

Split Character Variables into Meaningful Text

Savithri Jajam, Covance Inc., USA
Vamshi Matta, Covance Inc., USA
Lavanya Peddibhotla, Covance Inc., USA

ABSTRACT:

The vast majority of clinical trial data must meet the CDISC standard in order to go for a submission, and one of the primary criteria is to have 200 lengths for any character variable. One of the challenges to follow this is when we have data of more than 200 and we have to split it into multiple variables with meaningful text. To achieve this, we can use different approaches. Few methods will be discussed in this paper with examples.

INTRODUCTION:

All variables in SDTM datasets should have maximum length of 200. But some of the datasets might have more text than 200 characters. For example, Comments (CO), Deviations (DV). But still we need to split them into multiple variables and move extra variables into supplementary dataset because CDISC will not allow them in the parent or main dataset. To do this we have so many methods and approaches. Few of them are explained in this paper.

In this paper 4 methods are discussed. 3 simple procedures with different approaches and a macro to create any number of variables based on the length of the text.

Method 1:

In this method, first **LENGTH** function was used to know the length of the variable and **SUBSTR** was used to strip the text until needed. Used length function again to exclude any words ending close to 200 length.

But this method has limitations and can't use if text is very lengthy, and we need to manually check how many variables we have to derive.

Source Code:

```
data qcl;  
set source.dv;  
  describe_deviation=strip(compbl(describe_deviation));  
  length=length(describe_deviation);  
  c=substr(describe_deviation,201,1);  
  if .<length<=200 and missing(c) then dvterm=describe_deviation;  
  else if 200<length<=400 or not missing(c) then do;  
    if substr(describe_deviation,200,1)="" or substr(describe_deviation,201,1)=""  
  then do;  
    dvterm=substr(describe_deviation,1,200);  
    dvterm2=strip(substr(describe_deviation,201));  
  end;  
  else do;  
    length1=200-length(scan(substr(describe_deviation,1,200),-1,""));  
    dvterm=substr(describe_deviation,1,length1);  
    dvterm2=strip(substr(describe_deviation,length1+1,200));  
  end;  
end;  
run;
```

Picture 1:

△ DVTERM	△ dvterm2	△ Describe_Deviation	⊕ length
[Day 43] The subject'...	(13May2017 through...	[Day 43] The subject's...	224
Blood transfusion his...	document with and ...	Blood transfusion hist...	248
[0044-301A] Day 18...	was informed in adv...	[0044-301A] Day 183 ...	224
[0045-301A]Day71- ...		[0045-301A]Day71- T...	172
[0045-301A] Day71- ...		[0045-301A] Day71- ...	129
[0045-302A] Day 57-...		[0045-302A] Day 57- ...	157

Method 2:

In this method, first **LENGTH** function was used to know the length of the variable and **FINDC** was used to strip the text until needed. Example is shown on CM coding variable. But this method also has limitations and can't use if text is very lengthy, and we need to manually check how many variables we have to derive.

Source Code:

```
data adcm;
set cm;
len=length(Level_5_Text);
length orig $400. cmdecod cmdecod1 $200;
    orig=Level_5_Text;
    pos=findc(orig, ' ', -201);
    cmdecod=strip(substr(orig, 1, pos));
    cmdecod1=strip(substr(orig, pos+1));
run;
```

Picture 2:

△ Level_5_Text	⊕ len ↓	△ cmdecod	△ cmdecod1
ALANINE;ARGININE;CAL...	303	ALANINE;ARGININE...	DIBASIC;PROLINE;SE...
ALANINE;ARGININE;CAL...	303	ALANINE;ARGININE...	DIBASIC;PROLINE;SE...
ASCORBIC ACID;BIOTIN;...	263	ASCORBIC ACID;BI...	NOS;PYRIDOXINE;RE...
(RS)-3 METHYL-2-OXOVA...	230	(RS)-3 METHYL-2-O...	ACETATE;THREONIN...
(RS)-3 METHYL-2-OXOVA...	230	(RS)-3 METHYL-2-O...	ACETATE;THREONIN...
ANGELICA ACUTILOBA R...	201	ANGELICA ACUTIL...	ROOT
ASCORBIC ACID;BIOTIN;...	193	ASCORBIC ACID;BI...	
ARGININE;BETA-ALANIN...	184	ARGININE;BETA-AL...	

Method 3:

Used length, **SUBSTR** and **REVERSE** functions. Example is shown on DV coding variable.

In this method, first **LENGTH** function was used to know the length of the variable and **REVERSE** function was used to know the word position and again **SUBSTR** function used to strip. This method is similar to method 1 but used **REVERSE** function.

Sample Code:

```
data dv1;
set source.dv;
if length(Describe_Deviation)>200 then do;
  if substr(strip(Describe_Deviation),201,1)="" or
    substr(strip(Describe_Deviation),200,1)="" then do;
    dvterm=substr(strip(Describe_Deviation),1,200);
  end;
  else if substr(strip(Describe_Deviation),200,1)^="" then do;
    term1=substr(strip(Describe_Deviation),1,200);
    term2=reverse(term1);
    term3=scan(strip(term2),1,"");
    len1=200-length(term3);
    dvterm=substr(strip(term1),1,len1);
  end;
end;
else if length(Describe_Deviation)<=200 then dvterm=strip(Describe_Deviation);
*DVTERM2;
if length(Describe_Deviation)>200 then length=length(strip(Describe_Deviation))-200;
if length(Describe_Deviation)>200 and (substr(strip(Describe_Deviation),201,1)=""
or substr(strip(Describe_Deviation),200,1)="" ) then do;
  dvterm2=strip(substr(strip(Describe_Deviation),201,length));
end;
else if length(Describe_Deviation)>200 and substr(strip(Describe_Deviation),200,1)^=""
then do;
  term4=reverse(term3);
  dvterm2=strip(term4)||strip(substr(strip(Describe_Deviation),201,length));
end;
run;
```

Picture 3:

△ dvterm	△ dvterm2	△ Describe_Deviation ↑	⊕ len
0263-303A : site a...	missing on ICF.	0263-303A : site address is present ...	213
0351-301A: D15 v...		0351-301A: D15 visit:The PK post-d...	188
0351-301A: Rand...	arm.	0351-301A: Randomization visit:The...	204
0351-301A: The H...		0351-301A: The HB for randomizati...	100
0351-301A:The sit...		0351-301A:The site is performing ax...	176
0351-301A:The sit...		0351-301A:The site is performing ax...	180

Method 4:

This is a simple macro to split into unlimited variable based on the length of the text. Very easy to adopt and use. Only 3 macro parameters are used but can be updated and add output variable name as well.

In this macro also, as a first step used length function to know length of the variable and then calculated to how many times it must split and wrote a do loop to execute as many times as need.

Device related data was taken as example and the length of the text was very large. So it has to create 33 extra variables which was done quickly and effectively without manual effort. Log was also shown in pic 5 to show the length of the variable and how many variables it has to split.

Macro parameters:

```
%split(indsn= <input dataset name>,  
outdsn= <output dataset name>,  
invar= <input variable with large text>);
```

Sample Code:

```
%macro split(indsn=,outdsn=,invar=);  
  
proc sql feedback noprint;  
    select distinct max(length(strip(&invar))) into :maxlen from &indsn;  
    %let maxlen=%trim(%left(&maxlen.));  
    %put maximum length of deter is &maxlen.;  
quit;  
  
*****Maximum number of variables required for DEEVDCXX***;  
data a;  
    set &indsn;  
    &invar = compbl(compress(&invar, , 'kw'));  
    dleng=length(strip(&invar));  
    leng=floor(dleng/200);  
run;  
  
proc sql noprint feedback;  
select max(leng)+1 into : obs from a;  
quit;  
  
%put &obs;  
  
data &outdsn;  
set &indsn;  
length devtext $7000. ;  
dleng=length(strip(&invar));  
devtext=strip(&invar);  
  
%do i= 1 %to &obs;  
    length DEEVDC&i $200;  
    pos=findc(devtext, ' ', -201);  
    DEEVDC&i=strip(substr(devtext, 1, pos));  
    devtext=strip(substr(devtext,pos+1));  
%end;  
keep DEEVDC: &invar;  
run;  
  
%mend split;  
%split(indsn=de4,outdsn=de5,invar=Event_Description);
```

Picture 4:

var1	DEEVDC1	DEEVDC2	DEEVDC3
A giant contai...	A giant container ship ...	they hoped the ship co...	side.

Picture 5:

```
maximum length of deter is 400
NOTE: PROCEDURE SQL used (Total process time):
      real time          0.00 seconds
      cpu time           0.00 seconds
```

```
select MAX(A.leng) + 1
into :obs
from WORK.A;
```

```
NOTE: PROCEDURE SQL used (Total process time):
7
```

```
real time          0.00 seconds
cpu time           0.00 seconds
```

```
Number of variables to be splitted obs= 3
```

Summary:

- 1) These methods will identify length of the text and split into multiple variables and that can be moved into SUPP datasets as per CDISC standards.
- 2) Above 4 methods are easy to use in any folder and any dataset including Tables and Listings.
- 3) First 3 methods need manual programming effort but method 4 will work for any text and creates n number of variables without programming effort.
- 4) The macro can identify and/or split into additional variables in a readable way complying CDISC standards.
- 5) The scope of this macro is for all datasets in a specified library

References:

- 1) <https://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/FormsSubmission%20%20Requirements/ElectronicSubmissions/UCM511237.pdf>.
- 2) Richard Addy and Charity Quick. "BreakOnWord: A Macro for Partitioning Long Text Strings at Natural Breaks". Proceedings of the PharmaSUG 2014 Conference. Available at <http://www.pharmasug.org/proceedings/2014/CC/PharmaSUG-2014-CC20.pdf>.
- 3) SAS Institute Inc. 2011. SAS® 9.2 Language Reference: Dictionary, Fourth Edition. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/lrdict/64316/PDF/default/lrdict.pdf>.
- 4) Chunxia Lin. "Methods to Derive COVAL- COVALn in CO Domain". Proceedings of the PharmaSUG 2013 Conference. Available at <http://www.pharmasug.org/proceedings/2013/CC/PharmaSUG-2013-CC08.pdf>.

Acknowledgments:

Author would like to thank the employer Covance, Inc. for the approval to present this paper at this PHUSE.

CONTACT INFORMATION:

Your comments and questions are valued and encouraged. Contact the author at:

Savithri Jajam Covance, Inc.

savithri.jajam@covance.com

Vamshi Matta

vamshi.matta@covance.com

Lavanya Peddibhotla Covance, Inc.

lavanya.peddibhotla@covance.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.
Other brand and product names are trademarks of their respective companies.