PharmaSUG 2012 - Paper CC22

A Macro to Indent and Hyphenate Lengthy Text in PROC REPORT

Stanley Yuen, SimulStat Inc., San Diego, CA

ABSTRACT

A SAS programmer can have long text values in a character variable to wrap in multiple lines using the FLOW option and SPLIT character assignment in PROC REPORT. This option is limited in allowing additional leading blank characters as indentations for each line of output. It can also sometimes break up a word with the remaining characters in the next line of output. When there is flowing text, it is preferable to insert a hyphen to indicate a word fragment and avoid unsightly spacing in left-justified text. This paper discusses a macro that modifies a character variable to allow the FLOW option and SPLIT character to produce indented hyphenated text in PROC REPORT using Base SAS 9 functions in the data step. The macro is also flexible in designating different indentations for the first line of text compared to the other lines for each observation in a report.

INTRODUCTION

Lengthy text values of character variables can be broken up into multiple lines within its column using the FLOW option in PROC REPORT. Another benefit to this option is that it will observe any assigned SPLIT character. However, the option is limited because it does not maintain indents added to the start of the text for subsequent lines and it is unable to prevent words from being fragmented.

This paper introduces a macro that can be called within a data step to modify the contents of a character variable to insert user-specified indentations and hyphens to create wrapped lines and split words when used by PROC REPORT. Now when tables or listings for clinical trial submissions are generated using PROC REPORT, the text can be aligned properly and split words can be identified with hyphens.

MACRO PARAMETERS

The macro consists of six parameters and is to be called within a data step prior to using PROC REPORT. The user needs to define the name of the text variable that will be modified. The remaining five macro parameters have default values and can be adjusted according to the user's specifications. Table 1 summarizes all the macro parameters along with their definitions and default values.

The result from running this macro is a new variable consisting of the original character variable indented and hyphenated (if called) ready to be used in PROC REPORT. The name of this newly created variable is the name of the original variable with the prefix NEW_ added to it.

PARAMETER	DEFINITION	DEFAULT VALUE
VARIABLE	The user-defined text variable to be indented and hyphenated.	No default value. Must be defined by the user.
SPLIT	The character used to split the text in PROC REPORT.	Default is /.
COLUMN_WIDTH	The width of the column of the text variable in PROC REPORT.	Default is 25.
INDENT_FIRST	The number of spaces to indent the first row of text within an observation for PROC REPORT.	Default is 0.
INDENT_REST	The number of spaces to indent the remaining rows of text within an observation for PROC REPORT.	Default is the same as INDENT_FIRST.
HYPHEN	Option to separate words with a hyphen.	Default is YES.

Table 1. Macro parameters with definitions and default values.

MACRO SOURCE CODE

The following is the source code of the macro with bolded numbered steps to be interpreted in the next section of the paper.

```
  %macro indent(variable, split=%nrbquote(/), column_width=25, indent_first=0,
                indent_rest=&indent_first, hyphen=yes);

1  attrib new_&variable length=$200 piece length=$&column_width. rest length=$200;
   rest=&variable;

2  if not missing(rest) then do until (missing(rest));
3  indent=ifn(missing(new_&variable), &indent_first, &indent_rest);
   stringsize=&column_width-indent;

   select;
4a %if &hyphen=yes %then %str(
   when (notalpha(substr(rest, stringsize-1, 3))=0) do;
   piece=cat(substrn(repeat('', indent), 2), substr(rest, 1, stringsize-1), '-');
   cutoff=stringsize;
   end;);
4b %if &hyphen^=yes %then %str(
   when (notalpha(substr(rest, 1, stringsize))=0) do;
   piece=cat(substrn(repeat('', indent), 2), substr(rest, 1, stringsize));
   cutoff=stringsize+1;
   end;);
4c when (findc(char(rest, stringsize+1), '_-', 'bsp')) do;
   piece=cat(substrn(repeat('', indent), 2), substr(rest, 1, stringsize));
   cutoff=stringsize+1;
   end;
4d otherwise do;
   piece=cat(substrn(repeat('', indent), 2), substr(rest, 1, findc(substrn(rest, 1, stringsize),
   '_-', 'bsp')));
   cutoff=findc(substrn(rest, 1, stringsize), '_-', 'bsp')+1;
   end;
   end;

5  rest=strip(substr(rest, cutoff));
6  new_&variable=catt(new_&variable, piece, "&split");
   end;
   drop indent stringsize piece cutoff rest;
   %mend indent;
```

EXPLANATION OF MACRO ALGORITHM

The macro essentially performs three tasks: indent the text, hyphenate words at the end of each row if called, and add a split character. First, the macro stores a copy of the text value into a temporary variable called `rest`. Then it cuts a substring of the text value, indents and hyphenates the text, and then stores the result into another temporary variable called `piece` while retaining the remainder text in `rest`. Finally, a split character is added to the end of the text and stored in the variable `new_&variable`. This cycle repeats with each subsequent treated text added to `new_&variable` until there is no more text left to process in `rest`.

The following is a breakdown of the logic behind the numbered steps in the macro source code:

1. Define the attributes of the variables corresponding to the final new variable created to be displayed in PROC REPORT (`new_&variable`), the substring of text to be processed (`piece`), and the remaining text (`rest`).

   Initialize `rest` to be the same value as the original text variable that you want to indent and hyphenate.

2. The next steps are done in a DO loop that repeats until there is no more text in `rest` left to process.

3. Define two variables, `indent` and `stringsize`, respectively as the number of blanks characterizing the indentations and the number of characters that can be displayed in the column in PROC REPORT factoring in the indentations.
4. Process each string of text by doing one of the following and store the value in `piece`:
    a. If the user wants to separate words with a hyphen, first add the number of blanks from `indent` followed by the amount of text in `rest` up to one less than the cutoff number specified in `stringsize`. Finally, add a hyphen to the end of the string of text.

       The starting position of the next string of text in `rest` to be processed starts at the position number of where the hyphen is added and is stored in `cutoff`.

    b. If the user does not want to separate words with a hyphen and the string of text to be processed is all letters, then add the number of blanks from `indent` followed by the amount of text in `rest` up to the cutoff number specified in `stringsize`.

       The starting position of the next string of text in `rest` to be processed starts at the position number after `stringsize` and is stored in `cutoff`.

    c. If there's a natural break (blank, underscore, hyphen or punctuation mark) at the position number after `stringsize`, then add the number of blanks from `indent` followed by the amount of text in `rest` up to the cutoff number specified in `stringsize`.

       The starting position of the next string of text in `rest` to be processed starts at the position number after `stringsize` and is stored in `cutoff`.

    d. For all other strings of text, add the number of blanks from `indent` followed by the amount of text in `rest` up to the position number of the first natural break (blank, underscore, hyphen or punctuation mark) to the left of the cutoff number specified in `stringsize`.

       The starting position of the next string of text in `rest` to be processed starts at the position number after the natural break and is stored in `cutoff`.

5. Reset the value of `rest` to be the remaining text to be processed starting at the number stored in `cutoff`.
6. Finally, add the split value and store the processed text in the variable `new_&variable`.

NOTES AND LIMITATIONS OF MACRO

Some notes about the macro:

- In step 4a, a word that is to be separated with a hyphen is defined as having at a minimum 1 letter to the left and 1 letter to the right of the cutoff value stored in `stringsize`.
- The REPEAT function returns a character value repeated n+1 times, so the starting position of the indentation begins at position 2 in order to discount the extra blank added from the function.
- The SUBSTRN function returns a result with a length of zero if the value of the length is non-positive. A note to the log stating that the third argument is invalid is not written when this function is used compared to the SUBSTR function.
- The 'b' option in the FINDC function searches from right to left instead of from left to right. The 's' and 'p' options add space characters and punctuation marks to the list of characters that the FINDC function searches.

There are some limitations to this macro. Since the macro looks at just 3 characters to determine if a hyphen should be used, the hyphen may not necessarily be correct grammatically. Punctuation marks can potentially be wrapped to the next row of text if it is after the cutoff position. ODS usage is limited as the macro assumes monospaced fonts like Courier are used. If different fonts are designated, the output may not fully utilize the column space allocated for the text.

EXAMPLE 1

The following is an example of the application of the macro utilized for a table summary of adverse events. In this data step, the macro is run for the body system and preferred term variables `aesys` and `aeterm`. The resulting values are stored in the variables `new_aesys` and `new_aeterm`. A final variable `finalae` combines the values from both macro runs so that body system and preferred term values are displayed below one column in PROC REPORT. Output 1 illustrates what the PROC REPORT might look like for the `finalae` variable.

```
data aetable;
length finalae $100;
set ae;
by aesys aeterm;
%indent(variable=aesys);
%indent(variable=aeterm, indent_first=3);
if first.var1 then do;
finalvar=new_aesys;
output;
end;
finalvar=new_aeterm;
output;
run;
```

```
Body System Term
   Preferred Term           Treatment 1   Treatment 2   Treatment 3   Total
-------------------------   -----------   -----------   -----------   ----------
Subjects having at least    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
one adverse event

Blood and lymphatic syst-   xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
em disorders
   Iron deficiency anaem-   xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
   ia
   Thrombocytopenia         xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)

Cardiac disorders           xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
  Atrial fibrillation       xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
  Atrioventricular block    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
  second degree
```

Output 1. Sample AE output from PROC REPORT with some hyphenation and indentation highlighted.

EXAMPLE 2

The following sample code uses the same data step as Example 1 except the `indent_rest` macro variable is changed to a different value so that the indentation for the preferred term variable is different for the other rows compared to the first row. Output 2 shows how the PROC REPORT could appear for the `finalae` variable under this scenario.

```
data aetable;
length finalae $100;
set ae;
by aesys aeterm;
%indent(variable=aesys);
%indent(variable=aeterm, indent_first=3, indent_rest=2);
if first.var1 then do;
finalvar=new_aesys;
output;
end;
finalvar=new_aeterm;
output;
run;
```

```
Body System Term
   Preferred Term           Treatment 1   Treatment 2   Treatment 3   Total
-------------------------   -----------   -----------   -----------   ----------
Subjects having at least    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
one adverse event

Blood and lymphatic syst-   xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
em disorders
   Iron deficiency anaem-   xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
  ia
   Thrombocytopenia         xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)

Cardiac disorders           xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
  Atrial fibrillation       xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
  Atrioventricular block    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)    xx (xx.x%)
 second degree
```

Output 2. Sample AE output from PROC REPORT with the different indentation highlighted.

CONCLUSION

The macro described in this paper allows the user to designate and maintain the number of leading blanks for indentation while allowing the option of hyphenating words for character variables to be displayed in PROC REPORT. The indentations can also be customized to be different for the first row compared to the remaining rows within the same data point. The macro allows greater flexibility when using the FLOW option and split character designation in PROC REPORT and can enhance the appearance of tables or listings for clinical trial submissions.

CONTACT INFORMATION

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

Name:	Stanley Yuen
Enterprise:	SimulStat, Incorporated
Address:	4370 La Jolla Village Drive, Suite 400
City, State ZIP:	San Diego, CA, 92122
Work Phone:	(858) 546-4337
Fax:	(858) 546-4338
E-mail:	syuen@simulstat.com