Transposition of Variables with Different Formats

Peter Lin, AstraZeneca LP, Wilmington, DE

ABSTRACT
Sometimes with clinical trial data, several variables need to be transposed into a single variable with multiple observations. At the same time, the transposed variable needs to keep the different formatted values of the original variables. It is impossible to use PROC TRANSPOSE in one step to accomplish the transposition. This paper introduces a macro that transposes all variables (numeric or character) into one character variable while retaining the different formatted values of the original variables. If a variable does not have a format, its character value is kept in the transposed variable. Users can also define the name and the length of the transposed variable.

KEY WORDS
FORMAT, VARIABLE ATTRIBUTES, MACRO ARRAY, proc transpose

INTRODUCTION
When PROC TRANSPOSE is used to transpose several variables into a single variable with multiple observations, the attributes of the transposed variable will follow the rules below.

- If all variables that the procedure is transposing are numeric, the resulting transposed variable will be numeric. The transposed variable will only contain the unformatted values of the original variables.
- If all variables that the procedure is transposing are character, the resulting transposed variable will be character. The transposed variable will only contain the unformatted values of the original variables.
- If the procedure is transposing both character and numeric variables, the resulting transposed variable will be character. Furthermore, only the formatted values of the numeric variables are transposed; the unformatted values of the character variables stay in the transposed variable.

We wrote %m_tran macro to complement PROC TRANSPOSE. The macro transposes the selected variables into a character variable containing the formatted values, not the unformatted values as by default with PROC TRANSPOSE.

MACRO OVERVIEW
Essentially, there are four steps in the %m_tran macro. First, the %m_tran macro creates a data set, which contains the name of the variable(s) that the macro is going to transpose, variable format name and label. Subsequently, variable name, variable format and variable label are saved into a macro array. In the next step, the macro creates the corresponding variables for each variable that the macro is going to transpose and puts the formatted value into the newly created variables. Finally, PROC TRANSPOSE is used to transpose the created variables. The output data set contains variables in the BY group, variable containing former variable label, variable containing former variable name, and the transposed variable(s).

The macro is called with the following parameters:

%m_trans(dsin=, dsout=, byvar=, tranvar=, clength=, cname=)

dsin is the input dat set containing variables to be transposed. dsout is the output data set. By default, dsout is set to the input data set name. byvar is the name of the variable(s) to be the BY group. tranvar is the name of the variable(s) to be transposed. clength is the length of the transposed variable. cname is the user-defined name for the transposed variable.

THE CODE
%macro m_trans(dsin=, dsout=&dsin, byvar=, tranvar=, clength=, cname=);
%*********************************************************;
%* MACRO: m_trans *;
%* AUTHOR: Peter Lin *;
%* PARAMETER: *;
%* DSIN : Input data set name *
%* DSOUT : Output data set name *
%*********************************************************;
%* %****END MTRAN MACRO**
%mend m_trans;

%*  Default is the input data set name.  *
%*  BYVAR : Variable name(s) that *
%*            the macro uses to form *
%*            BY groups            *
%*  TRANVAR: Variable name(s)      *
%*            to transpose        *
%*  CLENGTH: Column length of      *
%*            transposed variable, *
%*  CNAME  : Column name of        *
%*            transposed variable  *
%*                                                        *
%*  USAGE:  %m_trans(dsin=qans      *
%*          ,byvar=centre subject   *
%*          ,tranvar=Q1-Q24 QA23-QA27 AGE)  *
%*  DESCRIPTION:                      *
%*                                                      *
%*  NOTES:                                  *
%***********************************************************;
data _tranvar;
  set &dsin;
  keep &tranvar;
run;

proc contents data=_tranvar noprint out=_temp (keep=name format format1 formatd type length label);
run;

data _temp;
  length fmtname $40;
  set _temp;
  if type=1 then do;
    if format='' then do;
      if formatl ^=0 then
        fmtname=left(put(formatl, best.)||'.'||left(put(formatd, 2.)));
      end;
    else if formatl^=0 then
      fmtname=trim(left(format))||
      trim(left(formatl))||'.';
    else fmtname=trim(left(format))||'.';
    end;
  else do;
    if format ^='' then do;
      if compress(format, '$')='' then
        fmtname=trim(left(format))||
        trim(left(put(formatl,best.)))||'.';
      else fmtname=trim(left(format))||'.';
    end;
  end;
run;

proc sql noprint;
  select name, fmtname, label
   into :vname1-:vname500,
         :vfmt1-:vfmt500,
from _temp
order by name;
quit;
%let nvar=&sqlobs;
data _temp(drop=%do i=1 %to &nvar;
   _&&vname&i %end; );
   %do i=1 %to &nvar;
      %if &i=1 %then %let cstr= ;
      %let cstr=&cstr &&vname&i;
   %end;
length &cstr $&clength;
set &dsin(rename=( %do i=1 %to &nvar;
   &&vname&i=_&&vname&i
   %end;));
   %do i=1 %to &nvar;
      label &&vname&i="&&vlbl&i";
      %if &&vfmt&i ^=%str() %then %do;
         &&vname&i=put(_&&vname&i,
         &&vfmt&i);
      %end;
      %else %do;
         &&vname&i=_&&vname&i;
      %end;
   %end;
run;
proc transpose data=_temp out=&dsout
   (rename=(COL1=&cname));
   by &byvar;
   var &tranvar;
run;
proc datasets library=work;
delete _temp _tranvar;
modify &dsout;
   label &cname='Transposed Variable';
quid; %mend m_trans;

**AN EXAMPLE**
The data set before transposition contains the following variables:

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Type</th>
<th>Format</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENTRE</td>
<td>Char</td>
<td></td>
<td>Study site number</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>Char</td>
<td></td>
<td>Subject number</td>
</tr>
<tr>
<td>AGE</td>
<td>Num</td>
<td></td>
<td>Subject age</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>Num</td>
<td>8.1</td>
<td>Subject weight (Lb)</td>
</tr>
<tr>
<td>ASS_DAT</td>
<td>Num</td>
<td>YYMMD10.</td>
<td>Assessment date</td>
</tr>
<tr>
<td>QA1</td>
<td>Char</td>
<td>$YESNO.</td>
<td>Any speech loss</td>
</tr>
<tr>
<td>QA2</td>
<td>Char</td>
<td></td>
<td>Number of speech loss episodes</td>
</tr>
<tr>
<td>QA3</td>
<td>Char</td>
<td>$YESNO.</td>
<td>Any vision loss</td>
</tr>
<tr>
<td>QA4</td>
<td>Char</td>
<td>$QAD.</td>
<td>Most recent vision loss episode</td>
</tr>
</tbody>
</table>
Example data set (DIRQ):

<table>
<thead>
<tr>
<th>CENTRE</th>
<th>SUBJECT</th>
<th>AGE</th>
<th>WEIGHT</th>
<th>ASS_DAT</th>
<th>QA1</th>
<th>QA2</th>
<th>QA3</th>
<th>QA4</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>0023</td>
<td>45</td>
<td>178.35</td>
<td>2002/06/19</td>
<td>Y</td>
<td>2</td>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>39</td>
<td>201.82</td>
<td>2002/05/28</td>
<td>Y</td>
<td>1</td>
<td>Y</td>
<td>1</td>
</tr>
</tbody>
</table>

The macro call:

%m_tran(ds=DIRQ, byvar=centre subject, tranvar=age weight ass_dat qa1-qa4, clength=30, cname=Answer);

The output data set by calling the macro:

<table>
<thead>
<tr>
<th>CENTRE</th>
<th>SUBJECT</th>
<th>_ NAME_</th>
<th>_ LABEL_</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>0023</td>
<td>AGE</td>
<td>Subject Age</td>
<td>45</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>WEIGHT</td>
<td>Subject weight (Lb)</td>
<td>178.4</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>ASS_DATE</td>
<td>Assessment date</td>
<td>2002/06/19</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>QA1</td>
<td>Any speech loss</td>
<td>Yes</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>QA2</td>
<td>Number of speech loss episodes</td>
<td>2</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>QA3</td>
<td>Any vision loss</td>
<td>Yes</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>QA4</td>
<td>Most recent vision loss episode</td>
<td>&gt;31 days ago</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>AGE</td>
<td>Subject Age</td>
<td>39</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>WEIGHT</td>
<td>Subject weight (Lb)</td>
<td>201.8</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>ASS_DATE</td>
<td>Assessment date</td>
<td>2002/05/28</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>QA1</td>
<td>Any speech loss</td>
<td>Yes</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>QA2</td>
<td>Number of speech loss episodes</td>
<td>1</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>QA3</td>
<td>Any vision loss</td>
<td>Yes</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>QA4</td>
<td>Most recent vision loss episode</td>
<td>8-31 days ago</td>
</tr>
</tbody>
</table>

The output data set by using the single PROC TRANSPOSE step:

<table>
<thead>
<tr>
<th>CENTRE</th>
<th>SUBJECT</th>
<th>_ NAME_</th>
<th>_ LABEL_</th>
<th>COL1</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>0023</td>
<td>age</td>
<td>Subject Age</td>
<td>45</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>weight</td>
<td>Subject weight (Lb)</td>
<td>178.4</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>ass_dat</td>
<td>Assessment date</td>
<td>2002/06/19</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>qa1</td>
<td>Any speech loss</td>
<td>Y</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>qa2</td>
<td>Number of speech loss episodes</td>
<td>2</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>qa3</td>
<td>Any vision loss</td>
<td>Y</td>
</tr>
<tr>
<td>101</td>
<td>0023</td>
<td>qa4</td>
<td>Most recent vision loss episode</td>
<td>2</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>age</td>
<td>Subject Age</td>
<td>39</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>weight</td>
<td>Subject weight (Lb)</td>
<td>201.8</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>ass_dat</td>
<td>Assessment date</td>
<td>2002/05/28</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>qa1</td>
<td>Any speech loss</td>
<td>Y</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>qa2</td>
<td>Number of speech loss episodes</td>
<td>1</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>qa3</td>
<td>Any vision loss</td>
<td>Y</td>
</tr>
<tr>
<td>113</td>
<td>0017</td>
<td>qa4</td>
<td>Most recent vision loss episode</td>
<td>1</td>
</tr>
</tbody>
</table>

CONCLUSION
The %m_tran macro is a tool to complement PROC TRANSPOSE. The macro transposes the variables defined by users into a character variable by BY group variables. If the variables that the macro is transposing have a formatted value, the transposed variable would contain the same formatted value. The transposed variable would contain the unformatted value if the variables do not have formats.

REFERENCES
ACKNOWLEDGEMENTS
The author would like to express his appreciation to Gary Cunningham for his helpful suggestions on this paper.

CONTACT INFORMATION
Your comments and questions are valued and encouraged.
Contact the author at:
Peter Lin
Quantitative Decision Sciences
AstraZeneca
Delaware 2B-115B
1800 Concord Pike
Wilmington DE 19850-5437
E-mail: peter.lin@astrazeneca.com