Paper 1732 - 2014

Automatic Statistical Association Analyses Using SAS Macro Programs

Justin Jia, Canadian Imperial Bank of Commerce (CIBC), Toronto, ON, Canada

Amanda Lin, Bell Canada, Toronto, ON, Canada

ABSTRACT

In the past, we often needed to perform huge numbers of repetitive and data-driven statistical association analyses to evaluate the performance of business events in terms of the event outcome or target response. These routine tasks were usually carried out manually by using Microsoft Excel, which was tedious, time-consuming and error-prone in nature.

In order to improve the work efficiency and analysis accuracy, we managed to automate the post-event analysis process by SAS® programming to replace the manual work with Excel. Through the use of SAS MACRO and other advanced skills, we successfully automated the complicated data-driven analyses with high efficiency and accuracy.

This paper will present and illustrate the creative analytical ideas and programming skills for developing the automatic analysis process, which can be extended to apply in a wide variety of fields such as clinical trial studies, scientific research, engineering process control, marketing research, risk management and business analytics.

INTRODUCTION

Statistical inference is the process of drawing conclusions from a random sample to a general population of interest. Using this technique, we can probe and understand the properties and characteristics of the population under study (which are usually not measurable due to large population size or other reasons) via some form of random sampling. Based on the probability distribution of data, we can then perform a variety of inferential analyses: use the statistics of the random sample to estimate the statistics of the population (mean, median, variance etc.), compare the difference of a given analysis variable in different populations, and test the correlation and association between different random variables. This inferential process is often referred as statistical hypothesis testing, which find important applications in a wide variety of fields such as scientific research, engineering process control, health study and clinical trials, marketing research etc. Hypothesis testing can be carried out on both continuous and categorical variables. For example, in clinical trial studies, Pearson's Chi-Square test is often used to test the effectiveness of new drugs. In this research, double blind experiments are conducted parallel on both a test group of patients who are treated with an active medicine and a control group of patients treated with a placebo. Then Pearson's Chi-Square test or Fisher's exact test is applied to investigate if an association exists between the treatment variable and the outcome variable. This methodology is also called contingency table analysis which is aimed to determine the association between the row and column variables. In scientific research and engineering process control, Student's t-test, ANOVA and non-parametric tests are employed to test the means or medians of a continuous analysis variable for different populations. In marketing research and risk management, statistical hypothesis testing is often used to assess the performance and effectiveness of implemented marketing strategies or risk management strategies. Student's t-test and ANOVA are used for testing on continuous response variables (the monetary values of sales, profits, turnovers, revenues etc.; the numeric values of customer response rates, deactivation rates and risk scores etc.), while Chi-Square test is for categorical response variables (customer response, churn, attrition, deactivation, default etc.).

In the past, we had plentiful routine tasks to conduct the post-event valuations to assess the productivity of marketing activities or the performance of implemented collections. For this purpose, we needed to conduct huge numbers of contingency table analyses by Chi-Square testing to test the statistical association between the treatment and outcome variables. Usually, we did it by manual testing with Excel, which was a repetitive, tedious and time-consuming work with low efficiency. Moreover, it was error-prone in nature due to the manual copy/paste and testing procedures. To overcome these drawbacks, we managed to utilize SAS programming to automate the analysis process and to replace the manual testing with Excel. Through the use of SAS macro and other advanced programming skills, we successfully automated the complicated statistical association analyses with amazing efficiency and accuracy: the tedious and time-consuming manual work, which usually cost an experienced analyst more than one week to do, could be completed within several minutes by the automatic SAS programs.

This paper will present and discuss the automatic analysis process using advanced SAS programming. The creative analytical ideas and programming skills can be used in a wide variety of fields such as clinical trial studies, scientific research, engineering process control, marketing research, risk management and business analytics.

THE REQUESTS OF STATISTICAL ASSOCIATION ANALYSES

In this work, the raw data of payment collection events were gathered and stored in numerous Excel files (in XLS or CSV format). Table 1 demonstrates the attributes of raw data in one example Excel file.

Table 1. Attributes of the raw collections data (File: HGB Collections 2012 Q4.csv)

Channel	Group	Contacted	Collected
H	Strategy A	37,213	75
K	Control	4,296	26
K	Strategy A	5,789	59
L	Control	4,003	19
L	Strategy B	38,654	271
M	Strategy A	4,831	85
M	Strategy B	5,210	32
N	Control	6,856	63
N	Strategy A	39,750	487
N	Strategy B	23,334	171
W	Strategy B	42,036	82

Channel: character variable, $10, the code of a collection channel.

Group: character variable, $30, the target customer group in a collection activity. “Strategy A” and “Strategy B”, collection targets were selected by different collection strategies. “Control”, targets were randomly selected from a large target database by simple random sampling, serving as a parallel comparison group.

Contacted: numeric, N8, the total number of targets contacted.

Collected: numeric, N8, the number of targets whose owed payments were successfully collected.

Analysis Requests:

For each collection channel, we needed to perform Chi-Square testing on the different target groups to determine if there exists an association between the implemented collection strategy and the collection outcome. Ideally, each channel was supposed to contain data for the 3 different target groups (Control, Strategy A, Strategy B). However, the real data could be incomplete and messy because of either the incomplete implementation of collection strategies or other issues during data collection. As a result, some channels may have only one or two target groups. For example, as shown in Table 1, Channels H and W have only one target group in each. They are invalid data for Chi-Square testing and we need ignore them in our analysis process. Channels K/L/M have two customer groups in each one. These data issues remarkably complicated the analysis process and rendered it to be data-driven in nature.

Under these circumstances, our analysis requests were:

“ If a channel has only one single target group, the data is invalid and useless in post-event analyses. We therefore ignore it and give it an ‘**Invalid Data**’ tag.

“ If a channel contains two target groups, we need do a 2×2 contingency table analysis by Chi-Square testing to determine if the row and column variables are associated with each other at the α=0.05 significance level. If an association exists between them, we need look into the relative risk and odds ratio of the Chi-Square testing, which serve as a measure of association to evaluate the difference between customer groups. Please note these two statistics are only applicable to the 2×2 contingency table analysis.

If a channel contains three customer groups, firstly we need to do a 3×2 overall contingency table analysis to investigate the association between the row and column variables. If they are associated with each other, we need further tests to find out which collection strategy produces the best performance. Because relative risk or odds ratio is a measure of association only applicable to 2×2 contingency table analyses, therefore we need to conduct three separate 2×2 Chi-Square subtests (i.e., Strategy A vs. Control, Strategy B vs. Control, Strategy A vs. Strategy B) to determine the best performer. Please note that the 3 ×2 overall analysis is necessary because it takes into account of interactions among target groups, therefore it can NOT be skipped and replaced by the three separate 2×2 subtests”.

Usually, we did these complicated contingency table analyses manually by using Excel, which was of low efficiency and error-prone because of the large amounts of Excel files and raw data. Therefore we strived to automate the analysis process by SAS programming.

To meet above analysis requirements, we have below challenges in the automation of the analysis process by SAS programming:

1. Payment collection channels contain varying target groups, we must instruct SAS program to differentiate and identify them correctly and accurately;
2. The analysis process is data-driven in nature. Different analyses are required depending on the structure of each channel (single, two, or three target groups?);
3. The final analytic reports must contain not only the analysis results, but also essential descriptive information, such as the test number and test date, the name of the source table and the collection channel, the Chi-Square test description etc. Without these identification and documentation information, the analysis results will be confusing and useless.

Hereby we present and illustrate the creative analytical ideas and programming skills for developing the automatic analysis process.

PART 1: READ IN RAW DATA FROM EXTERNAL EXCEL FILES

The initial step is to read in the raw data stored in numerous Excel files. Please note that some Excel files may use illegal symbols as file names, such as blank or dash (-). We need remove or convert them so that they meet the requirements of SAS naming conventions. Due to the large numbers of raw Excel data files, we must utilize SAS Macro to automatically read them into SAS data sets.

```
************* Part 1: Read in external Excel files from storage library***********;
options mprint symbolgen source2 missing= ' ' NOFMTERR;
libname A 'physical path/Data';
filename Raw 'physical path/Input';

******************** Method 1: Use directory functions. **********************;
data AAA (drop=RC);
length Memname In_Name Out_Name $60;

Did=dopen("Raw");
if did> 0 then do;
Num=dnum(did);
do J=1 to Num;
Memname=dread(did, J);
In_Name=scan(memname, 1, '.');
Out_Name=tranwrd(compress(In_Name),'-', '_');
output;
end;
end;
else do;
Msg=sysmsg();
end;

RC=dclose(did);
put 'Directory Opening Indicator Msg=' msg
    'Directory Closing Indicator RC='  RC;
run;
*************************** Method 2: Use SAS PIPE. ****************************;
filename Raw pipe 'ls  -l  /physical path/Input/';

data DirList;
infile Raw truncover;
input      mode $ 1-10  nlinks 12-14 user $ 16-23    group $25-32
           size 34-40   lastmod $ 42-53     MemName $ 54-253;
run;

data BBB;
length In_Name Out_Name $60;
set DirList(keep=MemName);
if _N_=1 then delete;
In_Name=scan(MemName, 1, '.');
Out_Name=tranwrd(compress(In_Name),'-', '_');
run;
```

```
************ Use Macro to read in the Excel files into SAS data sets. ***********;
%macro Read;
proc sql;
select count(*) into : N
from AAA;
%let N=&N;

select Memname, Out_Name into : Mem1-:Mem&N,  :OUT1- :OUT&N
from AAA;
quit;

%local I;
%do I=1 %to &N;
filename Mem&I "/physical path/&&Mem&I. ";
proc import  datafile=Mem&I  out= A.&&OUT&I  DBMS=CSV  REPLACE;
GETNAMES= yes;
Guessingrows=2000;
run;
data A.&&OUT&I;
length Source_Table $60;
set  A.&&OUT&I;
Source_Table="&&Mem&I.";
format group;
run;

proc append base=A.Raw_All data=A.&&OUT&I force;
run;

%end;
%mend;
%Read;
```

As illustrated above, we have two solutions to read in the raw data from external Excel files. The first method is to use SAS directory functions[1]. The DOPEN function opens the specified Raw directory and returns a directory identifier value, which is 0 if the directory can not be opened, otherwise the value will be greater than 0. The DNUM function can identify the number of members in the directory, and this value is passed to the DREAD function as the highest possible member number. A do-loop is utilized with the DREAD function to return the names of the Excel files sitting in the directory. The DREAD function will return a blank value if an error occurs during the process. Because Excel file names may contain illegal symbols or characters, we then use the SCAN function to retrieve the file names without suffix (.xls or .csv), the COMPRESS function to remove blanks and the TRANWRD function to replace the dash "-" symbol with underscore "_" symbol . In case that the directory can NOT be opened, SYSMSG function will return an error code or warning message text from processing the last dataset or external file function. Finally, the DCLOSE function is employed to close the directory opened by the DOPEN function, it will return a zero value if the operation is successful, otherwise it will return a non-zero value for unsuccessful closing. When a data step ends, all directories or members opened within a data step are closed automatically. Below is the printout of the created AAA dataset.

Table 2. Printout of the generated AAA data set.

MemName	In_Name	Out_Name	DID	Num	J	Msg	RC
HGB-Collections 2012 Q4.csv	HGB-Collections 2012 Q4	HGB_Collections2012Q4	1	3	1		
STC-Collections-2011.csv	STC-Collections-2011	STC_Collections_2011	1	3	2		
New AHV Collection 2012.csv	New AHV Collection 2012	NewAHVCollection2012	1	3	3		

An alternative way is through the use of the piping functionality of SAS. Piping is a channel of parallel communications between SAS and other applications[2]. For example, piping enables our SAS application to receive input from any Windows or UNIX command that writes to standard output and to route output to any UNIX command that reads from standard input. Therefore, we can utilize pipes to find the number of files in a specified directory: use the DIR line command for the Windows operating system, and the LS command for the UNIX system.

As shown in Method 2, we use the PIPE device type on the FILENAME statement to invoke the UNIX LS command, which creates a directory listing of all files in directories and subdirectories. The output of the LS command is sent to

the following DATA step through the RAW file reference. The DATA step then creates a data set named DirList, below is its print out.

Table 3. Printout of the created DirList data set.

Mode	Nlinks	User	Group	Size	Lastmod	MemName
total 6						
-rw-r--r--	1	justin	users	223	Oct 15 10:15	HGB-Collections 2012 Q4.csv
-rw-r--r--	1	justin	users	223	Oct 15 10:15	STC-Collections-2011.csv
-rw-r--r--	1	justin	users	265	Oct 15 10:15	New AHV Collection 2012.csv

The MemName variable have the names of the raw Excel files. We then use the following DATA step to remove or convert the illegal symbols in file names, which produces the BBB data set with the same MemName, In_Name and Out_Name variables as in the AAA data set.

Then the SAS Macro **%Read** is developed to read the external Excel files into SAS datasets. In this Macro, Proc SQL is applied to create macro variables for the following macro Do-Loop: the macro variable N counts the total number of observations in the directory listing AAA (or BBB) data set (namely the total number of raw Excel files); macro variables Mem1-MemN and Out1-OutN store the external Excel file names and corresponding output dataset names, respectively. Then a macro do-loop and the Proc Import procedure are utilized to read in the raw Excel files one by one, followed by the appending procedure to yield the final all-in-one A.Raw_All data set. Table 1 shows the partial contents of the A.Raw_All data set.

PART 2: PREPARE DATA FOR STATISTICAL ASSOCIATION ANALYSES

Once the raw Excel data have been read into SAS, we can start to perform statistical association analyses by Chi-Square testing. According to the analysis requests described above, the analysis process is data-driven in nature dependent on the structure of a specific channel. For this purpose, we need instruct SAS program to distinguish and identify the structure of each channel accurately and then perform the required analyses accordingly. It is a challenging task that requires both analytical thinking and creative solutions.

We have managed to achieve this goal by innovative SAS programming. Although the analysis process illustrated here is based on a simple three-group example, however, the analytical ideas and programming logic can be easily extended to more complicated cases as well.

```
********************* Part 2: Prepare data for Chi-Square testing.************;
proc sort data= A.Raw_All;   by Source_Table Channel;    run;

data Encoding;
set  A.Raw_All;
by Source_Table Channel;
if upcase(group)='CONTROL'           then  Grp_Code=0;
else if upcase(group)='STRATEGY A'   then  Grp_Code=1;
else if upcase(group)='STRATEGY B'   then  Grp_Code=2;
run;

proc sql;
create table Tagging as
select *, count(*) as Member, Sum(Grp_Code) as Sum_Code
from Encoding
group by Source_Table, Channel;
quit;


proc format fmtlib;
value category
0='1-Group Channel: Invalid Data !'
1='2-Group Channel: Control and Strategy A'
2='2-Group Channel: Control and Strategy B'
3='2-Group Channel: Strategy A and Strategy B.'
5='3-Group Channel: Control, Strategy A and Strategy B.';
```

```
value outcome
1='1. Yes'
2='2. No';

value grp_code
0='Control'
1=' Strategy A'
2=' Strategy B';
run;

data Start  G1_Invalid(keep= Source_Table Channel Contacted Category);
set  Tagging;
Cat= Member*Sum_Code;
Category=round( Member*Sum_Code/2);
if Member=1 then do;
Category=0;
output G1_Invalid;
end;
else output Start;
run;


proc sort data= Start(drop= Member Cat Sum_code) Out= Start_Sorted;
by Source_Table Channel Category;
run;

data Transform(drop=contacted collected I Yes No) ;
set  Start_Sorted ;
by Source_Table Channel Category;
Yes = Collected;
No=   Contacted - Collected;

array CNT(2) Yes No;
do I= 1 to 2;
Count=CNT(I);
Outcome= I;
output;
end;
format outcome outcome.  ;
run;
```

As shown above, we first sort the A.Raw_All data set by Source_Table and Channel, then create a new variable Grp_Code to encode the different target groups in the following DATA step. The variable Grp_Code has values of 0, 1, 2 for the Control, Strategy A and Strategy B customer groups respectively. The values need to be numeric rather than character since we will do calculations on them.

Then Proc SQL is employed to summarize the detail information of each channel. Please note remerging will occur here: the summary statistics will remerge back with the original detail data. However, this is what we intend to pursue rather than to avoid because the result helps to identify the different channel structures.

As shown in Table 4, the Member variable can determine that how many target groups exist in each channel, and the sum of Grp_Code can distinguish the different combinations of the 2-group channels (K, L, M). However, the Sum_Code cannot discriminate the Channel M from the 3-group Channel N since the values are 3 for both of them. Therefore the encoding variables Member and Sum_Code both do half of the whole job. To reach our goal, we multiply Member by Sum_Code to create a new variable Cat (Cat= Member*Sum_Code), this product variable can successfully differentiate all the combinations (Please refer to Table 4). We then divide it by 2 and round it to the simplest numbers (1, 2, 3, 5 respectively) and use a custom format to display the various channel structures. Furthermore, we split the raw data into two separate data sets: G1_Invalid is to store the invalid observations with a single customer group (the value of Category is set to 0), on which we perform no analyses. The Start data set is for the valid observations, below table demonstrates its contents.

Table 4. Partial printout of the generated Start dataset.

Group	Source_Table	Channel	Contacted	Collected	Grp_Code	Member	Sum_Code	Cat	Category
Control	HGB-Collections 2012 Q4.csv	K	4,296	26	0	2	1	2	1
Strategy A	HGB-Collections 2012 Q4.csv	K	5,789	59	1	2	1	2	1
Control	HGB-Collections 2012 Q4.csv	L	4,003	19	0	2	2	4	2
Strategy B	HGB-Collections 2012 Q4.csv	L	38,654	271	2	2	2	4	2
Strategy A	HGB-Collections 2012 Q4.csv	M	4,831	85	1	2	3	6	3
Strategy B	HGB-Collections 2012 Q4.csv	M	5,210	32	2	2	3	6	3
Control	HGB-Collections 2012 Q4.csv	N	6,856	63	0	3	3	9	5
Strategy A	HGB-Collections 2012 Q4.csv	N	39,570	487	1	3	3	9	5
Strategy B	HGB-Collections 2012 Q4.csv	N	23,334	171	2	3	3	9	5

Category: 1='2-Group Channel: Control and Strategy A'

2='2-Group Channel: Control and Strategy B'

3='2-Group Channel: Strategy A and Strategy B.'

5='3-Group Channel: Control, Strategy A and Strategy B.' ;

When we use Proc Freq to do Chi-square testing, this procedure requires that the input data be in a defined structure. Therefore we need transform the above data set to meet this requirement. We firstly sort the data by Source_Table, Channel and Category, then utilize a SAS Array and a Do-Loop to rotate it into the below data structure, which is appropriate for Chi-Square testing.

Table 5. Partial printout of the generated Transform data set (Channel N only) .

Group	Source_Table	Channel	Grp_Code	Category	Count	Outcome
Control	HGB-Collections 2012 Q4.csv	N	0	5	63	1. Yes
Control	HGB-Collections 2012 Q4.csv	N	0	5	6,793	2. No
Strategy A	HGB-Collections 2012 Q4.csv	N	1	5	487	1. Yes
Strategy A	HGB-Collections 2012 Q4.csv	N	1	5	39,083	2. No
Strategy B	HGB-Collections 2012 Q4.csv	N	2	5	171	1. Yes
Strategy B	HGB-Collections 2012 Q4.csv	N	2	5	23,163	2. No

PART 3: AUTOMATIC STATISTICAL ASSOCIATION ANALYSES

After the data preparation in Part 2, here comes the core part of the automatic analysis process. Below Macro programs are developed for the data-driven contingency table analyses.

```
***************** Part 3: Automatic Statistical Association Analyses.************;
%macro Q_Test;

****Split Transform dataset into 2- and 3-group datasets for different actions.***;
data G2  G3;
set transform;
if Category= 5 then output G3;
else output G2;
run;

********************** 2*2 Chi-Square Tests for 2-Group Channels.*****************;
proc freq data= G2  order=formatted;
format Grp_Code grp_code. Outcome outcome.;
by Source_Table Channel Category;
weight Count/ zeros;
tables Grp_Code*Outcome/chisq measures;
output out= G2_Test(keep= Source_Table Channel Category N P_PCHI _RRC1_ _RROR_
rename=(P_PCHI= p_Value _RRC1_=Relative_Risk _RROR_= OddsRatio))
       chisq measures;
run;
```

```
************* Overall 3*2 Chi-Square Tests for 3-Group Channels. ****************;
proc freq data= G3  order=formatted;
format Grp_Code grp_code. Outcome outcome.;
by Source_Table Channel Category;
weight Count/ zeros;
tables Grp_Code*Outcome/chisq measures;
output out=G3_Overall(keep= Source_Table Channel Category N P_PCHI  rename=(P_PCHI=
p_Value))  chisq ;
run;

proc sql;
create table For_Sub_Test as
select  a.*
from   G3   a, G3_Overall   b
where  a.Source_Table=b.Source_Table        and
       a.Channel=b.Channel                  and
       a.Category=b.Category                and
       b.p_Value <0.05 ;
quit;

***************************** 2*2 Chi-Square subtests. *************************;
%macro G3_Sub_Test;
proc sql;
select distinct Grp_Code into : GRP_Code_1 -: GRP_Code_3
from For_Sub_Test;
quit;
%put _user_;

%local I  J;
%do I=1 %to 3;
%do J=&I+1 %to 3;

proc freq data=  For_Sub_Test  order=formatted;
where Grp_Code  in ( &&GRP_Code_&I , &&GRP_Code_&J );
format Grp_Code grp_code. Outcome outcome.;
by Source_Table  Channel  Category;
weight Count/ zeros;
tables  Grp_Code*Outcome/chisq measures;
output out=G3_Subtest(keep= Source_Table Channel Category N P_PCHI _RRC1_ _RROR_
rename=(P_PCHI= p_value _RRC1_=Relative_Risk _RROR_= OddsRatio))  chisq measures;
run;

data G3_Subtest;
set  G3_Subtest;
Sub_Category= &&GRP_Code_&I  + &&GRP_Code_&J;
run;

proc append base=G3_SubTest_All   data= G3_Subtest  force;  run;

%end;
%end;
%mend;
%G3_Sub_Test;

proc format fmtlib;
value test
1='2*2 Test:   Strategy A vs. Control.'
2='2*2 Test:   Strategy B vs. Control.'
3='2*2 Test:  Strategy A vs. Strategy B.';

value subtest
1='2*2 Subtest:   Strategy A vs. Control.'
2='2*2 Subtest:   Strategy B vs. Control.'
```

```
3='2*2 Subtest:  Strategy A vs. Strategy B.';
run;

data Final;
length Test_Label  $60;
set G1_Invalid(in=A  rename=(Contacted=N) )  G2_Test(in=B)
    G3_Overall(in=C)                         G3_SubTest_All(in=D);

if A=1 then Test_Label=' Invalid Data';
if B=1 then Test_Label=put(Category, test.);
if C=1 then Test_Label=' 3*2 Overall Test:  Control, Strategy A, Strategy B.';
if D=1 then Test_Label=put(Sub_Category, subtest.);

drop Sub_Category ;
format Category Category. p_value 6.3 relative_risk oddsratio 6.2 ;
run;

%mend;
%Q_Test;
```

As illustrated above, we firstly split the Transform data set into two separate datasets: G2 for 2-Group channels and G3 for 3-Group channels. For 2-group channels, the analysis is simple and straightforward. The Proc Freq procedure is applied directly to perform 2×2 Chi-Square tests (by Source_Table, Channel and Category). The custom formats (GRP_CODE. and OUTCOME.) are applied to ensure the appropriate creation and analysis of contingency tables. The CHISQ and MEASURES options on the TABLES statement will instruct SAS to do Chi-square testing and output the relative risk and odds ratio. The testing results are then output to the G2_Test data set by the OUTPUT statement. Table 6 shows the structure and contents of the G2_Test data set.

Table 6. Partial printout of the created G2_Test data set.

Source_Table	Channel	Category	N	p_value	Relative_Risk	OddsRatio
HGB-Collections 2012 Q4.csv	K	1	10,085	0.025	1.68	1.69
HGB-Collections 2012 Q4.csv	L	2	42,657	0.097	1.48	1.48
HGB-Collections 2012 Q4.csv	M	3	10,041	0	2.87	2.90

For 3-group channels, the analysis process is much more complicated as per the analysis requests. At first, we apply Proc Freq directly on the G3 data set to conduct the overall 3×2 tests and the needed statistics are output to the G3_Overall dataset. In this case, it does NOT contain the Relative_Risk and OddsRatio variables because they are measures of associations for the 2×2 Chi-Square tests only. We then join the G3_Overall data set with the original G3 data set on the condition of p-value < 0.05, which eventually selects all the statistically significant 3-group channels and produces the For_Sub_Test data set for the following 2×2 subtests.

The separate 2×2 subtests are accomplished by the **%G3_Sub_Test** sub-macro program. In this program, Proc SQL is used to create three macro variables Grp_Code_1 to Grp_Code_3, which store the values of Grp_Code (0, 1, 2) for different customer groups. Then a macro Do-Loop is utilized to do the three separate 2×2 subtests via the subsetting WHERE statement. The analysis results including relative risk and odds ratio are output to the G3_Sub_Test data set. In the following DATA step, a numeric Sub_Category variable is created to flag the corresponding subtest. All the subtest results are appended together by the Proc Append procedure. Below tables show the structure and contents of G3_Overall and G3_Subtest_All data sets.

Table 7. Partial printout of the generated G3_Overall data set.

Source_Table	Channel	Category	N	p_Value
STC-Collections-2011.csv	Q	5	96,114	0
New AHV Collection 2012.csv	S	5	84,428	0
HGB-Collections 2012 Q4.csv	N	5	69,760	0

Table 8. Partial printout of the generated G3_SubTest_All data set (Channel N only).

Source_Table	Channel	Category	N	p_Value	Relative_Risk	OddsRatio	Sub_Category
HGB-Collections 2012 Q4.csv	N	5	46,426	0.03	1.34	1.34	1
HGB-Collections 2012 Q4.csv	N	5	30,190	0.12	0.80	0.80	2
HGB-Collections 2012 Q4.csv	N	5	62,904	0.00	1.68	1.69	3

Sub_Category: 1='2*2 Subtest: Strategy A vs. Control.'

2='2*2 Subtest: Strategy B vs. Control.'

3='2*2 Subtest: Strategy A vs. Strategy B.';

The last step is to stack all the 4 data sets together by using the DATA step. A new variable Test_Label is created to describe the conducted tests through the use of the IN contributor and created custom formats. Table 9 shows the structure and contents of the created Final data set.

Table 9. Partial printout of the generated Final dataset.

Source_Table	Channel	N	Category	p_Value	Relative_Risk	OddsRatio	Test_Label
HGB-Collections 2012 Q4.csv	H	37,213	0				Invalid Data
HGB-Collections 2012 Q4.csv	W	42,036	0				Invalid Data
HGB-Collections 2012 Q4.csv	K	10,085	1	0.025	1.68	1.69	2*2 Test: Strategy A vs. Control.
HGB-Collections 2012 Q4.csv	L	42,657	2	0.097	1.48	1.48	2*2 Test: Strategy B vs. Control.
HGB-Collections 2012 Q4.csv	M	10,041	3	0.000	2.86	2.90	2*2 Test: Strategy A vs. Strategy B.
HGB-Collections 2012 Q4.csv	N	69,760	5	0.000			3*2 Overall Test: Control, Strategy A, Strategy B
HGB-Collections 2012 Q4.csv	N	46,426	5	0.028	1.34	1.34	2*2 Subtest: Strategy A vs. Control.
HGB-Collections 2012 Q4.csv	N	30,190	5	0.122	0.80	0.80	2*2 Subtest: Strategy B vs. Control.
HGB-Collections 2012 Q4.csv	N	62,904	5	0.000	1.68	1.69	2*2 Subtest: Strategy A vs. Strategy B.

Category: 0='1-Group Channel: Invalid Data!'

1='2-Group Channel: Control and Strategy A'

2='2-Group Channel: Control and Strategy B'

3='2-Group Channel: Strategy A and Strategy B.'

5='3-Group Channel: Control, Strategy A and Strategy B.' ;

PART 4: ODS OUTPUT TO CREATE ANALYTIC REPORTS IN EXCEL FORMAT

After performing all the required association analyses, we need output the testing results to Excel to create analytic reports for clients. To improve the report layout and format, we apply SAS traffic-lighting techniques to highlight and enhance the reports.

```
********************* ODS Output to Create Analytic Reports. ******************;
proc format fmtlib;
value Conclusion
.='   '
low-<0.05='Association exists between analysis variables.'
other='No association  exists between analysis variables.';

value Fore_Color
low-<0.05='Red'
other='Black';

value Back_Color
0-<0.05='Skyblue'
other='White';
run;

proc sort data= Final;  by Source_Table Channel Test_Label;     run;

data A.Report;
length Test_No 3;
set Final;
by Source_Table Channel Test_Label;
Conclusion=p_value;
Test_No =_N_;
Test_Date= "&sysdate9. ";
format p_value 6.3  Conclusion conclusion. ;
run;
```

```
filename Report '/physical path/Post_Collection_Analyses_Report_&YYMM..xls';
ODS listing close;
ODS MSOFFICE2K file=report style=journal;
Title "Analytic Reports of Post-Collection Analyses by Chi-Square Testing.
Significance level: alpha =0.05.";

proc print data= A.Report noobs  label;
var  Test_No Test_Date Source_Table Channel Category N  Test_Label;
var  p_Value/style=[foreground=Fore_Color.  background=Back_Color.];
var  Relative_Risk OddsRatio ;
var  Conclusion/style=[foreground=Fore_Color.  background=Back_Color.];

label  Test_No='Test No.'                    Test_Date='Test Date'
       Source_Table='Source Table'           Channel='Collection Channel'
       Category='Channel Description'        N='Total Number of Subjects'
       Test_Label='Test Description'         p_Value='p-Value for Chi-Square Test'
       Relative_Risk='Relative Risk'         OddsRatio='Odds Ratio'
       Conclusion='Test Conclusion' ;
run;

ODS _All_ close;
ODS listing;
```

As illustrated above, we create the custom formats to tag and describe the Conclusion column based on its cell value (which is p-value actually). If the cell value is smaller than 0.05, we reject the H_0 null hypothesis of "No association exists between the analysis variables", and accept the H_1 alternative hypothesis of "Association exists between the analysis variables". Otherwise we accept the null hypothesis. Furthermore, we highlight the p-value and conclusion text with different fonts and fore/back-ground colors to draw attention via the traffic-lighting technique.

To generate the final analytic report, we first use Proc Sort to sort and arrange the test results in the sequence of Source_Table and Channel, then employ the DATA step to create three new variables: Conclusion, Test_No and Test_Date for documentation purposes. The ODS MSOFFICE2K and Proc Print procedures are utilized for report-writing, and the traffic-lighting of the p_Value and Conclusion columns is achieved by applying the custom formats to the Style= option. Table 10 demonstrates part of the analytic report.

Table 10. Partial Printout of the Generated Analytic Report in Excel Format.

Analytic Report of Post-Collection Association Analyses by Chi-Square Testing. Significance Level: alpha = 0.05.

Test No.	Test Date	Source Table	Collection Channel	Channel Description	Total Number of Subjects	Test Description	p-Value for Chi-Square Test	Relative Risk	Odds Ratio	Test Conclusion
15	17-Oct-13	HGB-Collections 2012 Q4.csv	H	1-Group Channel: Invalid Data !	37,213	Invalid Data				
16	17-Oct-13	HGB-Collections 2012 Q4.csv	K	2-Group Channel: Control and Strategy A	10,085	2*2 Test: Strategy A vs. Control.	0.025	1.68	1.69	Association exists between analysis variables.
17	17-Oct-13	HGB-Collections 2012 Q4.csv	L	2-Group Channel: Control and Strategy B	42,657	2*2 Test: Strategy B vs. Control.	0.097	1.48	1.48	No association exists between analysis variables.
18	17-Oct-13	HGB-Collections 2012 Q4.csv	M	2-Group Channel: Strategy A and Strategy B.	10,041	2*2 Test: Strategy A vs. Strategy B.	0.000	2.86	2.9	Association exists between analysis variables.
19	17-Oct-13	HGB-Collections 2012 Q4.csv	N	3-Group Channel: Control, Strategy A and Strategy B.	69,760	3*2 Overall Test: Control, Strategy A, Strategy B.	0.000			Association exists between analysis variables.
20	17-Oct-13	HGB-Collections 2012 Q4.csv	N	3-Group Channel: Control, Strategy A and Strategy B.	46,426	2*2 Subtest: Strategy A vs. Control.	0.028	1.34	1.34	Association exists between analysis variables.
21	17-Oct-13	HGB-Collections 2012 Q4.csv	N	3-Group Channel: Control, Strategy A and Strategy B.	30,190	2*2 Subtest: Strategy B vs. Control.	0.122	0.8	0.8	No association exists between analysis variables.
22	17-Oct-13	HGB-Collections 2012 Q4.csv	N	3-Group Channel: Control, Strategy A and Strategy B.	62,904	2*2 Subtest: Strategy A vs. Strategy B	0.000	1.68	1.69	Association exists between analysis variables.
23	17-Oct-13	HGB-Collections 2012 Q4.csv	W	1-Group Channel: Invalid Data !	42,036	Invalid Data				

As shown in Table 10, the analytic report created by the automatic analysis program provides valuable and insightful information for assessing the performance of collection activities. For example, it reveals that the success of collections is significantly associated with the collection strategy for Channel K at α= 0.05 significance level (p-value= 0.025), and the collection target group selected by the Strategy A is 68% more likely to pay back the owed payments than the randomly selected Control group (relative risk=1.68). Similar results are observed for Channel M, and the

customer pay-back rate in the Strategy A group is almost 2 times higher than that in the Strategy B group (relative risk=2.86) . However, no association is identified between the collection strategy and the collection outcome for Channel L at α= 0.05 significance level (p-value= 0.097). As for the 3-group channel N, the overall 3×2 test indicates the two analysis variables are associated with each other, and the three 2×2 subtests suggest that the Strategy A is the best player among them, which performs 68% better than the Strategy B (relative risk=1.68). Therefore, the developed SAS programs offer an efficient and accurate solution to analyze the huge numbers of business data and evaluate the performance of collection events.

CONCLUSION

SAS Macro is a very important and useful programming technique and it is often utilized to execute repetitive and data-driven operations. As presented in this paper, we have successfully utilized this technique to automate the data-driven statistical association analyses to replace the manual testing with Excel. The developed SAS programs provide a fast and accurate solution to assess the performance of collection events or marketing activities in terms of the collection outcome or target response. The illustrated analytical ideas and programming skills can be easily extended to more complicated cases and find prominent applications in a wide variety of fields such as clinical trial studies, scientific research, engineering process control, marketing research, risk management and business analytics.

REFERENCES

[1] SAS Support Website,
http://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a000209687.htm

[2] SAS Support Website
http://support.sas.com/rnd/scalability/connect/piping.html

ACKNOWLEDGEMENTS

Special thanks to Dr. Arthur Tabachneck for his valuable feedback and suggestions during reviewing this paper. We are very grateful to him for the great help and support. Thank you so much.

CONTACT INFORMATION

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

Justin Jia
Customer Marketing and Strategies, CIBC Canada
Email: justin.jia@cibc.com

Amanda Lin
Risk Management, Bell Canada
Email: amanda_shan_shan.lin@bell.ca