number of old variable**

**>names.**

**>This command not executed.**

**In order to make the above syntax work, you would need to**

**specify the variables before and after newvar to be autorecoded. Although not shown here, you can use the keyword to twice to specify the variables to be autorecoded (for example: a1 to a3 a4 to a5).**

**autorecode a1 to a3 a4 a5 /into b1 to b5.**

**NOTE: While the oneway, anova, manova and discriminant commands require both the independent and the dependent**

**variables to be numeric, the glm command can be used with a string independent variable.**

ARABPSYCHOLOGY.COM