

Yet Another More User-Friendly Text-Wrapping Algorithm

Dorothy E. Pugh, ClinTrials Research, Inc.
Susan M. Lach, ClinTrials Research, Inc.

ABSTRACT

In the clinical trials field, which often involves reporting comment fields in the form of large text fields, it is important to have a text-wrapping algorithm which requires 1) simple input, 2) minimal pre- and post-processing on the part of the programmer, and 3) can handle an observation with multiple large text fields. This algorithm requires three input parameters, no pre-processing, and its output is a SAS® data set containing the wrapped text and two new variables indicating 1) which lines were added to the data set as a result of the text-wrapping, and 2) which new observations were generated from the same old observation. The only post-processing the user needs to do is to make output of each variable conditional on the value of these new variables.

INTRODUCTION

In the clinical trials field, programmers frequently need to produce by-patient listings of raw clinical data. This often includes multiple large text fields (up to 200 characters), which must be wrapped to fit narrow columns. Therefore, there is a strong demand for text-wrapping algorithms that can manipulate text so that it can be printed in a tabular format without further elaborate processing, and which require a few simple input parameters that make no assumptions about the data other than the maximum length of the text fields.

THE ALGORITHM

This text-wrapping macro requires three input parameters: 1) a list of large text variables to be wrapped, 2) a list of their respective widths, 3) the data set containing the variables to be wrapped, and 4) an optional parameter containing the delimiters at which line breaks are to be made (the default being a blank). Input for the fourth parameter needs to be a list of characters inside quotes. Each list must be separated by blanks.

The macro first assigns the value OLDLINE = 1 to the input data set. Then it creates a temporary

data set for each text variable to be wrapped. Each of these temporary data sets contains one of the variables listed in the first parameter as well as all of the variables on the data set not contained in the first parameter list. The wrapping process, which is performed on each of these data sets, splits up each large text field by creating new observations that each contain a segment of the old string as well as copies of the unwrapped variable values. A new variable is added, NEWLINE, which is set to 1 on the first observation containing each text field, while the other lines have higher values.

When the wrapping process on these data sets is complete, then they are merged by OLDLINE and NEWLINE. OLDLINE has the same value for every observation that is part of the oldline, while NEWLINE numbers the lines within OLDLINE.

An example program calling this macro is illustrated in Figure 1. The output is illustrated in Figure 2.

THE CODE

```
%macro wrap(vars,widths,dataset,delim);

    %if &delim eq %then %let delim = ' ';

    *****
    * Get names and count variables to be *
    * wrapped.                            *
    *****

    %let i = 1;

    %do %while (%scan(&vars,&i,' ') ne);
        %let var&i = %scan(&vars,&i,' ');
        %if %scan(&widths,&i,' ') ne
            %then %let width&i =
                %scan(&widths,&i,' ');
            %else %let width&i =
                &&width%eval(&i-1);
        %let l %eval(&i+1);
    %end;

    %let numvars = %eval(&i-1);

    *****
    * For each variable, perform wrap.    *
    *****
```

```

%do i = 1 %to &numvars;

*****
* Construct list of variables to *
* drop (i.e., all variables except *
* the one being wrapped). *
*****;

%let varlist =;

%do j = 1 %to &numvars;
  %if &j ne &i %then
    %let varlist =
      &varlist &&var&j;
%end;

*****
* Do actual wrapping ONLY if *
* variable length is greater *
* column width. *
*****;

data &&var&i(drop=&varlist);

  set &dataset end=eof;
  retain maxlen 0;
  oldline = _n_;
  temp = &&var&i;

  if temp ne '' then do;

    if length(&&var&i) gt &&width&i
      then do;

        do until (length(temp) le 1);

          i = &&width&i;

          do until (substr(temp,i,1)
                    in (&delim) or i le 1);
            i = i - 1;
          end;

          &&var&i=substr(temp,1,i);
          output;
          temp=left(substr(temp,i+1);

          if length(temp)
            le &&width&i then do;
            &&var&i=trim(temp);
            output;
            temp = "";
          end;

        end;

      end;

    end;

  end;

end;

end;

else do;
  &&var&i = trim(temp);
  output;
end;

end;

run;

*****
* Add variables to be used to merge the data *
* sets with the individual wrapped variables *
* on them. Old lines get *
* sequentially. New lines created by *
* wrapping get values of NEWLINE > 1. *
*****;

data &&var&i;
  set &&var&i;
  by oldline;
  retain newline 0;
  if first.oldline then newline = 0;
run;

*****
* Merge all the datasets, each with a newly *
* wrapped variable, by the old line counter *
* (OLDLINE) and the new one *
* (NEWLINE). *
*****;

data &dataset;
  merge
    %do i = 1 %to &numvars;
      &&var&i(in=var&i)
    %end;
    ;
  by oldline newline;

  %do j = 1 %to &numvars;
    if ^var&j then &&var&j = "";
  %end;

run;

%mend wrap;

```

CONCLUSION

This macro illustrates a method of text-wrapping which not only splits the text to be wrapped into sub-strings but embeds it in an output data set in a way that enables it to be written easily to an output file with PUT statements.

AUTHOR CONTACTS

Dorothy E. Pugh
ClinTrials Research, Inc.
11000 Weston Blvd.
Cary, NC 27513
dpugh@clintrialsresearch.com

Susan M. Lach
ClinTrials Research, Inc.
11000 Weston Blvd.
Cary, NC 27513
slach@clintrialsresearch.com

Figure 1

```
options ls=70;

(Code here setting up libname of macro directory)

data a;
  length text1 text2 $100.;
  name = "Joe Blow ";
  age = "35 ";
  text1 = "A patient in our study who is remarkable for his blue ears";
  text2 = "He is otherwise an ordinary person who likes to garden.";
  output;

  name = "Jane Doe ";
  age = "46 ";
  text1 = "She has bright green antennae";
  text2 = "She likes badminton, bobsledding, and buying bestselling books.";
  output;
run;

proc print;
  var name age oldline newline text1 text2;
  title 'Output';
run;

%wrap(text1 text2,15 20,a)
```

Figure 2

Output						1
OBS	NAME	AGE	OLDLINE	NEWLINE	TEXT1	TEXT2
1	Joe Blow	35	1	1	A patient in	He is otherwise an
2	Joe Blow	35	1	2	our study who	ordinary person
3	Joe Blow	35	1	3	is remarkable	who likes to garden.
4	Joe Blow	35	1	4	for his blue	
5	Joe Blow	35	1	5	ears	
6	Jane Doe	46	2	1	She has	She likes
7	Jane Doe	46	2	2	bright green	badminton,
8	Jane Doe	46	2	3	antennae	bobsledding, and
9	Jane Doe	46	2	4		buying bestselling
10	Jane Doe	46	2	5		books.