Paper 088-2011

Taking the Proc Summary a Step Beyond

Priya Suresh, RTI International, RTP, NC

ABSTRACT

Proc Summary can be used to display cross-tabs of multiple variables to quickly identify unique patterns in the data. The summary output shows the unique pattern of input variables and the resultant value of the derived variable in an easily verifiable format. If the same code is applied to data over multiple time periods, the same patterns are repeated again and again. The next step beyond Proc Summary is to use exact pattern matching technique to weed out the verified patterns from prior time periods and display only the new patterns that occur in the data. This presentation describes the application of this technique and explains how SAS® is used to provide an elegant solution.

INTRODUCTION

The project I am on collects data throughout the year. The raw datasets are built annually. Multiple raw variables from these datasets are used to derive key analytical variables. We use Proc Summary to review and verify that the variables have been derived correctly. The summary will display selected raw variables in conjunction with the derived variable. We review the output from Proc Summary and check the patterns to make sure that all of the variations have been covered, and that the derived values are expected. If a new pattern occurs, it could mean a scenario has occurred that has not happened before. In that case, we do further investigation and decide either to correct the derivation (edit) program or add the new pattern to the list of approved ones.

REPITITION IS INEFFICIENT

The same derivation logic is applied year after year. Each annual dataset holds tens of thousands of records. Each summary output can be anywhere from a single page to 100 pages depending on the derived variable and the raw variables involved. The number of derived variables is also very high. You can imagine the magnitude of the review task. Since anomalies do occur in the data, it is imperative that we review the summaries for each dataset.

Reviewing and re-approving the patterns that existed before is literally a waste of time. So, the obvious next step is to weed out the patterns that have been approved in prior years. How does one go about weeding out the repeating patterns? In other words, how do we take the Proc Summary a step beyond?

THE BIG PICTURE

The technique I developed is as follows:

a) Write out the Proc Summary patterns for a specific set of variables from the current dataset
b) Compare that to previously reviewed and approved patterns for the same set of variables
c) Keep only the new patterns from the current dataset
d) Review the few new patterns and you're done!

PSEUDO CODE (OR IS IT INSTRUCTIONS FOR FORM 1040?)

a) Write out Proc Summary output of the current dataset into a temp SAS dataset.

```
PROC SUMMARY DATA=&Indsn  NWAY MISSING;
CLASS Var1 Var2 ...;
OUTPUT OUT= temp;
```

b) Split the temp SAS dataset into two SAS datasets (droping the _type_ variable):

- one that contains just the _freq_ (renamed actual_Freq_), let's call the dataset LclFreq, and
- the other without the _freq_ but just the data (pattern), let's call this one LclData.

```
data lclData (drop=actual_freq_)
     lclFreq (keep=actual_freq_)
     ;
       set temp (drop=_TYPE_  rename=(_freq_=actual_freq_));
run
```

c) Proc Export LclData (data pattern) dataset into a comma separated file (csv), current.csv

```
proc export data=lclData
   outfile='current.csv'
   dbms=dlm
   replace;
   delimiter=',';
run;
```

The first two records of 'current.csv' generated by Proc Export for the data example shown later in this paper is as follows:

```
ANLEVER,DARVTYLC,PERCTYLX,VICOLOR,ANLCARD,CODEINE,DEMEROL,DILAUD,FIORICET,FI
ORINAL,HYDROCOD,METHDON,MORPHINE,OXYCONTN,PHENCOD,PROPOXY,SK65A,STADOL,TALAC
EN,TALWIN,TALWINNX,TRAMADOL,ULTRAM,ANLNOLST,PR41CNT,ANALNEWA,ANALNEWB,ANALNE
WC,ANALNEWD,ANALNEWE
1,1,1,1,1,1,1,6,1,1,1,6,6,6,6,6,6,6,6,6,6,6,6,1,0,4046,9998,9998,9998,9998
```

d) Build the dataset, LclTwoVar, to contain:

- A giant text string variable, lcl_X, that holds the entire data pattern from the current.csv file (SAS variable can be up to $252 characters), and
- _freq_ value, renamed as Actual_Freq_, as a separate variable.

```
data lclOneVar;
       length lcl_X $252;
                                         /* Note: firstobs=2 */
       infile 'current.csv' truncover  LRECL=500 firstobs=2;

       input @1 lcl_X $252.;

       if lcl_X ne " " then  output lclOneVar;

data lclTwoVar;
       merge  lclOneVar
              lclFrq       /* from step b */
              ;
       /* Note there is no BY variable on this merge because the source
       for lclOneVar and lclFrq is the same dataset and hence, should be
       in the same order */

       output lclTwoVar;

proc sort data=lclTwoVar;
       by lcl_X;
```

e) The same steps (steps a through c) should have been done to the "master patterns" that have been approved in earlier rounds; again without the _freq_ variable

```
data lclMst;
              /* Note: same summary class statement executed on  the
              Master dataset and the data pattern output is in temp1Mst
              */
       set temp1Mst (drop=_TYPE_ _freq_);

       output lclMst;

proc export data=lclMst
       outfile='Master.csv'
       dbms=dlm
       replace;
       delimiter=',';
run;
```

f) Read each record of the Master CSV file as one giant text string variable, lcl_X, into the dataset LclMst

```
data lclMst ;
      length lcl_X $252;
                          /* Note firstobs=2 */
      infile 'Master.csv' truncover LRECL=500 firstobs=2;
      input @1 lcl_X  $252.;

      output lclMst;

proc sort data=lclMst;
      by lcl_X;
```

g) Merge the datasets from steps d and f (i.e. LclTwoVar and LclMst by lcl_X); and keep only the patterns that are in the current dataset, LclTwoVar, but not in the Master dataset, LclMst.

```
data lclMrg2Vars;
      merge lclMst (in=f1)
            lclTwoVar (in=f2);
            by lcl_X ;

             /* keep new patterns only in the merged dataset */
      if (f2 and not f1) then
             output lclMrg2Vars;
```

h) Re-read the first line that contains the variable names of the csv file from step c and write it out with the commas; add in the variable name, "actual_Freq_".

```
data lclWriteOut;
      set lclMrg2Vars;

      if _n_ =1 then
      do;
             /* Use the names of the variables from the first line of
             the original csv file to re-create the first line of the
             csv file with the new patterns */

                          /* This is firstobs=1 */
             infile 'Current.csv' truncover  LRECL=500 ;
             input  @1 Lcl_LineOne_1 $250.
                    @251 Lcl_LineOne_2 $250.
                    ;

             file 'Merged.csv';
             put    @1 Lcl_LineOne_1 $250.
                    @251 Lcl_LineOne_2 $250.
                    ",actual_freq_"
                    ;
      end;
      /* note step i continues here */
```

i) Append the "new" patterns from step g into the csv file in step h.

```
             /* note that this is being written to the same Merged.csv
             also */
      put lcl_X ","  actual_freq_;
```

j) Use Proc Import to make a "summary dataset" of the "new" patterns with the actual_Freq_ in it. Then use Proc Print to print out the summary patterns.

```
proc import datafile='Merged.csv'
        out=lclNewPatterns
        dbms=dlm
        replace;
        delimiter=',';
        getnames=yes;
proc print data=lclNewPatterns;
```

Note: In most situations you will find that Proc Import followed by Proc Print does not do a good job of lining up the variables on a single output line for the patterns. The main reason perhaps is that Proc Import defines the format of the variables to be Best32 or max allowable values and this makes the widths of the variables to be much larger than desired. So, the workaround is as follows:

k) Repeat step h but parse out the variable names and create the input statement.

```
data Null;

        length Lcl_LineOne_1 Lcl_LineOne_2 $250;
        retain Lcl_firstline  0;
        retain Lcl_LineOne_1 Lcl_LineOne_2;

        if _n_ =1 then
        do;
                /* Use the names of the variables from the first line of
                the original csv file to create the Input statement varlist
                */

                                /* This is firstobs=1 */
                infile 'Current.csv' truncover  LRECL=500 ;
                input  @1 Lcl_LineOne_1 $250.
                        @251 Lcl_LineOne_2 $250.
                        ;
                lcl_ALLVARS= Lcl_LineOne_1 || Lcl_LineOne_2;

                file 'Input_Class_Stmt_txt';

                put '%let CLASSVAR = ' ;

                do i=1 to 500;
                        call scan(lcl_ALLVARS, i, position, length);
                        if not position then leave;
                        lcl_name=substrn(lcl_ALLVARS, position, length);
                        put lcl_name ' ' @;
                end;
                put ' actual_freq_; ' ;
        end;
```

l) Need to do step i, which writes out the "new" patterns from step g into the csv file, Merged.csv.

```
data lclWriteOut;
        set lclMrg2Vars;

        file 'Merged.csv';
        put lcl_X ","  actual_freq_;
```

m) Instead of step j, Use a standard Data Step with input statement to read in the file from step l; This SAS dataset will have only the "new" patterns with the correct variable names including the actual_Freq_ variable. Then use Proc Summary again to print out the summary patterns so that all variables are lined up in one output line. But first generate the character length definitions to be included for the Data Step.

```
proc contents data=temp out=LCLcont noprint;
data _null_;
       set  lclCont;
       file 'Charlen.txt';
         if type=2 then /* if character then write variable with length*/
           do;
             put 'Length ' name '$' LENGTH ';';
           end;

data lclNewPatterns;
               /* Include any char variable length sets here */
       %include 'Charlen.txt';

               /* this defines the macro variable classvar used below */
       %include 'Input_Class_Stmt.txt';

               /* Note: by default DSD means comma delimiter */
       infile 'Merged.csv' truncover  LRECL=550 firstobs=2 DSD dlm=',';

               /* NOTE: This macro variable classvar is in the generated
              file, Input_class_stmt.txt done in step k */
       input &classvar ;

              /* it seems like proc export writes out an additional blank
              line! So, the conditional output here */
       if actual_freq_ ne  . then
              output lclNewPatterns;

proc print data= lclNewPatterns;
VAR &classvar; /* generated class stmt vars*/
```

Data Example: In this example, the master dataset has 65 cases with 5 unique patterns. The current input dataset has 53 cases with 4 unique patterns. Patterns 2 and 4 in the current dataset (marked with asterisks) exist in the master dataset and have been reviewed already. Patterns 1 and 3 are new ones and are to be reviewed.

Master Data Summary:

```
STUDY  - TEMP - Analgesics - 1 - MASTER
       D  P                 F  F  H     M  O                    T  T     A      A    A    A    A    A
    A  A  E  V  A  C  D     I  I  Y  M  O  X  P  P        T     A  R     N P    N    N    N    N    N
    N  R  R  I  N  O  E  D  O  O  D  E  R  Y  H  R     S  A  T  L  A  U  L R    A    A    A    A    A          _  _
    L  V  C  C  L  D  M  I  R  R  R  T  P  C  E  O  S  T  L  A  W  M  L  N 4    L    L    L    L    L          T  F
    E  T  T  O  C  E  E  L  I  I  O  H  H  O  N  P  K  A  A  L  I  A  T  O 1    N    N    N    N    N          Y  R
 O  V  Y  Y  L  A  I  R  A  C  N  C  D  I  N  C  O  6  D  C  W  N  D  R  L C    E    E    E    E    E          P  E
 b  E  L  L  O  R  N  O  U  E  A  O  O  N  T  O  X  5  O  E  I  N  O  A  S N    W    W    W    W    W          E  Q
 s  R  C  X  R  D  E  L  D  T  L  D  N  E  N  D  Y  A  L  N  N  X  L  M  T T    A    B    C    D    E          _  _

 1  1  1  1  1  1  1  1  6  6  6  1  6  6  1  6  6  6  6  6  1  6  6  6  1 0 9985 9985 9998 9998 9998 1073741823  1
 2  1  2  2  2  2 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  1 0  422 9998 9998 9998 9998 1073741823  1
 3  1  2  2  2  2 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  1 0 2003 9998 9998 9998 9998 1073741823  1
*4  1  2  2  2  2 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  1 0 4031 9998 9998 9998 9998 1073741823  1
*5 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 0 9991 9991 9991 9991 9991 1073741823 61
```

Current Data Full Summary:

```
STUDY  - TEMP - Analgesics - 1 - ORIGINAL ALL
       D  P                 F  F  H     M  O                    T  T     A      A    A    A    A    A
    A  A  E  V  A  C  D     I  I  Y  M  O  X  P  P        T     A  R     N P    N    N    N    N    N
    N  R  R  I  N  O  E  D  O  O  D  E  R  Y  H  R     S  A  T  L  A  U  L R    A    A    A    A    A          _  _
    L  V  C  C  L  D  M  I  R  R  R  T  P  C  E  O  S  T  L  A  W  M  L  N 4    L    L    L    L    L          T  F
    E  T  T  O  C  E  E  L  I  I  O  H  H  O  N  P  K  A  A  L  I  A  T  O 1    N    N    N    N    N          Y  R
 O  V  Y  Y  L  A  I  R  A  C  N  C  D  I  N  C  O  6  D  C  W  N  D  R  L C    E    E    E    E    E          P  E
 b  E  L  L  O  R  N  O  U  E  A  O  O  N  T  O  X  5  O  E  I  N  O  A  S N    W    W    W    W    W          E  Q
 s  R  C  X  R  D  E  L  D  T  L  D  N  E  N  D  Y  A  L  N  N  X  L  M  T T    A    B    C    D    E          _  _

 1  1  1  1  1  1  1  1  6  1  1  1  6  6  6  6  6  6  6  6  6  6  6  6  1 0 4046 9998 9998 9998 9998 1073741823  1
*2  1  2  2  2  2 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  1 0 4031 9998 9998 9998 9998 1073741823  1
 3  1  2  2  2  2 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99 99  1 1  425 3086 9998 9998 9998 1073741823  1
*4 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 91 0 9991 9991 9991 9991 9991 1073741823 50
```

New Patterns Only Summary of Current Data:

```
STUDY  - TEMP - Analgesics - MERGED
      D  P                     F   F   H       M   O                           T   T       A        A     A     A     A     A   a
   A  A  E  V  A   C   D       I   I   Y   M   O   X   P   P           T       A   R       N  P     N     N     N     N     N   l
   N  R  R  I  N   O   E   D   O   O   D   E   R   Y   H   R       S   A   T   L   A   U   L  R     A     A     A     A     A   _
   L  V  C  C  L   D   M   I   R   R   R   T   P   C   E   O   S   T   L   A   W   M   L   N  4     L     L     L     L     L   f
   E  T  T  O  C   E   E   L   I   I   O   H   H   O   N   P   K   A   A   L   I   A   T   O  1     N     N     N     N     N   r
O  V  Y  Y  L  A   I   R   A   C   N   C   D   I   N   C   O   6   D   C   W   N   D   R   L  C     E     E     E     E     E   e
b  E  L  L  O  R   N   O   U   E   A   O   O   N   T   O   X   5   O   E   I   N   O   A   S  N     W     W     W     W     W   q
s  R  C  X  R  D   E   L   D   T   L   D   N   E   N   D   Y   A   L   N   N   X   L   M   T  T     A     B     C     D     E   _

1  1  1  1  1  1   1   1   6   1   1   1   6   6   6   6   6   6   6   6   6   6   6   6   1  0  4046  9998  9998  9998  9998   1
2  1  2  2  2  2  99  99  99  99  99  99  99  99  99  99  99  99  99  99  99  99  99  99   1  1   425  3086  9998  9998  9998   1
```

CONCLUSION

This technique helps reduce the number of patterns the reviewer has to approve and is very useful for studies that implement the same edits over different years or datasets. I created some SAS® Macros to make it easier to apply this technique to the large number of summaries that we create and review. This technique could be extended to comparing datasets and not be limited to Proc Summary patterns. In this way, rather than reviewing Proc Summaries of small sets of variables, one could compare much larger sets of variables or the whole dataset (written pattern record not exceeding 252 bytes!) so that one can identify the new patterns/records for the current dataset.

REFERENCES

SAS® On-line Macro examples.

CONTACT INFORMATION

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

Author Name: Priya Suresh
Enterprise: RTI International
Address: 3040 Cornwallis Road
City state ZIP: RTP, NC 27709
Work Phone: 919 541-7428
Fax: 919 541-6178
Email: psoh@rti.org
Web: www.rti.org