

How to Convert String Variables into Numeric Ones

BASICS DATA ANALYSIS T-TEST ANOVA CHI-SQUARE TEST

You are here: [Home](#)

→ [Basics](#) → [SPSS](#)

[FAQ](#) → [How to](#)

[Convert String](#)

[Variables into Numeric](#)

[Ones?](#)

Converting an **SPSS string variable** into a numeric one is simple.

However, there's a huge **pitfall** that few people are aware of: string

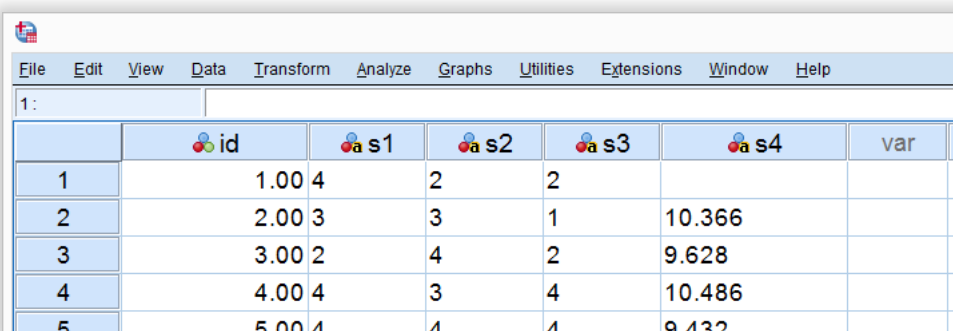
values that can't be converted into numbers result in **system missing**

values without SPSS throwing any error or warning.

This **can mess up your data** without you being aware of it. Don't believe

me? I'll demonstrate the problem -and the solution- on **convert-**

strings.sav, part of which is shown below.

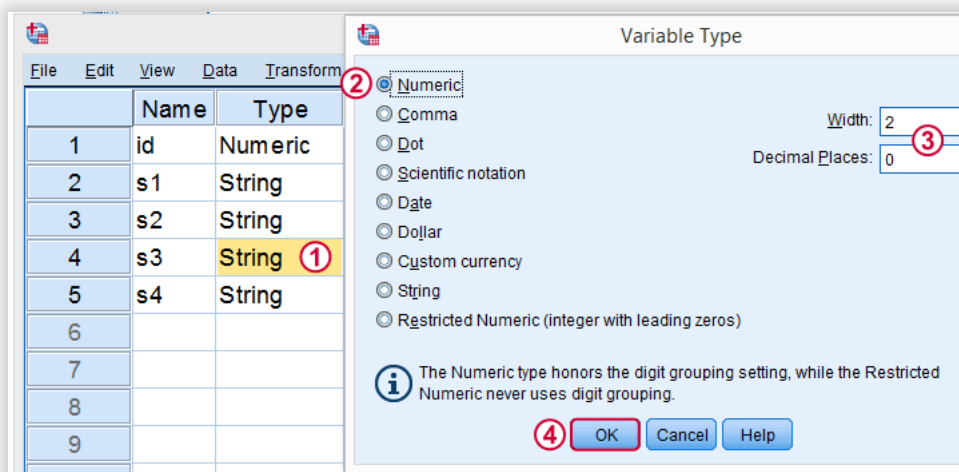


	id	s1	s2	s3	s4	var
1	1.00	4	2	2		
2	2.00	3	3	1	10.366	
3	3.00	2	4	2	9.628	
4	4.00	4	3	4	10.486	
5	5.00	4	4	4	9.432	

SPSS Strings to Numeric - Wrong Way

First off, you *can* convert a string into a numeric variable in **variable view**

as shown below.



Now, I never use this method myself because

AdChoices

SPSS Data Analysis

Free SPSS Software

Analyzing Data

- I can't apply it to **many variables** at once, so it may take way more effort than necessary;
- it doesn't generate any **syntax**: there's no **Paste** button and nothing's appended to my journal file;
- it can **mess up the data**. However, there's remedies for that.

So What's the Problem?

Well, let's *do it* rather than read about it. We'll

1. set empty cells as user missing values for s3;
2. convert s3 to numeric in variable view;
3. run descriptives on the result.

***Set empty string as user missing value for s3.**

missing values s3 ('').

***Inspect frequency table for s3.**

frequencies s3.

***Now manually convert s3 to numeric under variable vi**

***Inspect result.**

descriptives s3.

***N = 444 instead of 459. That is, 15 values failed to**

Result

s3

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1	29	6.0		
	1a	1	.2		
	2	115	24.0		
	2a	4	.8		
	3	159	33.1		
	3a	6	1.3		
	4	110	22.9		
	4a	3	.6		
	5	31	6.5		
	5a	1	.2		
	Total	459	95.6		
Missing		21	4.4		
Total		480	100.0		

Should be converted into 1?

We should have 459 valid values after converting to numeric.

Note that some values in our string variable have been flagged with "a". We probably want these to be converted into numbers. We have **459 valid values** (non empty cells).

Descriptives

Descriptive Statistics					
	N	Minimum	Maximum	Mean	Std. Deviation
s3	444	1	5	3.00	1.025
Valid N (listwise)	444				

After converting our variable to numeric, we ran some descriptives. Note that we only have N = 444. Apparently, **15 values failed to convert** - probably not what we want. And we usually **won't notice this problem** because we don't get any warning or error.

Conversion Failures - Simplest Solution

Right, so how can we **perform the conversion safely**? Well, we just

- inspected **frequency tables**: how many non empty values do we have before the conversion?

- **converted** our variable(s) to numeric;
- inspected N in a **descriptive statistics** after the conversion. If N is lower than the number of non empty string values (frequencies before conversion), then something may be wrong.

In our first example, the frequency table already suggested we must **remove the "a"** from all values before converting the variable. We'll do just that in a minute.

Although safe, I still think this method is too much work, especially for multiple variables. **Let's speed things up** by using some handy syntax.

SPSS - String to Numeric with Syntax

The fastest way to convert string variables into numeric ones is with the **ALTER TYPE** command.* It allows us to **convert many variables with a single line** of syntax.

The syntax below converts all string variables in one go. We then check a descriptives table. **If we don't have any system missing values, we're done.**

SPSS ALTER TYPE Example

***Close data without saving and reopen before proceeding**

***Convert all variables in one go.**

```
alter type s1 to s3 (f1) s4 (f6.3).
```

***Inspect descriptives.**

```
descriptives s1 to s4.
```

Note: using `alter type s1 to s4 (f1).` will also work but the decimal places for s4 won't be visible. This is why we set the correct f

format: `£6.3` means 6 characters including the decimal separator and 3 decimal places as in 12.345. Which is the format of our string values.

Result

	N	Minimum	Maximum	Mean	Std. Deviation
s1	480	1	5	3.04	.957
s2	461	1	5	3.02	.974
s3	444	1	5	3.00	1.025
s4	405	8.558	11.611	10.03869	.490713
Valid N (listwise)	357				

Since we've 480 cases in our data, we're done for `s1`. However, the other 3 variables contain system missings so we **need to find out why**. Since we can't undo the operation, let's close our data without saving and reopen it.

Solution 2: Copy String Variables Before Conversion

Things now become a bit more technical. However, readers who struggle their way through will learn a **very efficient solution** that works for many other situations too. We'll basically

- copy all string variables;
- convert all string variables;
- compare the original to the converted variables.

Precisely, we'll **flag non empty string values that are system missing** after the conversion. As these are at least suspicious, we'll call those conversion failures. This may sound daunting but **it's perfectly doable** if we use the right combination of commands. Those are mainly **STRING**, **RECODE**, **DO REPEAT** and **IF**.

Copy and Convert Several String Variables

***Close data without saving and reopen before proceeding**

***Copy all string variables.**

```
string c1 to c4 (a7).
```

```
recode s1 to s4 (else = copy) into c1 to c4.
```

***Convert variables to numeric.**

```
alter type s1 to s3 (f1) s4 (f6.3).
```

***For each variable, flag conversion failures: cases w**

```
do repeat #conv = s1 to s4 / #ori = c1 to c4 / #flags
if(sysmis(#conv) and #ori <> '') #flags = 1.
end repeat.
```

***If N > 0, conversion failures occurred for some vari**

```
descriptives flag1 to flag4.
```

Result

Descriptive Statistics					
	N	Minimum	Maximum	Mean	Std. Deviation
flag1	0				
flag2	0				
flag3	15	1.00	1.00		
flag4	52	1.00	1.00		
Valid N (listwise)	0				

Conversion failures:
system missing after conversion
but *not* empty before conversion.

Only flag3 and flag4 contain some conversion failures. We can visually inspect what's the problem by moving these cases to the top of our dataset.

***Visually inspect why values fail to convert.**

```
sort cases by flag3 (d).
```

***Some values flagged with 'a'.**

```
sort cases by flag4 (d).
```

***Some values flagged with 'a' through 'e'.**

Result

	id	s3	c3	c4	flag1
1	119.00	.3a	10.533d		
2	135.00	.4a	10.597		
3	157.00		9.694		
4	188.00		10.641		
5	194.00		10.168		
6	210.00		.3a	10.174	

Remove Illegal Characters, Copy and Convert

Some values are flagged with letters "a" through "e", which is why they fail to convert. We'll now fix the problem. First, we close our data without saving and reopen it. We then rerun our previous syntax but remove these letters before the conversion.

Syntax

***Close data without saving and reopen before proceeding**

***Copy all stringvars.**

```
string c1 to c4 (a7).
```



```
recode s1 to s4 (else = copy) into c1 to c4.
```

```
*Remove 'a' from s3.
```

```
compute s3 = replace(s3,'a','').
```

```
*Remove 'a' through 'e' from s4.
```

```
do repeat #char = 'a' 'b' 'c' 'd' 'e'.
```

```
compute s4 = replace(s4,#char,'').
```

```
end repeat.
```

```
*Try and convert variable again.
```

```
alter type s1 to s3 (f1) s4 (f6.3).
```

```
*Flag conversion failures again.
```

```
do repeat #conv = s1 to s4 / #ori = c1 to c4 / #flags
```

```
if(sysmis(#conv) and #ori <> '') #flags = 1.
```

```
end repeat.
```

```
*Inspect if conversion succeeded.
```

```
descriptives flag1 to flag4.
```

```
*N = 0 for all flag variables so we're done.
```

```
*Delete copied and flag variables.
```

```
delete variables c1 to flag4.
```

Result

Warnings

No statistics are computed because there are no valid cases.

Execution of this command stops.

All flag variables contain only (system) missings. This means that we no longer have any conversion failures; all **variables have been correctly converted**. We can now delete all copy and flag variables, save our data

and move on.

Thanks for reading!

Let me know what you think!

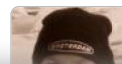
Done!

**Required field. Your comment will show up after approval from a moderator.*

This tutorial has 12 comments

By **Ruben Geert van den Berg** on January 19th, 2020

Interesting. I think this warning was added to recent SPSS versions as I don't remember seeing it when I wrote the article.



Expand [comment](#) | [all comments](#)

By Rogier on January 19th, 2020



The syntax:

string c1 to c4 (a7).

recode s1 to s4 (else = copy) into c1 to c4.

... is not working as s1...s4 and c1...c4 differ in length

AdChoices

[SPSS Data Analysis](#)

[Free SPSS Software](#)

[String Variables](#)

[1](#) ... [3](#)

Get In Touch!

SPSS Help (Netherlands)

Ruben Geert van den Berg

Sigma Plus Statistiek

LinkedIn

www.sigma-plus-statistiek.nl

Facebook

info@sigma-plus-statistiek.nl

SPSS Help (International)

SPSS tutorials

www.spss-tutorials.com

info@spss-tutorials.com

SPSS tutorials

© Copyright Protected 2019 by **Sigma Plus Statistiek**

Disclaimer

