A Technique to Create a Numeric Variable from a Character Variable
Hany Aboutaleb, Biogen Idec, Cambridge, MA

Abstract:

If you have a character variable in your dataset, and you wish to transfer this variable to numeric variable instead, this article can help you. I will describes and illustrate by example how to use Proc Sort, and Format with the help of SAS® Macro facility to create a new numeric variable in your data set from your character variable.

Overview:

Suppose you have five variables on your data, one of which is a character type, and the rest of the variable is numeric type. You wish to have the variables in numeric type to convert those variables to vertical or horizontal format using arrays or Proc Transpose. There are many ways to transfer a character variable to numeric variable, however all of those ways could have some problems (could lose some values from your character variable... etc.). There is a safe way to make the transformation without causing truncation to character variables. The technique involves sorting the data by this character variable, creating a counter for each unique value. Create a series of macro variables for each unique value of the character variable beginning with prefix _PRM_1 ... _PRM_n, where n is the total number of unique value for that character variable. Create a new numeric variable in your data and store the counter value in that numeric variable. You have the choice of keeping any missing value in this new numeric variable, or you can delete those missing values. Finally create a format value using the numeric variable and the macro variables _PRM_1..._PRM_n where we store the unique values of the character variable on those macro variables. As you will see we can create a new numeric variable with the help of Proc Format to store the character in format value.

Method:

• Find the dataset with a character variable you like to transfer to numeric variable.
• Identify if the variable has a format value.
• Choose a new name for the numeric variable and format value.
• Choose if you want to keep missing value.
• Choose if you want to create a new dataset.

Key steps:

• Reference the location of your data.
• The macro will check all of the values you supply to the macro variable during macro execution.
• The macro will sort the data by the character variable values.
• The macro will create macro variables for each unique value _PRM_1 ... _PRM_n, and one for the counter.
• The macro will create format using Proc Format to assign the value of the character variable using the macro variable _PRM_1 ... _PRM_n.
The Program:

Program Name: Chr2num.sas
Project Name : Biogen Idec

Programmer : Hany Aboutaleb
Creation Date: 11/11/03
Language : PC SAS

Purpose: This macro will transfer any character value to numeric

Remarks: Any comment or suggestions please email to: hany.aboutaleb@biogenidec.com

Here you can change the name of the sub-directory to the new locations.

Libname FMT 'C:\sasfmts';
Libname in1 'c:\prjt\datain';
Libname out1 'c:\prjt\dataout';

Set the options for SAS

Options Pageno=1 ls=150 ps=57 Mprint Mautosource Nofmterr;

The Macro starts with the macro call.
Input parameters:

DATA = input data name
    Default: _last_ last data set used.
    Example: test.

OUT= output data name
    Default: Final.
    Example: final.

OLDVAR = old variable name or the used variable name.
    Default: none
    Example: SUPERVIS

OLDFMT = old format name or the used format name.
    Default: none.
    Example: $10.

NEWVAR = New name for the new variable.
    Default: none
    Example: numeric.

NEWFMT = New name for the new format for the new variable.
    Default: none.
    Example: wild.

RENAME = if you wish to rename the new variable to the old one.
    Default: NO.
    Example: NO.

MISSING= if you like to keep missing values in the new variable values.
    Default: ‘.’
    Example: ‘.’

KEEPMISS= if you like to keep missing values from the old variable values.
    Default: NO.
Example: NO.

```sas
%MALLOC chr2num DATA=_LAST_,OUT=final,OLDVAR=,OLDFMT=,NEWVAR=,
NEWFMT=,RENAME=NO,MISSING=,KEEPMISS=);
```

The next statement will determine whether you have assigned valid value to all of the macro variables and assign a default value to those you left blank

```sas
%LOCAL I DATAO PERIOD;
%LET BLANK=%STR( );
%IF %LENGTH(&MISSING)>0 %THEN %DO;
  %IF &MISSING=" " %THEN
    %LET MISSING=%STR( );
%END;
%LET PERIOD=%STR(.);
%IF %INDEX(&NEWFMT,.)>0 %THEN
  %LET NEWFMT= %SUBSTR( &NEWFMT,1,%LENGTH(&NEWFMT)-1);
%IF %INDEX(&OLDFMT,.)=0 %THEN %LET OLDFMT=&OLDFMT.
%IF %LENGTH(&OUT)>0 %THEN %LET DATAO=&OUT;
%ELSE %LET DATAO=&DATA;
```

Next sort the data with the character variable values.

```sas
PROC SORT DATA=&DATA;
  BY &OLDVAR;
```

This data step will count the unique value and store it in the macro variable __CNT___. It will create macro variables to store each unique value of the character variable in the macro variables __PRM__1 ... __PRM__n where n is the total number of the unique values of this character variable. It also checks if the user chooses to keep the missing value and assign this missing value to the new numeric variable before storing the counter value in the new numeric variable. If the user did not supply the macro call with a name for the new numeric variable, the macro will drop the character variable and assign the same name to the new numeric variable.
DATA &DATAO;SET &DATA END=DONE;
BY &OLDVAR;
LENGTH __PRM___ $ 8;
RETAIN __CNT___ 0;
IF FIRST.&OLDVAR THEN DO;
  __CNT___ = __CNT___ +1;
  __PRM___ = '__PRM'||TRIM(LEFT(__CNT___));
  CALL SYMPUT (__PRM___,PUT(&OLDVAR,&OLDFMT));
END;

%IF %LENGTH(&KEEPMISS)=0 %THEN %DO;
  IF &OLDVAR=' ' AND PUT(&OLDVAR,&OLDFMT)=''
    THEN &NEWVAR = . ;
  ELSE &NEWVAR=__CNT___;
%END;
%ELSE %DO;
  &NEWVAR=__CNT___;
%END;

%IF %UPCASE(&RENAME)=YES %THEN %DO;
  RENAME &OLDVAR=&NEWVAR
  &NEWVAR=&OLDVAR;
%END;
DROP __CNT___ __PRM___ ;
IF DONE THEN CALL SYMPUT ('__CNT___',TRIM(LEFT(__CNT___)));   RUN;

Here is the logic to rename the data step to the new name chosen by the user.

%IF %UPCASE(&RENAME)=YES %THEN %DO;
  DATA &DATAO;SET &DATA;
  DROP &NEWVAR;
%END;

Now the time comes to create a formatted value for the new numeric variable.

PROC FORMAT; VALUE &NEWFMT
  %DO I=1 %TO &_CNT___;
    &I="&&__PRM&I"
  %END;
%IF %LENGTH(&MISSING)>0 %THEN %DO;
    .="&MISSING"
%END;
%STR();
RUN;

Macro end.

%MEND chr2num;
Sample Data:

<table>
<thead>
<tr>
<th>IDNUM</th>
<th>SUPERVIS</th>
<th>WAGERATE</th>
<th>WAGECAT</th>
<th>JOBCODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1122</td>
<td>JOHNSON</td>
<td>14.20</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>1131</td>
<td>JOHNSON</td>
<td>13.30</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>1132</td>
<td>YOUNG</td>
<td>13.75</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>1124</td>
<td>FLETCHER</td>
<td>19.21</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Sample Macro call:

```sas
%Chr2num(DATA=_LAST_,OUT=FINAL, OLDVAR=SUPERVIS,OLDFMT=$10., NEWVAR=NUMERIC,NEWFMT=WILD.,RENAME=NO,MISSING=' ',KEEPMISS=NO);
*** Without format *******;
proc print;
title 'The data after the macro W F';
run;
*** With format *******;
proc print;
title 'The data after the macro with F';
format numeric wild.;
run;
```

Sample output without using the format:

<table>
<thead>
<tr>
<th>IDNUM</th>
<th>SUPERVIS</th>
<th>WAGERATE</th>
<th>WAGECAT</th>
<th>JOBCODE</th>
<th>NUMERIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1124</td>
<td>FLETCHER</td>
<td>19.21</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>1122</td>
<td>JOHNSON</td>
<td>14.20</td>
<td>2</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>1131</td>
<td>JOHNSON</td>
<td>13.30</td>
<td>1</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>1132</td>
<td>YOUNG</td>
<td>13.75</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

Sample output using the format:

<table>
<thead>
<tr>
<th>IDNUM</th>
<th>SUPERVIS</th>
<th>WAGERATE</th>
<th>WAGECAT</th>
<th>JOBCODE</th>
<th>NUMERIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1124</td>
<td>FLETCHER</td>
<td>19.21</td>
<td>3</td>
<td>6</td>
<td>FLETCHER</td>
</tr>
<tr>
<td>1122</td>
<td>JOHNSON</td>
<td>14.20</td>
<td>2</td>
<td>8</td>
<td>JOHNSON</td>
</tr>
<tr>
<td>1131</td>
<td>JOHNSON</td>
<td>13.30</td>
<td>1</td>
<td>7</td>
<td>JOHNSON</td>
</tr>
<tr>
<td>1132</td>
<td>YOUNG</td>
<td>13.75</td>
<td>2</td>
<td>7</td>
<td>YOUNG</td>
</tr>
</tbody>
</table>

CONCLUSION:

A great deal of flexibility can be gained by using this macro to change SAS variable from character to numeric.
TRADEMARKS:

SAS® is a registered trademark of the SAS Institute, Inc., in the USA and in other countries. Windows® is a registered trademark of the Microsoft Corporation.

Acknowledgment:

The author would like to thank Ms. Tina Casteris for her review and valuable comments.

REFERENCES:


CONTACT:

Hany Aboutaleb
BiogenIdec.
14 Cambridge Center
Cambridge MA 02142
Tele. #: (617) 914-7125
Fax: (617) 679-3280
Internet: hany.aboutaleb@biogenidec.com