ABSTRACT

This study is intended to assist Analysts to generate the best of variables using simple arithmetic operators (square root, log, loglog, exp and rcp) and such as monthly amount paid, daily number of received customer service calls, or annual number total sales for a specific product. During a statistical data modeling process, Analysts are often confronted with the task of computing derived variables using the existing available variables. The advantage of this methodology is that the new variables may be more significant than the original ones. This paper gives a new way to compute all the possible variables using a set of math transformation. The codes include many SAS® features that are very useful tools for SAS programmers to incorporate in their future codes. Such as %SYSFUNC, SQL, %INCLUDE, CALL SYMPUT, %MACRO, SORT, CONTENTS, MERGE, MACRO _NULL_, as well as %DO …%TO … and many more. I demonstrate the syntax and mechanics of the macro using following examples.

INTRODUCTION

These codes are used for logistical regression model.

There are four steps to transformation one variable into 5 derived variables and pick the best one based on the information value and store it into final data.

Table 1, sale data –age_of_house. sold indicator, location, bed_room, style, listdaystosale and location.

<table>
<thead>
<tr>
<th>sold indicator</th>
<th>size_sqft</th>
<th>location</th>
<th>bed_rooms</th>
<th>bath_room</th>
<th>style</th>
<th>listdaysTosale</th>
<th>location</th>
<th>age_of_house</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>2393</td>
<td>chicago</td>
<td>3</td>
<td>2</td>
<td>condo</td>
<td>130</td>
<td>suburbur</td>
<td>50</td>
</tr>
<tr>
<td>no</td>
<td>3628</td>
<td>Houston</td>
<td>5</td>
<td>3</td>
<td>single family</td>
<td>200</td>
<td>city</td>
<td>20</td>
</tr>
<tr>
<td>yes</td>
<td>2200</td>
<td>Washinton</td>
<td>3</td>
<td>2</td>
<td>condo</td>
<td>100</td>
<td>city</td>
<td>60</td>
</tr>
<tr>
<td>no</td>
<td>2000</td>
<td>chicago</td>
<td>3</td>
<td>2</td>
<td>town house</td>
<td>120</td>
<td>city</td>
<td>60</td>
</tr>
<tr>
<td>yes</td>
<td>1890</td>
<td>Houston</td>
<td>4</td>
<td>3</td>
<td>single family</td>
<td>100</td>
<td>suburbur</td>
<td>87</td>
</tr>
<tr>
<td>yes</td>
<td>1600</td>
<td>Washinton</td>
<td>2</td>
<td>1</td>
<td>town house</td>
<td>80</td>
<td>city</td>
<td>70</td>
</tr>
<tr>
<td>yes</td>
<td>2400</td>
<td>chicago</td>
<td>3</td>
<td>1</td>
<td>single family</td>
<td>110</td>
<td>suburbur</td>
<td>50</td>
</tr>
<tr>
<td>yes</td>
<td>2500</td>
<td>Houston</td>
<td>4</td>
<td>2</td>
<td>single family</td>
<td>120</td>
<td>suburbur</td>
<td>35</td>
</tr>
<tr>
<td>yes</td>
<td>3600</td>
<td>Washinton</td>
<td>3</td>
<td>2</td>
<td>town house</td>
<td>130</td>
<td>city</td>
<td>50</td>
</tr>
<tr>
<td>yes</td>
<td>3500</td>
<td>chicago</td>
<td>6</td>
<td>2</td>
<td>single</td>
<td>250</td>
<td>suburbur</td>
<td>25</td>
</tr>
</tbody>
</table>
Step 1. Five mathematical function of macro to transform one variable into 5 new variables.

**Macro 1, Log transformation.** See Figure 1-a

```sas
%MACRO LOG_FUNC(VN, LO_LIM=, HI_LIM=, D=&DELTA);
/* FUNCTION VNL=LOG(VN-LO_LIM+D) */
%LET VNN1 = &VN_6&SX1;
IF &VN < 0 THEN &VNN1 = 0;
ELSE &VNN1 = LOG(&VN+&D);
%IF &HI_LIM NE %THEN %DO;
  IF &VN > &HI_LIM THEN &VNN1 = LOG(&HI_LIM+&D);
%END;
IF LENG <= 34 THEN VARLAB1 = SUBSTR(VARLAB, 1, LENG)||' (LOG)';
ELSE VARLAB1 = SUBSTR(VARLAB, 1, 34)||' (LOG)';
%MEND
```

**Macro 2, Log log transformation.**

```sas
%MACRO LLG_FUNC(VN, LO_LIM, HI_LIM=, D=&DELTA);
/* FUNCTION VNL=LOG(LOG(VN+D+1)) */
%LET VNN1 = &VN_6&SX2;
IF &VN < 0 THEN &VNN1 = LOG(LOG(3));
ELSE &VNN1 = LOG(LOG(&VN+&D+2));
%IF &HI_LIM NE %THEN %DO;
  IF &VN > &HI_LIM THEN &VNN1 = LOG(LOG(&HI_LIM+&D+2));
%END;
IF LENG <= 33 THEN VARLAB2 = SUBSTR(VARLAB, 1, LENG)||' (LLOG)';
```

---

**Figure 1-a.** See log transformation.
ELSE VARLAB2 = SUBSTR(VARLAB,1,33)||" (LLOG)";

%MEND LLG_FUNC;

Macro 3, Exponential transformation.
%Macro EXP_FUNC(VN,LO_LIM=,HI_LIM=,A=0,B=);
/* FUNCTION VNE=EXP(A*VN) */;
%let VNN1 = &VN_6&Sx3;
%let RANGE = %sys Evalf(&B-&A);
%if &RANGE = 0 %then %let RANGE = 1;
if &VN < 0 then &VNN1 = -1;
else &VNN1 = -exp((-1) * &VN / &RANGE);
%if &HI_LIM NE %then %do;
if &VN > &HI_LIM then &VNN1 = -exp((-1) * &HI_LIM / &RANGE);
%end;
&VNN1 = round(&VNN1,0.00001);
if LENG <= 34 then VARLAB3 = substr(VARLAB,1,LENG)||' (EXP)';
else VARLAB3 = substring(VARLAB,1,34)||' (EXP)';
%MEND EXP_FUNC;

Macro 4, Square Root transformation.
%Macro SQR_FUNC(VN,LO_LIM=,HI_LIM=,A=0);
%let VNN1 = &VN_6&Sx4;if &VN < 0 then &VNN1 = 0;
else &VNN1 = sqrt(&VN+A);
%if &HI_LIM NE %then %do;
if &VN > &HI_LIM then &VNN1 = sqrt(&HI_LIM+A);
%end;
if LENG <= 33 then VARLAB4=SUBSTR(VARLAB,1,LENG)||' (SQRT)';
else VARLAB4 = SUBSTR(VARLAB,1,33)||' (SQRT)';
%MEND SQR_FUNC;

Macro 5, RCP transformation see figure 1-b.

\[
\begin{align*}
\text{rcp_age_of_house} \\
\begin{array}{c}
\begin{array}{c}
0 \\
-0.01 \\
-0.02 \\
-0.03 \\
-0.04 \\
-0.05 \\
-0.06 \\
100000 \\
200000 \\
300000 \\
400000 \\
500000 \\
\end{array}
\end{array}
\end{align*}
\]

Figure 1-b, RCP function.

%Macro RCP_FUNC(VN,LO_LIM=,HI_LIM=,D=&DELT)

Figure 1-b, RCP function.

%Macro RCP_FUNC(VN,LO_LIM=,HI_LIM=,D=&DELT)
%VNAME_N(&VN,VN_6)
%LET VNN1 = &VN_6&SX5;
IF &VN < 0 THEN &VNN1 = -1;
ELSE &VNN1 = -1 / (&VN+&D);
%IF &HI_LIM NE &THEN %DO;
IF &VN > &HI_LIM THEN &VNN1 = -1 / (&HI_LIM+&D);
%END;
IF LENG <= 34 THEN VARLAB5 = SUBSTR(VARLAB,1,LENG)||' (RCP)';
ELSE VARLAB5 = SUBSTR(VARLAB,1,34)||' (RCP)';
%MEND RCP_FUNC;

Step 2, After variable transforming, call mapp macro to get each variable into same label and short type of variable.

%MACRO MAPP(DATA_IN=,DATA_OUT=,VARIABLES=,PCNTILE=);
%GLOBAL VNN1;
DATA &DATA_OUT;
LENGTH VARLAB VARLAB1-VARLAB5 $40;
SET &DATA_IN;
%DO II=1 %TO &N_VAR;
%IF (&&VAR&II NE &BAD AND &&VAR&II NE &WEIGHT_) %THEN %DO; CALL LABEL(&&VAR&II,VARLAB);
LENG=LENGTH(VARLAB);%LOGFUNC(&&VAR&II,LO_LIM=&&P&II._&LEFT_PCT,H
I_LIM=&&P&II._&RITE_PCT);
%LET VAR&II._1=&VNN1;
%LLGFUNC(&&VAR&II,LO_LIM=&&P&II._&LEFT_PCT,HI_LIM=&&P&II._&RITE_PCT);
%LETVAR&II._2=&VNN1;
%EXPFUNC(&&VAR&II,LO_LIM=&&P&II._&LEFT_PCT,HI_LIM=&&P&II._&RITE_PCT);
%LETVAR&II._3=&VNN1;
%SQRFUNC(&&VAR&II,LO_LIM=&&P&II._&LEFT_PCT,HI_LIM=&&P&II._&RITE_PCT);
%LETVAR&II._4=&VNN1;
%RCPFUNC(&&VAR&II,LO_LIM=&&P&II._&LEFT_PCT,HI_LIM=&&P&II._&RITE_PCT);
%LETVAR&II._5=&VNN1;
%DO JJ=1 %TO 6;
CALL SYMPUT("LAB&&II._&JJ",VARLAB&JJ);
%END;
%END;
%END;
%IF &DROP_ALL=1 %THEN %DO;
DROP &&VARIABLES LENG VARLAB VARLAB1-VARLAB5;
%END;
%ELSE %DO;
DROP LENG VARLAB VARLAB1-VARLAB5;
%END;
RUN;
%MEND MAPP;

This mapp macro can transfer one variable into 5 new derived variables with 5 new label see table 1A.
Example 2, Table 1A,

<table>
<thead>
<tr>
<th>Original variable</th>
<th>log</th>
<th>loglog</th>
<th>sqrt</th>
<th>exp</th>
<th>rcp</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a₁</td>
<td>a₂LL</td>
<td>a₃</td>
<td>a₄</td>
<td>a₅</td>
</tr>
<tr>
<td>b</td>
<td>b₁</td>
<td>b₂LL</td>
<td>b₃</td>
<td>b₄</td>
<td>b₅</td>
</tr>
</tbody>
</table>

Step 3, call logistical variable and pick the most significantly variables. See example3.

```sas
%MACRO SLCTVAR(SDATA,INDSN,DROPFILE,NUMKP=1,SLEVEL=0.05);
DATA SUBSET;
SET &INDSN;
  %IF &PORTION NE %THEN %DO;
    IF RANUNI(374521) <= &PORTION THEN OUTPUT;
  %END;
RUN;
%DO II=1 %TO &N_VAR;
  %IF &&P&II._&RITE_PCT NE . %THEN %DO;
    PROC LOGISTIC DATA=SUBSET(KEEP=&BAD &&VAR&II &&VAR&II._1 &&VAR&II._2 &&VAR&II._3 &&VAR&II._4 &&VAR&II._5 &WEIGHT_)
    OUTEST=EST1 NOPRINT;
    MODEL &BAD =&&VAR&II &&VAR&II._1 &&VAR&II._2 &&VAR&II._3 &&VAR&II._4 &&VAR&II._5 / SELECTION=FORWARD STOP=&NUMKP
    SLE=&SLEVEL;
    %IF &WEIGHT_ NE %THEN %DO;
    WEIGHT &WEIGHT_;
    %END;
  %END;
RUN;
DATA ONEVAR(KEEP=DROPVAR); LENGTH DROPVAR $8;
SET EST1(WHERE=(_TYPE_='PARMS'));
ARRAY VNNS &&VAR&II._1 &&VAR&II._2 &&VAR&II._3 &&VAR&II._4 &&VAR&II._5;
DO K=1 TO DIM(VNNS);
  IF VNNS{K} <= .Z THEN DO;
    CALL VNAME(VNNS{K},DROPVAR); OUTPUT;
  END;
END;
RUN;
%END;
%ELSE %DO;
DATA ONEVAR(KEEP=DROPVAR);
LENGTH DROPVAR $8;
ARRAY VNNS &&VAR&II &&VAR&II._1 &&VAR&II._2 &&VAR&II._3 &&VAR&II._4 &&VAR&II._5;
DO K=1 TO DIM(VNNS);
  VNNS{K} = 0;
  CALL VNAME(VNNS{K}, DROPVAR);
  OUTPUT;
END; STOP;
RUN;
```
%END;
PROC APPEND BASE=ROPLIST DATA=ONEVAR FORCE;
RUN;
%END;

DATA _NULL_;  
SET DROPLIST END=LAST;  
FILE "DROPFILE.dat";
IF _N_=1 THEN PUT '%MACRO DROPLIST;';
PUT DROPVAR;  
IF LAST THEN PUT '%MEND DROPLIST;';
RUN;

%INCLUDE "dropfile.dat";

DATA &SDA &T;&A;  
SET &INDSN;
LABEL %DO II=1 %TO &N_VAR; &&VAR&II._1="&&LAB&II._1" 
&&VAR&II._2="&&LAB&II._2" &&VAR&II._3="&&LAB&II._3" &&VAR&II._4="&&LAB&II._4" 
&&VAR&II._5="&&LAB&II._5"  %END;
%STR(;);
DROP %DROPLIST;
RUN;

%MEND SLCTVAR;

Example 2, droplist of this macro.

droplist
 a_l
 b_s
 c_e
 d_r

Step 4, call lgt_plot and calculate each variable for the information value and log odds.

%macro lgt_plot(dsn,bad,varname);
data temp;
set &dsn (keep=&bad &varname) end=last;retain n_bads 0;
format odds 6.2 ;
length odds_a $5;
level=&varname;
if &bad then n_bads=n_bads+1;
if last then do;
call syput('n_bads',trim(left(n_bads)));  
call syput('n_obs',trim(left(_n_)));  
odds=(n_bads)/(_n_-n_bads+0.1);
odds_a=trim(left(odds));
call syput('o_odds', odds_a);
end;
drop n_bads odds;
run;

proc means data=temp noprint;
   var &varname &bad; class level;
   output out=temp2 min=min min2 sum=&varname &bad max=max max2;
run;

sql noprint;
   select count(*) into :macro1 from temp;
quit;

data temp3;
set temp2(firstobs=2) end=last;
   format odds 3. logodds log_b_g 6.2 info 7.5;
   retain tot_info 0;
   odds=(&bad+0.1)/(s_wt-&bad+0.1);
   logodds=log(odds);
   WOE=logodds;
   log_b_g=logodds;

   p_good=(s_wt-&bad+0.1)/(&n_obs&n_bads+0.1); p_bad=(&bad+0.1)/(&n_bads+0.1);
   if info = . then info = 0;
   tot_info = tot_info + info;
   if last then do;
      if tot_info <= 0.00001 then tot_info = 0.00001;
      tot_info = round(tot_info,0.00001);
      call symput('tot_info',tot_info);
   end;

data temp3;
set temp3;
   format info_rt 4.1;
   info_rt = info/(&tot_info);
run;

proc means data=temp3 noprint;
   var logodds &varname;
   output out=null css=logodds &varname;
run;

proc corr data=temp3 noprint outp=corr1;
   var logodds &varname;
run;

Data out.corr1;
   set corr1;
   length var $32 corr_a $5;
   format corr &varname 4.2;
   if _type_='CORR' and logodds = 1; 
   var="&varname";
   corr=&varname;
   info=&tot_info;
   corr_a=trim(left(corr));
call symput("&varname",trim(left(corr_a)));  
keep var corr info;  
run;  
%mend lgt_plot(dsn,bad,varname);  

Example3, final variable list and related measurement  

<table>
<thead>
<tr>
<th>variable</th>
<th>information value</th>
<th>correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>size_sqft_r</td>
<td>0.2</td>
<td>0.45</td>
</tr>
<tr>
<td>location</td>
<td>0.1</td>
<td>0.35</td>
</tr>
<tr>
<td>bed_rooms_l</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>bath_room</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>style</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>listdaysTosale</td>
<td>0.7</td>
<td>0.5</td>
</tr>
<tr>
<td>age_of_house</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>price</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

This final list of significant variable list can use to plug in the final logistical model. This example shows the ranking order of the derived variables with dependent variable.

CONCLUSION  
This method of computing the derived variables can be useful for modeling purposes and save time during the variable creation stage. These macros are very straightforward but the user needs a strong understanding of SAS programming especially in SAS MACRO in order to fully take advantage of the codes.

Although not shown in the examples above, the macro will work for nonlinear variable (i.e. linear from the alternative model). In addition to normal bug fixing, future work on the macro may include improving the overall coding efficiency and enhancing the handling of missing values. Fitting complex models and/or models to large data sets is the chief challenge because of the iterative nature of the macro. My hope is that in the not-too-distance future, the parametric will be built-in component.

REFERENCES  


ACKNOWLEDGMENTS

We appreciate the feedback, suggestions and encouragement from my co-worker, Ming Zhang, Alicia, Anani and Joel. We also thank my Manager, Mamta Kakkar and Yang and Kotha, for their support and patience.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at nanyhu@discover.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

The views and opinions expressed here are those of the authors and do not necessarily reflect the views and opinions of Discover Bank. Discover Bank is not, by means of this article, providing technical, business, or other professional advice or services and is not endorsing any of the software, techniques, approaches, or solutions presented herein. This article is not a substitute for professional advice or services and should not be used as a basis for decisions that could impact your business.