A Macro for Managing Long Text Strings and Producing Pretty SAS reports

Cécilie Birner, Metropolitan Transportation Commission, Oakland CA

Abstract:
A 200 character text string can be unwieldy in a SAS report. SAS reporting functions truncate text strings and generally are not flexible enough to please detail oriented managers. This talk will illustrate a macro that untangles these long text strings. They will be transformed into bits which wrap at appropriate places. The paper will also cover some wrong turns and right turns taken in approaching this common report problem.

1) Introduction

Ever had to do a report but had one variable that was over too long? Most SAS procedures can not handle long variables. Proc report is very flexible, but can not handle all reports you may need to do. In addition, there are some limitations to its text-wrapping abilities. This paper shows one way to get around that problem by using a SAS macro and the data step.

2) Examples of the Problem

Let's say you have transportation data set with several variables including a project description. If the description is longer than one conventional line (80 Characters for portrait or 120 characters for landscape) your reports will look very bad and omit important information. Only the first line will print therefore producing both unprofessional and annoying reports. For example the project description:

```
Build a purple staircase from Earth to Barney's Magical Clubhouse.  Also include a
large purple elevator (strong enough for Barney when he is BIG!) for our super-dee-
dooper wheelchair-using friends.
```

truncates to the following with proc print:

```
Build a purple staircase from Earth to Barney's Magical Clubhouse.  Also include a lar
```

This is clearly not acceptable for a formal report.

3) Macro fix:

What I did was the following:

A) Figure out the maximum length each field could be
B) Estimated out how the page will be laid out
C) Plugged the numbers I needed into the macro
D) Ran the macro and incorporated the nicely generated formatted bits into the datastep
E) Had a supervisor very pleased with the readable reports.

The field I needed to wrap was a 200 character project description. I needed to have it go into up to three lines. Therefore, I needed to split it up into three segments of 70. Therefore, each observation needs to be cut into one, two, or three pieces. The pieces are called despart1, despart2, and despart3 for the first, second, and third bits receptively. The last variable you need is the total number of parts each observation was cut up to.

The first thing the macro does is check to see if the field is less 70 characters or fewer. If the macro is less than 70, it does not need to do anything.

```
lengd=length(descript);
** one line descriptions;
if lengd le 70 then do;
   despart1=descript;
   d_lines='1';
end;
```

If the macro is greater than 70 characters, then you need to figure out the cleanest place to cut the text string. The macro goes out to the maximum length the part can be can and then counts backwards until it finds an appropriate place to wrap.

```
** description over two lines;
      else if lengd gt 70 then do;
* check to see if you can wrap at maximum character'
      nextchar=substr(descript,71,1);
            cutpt=substr(descript,70,1);
            if cutpt in(' ',';','.','-',',','!')
            then do;
** output correctly cut first part;
            despart1=substr(descript,1,71);

** output rest of line for further chopping;
            temppart=substr(descript,70);
            end;

** otherwise, find good place to wrap;
       else do;
       do i=69 to 1 by -1 until (cutpt in (' ',';','.','-','!'));
            cutpt=substr(descript,i,1);
            nextchar=i+1;
        end;
          ** output nicely cut first part;
despart1=substr(descript,1,i);
```

The key of this program is the 'do until' loop in bold above.

You can now output the remainder to a temporary variable and start the process again.

```
temppart=substr(descript,nextchar);
```

remember to drop variables you won't need anymore.

```
** drop vars no longer needed;
drop i temppart nextchar lengd nextchar cutpt;
```

This process can be repeated as many times as you need to truncate you data.
Finally, you can the use the parts in a very pretty data step:

```
Data _null_;
      set chopped;
file print;                                        ** send data to listing;
put @10 despart1;
if d_lines in (2,3) then do;
   put @10 despart2;
 end;
if d_lines=3 then do;
 put@10 despart3;
end;
put ' ' / / ;

; run;
```

The macro allows us to print out a nicely wrapped project description in a report:

```
Build a purple staircase from Earth to Barney's Magical Clubhouse.
Also include a large purple elevator (strong enough for Barney when
he is BIG!) for our super-dee-dooper wheelchair-using friends.
```

Note: if you have any numbers if your text, avoid using the comma as a delimiter. It can result in splitting a dollar figure.

4) Other Options

If a SAS procedure works for you please use it! SAS offers many wonderful procedures and there is no reason to re-invent the virtual wheel.

Another option is to output reports in HTML tables. HTML is a flexible format and can automatically wrap text.

Lastly, you can always export SAS data to a database file program. Note that the SQL in database programs is not as strong as the SAS programming language for complicated data. You may not have a program log which will make error checking extremely difficult. Lastly, database programs are not a mathematically sophisticated as statistical software. You will need to research how it handles missing variables and the calculation of means and other mathematical processes well documented in the SAS system.

Acknowledgements

Thanks to Ned Lewis and Gao Liu for helping me try to get a SAS procedure to work for my report.
Special thanks to SAS institute for being so responsive to my requests for help.
Thanks also to Phil Spector for his excellent statistical computing instruction.

Contacting the Author

Cecilie Birner can be reached at: cbirner@mtc.ca.gov (work) or cecilie@igc.org (personal).
Her web page (done using this macro) is: www.mtc.ca.gov/projects/tip/tipind.htm