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.

Plot of Y*X. When Y is dichotomous.

Plot of Y*X. When Y is continuous.

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 curve. 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

```sas
/*/********************************************************* *
 Univariate Plot Program 1
 Fixed Range
 KeyBank
 *********************************************************/
%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= &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=axis3 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";
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)
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;
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:

<table>
<thead>
<tr>
<th>Obs</th>
<th>MTD_CHRG_</th>
<th>MTD_PAPR_</th>
<th>MTD_PAPR_</th>
<th>attritor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BACK_Q</td>
<td>MTD_CR_A</td>
<td>DR_Q</td>
<td></td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>VarName</th>
<th>RangeMin</th>
<th>RangeMax</th>
<th>RangeValue</th>
<th>Count</th>
<th>Rate</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>0</td>
<td>0</td>
<td>0 - 0</td>
<td>127105</td>
<td>13.13</td>
<td>16689</td>
</tr>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>1</td>
<td>1</td>
<td>1 - 1</td>
<td>6259</td>
<td>7.89</td>
<td>494</td>
</tr>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>2</td>
<td>2</td>
<td>2 - 2</td>
<td>1931</td>
<td>8.75</td>
<td>169</td>
</tr>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>3</td>
<td>3</td>
<td>3 - 3</td>
<td>838</td>
<td>7.99</td>
<td>67</td>
</tr>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>4</td>
<td>4</td>
<td>4 - 4</td>
<td>492</td>
<td>7.18</td>
<td>55</td>
</tr>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>5</td>
<td>5</td>
<td>5 - 5</td>
<td>299</td>
<td>7.69</td>
<td>20</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>0</td>
<td>500883</td>
<td>0 - 500883.27</td>
<td>131435</td>
<td>12.69</td>
<td>16673</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>501262</td>
<td>1002086</td>
<td>501261.6 - 1002086.46</td>
<td>2352</td>
<td>12.15</td>
<td>215</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>1003316</td>
<td>1502753</td>
<td>1003315.56 - 1502752.7</td>
<td>989</td>
<td>11.15</td>
<td>81</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>1503391</td>
<td>2003761</td>
<td>1503390.65 - 2003761.2</td>
<td>563</td>
<td>12.80</td>
<td>61</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>2008737</td>
<td>2505228</td>
<td>2008736.94 - 2505228.0</td>
<td>433</td>
<td>12.39</td>
<td>36</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>2505496</td>
<td>3006245</td>
<td>2505496.12 - 3006245.4</td>
<td>307</td>
<td>13.36</td>
<td>38</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>3011859</td>
<td>3507576</td>
<td>3011858.56 - 3507575.9</td>
<td>211</td>
<td>11.22</td>
<td>15</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>3509284</td>
<td>4000000</td>
<td>3509283.75 - 4000000</td>
<td>137</td>
<td>12.60</td>
<td>13</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>4011158</td>
<td>4507140</td>
<td>4011158.16 - 4507140</td>
<td>130</td>
<td>11.15</td>
<td>10</td>
</tr>
</tbody>
</table>
Figure 1

Count and Response Rate for 'MTD_CHRG_BACK_Q'

Figure 2

Count and Response Rate for 'MTD_CR_A'

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 Figure 2, 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.

![Count and Response Rate for MTD_CR_A](image)

Figure 3

Figure 3 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 Figure 5:

<table>
<thead>
<tr>
<th>VarName</th>
<th>RangeMin</th>
<th>RangeMax</th>
<th>RangeValue</th>
<th>Count</th>
<th>Rate</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>0</td>
<td>0</td>
<td>0 - 0</td>
<td>127105</td>
<td>13.13</td>
<td>16689</td>
</tr>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>1</td>
<td>5</td>
<td>1 - 5</td>
<td>9819</td>
<td>8.19</td>
<td>805</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>0.01</td>
<td>2.28</td>
<td>0.01 - 2.28</td>
<td>4083</td>
<td>18.62</td>
<td>760</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>2.29</td>
<td>720</td>
<td>2.29 - 720</td>
<td>13664</td>
<td>14.94</td>
<td>2041</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>720.09</td>
<td>2511.7</td>
<td>720.09 - 2511.7</td>
<td>13669</td>
<td>12.75</td>
<td>1743</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>2511.95</td>
<td>5729</td>
<td>2511.95 - 5729</td>
<td>13669</td>
<td>9.83</td>
<td>1344</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>---------</td>
<td>----------------</td>
<td>---------</td>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>5729.5</td>
<td>9999.88</td>
<td>5729.5 - 9999.88</td>
<td>12384</td>
<td>8.83</td>
<td>1094</td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>10000</td>
<td>10000</td>
<td>10000 - 10000</td>
<td>62073</td>
<td>8.79</td>
<td>5456</td>
</tr>
</tbody>
</table>

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).
### Figure 6

<table>
<thead>
<tr>
<th>Variable</th>
<th>Before Score</th>
<th>Before Pr &gt; Chi-Square</th>
<th>After Score</th>
<th>After Pr &gt; Chi-Square</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTD_CHRG_BACK_Q</td>
<td>77.8</td>
<td>0.1276</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTD_CHRG_BACK_Q_I</td>
<td>167.8913</td>
<td>&lt;0.0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTD_CR_A</td>
<td>0.5224</td>
<td>0.1921</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MTD_CR_A_T</td>
<td>151.7017</td>
<td>&lt;0.0001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 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


### 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.