SAS code to compute a set of new variables from Time Series variables

Anani K. Hoegnifioh, Risk Assessment Group, US Cellular, Chicago, IL

Abstract

This study is intended to assist analysts in generating maximum number of variables using simple arithmetic operators (addition, difference, minimum, maximum, counts, and division) and one or more historical data sets. These data sets might include monthly amount paid, daily number of received customer service calls, weekly worked hours on a project, or annual total sale of a specific product. During a statistical modeling process, analysts are often confronted with the task of computing derived variables using the existing available variables. The advantage of this methodology is that the new variables may be analytically useful than the original ones. This paper provides a way to compute all the possible variables using a set of historical data such as daily, weekly, monthly, or yearly information using the simple arithmetic operators (addition, difference, minimum, maximum, counts, and division) between two or more period of an observed time-series variable and aggregate them into a derived variable. An arithmetic formula was developed to evaluate the number of possible new variables from two parameters (number of observed period and type of operators. The code includes many SAS features that are very useful tools for SAS programmers to incorporate in their future code such as %SYSFUNC, SQL, %INCLUDE, CALL SYMPUT, %MACRO, DICTIONARY.XXXX (where XXXX can be TABLE, COLUMN), procedures (SORT, CONTENTS), and data steps (MERGE, _NULL_) and many more.

Introduction

Statistical data modeling process often leads to the computation of new derived variables using the existing variables. The main reason for doing so is that the derived data may constitute potentially stronger predictors for the models to be developed. This project is intended to assist modelers in generating maximum number of variables using simple

arithmetic operators (addition, difference, minimum, maximum, counts, and division) and a given characteristic that can be observed historically (daily, weekly, monthly, or annually). An arithmetic formula was developed to calculate the maximum number of new variables that can be generated. The paper is divided into two different parts.

The first part contains the historical data into a modeling data format and labels the variable based on the time period. Once this section is completed then the second part will compute the new derived variables.

Part One

Let's assume that a modeler was asked to develop a telecommunication statistical model. During the Exploratory Data Analysis, it was observed that the available data set contains a set of variables such as amount paid and number of minutes used for the last 6 months for each account in the dataset. Table 1 is an example of variables in the original dataset.

Table 1

ID	DATE	TEST_A	TEST_B
83382960	10/1/2010	21.4	20.4
83382960	11/1/2010	21.5	20.4
83382960	12/1/2010	21.7	20.5
83382960	1/1/2011	21.5	20.6
83382960	2/1/2011	21.6	20.4
83382960	3/1/2011	21.6	20.4
83382961	10/1/2010	21.8	20.5
83382961	11/1/2010	21	20.2
83382961	12/1/2010	21.5	20
83382961	1/1/2011	21.6	20.7
83382961	2/1/2011	21.4	20.3
83382961	3/1/2011	21.8	20.4

In this situation, let:

N = number of original variables (in this case, N = 2)

PERIODICVARS = repetitive variables (TEST_A and TEST_B)

NBPERIODS = number of repetitive variables (NBPERIODS = 6 months)

NONPERIODICVARS = static variables - one time variables - (ID, DATE)

The first part of the code transforms the original data set into the format shown below in table 2.

Table 2

ID	STDATE	EDDATE	TEST_A1	TEST_A2	TEST_A3	TEST_A4	TEST_A5	TEST_A6	TEST_B1	TEST_B2	TEST_B3	TEST_B4	TEST_B5	TEST_B6
83382960	10/1/2010	3/1/2011	21.5	21.6	21.7	22	22.3	22.3	21.2	21.8	21.6	21.9	22.1	22.2
83382961	10/1/2010	3/1/2011	20.3	20.4	20.5	20.8	21.1	21.1	20.2	20.5	20.3	20.3	20.4	21.2

The code includes many SAS features that are very useful tools for SAS programmers to incorporate in their future code such as %SYSFUNC, SQL, %INCLUDE, CALL SYMPUT, %MACRO, DICTIONARY.XXXX (where XXXX can be TABLE, COLUMN), SORT, CONTENTS, MERGE, MACRO _NULL_, as well as %DO … %TO … and many more. Once the dataset shown in table 2 is completed, the second part is ready to run. The code generates all the meaningful possible combinations of variables using the following arithmetic operators: sum, average, difference, Minimum, Maximum, count, and percentage of difference using each array variables. Table 3 shows the total number of combinations one can obtain from a set of array variables giving one set of array variables for different numbers of periods.

Table 3

DESCRIPTIVE COMPUTATION	DESCRIPTIVE FORMULA	CONDENSED FORMULA
MAXIMUM NUNBER OF PERIODS (N)	N	
NUMBER OFCOMBINATIONS (P)	P	
NUMBER OF PERIODS (FROM 1 TO N)	1, 2, 3, …, N	
NON MISSING VALUES FOR ALL PERIODS	1	1
ORIGINAL VARIABLES (INDEXED)	N	N
DIFFERENCE VARIABLES (POSSIBLE PAIRS)	FACT(N)/(FACT(2)*FACT(N-2))	B
% DIFFERENCE VARIABLES (POSSIBLE PAIRS)	FACT(N)/(FACT(2)*FACT(N-2))	B
MINIMUM VARIABLES (POSSIBLE COMBINATION)	FACT(N)/(FACT(P)*FACT(N-P))	A
MAXIMUM VARIABLES (POSSIBLE COMBINATION)	FACT(N)/(FACT(P)*FACT(N-P))	A
AVERAGE VARIABLES (POSSIBLE COMBINATION)	FACT(N)/(FACT(P)*FACT(N-P))	A
SUMMATION VARIABLES (POSSIBLE COMBINATION)	FACT(N)/(FACT(P)*FACT(N-P))	A
TOTAL NUMBER OF NEW VARIABLES = TOTAL		4A+2B+N+1

$$TOTAL = 4\left(\frac{N!}{P!*(N-P)!}\right) + 2\left(\frac{N!}{(N-2)!*(P-2)!}\right) + N + 1$$

Part Two

With our method of computation, using one characteristic observed for n periods, one can derive the total number of new variables. The next step will be to check their

predictability using different data exploration techniques such as Exploratory Data Analysis, or Data Mining or any other methods to select the potential useful attributes for the further analysis. Table 4 is a summary of the potential number of variables that can be derived from an historical observation of a single variable for N consecutive periods (yearly, quarterly, monthly, weekly, daily etc…)

Table 4

Total Number Of Derived Variables A Single Time Series Variable Observed For N Periods												
N =	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
NMS(P) =	1	1	1	1	1	1	1	1	1	1	1	1
ORIGINAL =	1	2	3	4	5	6	7	8	9	10	11	12
DIF(P) =	0	1	3	6	10	15	21	28	36	45	55	66
PDF(P) =	0	1	3	6	10	15	21	28	36	45	55	66
MIN(P) =	0	1	4	11	26	57	120	247	502	1013	2036	4083
MAX(P) =	0	1	4	11	26	57	120	247	502	1013	2036	4083
AVG(P) =	0	1	4	11	26	57	120	247	502	1013	2036	4083
SUM(P) =	0	1	4	11	26	57	120	247	502	1013	2036	4083
TOTAL =	**2**	**9**	**26**	**61**	**130**	**265**	**530**	**1053**	**2090**	**4153**	**8266**	**16477**

The section of the code that creates all the derived variables (see example in table 5) are in a separate macro and are called by the main code via a %INCLUDE statement. The code can be found in the appendix. The rest of the presentation will be a short demo of how the code works.

Table 5

Name of Derived Variables from a period N = 3 using one Variable					Total Per Category
NON MISSING	NMSTEST_A				1
ORIGINAL	TEST_A1	TEST_A2	TEST_A3		3
DIFFERENCE	DIFTEST_A12	DIFTEST_A13	DIFTEST_A23		3
% DIFFERENCE	PDFTEST_A12	PDFTEST_A13	PDFTEST_A23		3
MINIMUM	MINTEST_A12	MINTEST_A13	MINTEST_A23	MINTEST_A123	4
MAXIMUM	MAXTEST_A12	MAXTEST_A13	MAXTEST_A23	MAXTEST_A123	4
AVGERAGE	AVGTEST_A12	AVGTEST_A13	AVGTEST_A23	AVGTEST_A123	4
SUMMATION	SUMTEST_A12	SUMTEST_A13	SUMTEST_A23	SUMTEST_A123	4
					26

Conclusion

This method of computing the derived variables can be useful for modeling purposes and save time during the variable creation stage. The macros are very straightforward but the user needs a strong understanding of SAS programming especially in SAS MACRO in order to fully take advantage of the code.

Acknowledgements:

I would like to thank Stephen E. Brooks for his continuous assistance during the coding period. My sincere gratitude goes also to my Manager Joel Walker and my teammates (P. Arroyo, X. Hu, J. Marek, and K. Strzalka) for their encouragement and advice.

Contact Information:

Your comments and questions are encouraged. Please contact the author at the following address:

Anani K. Hoegnifioh – Statistical Analyst (Anani.hoegnifioh@uscellular.com)
U.S. Cellular®
8410 West Bryn Mawr Ave. Suite 700
Chicago, IL 60631

Reference

1. Delwiche, Lora D., and Susan J. Slaughter. *The Little SAS Book: a Primer: a Programming Approach*. Cary, NC: SAS Institute, 2008.
2. *SAS 9.2 Macro Language: Reference.* Cary, NC: SAS Publishing, Feb 2009.
3. Burlew, Michele M. *SAS Macro Programming Made Easy*. Cary, NC: SAS Institute, 2006.

SAS Code

1. Main Code
2. Computational Macro