A SAS® Macro that Creates Numeric Decimal Formats

John Q. Zhang AT&T at Basking Ridge, NJ USA

ABSTRACT

SAS Macro function, %EVAL() function, performs integer arithmetic only. Calculations on fractions or decimal numeric are definitely not allowed. This paper introduces a macro that has managed to avoid the restriction and creates numeric formats that have adjustable ranges of decimal numbers. The numeric formats are user-defined. The range specifications (including numeric ceiling and floor) can be handled or adjusted by manipulating the concerning macro parameters. The paper shows definitely how powerful and flexible SAS Macro language really is. The SAS products concerned in this paper are mostly SAS/Base and/or Macro. With respect to operating systems, there is not supposed to have any limit to the feasibility of the macro as long as SAS Macro facility is available. The intended audience should hopefully have some experience in programming in SAS/Base (including Macro Language). The skill level of the users is preferably intermediate. The rationale presented on the paper can be extended to some operations other than SAS formats in case calculations on fractions in SAS Macro processing are confronted with.

THE IDEA AND OPERATION

SAS has documented a sample program of SAS Macro that can set up a series of individual numbers for PROC PLOT operation. Those numbers look like 1.0, 1.1,…,1.8, 1.9, etc. However, that program has to be further elaborated before range specifications can possibly be handled. Some of the sample programs of SAS Macro can take care of numeric range specifications for PROC FORMAT but there are no decimals or fractions involved. In reality, it is not unusual for SAS users to manually set up considerably large ranges for numeric fractions and decimals, especially an input control data set is not available. The work will be time-consuming and boring. Therefore it is time to turn to SAS Macro, since SAS Macro is designed to effectively handle repetitive operation. To evade the pitfall that %EVAL() doesn't take fractions or decimals, nested Do loops of SAS Macro have to be utilized to create 2 groups of integer, then combine those 2 groups to form decimal numbers by concatenation.
However, life is not that easy. If the macro is set up simply by concatenating 2 groups of integer the output will not be what is expected as the following part of SAS log shows:

```
%MACRO TNT(FSTOP,AMT,SECTOP);
%*GLOBAL I J X Y;
%DO I=-3 %TO 3;
 %DO J=0 %TO &SECTOP %BY &AMT;
 %LET X=&I..&J;
 %LET K=%EVAL(&J+&AMT);
 %LET Y=&I..&K;
   &X-<&Y="&X-<&Y"
 %END;
%END;
%MEND;

PROC FORMAT;
 VALUE N_25FMT
 %TNT(75,25,50);
ERROR: Start is greater than end: -3-<-
3.25.
MPRINT(TNT):   -3.0-<-3.25="-3.0-<-3.25"
-3.25-<-3.50="-3.25-<-3.50"-3.50-<-
3.75="-3.50-<-3.75" -2.0-<-2.25="-2.0-<-
2.25"-2.25-<-2.50="-2.25-<-2.50" -2.50-
<-2.75="-2.50-<-2.75"-1.0-<-1.25="-1.0-
<-1.25" -1.25-<-1.50="-1.25-<-1.50"-
1.50-<-1.75="-1.50-<-1.75" 0.0-
<0.25="0.0-<0.25" 0.25-<0.50="0.25-
<0.50"0.50-<0.75="0.50-<0.75" 1.0-
<1.25="1.0-<1.25" 1.25-<1.50="1.25-
<1.50"1.50-<1.75="1.50-<1.75" 2.0-
<2.25="2.0-<2.25" 2.25-<2.50="2.25-
<2.50"2.50-<2.75="2.50-<2.75" 3.0-
<3.25="3.0-<3.25" 3.25-<3.50="3.25-
<3.50"......
NOTE: The previous statement has been
deleted.
RUN;
NOTE: The SAS System stopped processing
this step because of errors.
```

Two problems are brought up on the above program, the explicit one and the implicit one. The explicit one is the error message, "Start is greater than end: -3-<-3.25". The problem keeps coming up until the end of negative part of the ranges (the bold part of the text above). The second problem is that there is a gap between the underlined ranges. The gap also makes the negative part of the format deranged. So the format is incomplete even though the existing gap does not hurt the SAS statements. Two sets of DO LOOP have to be established to handle both negative ranges and positive ones. To eliminate the gap, a macro statement has to be inserted between the 2 macro DO LOOPs (see the bold text in the macro).

```
%MACRO TNT(FSTOP,AMT,SECTOP);
 %*GLOBAL I J X Y;
 %DO I=-3 %TO 0;
```

```
  %DO J=&FSTOP %TO &AMT %BY %EVAL(&AMT*-
1);
  %LET X=&I..&J;
  %LET K=%EVAL(&J-&AMT);
  %LET Y=&I..&K;
    &X-<&Y="&X-<&Y"
  %END;
 &I..0-<%EVAL(&I+1).&FSTOP="&I..0-
<%EVAL(&I+1).&FSTOP"
  %END;
%DO I=0 %TO 3;
 %DO J=0 %TO &SECTOP %BY &AMT;
  %LET X=&I..&J;
  %LET K=%EVAL(&J+&AMT);
  %LET Y=&I..&K;
    &X-<&Y="&X-<&Y"
  %END;
 &I..&FSTOP-<%EVAL(&I+1).00="&I..&FSTOP-
<%EVAL(&I+1).00"
%END;
%MEND;
PROC FORMAT;
 VALUE N_25FMT
%TNT(75,25,50);
ERROR: Start is greater than end: 0.75-
<0.5.
MPRINT(TNT):   -3.75-<-3.50="-3.75-<-
3.50" -3.50-<-3.25="-3.50-<-3.25"-3.25-
<-3.0="-3.25-<-3.0" -3.0-<-2.75="-3.0-<-
2.75" -2.75-<-2.50="-2.75-<-2.50"-2.50-
<-2.25="-2.50-<-2.25" -2.25-<-2.0="-
2.25-<-2.0" -2.0-<-1.75="-2.0-<-1.75"-
1.75-<-1.50="-1.75-<-1.50" -1.50-<-
1.25="-1.50-<-1.25"-1.25-<-1.0="-1.25-<-
1.0" -1.0-<0.75="-1.0-<0.75" 0.75-
<0.50="0.75-<0.50"0.50-<0.25="0.50-
<0.25" 0.25-<0.0="0.25-<0.0" 0.0-
<1.75="0.0-<1.75"0.0-<0.25="0.0-<0.25"
0.25-<0.50="0.25-<0.50" 0.50-
<0.75="0.50-<0.75"0.75-<1.00="0.75-
<1.00" 1.0-<1.25="1.0-<1.25" 1.25-
<1.50="1.25-<1.50"......
NOTE: The previous statement has been
deleted.
RUN;
NOTE: The SAS System stopped processing
this step because of errors.
```

The SAS log shows that the gap is gone. The inconsistency of the ranges, however, still persistently exists. The problem appears at the boundary of negative ranges and positive ranges. A special arrangement seems needed to solve the problem. Another set of DO LOOP has been put in between the groups above mentioned, to handle from '-1' to '0' part of the format.

```
OPTIONS MPRINT NOSYMBOLGEN NOMLOGIC
LS=80;

    %MACRO TNT(FSTOP,AMT,SECTOP);
    %*GLOBAL I J X Y;

      %DO I=-3 %TO -2;
        %DO J=&FSTOP %TO &AMT %BY
%EVAL(&AMT*-1);
         %LET X=&I..&J;
         %LET K=%EVAL(&J-&AMT);
         %LET Y=&I..&K;
           &X-<&Y="&X-<&Y"
         %END;
 &I..0-<%EVAL(&I+1).&FSTOP="&I..0-
<%EVAL(&I+1).&FSTOP"
      %END;

       %DO I=-1 %TO 0;
         %DO J=&FSTOP %TO &AMT %BY
%EVAL(&AMT*-1);
          %LET X=&I..&J;
          %LET K=%EVAL(&J-&AMT);
          %LET Y=&I..&K;
          %IF &I=0 %THEN %DO;
           %IF &J^=0 AND &K^=0 %THEN -
&X-<-&Y="-&X-<-&Y";
           %ELSE %IF &J^=0 AND &K=0
%THEN -&X-<&Y="-&X-<&Y";
          %END;
          %ELSE &X-<&Y="&X-<&Y";
         %END;
         %IF &I^=0 %THEN
          &I..0-<-
%EVAL(&I+1).&FSTOP="&I..0-<-
%EVAL(&I+1).&FSTOP";
       %END;

       %DO I=0 %TO 3;
         %DO J=0 %TO &SECTOP %BY &AMT;
          %LET X=&I..&J;
          %LET K=%EVAL(&J+&AMT);
          %LET Y=&I..&K;
            &X-<&Y="&X-<&Y"
         %END;
         &I..&FSTOP-
<%EVAL(&I+1).00="&I..&FSTOP-
<%EVAL(&I+1).00"
        %END;
     %MEND;

     PROC FORMAT;
      VALUE N_25FMT
      %TNT(75,25,50);/*
      VALUE N125FMT
      %TNT(875,125,750);
      VALUE N_10FMT
      %TNT(90,10,80);*/
MPRINT(TNT):   -3.75-<-3.50="-3.75-<-
3.50" -3.50-<-3.25="-3.50-<-3.25"
-3.25-<-3.0="-3.25-<-3.0" -3.0-<-2.75="-
3.0-<-2.75"
-2.75-<-2.50="-2.75-<-2.50"
-2.50-<-2.25="-2.50-<-2.25" -2.25-<-
2.0="-2.25-<-2.0"
-2.0-<-1.75="-2.0-<-1.75"
-1.75-<-1.50="-1.75-<-1.50" -1.50-<-
1.25="-1.50-<-1.25"
-1.25-<-1.0="-1.25-<-1.0" -1.0-<-0.75="-
1.0-<-0.75"
```

```
-0.75-<-0.50="-0.75-<-0.50"
-0.50-<-0.25="-0.50-<-0.25" -0.25-
<0.0="-0.25-<0.0" 0.0-<0.25="0.0-<0.25"
0.25-<0.50="0.25-<0.50" 0.50-
<0.75="0.50-<0.75" 0.75-<1.00="0.75-
<1.00"
1.0-<1.25="1.0-<1.25" 1.25-<1.50="1.25-
<1.50" 1.50-<1.75="1.50-<1.75"
1.75-<2.00="1.75-<2.00" 2.0-<2.25="2.0-
<2.25" 2.25-<2.50="2.25-<2.50"
2.50-<2.75="2.50-<2.75" 2.75-
<3.00="2.75-<3.00" 3.0-<3.25="3.0-<3.25"
3.25-<3.50="3.25-<3.50" 3.50-
<3.75="3.50-<3.75" 3.75-<4.00="3.75-
<4.00"
NOTE: Format N_25FMT has been output.
RUN;
QUIT;
```

CONCLUSION

Now the format, N_25FMT, is ready to work. The macro TNT makes the format quite flexible. First of all, the ceiling and floor of the ranges (here is -3 and 3) are adjustable. As long as resources allow, the range can be -100,000 to 100,000. The macro itself does not present any confinement to limit set-up. Secondly, the intervals of ranges are also adjustable. The example above uses interval value of 0.25. The frequently used interval should be 0.25, 0.125, 0.1, 0.2, etc. The macro will have to be further modified for the intervals like 0.0625, which is a half of 0.125, or 0.5 (the modifications are available on Appendix I & II).

A question may probably be raised here. Should we use the regular SAS statements to create a numeric format or use SAS Macro to do the job? Regular SAS statements are definitely more flexible than macro statements in view that there is no restriction on the calculations of fractions or decimals as %EVAL() function is used. The format created by macro TNT can be stored in a library as a permanent numeric format, so can the same format created by the regular SAS statements. However, a program generator will be necessary on which 2 DATA Steps are usually needed to create a format and write it out to a flat file (a SAS program) with a PROC FORMAT step on it, if regular SAS DATA Steps and PROC Steps are chosen. Then retrieve the flat file and run the SAS program to output the format. Take the following example, a numeric format with an interval of 0.3, which macro TNT is unable to handle:

```
OPTIONS LS=80;

DATA LOOP;
DO I=0 TO 10 BY .30;
J=I+0.3;
FMT=COMPRESS(PUT(I,5.2)||'-
<'||PUT(J,5.2)
    ||'='||'"'||PUT(I,5.2)||'-
<'||PUT(J,5.2)||'"');
OUTPUT;
END;
RUN;

DATA _NULL_;
SET LOOP END=END;
IF _N_=1 THEN PUT @1 'PROC FORMAT;' /@2
'VALUE NUMFMT';
PUT @4 FMT;
IF END THEN PUT @1 'RUN;' /@1 'QUIT;';
RUN;
QUIT;
```

The purpose of the paper is to explore the extensive utility of SAS Macro language. In terms of the specific format creation, utilizing SAS Macro or the regular SAS DATA Step and PROC step depends on personal preference and, more importantly, performance consideration.

This paper is a tentative experiment on the feasibility of implementation of SAS Macro for format creation. Any comments, suggestions and recommendations will certainly be appreciated.

ACKNOWLEDGMENTS

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

CONTACT INFORMATION

John Q. Zhang
Tel: (908) 221-4616 (O)
Email: jzhang79@.yahoo.com

Appendix I : The Macro that Produces a Format with Interval of 0.5

```
OPTIONS MPRINT NOSYMBOLGEN NOMLOGIC LS=80;

%MACRO TST2(START,END);
%GLOBAL I H J K X Y;
  %DO I=&START %TO -2;
    %DO J=0 %TO 5 %BY 5;
     %IF &J=0 %THEN %DO;
       %LET K=5;
       %LET H=%EVAL(&I+1);
     %END;
     %ELSE %IF &J=5 %THEN %DO;
       %LET K=0;
       %LET I=%EVAL(&I+1);
     %END;
      %LET X=&I..&J;
      %LET Y=&H..&K;
      &X-<&Y="&X-<&Y"
    %END;
    %LET I=%EVAL(&I-1);
  %END;

  %DO I=-1 %TO 0;
    %DO J=0 %TO 5 %BY 5;
     %IF &J=0 %THEN %DO;
       %LET K=5;
       %LET H=%EVAL(&I+1);
     %END;
     %ELSE %IF &J=5 %THEN %DO;
       %LET K=0;
       %LET I=%EVAL(&I+1);
     %END;
      %LET X=&I..&J;
      %LET Y=&H..&K;
       %IF &H=0 & &K=5 %THEN &X-<-&Y="&X-<-&Y";
       %IF &I=0 & &J=5 %THEN -&X-<&Y="-&X-<&Y";
    %END;
  %END;

  %DO I=0 %TO &END;
    %DO J=0 %TO 5 %BY 5;
     %IF &J=0 & &I=0 %THEN %DO;
      %LET H=0;
      %LET K=5;
     %END;
     %ELSE %IF &J=5 %THEN %DO;
      %LET K=0;
      %LET H=%EVAL(&I+1);
     %END;
     %ELSE %IF &J=0 %THEN %LET K=5;
      %LET X=&I..&J;
      %LET Y=&H..&K;
      &X-<&Y="&X-<&Y"
    %END;
  %END;
%MEND;

PROC FORMAT;
```

```
   VALUE N_05BFMT
   %TST2(-3,3);
MPRINT(TST2):   -3.0-<-2.5="-3.0-<-2.5" -2.5-<-2.0="-2.5-<-2.0"
-2.0-<-1.5="-2.0-<-1.5" -1.5-<-1.0="-1.5-<-1.0" -1.0-<-0.5="-1.0-<-0.5"
-0.5-<0.0="-0.5-<0.0" 0.0-<0.5="0.0-<0.5" 0.5-<1.0="0.5-<1.0"
1.0-<1.5="1.0-<1.5" 1.5-<2.0="1.5-<2.0" 2.0-<2.5="2.0-<2.5" 2.5-<3.0="2.5-
<3.0"
3.0-<3.5="3.0-<3.5" 3.5-<4.0="3.5-<4.0"
NOTE: Format N_05FMT has been output.
```

Appendix II : The Macro that Produces a Format with Interval of 0.625

```
OPTIONS MPRINT NOSYMBOLGEN NOMLOGIC;

%MACRO TST3(STOP,FIN,AMT);
%GLOBAL J K X Y;
  %DO I=-3 %TO -2;
    %DO J=%EVAL(&FIN+&AMT) %TO &AMT %BY %EVAL(&AMT*-1);
    %IF &J=-&AMT OR &J=&AMT %THEN %LET J=0&AMT;
     %LET X=&I..&J;
     %LET K=%EVAL(&J-&AMT);
    %IF &K=-&AMT OR &K=&AMT %THEN %LET K=0&AMT;
     %LET Y=&I..&K;
      &X-<&Y="&X-<&Y"
    %END;
   &I..0-<%EVAL(&I+1).%EVAL(&FIN+&AMT)="&I..0-
   <%EVAL(&I+1).%EVAL(&FIN+&AMT)"
  %END;

  %DO I=-1 %TO 0;
    %DO J=%EVAL(&FIN+&AMT) %TO &AMT %BY %EVAL(&AMT*-1);
     %IF &J=-&AMT OR &J=&AMT %THEN %LET J=0&AMT;
     %LET X=&I..&J;
     %LET K=%EVAL(&J-&AMT);
     %IF &K=-&AMT OR &K=&AMT %THEN %LET K=0&AMT;
     %LET Y=&I..&K;
    %END;
    %IF &I^=0 %THEN &X-<&Y="&X-<&Y" ;
    %ELSE %IF &I=0 %THEN %DO;
      %IF &K=0 %THEN -&X-<&Y="-&X-<&Y";
      %ELSE -&X-<-&Y="-&X-<-&Y";
    %END;
  %END;

  %DO I=0 %TO 1;
    %DO J=0 %TO &FIN %BY &AMT;
    %IF &J=&AMT %THEN %LET J=0&AMT;
     %LET X=&I..&J;
     %LET K=%EVAL(&J+&AMT);
    %IF &K=&AMT %THEN %LET K=0&AMT;
     %LET Y=&I..&K;
      &X-<&Y="&X-<&Y"
    %END;
    %LET TOP=%EVAL(&FIN+&AMT);
    &I..&TOP-<%EVAL(&I+1).000="&I..&TOP-<%EVAL(&I+1).000"
```

```
     %END;
       -1.0-<-0.9375="-1.0-<-0.9375"
   %MEND;
   PROC FORMAT;
     VALUE N625FMT
     %TST3(2,8750,625);
MPRINT(TST3):   -3.9375-<-3.8750="-3.9375-<-3.8750" -3.8750-<-3.8125="-
3.8750-<-3.8125"
-3.8125-<-3.7500="-3.8125-<-3.7500" -3.7500-<-3.6875="-3.7500-<-3.6875"
-3.6875-<-3.6250="-3.6875-<-3.6250" -3.6250-<-3.5625="-3.6250-<-3.5625"
-3.5625-<-3.5000="-3.5625-<-3.5000" -3.5000-<-3.4375="-3.5000-<-3.4375"
-3.4375-<-3.3750="-3.4375-<-3.3750" -3.3750-<-3.3125="-3.3750-<-3.3125"
-3.3125-<-3.2500="-3.3125-<-3.2500" -3.2500-<-3.1875="-3.2500-<-3.1875"
-3.1875-<-3.1250="-3.1875-<-3.1250" -3.1250-<-3.0625="-3.1250-<-3.0625"
-3.0625-<-3.0="-3.0625-<-3.0" -3.0-<-2.9375="-3.0-<-2.9375" -2.9375-<-
2.8750="-2.9375-<-2.8750"
-2.8750-<-2.8125="-2.8750-<-2.8125" -2.8125-<-2.7500="-2.8125-<-2.7500"
-2.7500-<-2.6875="-2.7500-<-2.6875" -2.6875-<-2.6250="-2.6875-<-2.6250"
-2.6250-<-2.5625="-2.6250-<-2.5625" -2.5625-<-2.5000="-2.5625-<-2.5000"
-2.5000-<-2.4375="-2.5000-<-2.4375" -2.4375-<-2.3750="-2.4375-<-2.3750"
-2.3750-<-2.3125="-2.3750-<-2.3125" -2.3125-<-2.2500="-2.3125-<-2.2500"
-2.2500-<-2.1875="-2.2500-<-2.1875" -2.1875-<-2.1250="-2.1875-<-2.1250"
-2.1250-<-2.0625="-2.1250-<-2.0625" -2.0625-<-2.0="-2.0625-<-2.0" -2.0-<-
1.9375="-2.0-<-1.9375"
-1.0625-<-1.0="-1.0625-<-1.0" -0.0625-<0.0="-0.0625-<0.0" 0.0-<0.0625="0.0-
<0.0625"
0.0625-<0.1250="0.0625-<0.1250" 0.1250-<0.1875="0.1250-<0.1875" 0.1875-
<0.2500="0.1875-<0.2500"
0.2500-<0.3125="0.2500-<0.3125" 0.3125-<0.3750="0.3125-<0.3750" 0.3750-
<0.4375="0.3750-<0.4375"
0.4375-<0.5000="0.4375-<0.5000" 0.5000-<0.5625="0.5000-<0.5625" 0.5625-
<0.6250="0.5625-<0.6250"
0.6250-<0.6875="0.6250-<0.6875" 0.6875-<0.7500="0.6875-<0.7500" 0.7500-
<0.8125="0.7500-<0.8125"
0.8125-<0.8750="0.8125-<0.8750" 0.8750-<0.9375="0.8750-<0.9375" 0.9375-
<1.000="0.9375-<1.000"
1.0-<1.0625="1.0-<1.0625" 1.0625-<1.1250="1.0625-<1.1250" 1.1250-
<1.1875="1.1250-<1.1875"
1.1875-<1.2500="1.1875-<1.2500" 1.2500-<1.3125="1.2500-<1.3125" 1.3125-
<1.3750="1.3125-<1.3750"
1.3750-<1.4375="1.3750-<1.4375" 1.4375-<1.5000="1.4375-<1.5000" 1.5000-
<1.5625="1.5000-<1.5625"
1.5625-<1.6250="1.5625-<1.6250" 1.6250-<1.6875="1.6250-<1.6875" 1.6875-
<1.7500="1.6875-<1.7500"
1.7500-<1.8125="1.7500-<1.8125" 1.8125-<1.8750="1.8125-<1.8750" 1.8750-
<1.9375="1.8750-<1.9375"
1.9375-<2.000="1.9375-<2.000" -1.0-<-0.9375="-1.0-<-0.9375"
NOTE: Format N625FMT has been output.
```