Paper CC01

Compress It More

Linda Palm Simonsson, I-Mind, Lund, Sweden

ABSTRACT

Programmers have been using the COMPRESS functions for years, but with SAS 9.1.3, the function got more flexible and useful with its new argument. Today we can easily remove tabulations, character returns etc. that used to be a real pain to find and remove. Other possibilities are to remove all letters and keep only numbers and punctuations to be able to look at e.g. medication doses hidden in the text. This paper will give an overview of the new argument and some useful examples.

INTRODUCTION

There have always been problems with finding strange and hidden characters like tabulations and carriage return characters in text variables. The use of eCRFs has made this issue more common. Data is now being entered by PIs and study nurses who do not have the same data entry skills as dedicated data entry staff working with paper CRF.

OVERVIEW OF THE PROCEDURE

Syntax

```
COMPRESS(<source><, chars><, modifiers>)
```

Arguments

source

specifies a character constant, variable or expression from which specified characters will be removed.

chars

specifies a character constant, variable or expression that initializes a list of characters. By default, the characters in this list are removed from the ***source*** argument. If you specify the K modifier in the third argument, then only the characters in this list are kept in the result.

modifier

specifies a character constant, variable or expression in which each non-blank character modifies the action of the COMPRESS function. Blanks are ignored.

modifier	Description
a	adds alphabetic characters to the list of characters
c	adds control characters to the list of characters.
d	adds digits to the list of characters
f	adds the underscore character and English letters to the list of characters
g	adds graphic characters to the list of characters
h	adds a horizontal tab to the list of characters
i	ignores the case of the characters to be kept or removed
k	keeps the characters in the list instead of removing them.
l	adds lowercase letters to the list of characters.
n	adds digits, the underscore character, and English letters to the list of characters.
o	processes the second and third arguments once rather than every time the COMPRESS function is called. Using the O modifier in the DATA step (excluding WHERE clauses), or in the SQL procedure, can make COMPRESS run much faster when you call it in a loop where the second and third arguments do not change.
p	adds punctuation marks to the list of characters.
s	adds space characters (blank, horizontal tab, vertical tab, carriage return, line feed, and form feed) to the list of characters
t	trims trailing blanks from the first and second arguments.

modifier	Description
u	adds uppercase letters to the list of characters.
w	adds printable characters to the list of characters. Characters to be considered as non-printable by SAS, depends on TRANTAB system option settings.
x	adds hexadecimal characters to the list of characters

OLD FUNCTIONALITY

The old functionality that most programmers are used to is to remove blanks or specific characters, like in the following example:

```
data test;
  OldVar='ABC abc';
  *remove blanks*;
  NewVar1=Compress(OldVar);
  *remove every a and A in the text string*;
  NewVar2=Compress(OldVar,'aA');
  *remove every a and A and compress existing blanks in the text string*;
  NewVar3=Compress(OldVar,' aA');
run;

OldVar    NewVar1   NewVar2   NewVar3
ABC abc   ABCabc    BC bc     BCbc
```

NEW FUNCTIONALITY

The new functionality is the use of the modifier with which we can e.g. keep instead of delete, or remove all numbers or letters. For the purpose of looking at different ways of using it we have created a variable named Text:

```
Text=" the  abc 123/%!("||'09'x||")?+ qWerty § Ref 19.3-20,9 __";
```

‘09’x puts a tabulation in the text and the variable will print like the following

```
TEXT=the  abc 123/%!(      )?+ qWerty § Ref 19.3-20,9 __
```

.

Table 1 Examples - TEXT=the abc 123/%!( )?+ qWerty § Ref 19.3-20,9 __

#	SAS code	Will make TEXT2 result in[*]	Explanation
1	`Text2=compress(Text);`	`theabc123/%!(  )?+qWerty§Ref19.3-20,9__`	Removes the blanks but not the tabulation (the old functionality)
2	`Text2=compress(Text,,'kw') ;`	`the  abc 123/%!()?+ qWerty § Ref 19.3-20,9 __`	To remove tabulations, we can use a combination of two modifiers k (keep) and w (writable)
3	`Text2=compress(Text,,'w');`	`        `	Using only the w as a modifier will keep tabulations, character returns etc. that we cannot see.
4	`Text2=compress(Text,,'ka');`	`theabcqWertyRef`	ka as a modifier makes us keep all alphabetic characters
5	`Text2=compress(Text,' ','ka');`	`the  abc  qWerty  Ref`	To keep alphabetic characters and spaces add the ‘ ‘ to the second argument
6	`Text2=compress(Text,,'a');`	`123/%!(        )?+  §  19.3-20,9 __`	a as a modifier is used to remove all alphabetic characters
7	`Text2=compress(Text,,'ad');`	`/%!(    )?+  §  .-, __`	ad as a modifier is used to remove all alphabetic characters together with digits
8	`Text2=compress(Text,,'kpd');`	`123/%!()?+§19.3-20,9__`	kpd as a modifier is used to keep all punctuations and digits
9	`Text2=compress(Text,',./- ','kd');`	`123/    19.3-20,9`	With kd as a modifier is used to keep all digits
10	`Text2=compress(Text,'0123456789,./ ','k');`	`123/    19.3-20,9`	With k as a modifier you can specify the exact characters/digit to keep. Adding k as a modifier can be seen as the opposite of the old functionality
11	`Text2=compress(Text,,'u');`	`the  abc 123/%!(      )?+ qerty § ef 19.3-20,9 __`	With u as the modifier, all uppercase letters are removed
12	`Text2=compress(Text,'','ku');`	`W  R`	With ku as the modifier and ‘’ specified as second argument, uppercase letters and spaces are kept
13	`Text2=compress(Text,,'ku');`	`WR`	With ku as the modifier and no second argument, ONLY uppercase letters are kept

[*] Tabulation is represented with

RAW-DATA TO SDTM

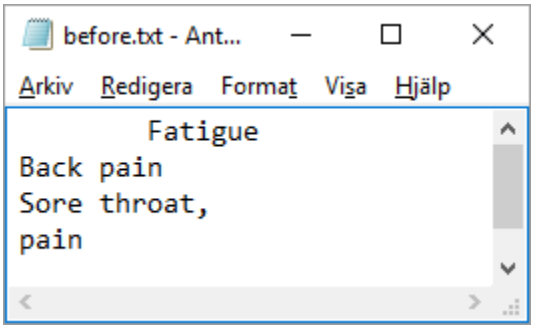
In raw data there are sometimes tabulations, carriage returns or other invisible characters wrongly entered in a text field (e.g. in AETERM or CMTRT). In a perfect world this is removed in the data entry system before database closure, but if left in the raw-data we are most often required to remove it by programming. If we know the hexadecimal code for a specific invisible character, we can remove them one by one like:

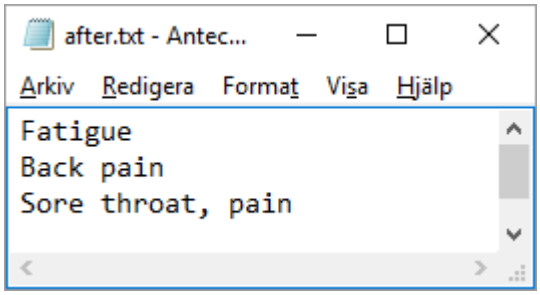
```
AETERM =compress(AETERM,'09'x);
```

This will solve the problem for Fatigue in the example below, Table 2, but if we want to remove the carriage return character in "Sore throat, pain" we need to find out the hexadecimal code for it and use the COMPRESS function again. However, if we use the modifier kw we don't have to mention specific hexadecimal codes since the modifiers kw will keep all the printable characters. This will make the code easier and more robust.

```
AETERM=compress(AETERM,,'kw');
```

Table 2 Remove invisible characters





COMBINATION TREATMENTS

Prior and concomitant medications are in some cases hard to collect in a straight forward way for combination treatments, and the doses are due to this other collected together with the trade name, like in Table 3. If we for some reason need to calculate the total daily dose of a certain substance for a subject, this can be a problem, but solvable with help of the modifiers in the compress function.

Table 3 Combination treatments in CM

#	USUBJID	CMTRT	CMDOSTOT	CMDOSU	CMDOSFRQ
1	ABC101	COMB 100/50 UG			BID
2	ABC101	COMB 100/50UG	200.0	ug	BID
3	ABC102	COMB 120/60 UG			QD
4	ABC102	COMB	0.1	mg	BID
5	ABC103	COMB 0.1/0.05 MG			BID

For observation 2 and 4 we have the total daily dose, but this needs to be calculated for the rest of the observations. To do this we can follow some simple steps

1) Select the treatment like
   a. `where index(CMTRT,'COMB'); or`
   b. `where ATCCODE in (<…>); or`
   c. `where CMDECOD in (<…>);`
2) Look at the patterns in CMTRT and decide what you need to keep to be able to look at the doses. In this case we need to keep digits and punctuations
   a. `Dose_in_Text=compress(CMTRT,'0123456789./','k'); or`
   b. `Dose_in_Text=compress(CMTRT,,'kpd');`
3) Keep the first dose only
   a. `First_Dose=input(scan(Dose_in_Text,1,'/'),best12.);`
4) The unit is found in either CMDOSU or last in CMTRT. To find the last word with no digits we can use compress together with scan
   a. `Last_Word=compress(scan(cmtrt,-1),,'d');`

5) The Unit can be created like
   a. `if cmdosu ne '' then unit=cmdosu; else if unit='' and upcase(last_word) in ('MG' 'UG') then unit=lowcase(last_word);`
6) Calculate the total daily dose for each observation like
   a.
```
if unit='mg' then Conv=1000; else
if unit='ug' then Conv=1;
if CMDOSTOT ne . then Total_Daily_Dose_in_ug=CMDOSTOT*conv;
else do;
  if CMDOSFRQ='QD' then Total_Daily_Dose_in_ug=First_Dose*conv; else
  if CMDOSFRQ='BID' then Total_Daily_Dose_in_ug=First_Dose*conv*2;
end;
```

Table 4 Combination treatments – Calculations done

#	DOSE_IN_TEXT	FIRST_DOSE	LAST_WORD	UNIT	CONV	TOTAL_DAILY_DOSE_IN_UG
1	100/50	100	UG	ug	1	200
2	100/50	100	UG	ug	1	200
3	120/60	120	UG	ug	1	120
4			COMB	mg	1000	100
5	0.1/0.05	0.1	MG	mg	1000	200

CONCLUSION

The new argument, the modifier, can be of help both in exploring text-strings for specific patterns/character or in getting rid of specific types of characters in a simple way.

REFERENCES

http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a000212246.htm

ACKNOWLEDGMENTS

Thanks to all my colleagues at I-Mind Consulting.

RECOMMENDED READING

Non Printable & Special Characters: Problems and how to overcome them,
http://www.lexjansen.com/nesug/nesug10/ff/ff04.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Linda Palm Simonsson
I-Mind Consulting
Magle Stora Kyrkogata 7A
SE-223 50 Lund, Sweden
Work Phone: +46 46 10 25 05
Email: linda.simonsson@i-mind.se
Web: www.i-mind.se