Data Exploration for Sweet Spots by Using Heat Maps in SAS®

Alec Zhixiao Lin, LoanDepot, Foothill Ranch, CA

ABSTRACT

In business or in research, we often seek to find sweets spots in a population, such as a cluster of customers with unusually high activities of fraud or an area suggesting an extreme potential for complex organic molecule formation. Expert opinions and past experiences are two common sources for drawing such insights, but in current age new data constantly emerge with new insights hidden for discovery. Screening all variables in pair is a useful but arduous practice to find patterns in data. This paper introduces an efficient process in SAS that will automatically pair up two variables for cross tabulation analysis. The heat maps from PROC SGPLOT will help with the discovery of sweet spots in the data.

INTRODUCTION

Many companies use data analysis to closely identify and target micro customer segments with similar traits with respect to life cycle status, earnings power, and perceived needs in order to improve marketing efficiency, increasing customer loyalty and preventing or reducing fraud. Socio-demographic attributes combined with behavioral information such as transaction records are commonly used for conducting such research internally. In the digital age new data constantly emerge and are being created, so relying on past business experience or expert opinion might not situate on in a winning position in the market. New learning from data is constantly sought after. Data exploration by using clustering and classification to find behavioral patterns in customers is a common practice.

Beyond analytics related to commercial activities, exploratory data analysis enables a scientist to quickly develop a more refined, testable hypothesis and rigorous experimental design for subsequent hypothesis testing.
While sifting through data of a very large scale, an emergence pattern can often lead a chance discovery into a testable theory.

Visualization is a very important routine in conducting exploratory data analysis. “Sweet spots” based on a single attributes can be easily identified by conducting a bivariate analysis between a potential independent variable and a target outcome such as marketing response, risk or revenue. For example, applicants with recent delinquency records are usually rejected by many credit card companies. Screening two attributes together is more challenging. Without a priori knowledge, one usually have little idea of which two variables should be paired for the analysis. This paper introduce a process in SAS that will automatically go through all potential variables and examine all overlays between two variables. A series of heat maps will be generated for all overlays. Analysts and screen through these graphs to come up with ideas for business rules.

THE METHOD

If one has ten variables to explore, there are altogether 44 combinations[1]. Number of combination will increase exponentially if number of variable increases. Prescreening variables by using bivariate analysis will reduce the processing time for examination of variables in pair.

Step 1 - Single-attribute screening

For character variables, PROC MEANS usually reveals a flag or value that shows highest outcome. For numeric variables, PROC RANK and PROC MEANS can be used to determine segments of records that show higher performance. For example, in credit underwriting, a high number of inquiries – e.g., > 3 – usually suggests a higher risk. To process both types of variables we suggest using Information Value (IV) that will analyze the predictive power for both binary and continuous outcomes without considering the monotonicity of a variable.

Many time a bivariate analysis of a single attribute in relation to an outcome might not yield powerful enough insights. In this case, cross tabulation is usually a recommendable practice for overlaying one variable on top another in hope of finding patterns. Compared to regression models, business rules based on this finding is more straightforward to be communicated to business peers. Also, some sweet spots in the data carries statistical significance, but not powerful enough to be used as a dummy variable in a model. For example, in a sample of 100,000 records, if a certain segments of 60 records shows a much higher risk, a dummy variable for this segment might not be picked up by the model.

This paper introduces a SAS process that will automatically do the following:

[1] One variable can pair with other variables for 9 times and cannot pair with itself, so the number of combination is (9+2) x 5=45.

- Automatically pair all numeric and character variables.
- Examine the cross tabulation of the pair of variables in relation to the outcome. The SAS program retains the flexibility when only numeric or character variables are available to use.
- Use PROC SGPLOT to automatically generate heat maps to sweet spots to emerge.

The number of heat maps produced depends on number of variables. Increase the number of variable will exponentially increases the number of image files produced. For example, 3 attributes will generate 3+2+1=6 output images, and 10 attributes will generate 55 images, and 50 attributes will generate 1275 graphs. In order to reduce the number of graphs without losing potential insights from the data, I suggest users run some preliminary examinations based on the following considerations:

- Reduce number of variables with high overlap in business meanings. For example, in evaluating sales performance, sales volume in last 3, 6 or 12 months are highly correlated. Analysis on them yields very similar, if not same, insights. Retain one such variable is usually sufficient.
- Run a variable reduction process to choose those with higher Information Values (IV) for the subsequent round of overelay analysis. A variable showing "flat" performance in relation to a target in interest usually will not be useful unless significant intereration exists between two variables.

The SAS Program

Suppose a user already has a sample prepared for the analysis, he/she only needs to define the following macro values to run the program:

```
libname your_lib "&libstr\zlin\trade data\data";  /* data library*/
%let datalib=your_lib;        /* libname corresponding the above */
%let inset=check_trade;       /* data set to be used */
%let y=bad;                   /* Target variable */
%let yformat=percent6.2;      /* format of target variables */
%let vartxt=
xchar1 xchar2 xchar3
;                             /* list of all character variables.
                                 Leave it blank if none. */
%let varnum=
xnum1 xnum2 xnum3 xnum4 xnum5
;                             /* list of all numeric variables.
                                 Leave it blank if none. */
%let binnum=10;                /* number of bins for numeric variables */
%let graphfolder=&libstr\zlin\trade data\output;
                              /* folder for output file */
%let graphname=check_overlay; /* name of the output file in pdf */
```

The following macro values define the color patterns of the heat maps. In our example, green suggests segments of lowest risk. It graduates to yellow and orange as the risk increases, and reaches red for the segments of highest risk.

```
%let missingnum=-9999999999;              /* filler for missing numeric value */
%let missingchar=_MISSING_;               /* filler for missing character value */
%let heaty=(green yellow orange red);     /* color pattern for performance data */
%let heatdist=TwoColorRamp;               /* color pattern for record distribution */
```

After gaining some hands-on experience, users can experiment with above macro values to find color patterns that suit his/her preference.

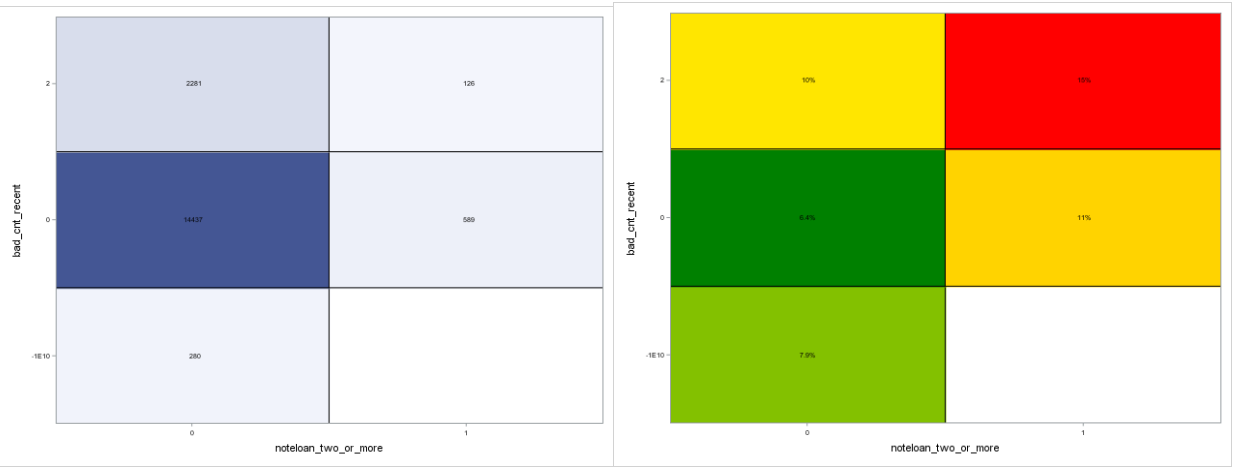
Users do not need to make any changes to the rest of the SAS program.

THE OUTPUT

PROC SGPLOT plot produces high-resolution graphs that yields very straightforward insights from the data and that can be used for illustration or presentation. Multiple options exists for produce cross
The primary output is a series of pairs of graphs. Each pair of graphs contains the following:

- Cross tabulate of record distribution
- Performance by the bins

For character attributes, the axis uses the values/flags as the ticker. For numeric attributes, the ticker for a segment is the median value of the bins. Users can use the two graphs to determine whether the strongest-colored segment can be picked to design a business rule. In this example, the business rule can be designed as follows:



The graph on the left shows the cross-tabulate distribution by cell, while graph on the right shows the performance of each cell. A blank cell means no records exists there. Users can decide whether the performance in one- usually - or more cells can be a potential business rule based on the following considerations:

- Whether the performance of the segment is high enough.
- Whether the segment contains enough records for a valid inference to be drawn.

For character attributes, the axis shows its values or flags. For numeric variables, the axis shows the median value of the associated cell. The SAS output also includes associated SAS code that can be used to design business rules.

Use it for anomaly detection, and then form intuitive business rules based on the detection. Analysts can use the segment as either a dummy variable in regression or for designing business rules.

Just find pattern is not enough. One need to think the causes that drive the pattern. That is, we need to explain the pattern. In scientific research, this usually will make discovery of new theories possible.

CONCLUSION

The renowned philosopher Aristotle (384-322 BC) was one of the first to approach knowledge discovery through rigorous, systematic observation, This paper introduces a process in SAS that makes this observation a bit easier for possible discovery of “sweet spots”.

REFERENCES

Lin, Alec Zhixiao, Entropy-Based Measures of Weight of Evidence and Information Value for Variable Reduction and Segmentation for Continuous Dependent Variables (SUGI 2015)
http://support.sas.com/resources/papers/proceedings15/3242-2015.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at
Alec Zhixiao Lin
VP of Modeling
Loan Depot
26642 Towne Center Drive

Foothill Ranch, CA 92610
Email: alecindc@gmail.com
Web: www.linkedin.com/pub/alec-zhixiao-lin/25/708/261/

APPENDIX

```
libname your_lib "&libstr\zlin\trade data\data";  /* data library*/
%let datalib=your_lib;        /* libname corresponding the above */
%let inset=check_trade;       /* data set to be used */
%let y=bad;                   /* Target variable */
%let yformat=percent6.2;      /* format of target variables */
%let vartxt=
xchar1
xchar2
xchar3
;                             /* list of all character variables.
                                 Leave it blank if none. */
%let varnum=
xnum1
xnum2
xnum3
xnum4
xnum5
;                             /* list of all numeric variables.
** list of numeric variables;
                    ** keep it null if no numeric variables;
%let binnum=8;      ** number of bins for numeric variables;
%let y=dq;          ** target variable y;
%let yformat=percent6.2;    ** format of target variable;
%let missingnum=-9999999999;  ** for missing numeric variables;
%let missingchar=_MISSING_;   ** for missing categorical variables;
%let heaty=(green yellow orange red);  ** color pattern for heat maps;
%let heatdist=TwoColorRamp;            ** color pattern for record distributions;
%let graphfolder=&libstr\zlin\legacy SAS programs\output; ** folder for output;
%let graphname=check_overlay;     ** file name for heat maps in pdf;


** count numeric variables;
data check_contents;
retain &varnum;
set &datalib..&inset(keep=&varnum obs=1); run;

proc contents data=check_contents varnum out=check_contents2; run;
proc sort data=check_contents2(keep=name varnum)
out=checkfreq(rename=(name=tablevar)); by varnum; run;

data varcnt; set checkfreq; varcnt+1; run;

proc sql; create table vcnt as select count(*) as vcnt from varcnt; quit;
data _null_; set vcnt; call symputx('numcnt', vcnt); run;

proc sql noprint; select tablevar into :vnum1-:vnum&numcnt from varcnt; quit;
proc sql noprint; select tablevar into :wnum1-:wnum&numcnt from varcnt; quit;

** count character variables;
data check_contents;
```

```
retain &vartxt;
set &datalib..&inset(keep=&vartxt obs=1); run;

proc contents data=check_contents varnum out=check_contents2; run;
proc sort data=check_contents2(keep=name varnum)
out=checkfreq(rename=(name=tablevar)); by varnum; run;

data varcnt; set checkfreq; varcnt+1; run;

proc sql; create table vcnt as select count(*) as vcnt from varcnt; quit;
data _null_; set vcnt; call symputx('txtcnt', vcnt); run;

proc sql noprint; select tablevar into :vtxt1-:vtxt&txtcnt from varcnt; quit;
proc sql noprint; select tablevar into :wtxt1-:wtxt&txtcnt from varcnt; quit;


%macro xtabnum(x1, x2);
proc rank data=&datalib..&inset groups=&binnum out=xdata;
var &x1 &x2;
ranks rankyaxis rankxaxis; run;

data xdata; set xdata;
rankyaxis=rankyaxis+1;
rankxaxis=rankxaxis+1;
if rankyaxis=. then do; rankyaxis=0; &x1=&missingnum; end;
if rankxaxis=. then do; rankxaxis=0; &x2=&missingnum; end;
run;

proc summary data=xdata nway;
var &y;
class rankyaxis rankxaxis/ missing order=data;
output out=tempcheckdata
       sum=y_sum;
run;

proc sql noprint;
select case when min(y_sum/_freq_) ge 0 and max(y_sum/_freq_) le 1 then
       int(1000/sum(max(y_sum/_freq_/2), min(y_sum/_freq_/2))) else 1000 end into
:multiplier
from tempcheckdata; quit;

data check_mean_cnt;
set tempcheckdata;
y_mean=y_sum/_freq_;
y_n=_freq_;
y_mean_int=int(y_mean*&multiplier);
if y_mean_int in (0, 1) then y_mean_int=1;

format y_mean &yformat;
informat y_mean &yformat;
run;

proc means data=xdata median min max nway noprint;
class rankyaxis;
var &x1;
output out=check_x1(drop=_type_ _freq_)
    median=&x1 min=min_yaxis max=max_yaxis; run;

proc means data=xdata median min max nway noprint;
class rankxaxis;
var &x2;
output out=check_x2(drop=_type_ _freq_)
    median=&x2 min=min_xaxis max=max_xaxis;
```

```
    run;

proc sql;
create table graph_data_num
as select a.*,
          b.&x1, min_yaxis, max_yaxis,
               c.&x2, min_xaxis, max_xaxis
from check_mean_cnt a,
     check_x1 b,
     check_x2 c
where a.rankyaxis=b.rankyaxis
  and a.rankxaxis=c.rankxaxis; quit;

proc sort data=graph_data_num; by &x2 &x1; run;

ods layout Start width=10in height=8in;
ods region x=0% y=0% width=50% height=30%;
proc sgplot data=graph_data_num NOAUTOLEGEND ;
XAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_n discretex discretey colormodel=&heatdist outline;
text x=&x2 y=&x1 text=y_n / textattrs=(size=5pt);
title;
footnote;
run;

ods region x=50% y=0% width=50% height=30%;
proc sgplot data=graph_data_num NOAUTOLEGEND ;
XAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_mean_int discretex  discretey colormodel=&heaty outline;
text x=&x2 y=&x1 text=y_mean / textattrs=(size=5pt);
title;
footnote;
run;
quit;
ods layout end;
%mend;


%macro xtabtxt(x1, x2);
data xdata; set &datalib..&inset;
if compress(&x1)=' ' then &x1="&missingchar";
if compress(&x2)=' ' then &x2="&missingchar";

rankyaxis=&x1;
rankxaxis=&x2;
run;

proc summary data=xdata nway;
var &y;
class rankyaxis rankxaxis/ missing order=data;
output out=tempcheckdata
       sum=y_sum;
run;

proc sql noprint;
select case when min(y_sum/_freq_) ge 0 and max(y_sum/_freq_) le 1 then
       int(1000/sum(max(y_sum/_freq_/2), min(y_sum/_freq_/2))) else 1000 end into
:multiplier
from tempcheckdata; quit;

data gragh_data_txt(rename=(rankyaxis=&x1 rankxaxis=&x2));
```

```
set tempcheckdata;
y_mean=y_sum/_freq_;
y_n=_freq_;
y_mean_int=int(y_mean*&multiplier);
if y_mean_int in (0, 1) then y_mean_int=1;

format y_mean &yformat;
informat y_mean &yformat;
run;

proc sort data=gragh_data_txt; by y_mean; run;

ods layout Start width=10in height=8in;
ods region x=0% y=0% width=50% height=30%;
proc sgplot data=gragh_data_txt NOAUTOLEGEND ;
XAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_n discretex discretey colormodel=&heatdist outline;
text x=&x2 y=&x1 text=y_n / textattrs=(size=5pt);
title;
footnote;
run;

ods region x=50% y=0% width=50% height=30%;
proc sgplot data=gragh_data_txt NOAUTOLEGEND ;
XAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_mean_int discretex  discretey colormodel=&heaty outline;
text x=&x2 y=&x1 text=y_mean / textattrs=(size=5pt);
title;
footnote;
run;
quit;
ods layout end;
%mend;


%macro xtabtxtnum(x1, x2);
** put character var in xaxis;
data xdata; set &datalib..&inset;
if compress(&x2)=' ' then &x2="&missingchar";
rankxaxis=&x2;
run;

proc rank data=xdata groups=&binnum out=xdata;
var &x1;
ranks rankyaxis; run;

data xdata; set xdata;
if rankyaxis=. then do; rankyaxis=0; &x1=&missingnum; end; run;

proc summary data=xdata nway;
var &y;
class rankyaxis rankxaxis/ missing order=data;
output out=tempcheckdata
       sum=y_sum;
run;

proc sql noprint;
select case when min(y_sum/_freq_) ge 0 and max(y_sum/_freq_) le 1 then
       int(1000/sum(max(y_sum/_freq_/2), min(y_sum/_freq_/2))) else 1000 end into
:multiplier
from tempcheckdata; quit;
```

```
data check_mean_cnt(rename=(rankxaxis=&x2));
set tempcheckdata;
y_mean=y_sum/_freq_;
y_n=_freq_;
y_mean_int=int(y_mean*&multiplier);
if y_mean_int in (0, 1) then y_mean_int=1;

format y_mean &yformat;
informat y_mean &yformat;
run;

proc means data=xdata median min max nway noprint;
class rankyaxis;
var &x1;
output out=check_x1(drop=_type_ _freq_)
    median=&x1 min=min_yaxis max=max_yaxis; run;


proc sql;
create table gragh_data_txtnum
as select a.*,
          b.&x1, min_yaxis, max_yaxis
from check_mean_cnt a,
     check_x1 b
where a.rankyaxis=b.rankyaxis; quit;

proc sort data=gragh_data_txtnum; by &x1 y_mean; run;

ods layout Start width=10in height=8in;
ods region x=0% y=0% width=50% height=30%;
proc sgplot data=gragh_data_txtnum NOAUTOLEGEND ;
XAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_n discretex discretey colormodel=&heatdist outline;
text x=&x2 y=&x1 text=y_n / textattrs=(size=5pt);
title;
footnote;
run;

ods region x=50% y=0% width=50% height=30%;
proc sgplot data=gragh_data_txtnum NOAUTOLEGEND ;
XAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_mean_int discretex  discretey colormodel=&heaty outline;
text x=&x2 y=&x1 text=y_mean / textattrs=(size=5pt);
title;
footnote;
run;
quit;
ods layout end;
%mend;

ods pdf file="&graphfolder\&graphname..pdf" style=myfont;
** cross tab and heat maps between categorical variables;
%macro dealtxt;
%if %sysfunc(countw(&vartxt dummymiss)) > 1 %then %do;
       %macro overlaytxt;
       %do m=1 %to &txtcnt;
         %do n=1 %to &txtcnt;
           %if &m < &n %then %do;
           %xtabtxt(&&vtxt&m, &&wtxt&n);
           %end;
```

```
      %end;
    %end;
    %mend overlaytxt;
       %overlaytxt;
%end;
%mend;
%dealtxt;


** cross tab and heat maps between categorical variables and numeric variables;
%macro dealnum;
%if %sysfunc(countw(&varnum dummymiss)) > 1 %then %do;
       %macro overlaynum;
       %do i=1 %to &numcnt;
         %do j=1 %to &numcnt;
          %if &i < &j %then %do;
            %xtabnum(&&vnum&i, &&wnum&j);
          %end;
         %end;
    %end;
    %mend overlaynum;
    %overlaynum;
%end;
%mend;
%dealnum;

** cross tab and heat maps between numeric variables;
%macro dealtxtnum;
%if %sysfunc(countw(&vartxt dummymiss)) > 1 and %sysfunc(countw(&varnum dummymiss)) >
1 %then %do;
       %macro overlaytxtnum;
       %do a=1 %to &numcnt;
         %do b=1 %to &txtcnt;
          %xtabtxtnum(&&vnum&a, &&wtxt&b);
      %end;
    %end;
    %mend overlaytxtnum;
       %overlaytxtnum;
%end;
%mend;
%dealtxtnum;
ods pdf close;
```