

Evaluating Continuous Variable Transformations in Logistic Regression

Maggie (Yu) Zhang, Shaolin Chen, Suzanne C. Rain, KeyBank, Cleveland, OH

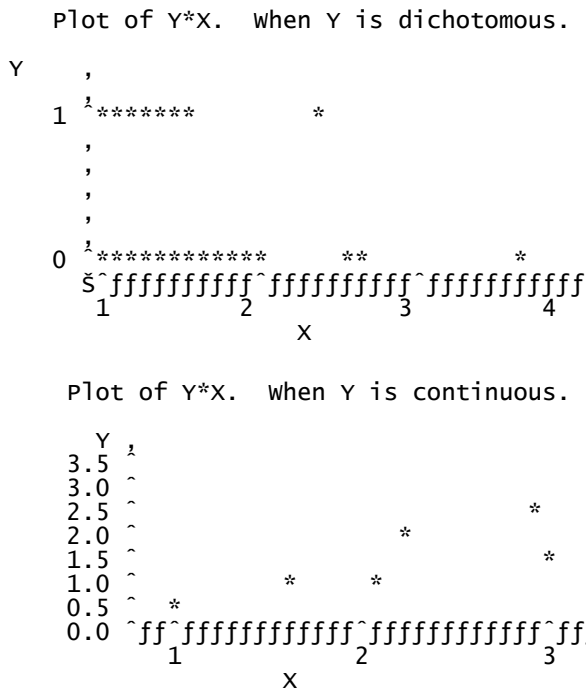
ABSTRACT

In modeling using logistic regression, the appropriate transformations on continuous variables are necessary to optimize the model predictiveness. The purpose of this paper is to demonstrate a method to automatically generate univariate plots for each input variable so that users can visually detect appropriate transformations. The application of this program is demonstrated using real examples from Keybank.

INTRODUCTION

In the marketing area of banking, logistic regression is widely used in modeling when the response variable is a bivariate action such as a response to an offer, a default on a loan, or closure of an account.

Variable transformation is an important technique to create robust models using logistic regression. Because the predictors are linear in the log of the odds, it is often helpful to transform the continuous variables to create a more linear relationship. To determine the best transformation of a continuous variable, a univariate plot is very helpful. Remember the nice univariate plot of Y variable against X variable in linear regression? This is not easily attained, because Y is dichotomous in logistic regression.



Without such a visual detection tool, variable transformation is a challenge in logistic regression. There are different recommended solutions. Among them, one is to create several variations (in forms of squared, cubed, or logged transformations etc.) and use a forward logistic regression to select the best fit. Another solution is to break all continuous variables into segments and treat them as categorical variables. This may work well to pick up nonlinear trends. The biggest drawback is that it loses the benefit of the linear trend relationship in the curve¹. It also may lead to over fitting.

The solution put forth in this paper is to create univariate plots in logistic regression, by observing grouped dependent variable Y against grouped independent variables X.

METHOD

To graph the dichotomous Y values against X values, X variables are broken into groups. Then the mean of Y within each group is calculated and is plotted, so that the grouped rate of change in Y is observed along with the change of grouped X.

As more groups are created, more detailed changes in rate of Y will be captured. However, the number of X bins must also take into consideration the total count within each group to allow for a reliable representation of that bin.

There are two ways to systematically group X variables. One is to divide the range of X evenly into N bins, so that each bin has 1/N of whole range, the other is to create bins by count, so that each bin has 1/N of the population.

The disadvantage of the first one is that for a very skewed dataset, most of the observations will be captured by one bin leaving the remaining bins to only capture a few outlying observations. The second way, on the other hand, by forcing bins to contain 1/nth of the population, the range of each bin can vary greatly.

When trying to understand Y's relationship with X, it is recommended to consider both methods.

When developing a model with many potential independent variables to consider, this graphing process can be painful if done individually. The SAS® MACRO program presented here is a simple tool to facilitate univariate analysis with automatic graphing process for variable selection and transformation.

THE SAS® APPLICATION

```

/*****
Univariate Plot Program 1
Fixed Range
KeyBank
*****/
/*****Define Parameters *****/
%let Input_DataSet=emdata.check2_build;
%let Output_DataSet=emdata.paper_Graph1;
%let X_Variables=
MTD_CHRG_BACK_Q
MTD_CR_A
;
%let Y_Variable=close;
%let lowestpctl=1;
%let highestpctl=99;
%let Bins=10;
/*****Global Statement and Options*****/
option mprint symbolgen;
legend1 label=none
position=(top center inside)
offset=(-2 cm,)
mode=share;
legend2 label=none
position=(top center inside)
offset=(2 cm,)
mode=share;
/*****Macro to Create Deciles, Plot Graphs, Output Dataset *****/
%macro CreateDeciles(X, Y, P_low, P_high, Bin);
data emdata.current(Keep=&X &Y);
set &Input_DataSet;
****1. Create Deciles****
proc Univariate data = emdata.current noprint;
var &X;
output out=emdata.temp(keep=P0 P&P_low P&P_high P100)
PctlPre= P
pctlpts= 0 &P_low &P_high 100;
data _null_;
set emdata.temp;
call Symput('Min',P0);
call Symput('Max',P100);
call Symput('P_low_value',P&P_low.);
call Symput('P_high_value',P&P_high.);
call Symput('StandardBinRange', round((( P&P_high - P&P_low)/&Bin), 0.001));

Data emdata.Outliers_On_Left
emdata.Outliers_On_Right
emdata.current1
;
set emdata.current;
if &X < &P_Low_Value
then output emdata.Outliers_On_Left;
else if &X > &P_High_Value
then output emdata.Outliers_On_Right;
else output emdata.current1;

```

```

data emdata.current1;
  set emdata.current1;
  Bin_No= Int((&X-&Min)/&StandardBinRange)+1;
  if Bin_No=11 then Bin_No=10;

proc sql;
  create table emdata.current2 as
  select Min(&X) as RangeMin,
         Max(&X) as RangeMax,
         Count(*) as Count,
         Avg(&Y)*100 as ResponseRate,
         Sum(&Y) as ResponseCount,
  Bin_No
  from emdata.current1
  group by Bin_No;

data emdata.current2;
  set emdata.current2;
  format RangeValue $50.;
  format Varname $50.;
  RangeValue= right(Round(RangeMin,0.01)) ||
" - " || left(Round(RangeMax,0.01));
  Varname = "&X";

*****2. Plot Graphs *****;
filename output "&X..gif";
goptions dev=gif gsfname=output gsfmode=replace;

data emdata.addn;
  set emdata.current2;
  xord=_n_;

data emdata.xname;
  set emdata.addn(rename=(xord=start rangevalue=label));
  fmtname='dataord';
  type='N';
  keep fmtname label start type;

proc format cntlin=emdata.xname;

symbol1 interpol=needle c=blue width=40 value=none;

symbol2 interpol=j value=star c=red;

axis1 label=( font=swiss height=1.5 "&X") offset=(5) value=(height=1) split=" - "
order=(1 to 10 by 1);

axis2 label=(angle=90 font=swiss h=1.5 'Bin Count' ) ;
axis3 label=(angle=90 font=swiss h=1.5 'Response Rate') ;

proc gplot data=emdata.addn gout=&X;
  plot count*rangevalue/overlay legend=legend1 haxis=axis1 vaxis=axis2;
  plot2 Responserate*rangevalue/overlay legend=legend2 haxis=axis1 vzero
vaxis=axis3 ;
  format xord dataord.;
  title1 h=1.5 f=swiss "Count and ResponseRate for '&X'";
/*****3. Output Dataset *****/
data &Output_DataSet;
  set &Output_DataSet
  emdata.Empty
  emdata.current2;
  run;
%mend createDeciles;
%macro ParseAndPassInterval(X,Y,P_low,P_high,Bin);
  data emdata.Empty;
  format VarName $50.;
  format RangeMin 12.2;
  format RangeMax 12.2;
  format RangeValue $50.;
  format Count 20.;

```

```

format ResponseRate 8.4;
format ResponseCount 20.;

data &Output_DataSet;
set emdata.Empty;
if _N_=1 then delete;
%let i=1;
%do %while(%length(%scan(&X,&i," ")) >0);
    %CreateDeciles(%scan(&X,&i," "),&Y,&P_low,&P_high,&Bin);
    %let i = %eval(&i+1);
%end;

%mend ParseandPassInterval;

%ParseandPassInterval(&X_Variables,&Y_Variable,&lowestpctl,&highestpctl,&Bins);
/*****
Univariate Plot Program 2
Fixed Count
KeyBank
*****/

/*****Define Parameters*****/
%let Input_DataSet=emdata.check2_build;
%let Output_DataSet=emdata.paper_Graph2;
%let X_Variables=
MTD_CHRG_BACK_Q
MTD_CR_A
;
%let Y_Variable=close;
%let lowestpctl=2;
%let highestpctl=98;
%let Bins=10;
/*****Globle Statement and Option*****/
option mprint symbolgen;
legend1 label=none
    position=(top center inside)
    offset=(-2 cm,)
    mode=share;
legend2 label=none
    position=(top center inside)
    offset=(2 cm,)
    mode=share;

/*****Macro to Create Deciles, Plot Graphs, Output Dataset*****/
%macro CreateDeciles(X,Y,P_low,P_high,Bin);
    data emdata.current (Keep=&X &Y);
    set &Input_DataSet;
    *****.1. Create Deciles*****.
proc Univariate data = emdata.current noprint;
    var &X;
    output out=emdata.temp(keep=P0 P&P_low P&P_high P100)
    PctlPre= P
    pctlpts= 0 &P_low &P_high 100;

data _null_;
    set emdata.temp;
    call Symput('Min',P0);
    call Symput('Max',P100);
    call Symput('P_low_value',P&P_low.);
    call Symput('P_high_value',P&P_high.);
    call Symput('StandardBinRange', Round(((P&P_high - P&P_low) /&Bin), 0.001));

Data emdata.Outliers_On_Left
emdata.Outliers_On_Right
emdata.current1
;
set emdata.current;
if &X < &P_Low_Value
then output emdata.Outliers_On_Left;

```

```

        else if &X > &P_High_Value
        then output emdata.Outliers_On_Right;
        else output emdata.current1;

proc rank data=emdata.current1 groups=&Bin out=emdata.current1;
    var &X;
    ranks dec;
    run;
proc sql;
    create table emdata.current2 as
    select Min(&X) as RangeMin,
           Max(&X) as RangeMax,
           Count(*) as Count,
           Avg(&Y)*100 as ResponseRate,
           Sum(&Y) as ResponseCount
    from emdata.current1
    group by dec;

data emdata.current2;
    set emdata.current2;
    format RangeValue $50.;
    format Varname $50.;
    RangeValue= right(Round(RangeMin, 0.01)) ||
" - " || left(Round(RangeMax,0.01));
    Varname = "&X";
*****2. Plot Graphs *****;
filename output "&X.gif";
options dev=gif gsfname=output gsfmde=replace;

data emdata.addn;
    set emdata.current2;
    xord=_n_;

data emdata.xname;
    set emdata.addn(rename=(xord=start rangevalue=label));
    fmtname='dataord';
    type='N';
    keep fmtname label start type;

proc format cntlin=emdata.xname;

symbol1 interpol=needle
        c=blue width=40
        value=none;
symbol2 interpol=j
        value=star
        c=red;

axis1 label=( font=swiss height=1.5 "&X") offset=(5)
split=" - " order=(1 to 10 by 1)
value=(height=1);
axis2 label=(angle=90 font=swiss h=1.5 'Bin Count' ) ;
axis3 label=(angle=90 font=swiss h=1.5 'Response Rate');
proc gplot data=emdata.addn gout=&X;
    plot count*xord/overlay legend=legend1 haxis=axis1 vaxis=axis2;
    plot2 Responserate*xord/overlay legend=legend2 haxis=axis1 vzero
vaxis=axis3 ;
format xord dataord.;

    title1 h=1.5 f=swiss "Count and ResponseRate for '&X'";
run;
*****3. Output Dataset *****;
data &Output_DataSet;
    set &Output_DataSet
        emdata.Empty
        emdata.current2;
    run;
%mend createDeciles;
%macro ParseAndPassInterval( X, Y, P_low, P_high, Bin);
data emdata.Empty;

```

```

format VarName $50.;
format RangeMin 12.2;
format RangeMax 12.2;
format RangeValue $50.;
format Count 20.;
format ResponseRate 8.4;
format ResponseCount 20.;

data &Output_DataSet;
  set emdata.Empty;
  if _N_=1 then delete;
  %let i=1;
  %do %while(%length(%scan(&X,&i," ")) >0);
    %CreateDeciles(%scan(&X,&i," "),&Y,&P_low,&P_high,&Bin);
    %let i = %eval(&i+1);
  %end;
%mend ParseandPassInterval;
%ParseandPassInterval(&X_Variables,&Y_Variable,&lowestpctl,&highestpctl,&Bins);
*****;

```

Program One and Two are essentially the same, but use different ways to create deciles (by range or by count) as stated above. The input parameters to both programs consist of:

1. The name of the input data set (&Input_DataSet), which contains all the observations of X variables and Y variable;
2. The name of output dataset (&Output_DataSet), which stores the output results for further analysis;
3. The name of the response variable (&Y_Variable);
4. The list of X variables to create univariate plots (&X_Variables);
5. And the number of segments to use to group X variables (&Bins).

To build more flexibility into this MACRO and allow users to do “Capping” and “Flooring”, we added two more parameters:

6. The lowest z percentiles of X to be excluded (&lowestpctl)
7. The highest w percentiles of X to be excluded (&highestpctl)

The output columns from both programs consist of:

1. The X variable name (&VarName);
2. The minimum value of each decile (&RangeMin);
3. The maximum value of each bin (&RangeMax);
4. The count of observations within each bin (&Count);
5. The response rate within each bin (&ResponseRate); and
6. The count of responders (&ResponseCount)

EXAMPLES AND DEMONSTRATIONS

In a project at Keybank to build an attrition model to predict the closure of Small Business Checking accounts, a modeling data set was created with 1,200 variables and millions of records. The variables in the final data were collected from both internal and external sources. Data types varied including firmagraphic, transactional, and econometric data. Trend variables were also derived to observe the change in values in many of the primary variables. The response variable was an indicator of attrition with a value of 0 or 1.

After developing the dataset of eligible variables, the field had to be narrowed to the stronger contenders. In a smaller dataset with fewer variables, it may not be necessary to reduce the number of variables for final eligibility in the model.

At this point in the project, univariate analysis was used to explore potential transformations for the variables and to detect potentially significant variables for the final model. Given the large number of variables, this could have been very time-consuming. With the SAS® MACRO created above, this graphing step now became relatively easy.

A portion of input modeling dataset emdata.check2_build looks like the following:

	MTD_CHRG_		MTD_PAPR_	
Obs	BACK_Q	MTD_CR_A	DR_Q	. attritor

```

1      0      1.73      0      .      .      0
2      0      0.00      3      .      .      0
3      0      0.00      0      .      .      0
4      0      0.00      0      .      .      0
5      0      4968.37     14     .      .      0
6      2     29261.65     21     .      .      1
7      0     1221.35      7      .      .      0
8      0     14257.97      8      .      .      0
9      0     19869.90     33     .      .      1
10     0     2622.41      4      .      .      0
11     0     1800.15      0      .      .      0
12     0      214.76      2      .      .      0
13     0     20519.04     69     .      .      0
14     0     89784.32     74     .      .      0
15     0      0.00       2      .      .      0
16     0     173179.70     66     .      .      0
17     0     24000.00    165     .      .      0
18     0     100962.00      0      .      .      0
19     0      989.34      0      .      .      0
20     1      6475.91     20     .      .      0
21     0      5476.90      7      .      .      0
22     0     89539.05     91     .      .      0
. . .
. . .
. . .
. . .

```

Define input parameter in SAS® MACRO as below:

```

%let Input_DataSet=emdata.check2_build;
%let Output_DataSet=emdata.paper_Graph1;
%let X_Variables=

MTD_CHRG_BACK_Q
MTD_CR_A

;
%let Y_Variable=attritor;
%let Bins=10;
%let lowestpctl=2;
%let highestpctl=98;

```

The codes above indicate to the MACRO to generate 10-bin univariate graphs on variables “MTD_CHRG_BACK_Q” and “MTD_CR_A” in the input data set “emdata.check2_build”. The binary dependent variable is called “attritor” in the same input dataset. The result is output to a dataset called “emdata.paper_Graph1”. To exclude the outliers that are single or low frequency occurrences and far from the mean, the capping and flooring rule is set to be 2%. That causes the Macro to only consider the values between the 2nd percentile and the 98th percentile to generate the univariate graphs.

The SAS® MACRO program generates the results as below in figure1 and figure2:

VarName	RangeMin	RangeMax	RangeValue	Response Count	Response Rate	Response Count
MTD_CHRG_BACK_Q	0	0	0 - 0	127105	13.13	16689
MTD_CHRG_BACK_Q	1	1	1 - 1	6259	7.89	494
MTD_CHRG_BACK_Q	2	2	2 - 2	1931	8.75	169
MTD_CHRG_BACK_Q	3	3	3 - 3	838	7.99	67
MTD_CHRG_BACK_Q	4	4	4 - 4	492	7.18	55
MTD_CHRG_BACK_Q	5	5	5 - 5	299	7.69	20
MTD_CR_A	0	500883	0 - 500883.27	131435	12.69	16673
MTD_CR_A	501262	1002086	501261.6 - 1002086.46	2352	12.15	215
MTD_CR_A	1003316	1502753	1003315.56 - 1502752.7	989	11.15	81
MTD_CR_A	1503391	2003761	1503390.65 - 2003761.2	563	12.80	61
MTD_CR_A	2008737	2505228	2008736.94 - 2505228.0	433	12.39	36
MTD_CR_A	2505496	3006245	2505496.12 - 3006245.4	307	13.36	38
MTD_CR_A	3011859	3505076	3011858.56 - 3505075.9	211	11.22	15
MTD_CR_A	3509284	4000000	3509283.75 - 4000000	137	12.60	13
MTD_CR_A	4011158	4507140	4011158.16 - 4507140	130	11.15	10

Figure 1

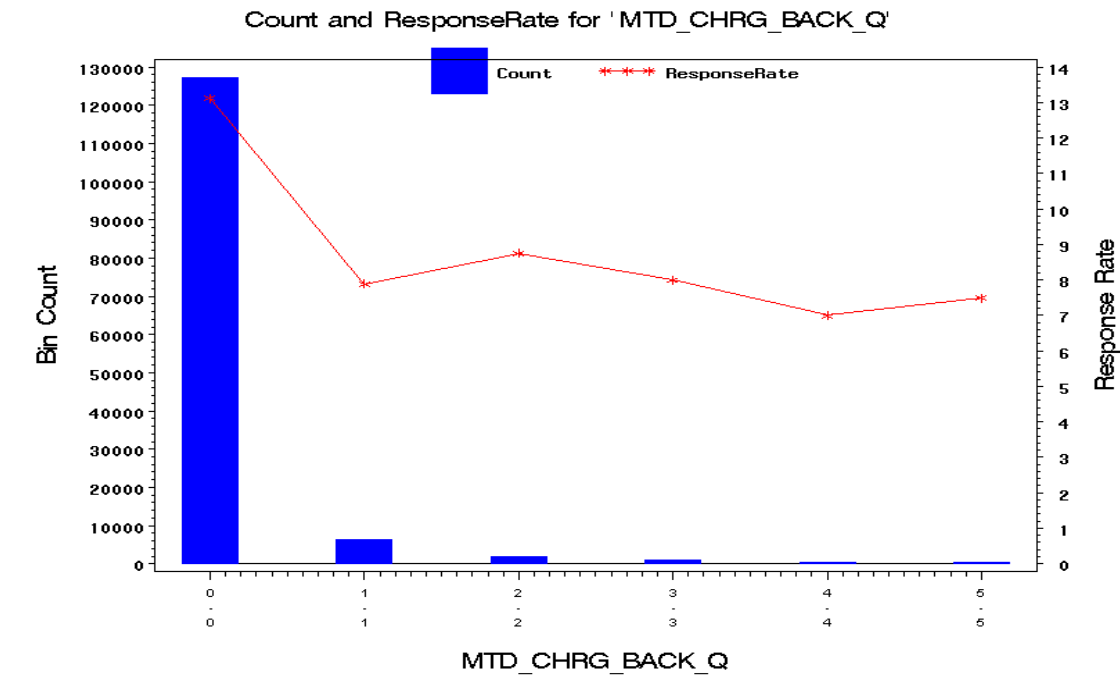


Figure 2

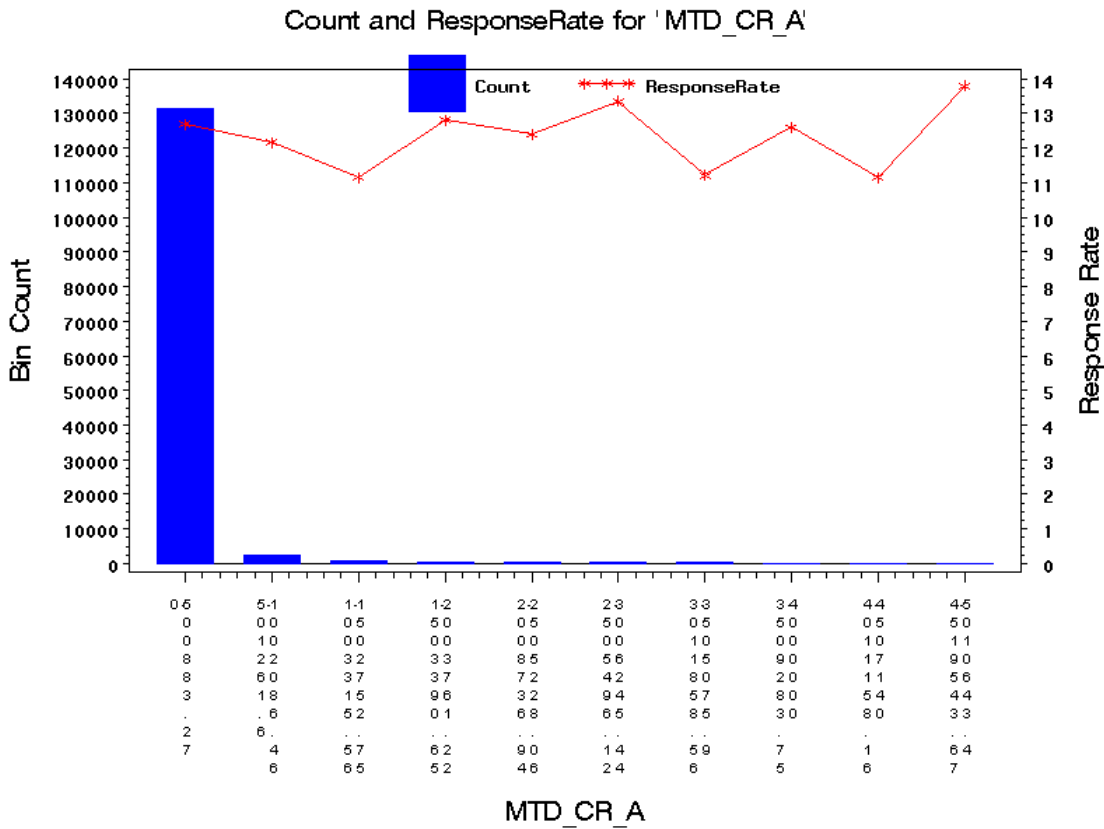


Figure 2 (cont)

The bars in the graph representing the counts of observations in each bin are left aligned to the “Bin Count” axis. The star line in the graph representing the attrition rate on each bin is right aligned to the “Response Rate” axis.

From the univariate graphs in Figure2, it's easily to detect that for variable `MTD_CHRG_BACK_Q` (Month-to-date check charge back quantity), most of the customers don't have charged-back checks. This is indicated by the bar on the first bin where `MTD_CHRG_BACK_Q=0`. There is a natural drop in attrition rate as long as the quantity is greater than 0 (13% vs. 8%). Therefore it's appropriate to transform this variable into categorical variable as

```
MTD_CHRG_BACK_Q_I=0;
If MTD_CHRG_BACK_Q>0 then MTD_CHRG_BACK_Q_I=1.
```

For variable `MTD_CR_A` (Month-to-date credit amount), there does not appear to be an obvious trend at a glance. The attrition rate jumps back and forth within the range of 12% to 14%. However, the majority (>99%) of customers have monthly credit amount under \$500K while the range of variable value is from \$0 to \$4.5 million. The bar (representing the counts) is very high on the first bin and it is unclear if there may be differentiation within that group. The counts in lower deciles are so low and on the outlying edges, that it is difficult to interpret their values. In this situation, the second graphing method can be used to force binning based on the counts rather than the range.

The univariate plot generated using method 2 is shown in figure 3. Instead of each bin having the equal range, the count within each bin is equal (except for ties), leaving the range of the bin to be varied. For example, the last bin in the new univariate plot has about 1/10 of the population, but range from 114k to 2 million, which accounts for 94% of the whole range of variable.

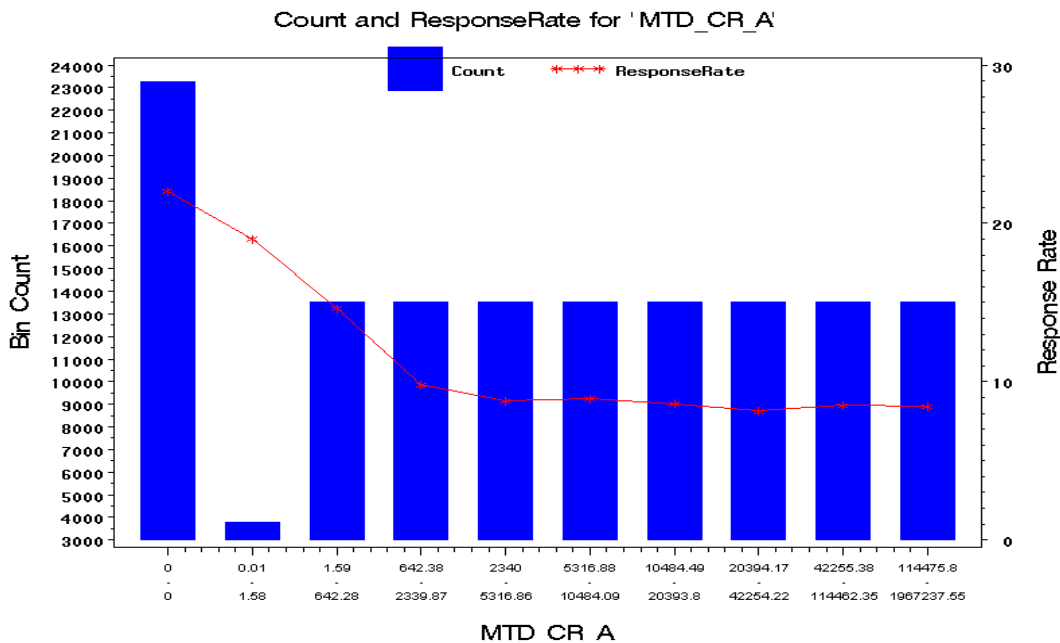


Figure 3

Figure3 gives a more detailed look at `MTD_CR_A` < 500k. It is now apparent that credit amount and attrition rate has a negative correlation. When the credit amount is greater than the value around 5th decile, the attrition rate levels off. A transformation to capture this is as follows:

```
MTD_CR_A_T= MTD_CR_A;
If MTD_CR_A> 5500 then MTD_CR_A_T= 5500;
```

Now re-running the same SAS® MACRO on the **transformed** variables, the new results are illustrated in Figure 4 and Figure5:

Response VarName	RangeMin	RangeMax	RangeValue	Count	Rate	Count
MTD_CHRG_BACK_Q	0	0	0 - 0	127105	13.13	16689
MTD_CHRG_BACK_Q	1	5	1 - 5	9819	8.19	805
MTD_CR_A	0	0	0 - 0	23259	22.01	5120
MTD_CR_A	0.01	2.28	0.01 - 2.28	4083	18.62	760
MTD_CR_A	2.29	720	2.29 - 720	13664	14.94	2041
MTD_CR_A	720.09	2511.7	720.09 - 2511.7	13669	12.75	1743

MTD_CR_A	2511.95	5729	2511.95 - 5729	13669	9.83	1344
MTD_CR_A	5729.5	9999.88	5729.5 - 9999.88	12384	8.83	1094
MTD_CR_A	10000	10000	10000 - 10000	62073	8.79	5456

Figure 4

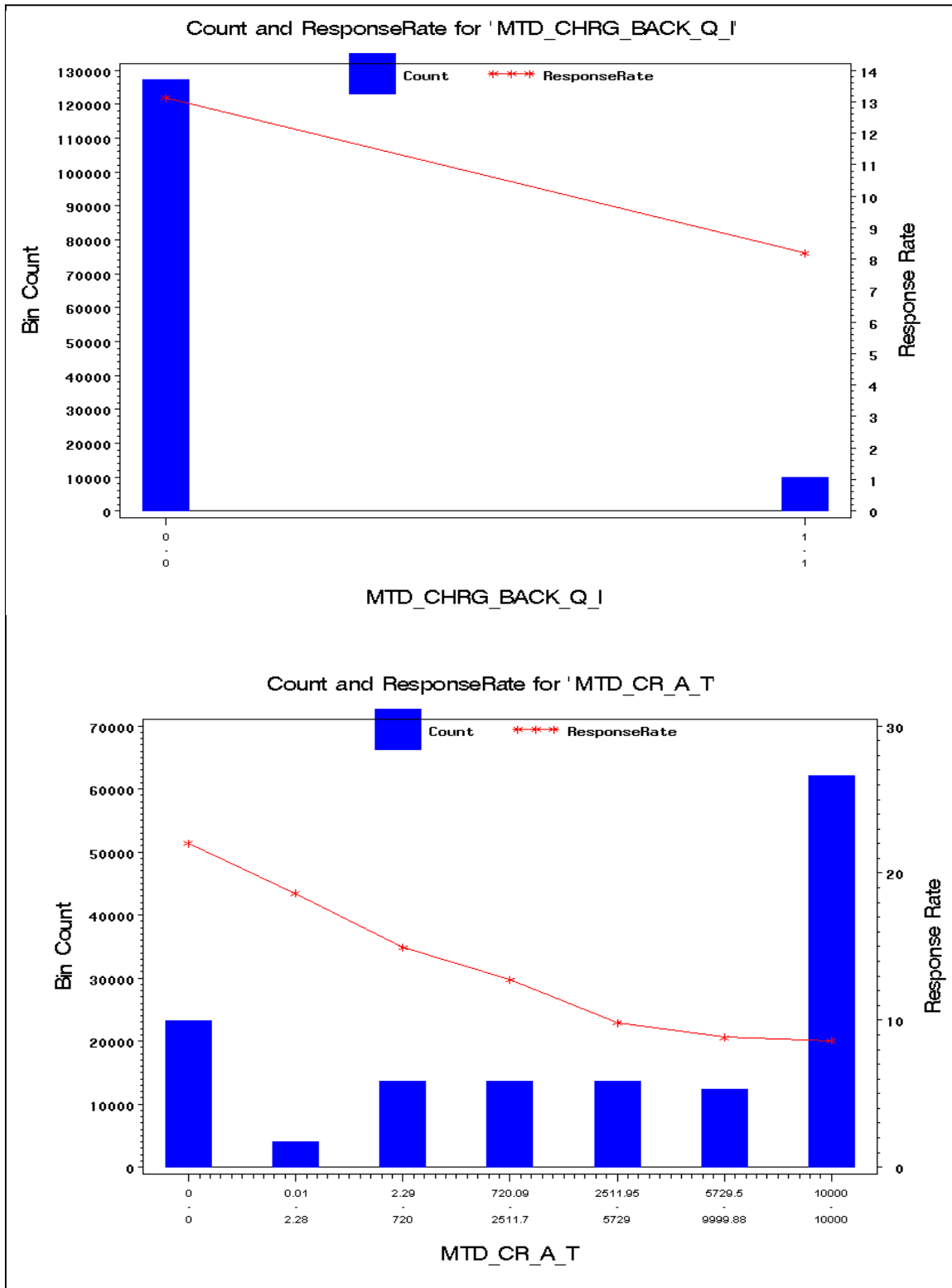


Figure 5

In addition to the obvious trend indicated by the new univariate plot, the comparison of p-value from univariate logistic regression results before and after transformations demonstrates that the transformation significantly increases the predicting power of both variables (see Figure 6).

Score Variable	Before After	
	Pr > Chi-Square	Chi-Square
MTD_CHRG_BACK_Q	77.8	0.1276
MTD_CHRG_BACK_Q_I	167.8913	<0.0001
MTD_CR_A	0.5224	0.1921
MTD_CR_A_T	151.7017	<0.0001

Figure 6

CONCLUSIONS

This SAS® MACRO is a very handy tool in the variable transformation step of modeling using logistic regression. Using the SAS® MACRO, various views of the X variable are systematically considered. This allows for the discovery of important transformations even in environments with very large datasets. Real examples from Keybank show that with this visual tool, variable transformations become more apparent and lead to very significant improvements in their contributions to logistic regression models.

REFERENCES

1.Olivia Parr Rud. Data Mining Cookbook. P88-90.

CONTACT INFORMATION

Your Comments and questions are valued and encouraged. Contact the authors at:

Maggie Zhang:216-689-3290

Simon Chen: 216.689.0358

Suzanne Cashman Rain: 216-689-4565

KeyBank

Mailcode: OH-01-27-0302

127 Public Square

Cleveland, OH 44114

Email: Maggie_Zhang@keybank.com

Simon_Chen@keybank.com

Suzanne_C_Rain@keybank.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.