Paper 79-25

A CASE-TRANSFORMING MACRO FOR MORE READABLE TEXT

Sylvia Tze, Medtronic Inc., Minneapolis, MN

ABSTRACT

In data entry, text fields are usually entered in uppercases. When printing out these text fields, one would prefer having them in the form of normal writing, combining upper and lowercases. One example is comments fields. Having words in upper and lower cases will enhance readability, conformity, and consistency with the rest of the report. This paper introduces a macro that would transform text fields from uppercases to a format of upper and lower cases. The macro is written in such a way that the first letter of each sentence is preserved as uppercase. Exceptions are taken into consideration when acronyms and abbreviations in uppercases are intentional. Users can provide their own list of acronyms and abbreviations to customize usage.

INTRODUCTION

SAS provides functions uppercase() and lowcase() to convert text to upper and lower cases respectively. This macro takes advantage of these two functions. The first word of each sentence is preserved as uppercase while the rest is changed to lowercase. The plan is to extract individual words from a sentence, perform the necessary conversion, and then assemble the words back to form a single sentence again.

THE MACRO

The macro is being called with the following parameters:

```
%utolcase(indata=,name=,outdata=&indata);
```

where **indata**, **name**, and **outdata** are keyword parameters; **indata** is the input data set containing **name** which is the text variable. By default, **outdata** is set to the input data set. By doing so, the text variable is being written over by the conversion. An output data set name can be specified to prevent the overwriting.

First of all, we extract individual words from a sentence. To do that, the scan function is used:

```
word=scan(&name,i,’ ‘);
```

where **&name** is the name of the variable to be converted and is being passed as a parameter, **i** is the counter of a do-loop to continue extracting until the end of sentence is reached. The quotes **’ ‘** specifies that space is being used as the delimiter here.

The first word can be handled using the following code, with the converted text assigned to the variable tempc:

```
tempc = left(substr(word,1,1)||
             lowcase(substr(word,2,length(word))));
```

while the rest of the words can be done in the following way:

```
tempc = lowcase(word);
```

The first letter of each sentence will then be kept uppercased and the rest being converted to lowercase. Individual words will be strung back together, separated with blanks, while each word is being converted.

```
newname = left(trim(newname)|| ’ ‘||left(tempc);
```

To determine if a word is in the beginning of a sentence, the macro will look at the previous word and see if it is followed by a period, question mark or exclamation mark. If so, the first letter of the following word will be kept uppercased. A variable called **lword** is created to keep track of whether the previous word is the last word of a sentence as indicated by its ending with a period, question mark or exclamation mark. The word is reversed and the first character examined.

```
if substr(left(reverse(word),1,1) in (‘.’,’!’,’?’)
  then lword=1;      /* last word of a sentence */
  else lword=0;      /* not the last word */
```

Next, acronyms and abbreviations are looked at. These are the words that we want to stay uppercased, without any conversion. A variable called **inlist** is created to indicate this. The extracted word is compressed to get rid of punctuations that could confuse SAS from identifying them. The macro then checks to see if it is among the list of words provided in the group. If not, the word is handled as described above. Otherwise, the word is concatenated to the existing string untouched.

Here is the finished macro:

```
%macro utolcase(indata=,name=,outdata=&indata);

 data &outdata(rename=(newname=&name));
   length word $15. newname $200.;
   retain i 1 lword 1;
   set &indata;
   word=scan(&name,i,’ ‘);
   newname=‘ ‘;

   /* start scanning for words */
   /* finding out if they are in the capital list */

   do while (word ne ‘ ‘);
     if compress(word,’.,!?:’) in (&caplist,’I’)
       then inlist=1;
       else inlist=0;

     /* not in list, start converting */
```

```
      if not inlist then do;

        /* it’s the first word of a sentence */
        /* keep the first letter uppercased */

        if lword=1 then do;
            tempc=left(substr(word,1,1)||
            lowcase(substr(word,2,length(word))));
            newname=trim(left(newname))|| ’ ‘||
                                left(tempc);
            lword=0;
        end;

        /* not the first word of a sentence */
        /* change everything to lowercase */

        else do;
            tempc=lowcase(word);
            newname=trim(left(newname))|| ’ ‘||
                                left(tempc);
        end;
      end;

      /* in list, leave it as is */

      else do;
         newname=trim(left(newname))|| ’ ‘||
                              left(word);
      end;

      i=i+1;

      /* if the word ends with ‘.’, ‘!’, or ‘?’       */
      /* then it is the last word of a sentence */

      if (substr(left(reverse(word)),1,1)
        in(‘.’,’!’,’?’))
        then lword=1;
        else lword=0;
      word=scan(&name,i,’ ‘);
    end;

    i=1;
    lword=1;
    newname=left(newname);
    drop &name i tempc word lword inlist;

run;

%mend;
```

EXAMPLE

This sample data set **charfn** contains two variables **fn** and **detail**. **Fn** contains SAS function names and **detail** contains their uses.
The macro call to invoke the macro to convert variable **detail** to mixed cases, overwriting the original field would be :

%utolcase(indata=charfn,name=detail);

The macro call to invoke the macro to convert variable **detail** to mixed cases, preserving the original data set would be :

```
%utolcase(indata=charfn,name=detail,
          outdata=ocharfn);
```

A new data set is created which contains the variable **detail** converted to mixed cases.

Here is the data set before the call:

fn	**detail**
SCAN	SELECTS THE NTH TERM FROM THE SOURCE. THE SOURCE CAN BE ANY CHARACTER EXPRESSION.
LEFT	LEFT-ALIGNS BY MOVING ANY LEADING BLANKS TO THE END OF THE VALUE.
TRIM	TRIMS TRAILING BLANKS. THE TRIM FUNCTION DOES NOT AFFECT THE WAY A VARIABLE IS STORED.
LENGTH	ASSIGNS A LENGTH YOU SPECIFY TO THE CHARACTER VARIABLE.
INDEX	SEARCHES FOR A PATTERN OF CHARACTERS.
INDEXC	SIMILAR TO INDEX! IT FINDS THE 1ST OCCURRENCE OF ANY ONE OF A SET OF CHARACTERS.
REVERSE	REVERSES CHARACTERS, HOWEVER, BEHAVES DIFFERENTLY WITH SCL VARIABLES.

We will define our list of words to stay capped with the macro list, **&caplist**.

%let caplist = ‘TRIM’,’FUNCTION’,’SCL’,’INDEX’;

The macro call will follow the above **%let** statement.

The resulting data set looks like this:

fn	**detail**
SCAN	**S**elects the nth term from the source. **T**he source can be any character expression.
LEFT	**L**eft-aligns by moving any leading blanks to the end of the value.
TRIM	**T**rims trailing blanks. **T**he **TRIM FUNCTION** does not affect the way a variable is stored.
LENGTH	**A**ssigns a length you specify to the character variable.
INDEX	**S**earches for a pattern of characters.
INDEXC	Similar to **INDEX**! **I**t finds the 1st occurrence of any one of a set of characters.
REVERSE	**R**everses characters, however, behaves differently with **SCL** variables.

The bold letters indicate the desired case.

CONCLUSION

The **%utolcase** macro is a handy tool to convert uppercased text to the more readable, and presentable mixed cased. The macro recognizes the first letter of a sentence and keeps it uppercased. It also allows a list to be specified for special words to stay uppercased, as intended.

AUTHOR CONTACT

Sylvia Tze, SAS Programmer
Medtronic, Inc.
4000 Lexington Avenue North
Mail Stop X230
Shoreview, MN 55126-2983
Work Phone: (612) 514-9847
E-Mail : sylvia.tze@medtronic.com