A MENU-DRIVEN TABLE GENERATOR

Joy Y. S. Jair, Kaohsiung Refinery, CPC
Rex C. T. Wu, Kaohsiung Refinery, CPC

Abstract

Instead of coding a SAS program each time when a demand is requested, a user friendly, menu-driven table generator is introduced. In the generator, users can define their tables by just selecting some conditions and interested items he needed in the menu. The desired tables are produced to a great degree of flexibility according to the variate selections. SAS/AF ® and SAS ® macro facilities incorporated with PROC TABULATE are fully utilized to meet the versatile requirements. A personnel system is used as an example and is under normal operation now. The results show great success.

Preface

SAS PROC TABULATE constructs table of descriptive statistics from class variables, analysis variables and keywords for statistics. It is not uncommon that the demands to produce different kinds of tables from the end users have become the major loads of an application programmer. Instead of coding program for each demand, a menu-driven table generator is needed for most of the cases. We can display the most frequently used items on the menu by using SAS/AF. After the user selects the desired conditions and classes, we will use SAS macros to convert them into a complete SAS program which will then generate the desired tables. In this way, we only have to write one program to meet variate kinds of demands. As a result, the end users not only get their wanted tables ahead of schedule, but the Information Systems Department personnel also be free from constant interruptions for adhoc demands. Thus the overall productivity is increased.

System Structure

The system structure of the table generator is shown below.

When the "Generate Tables" option is selected, the system will show the selection menu which will guide the user to select the conditions and interesting items he needed. After the completion of the selection, the required tables will be generated automatically. When the "Quit" option is selected, the system will quit and return to TSO.

System Description

An existing personnel system is selected as our example. Since our personnel system is in Chinese System and the SAS System also supports DBCS (Double Bytes Character Set)
so we use Chinese to design the menu. Thus all the tables generated are in Chinese, too. In this article, we translated the menu into English for explanation purpose, but the sample output is in Chinese to show the originality.

In the menu, there are four selection items divided into five maps as can be seen from Fig. 1 to Fig. 5. The four items are Column Selection, Row Selection, Condition Selection and Output Selection. In the Column/Row Selection, the user can select Column/Row classes he wanted in the table.

Within the Column/Row Selection, the selection items are classified into several groups in vertical direction (e.g. Employee Status, Sex,..., etc.) which will be treated as concatenation elements. Within each group, there are up to ten selectable items which will be treated as cross elements when selected. In order to handle the Column/Row Selection, there are four macros which will be described briefly as follows:

1) %TABLE : will generate the TABLE statement of PROC TABULATE. The sequence of the cross elements of each group can be specified by user using the digits from 0 to 9. It also generates the CLASS variables and VAR variables.

2) %INTERVAL : will calculate and generate the class range of class variables for those numerical items which the user specifies the starting, ending and interval information (e.g. Age).

3) %RANGE : will classify those non-numerical variables which are specified by High-Low limits to the appropriate classes (e.g. Salary Rate).

4) %FORMAT : will generate PROC FORMAT which will show the range type class variables described above in Chinese specified by the user.

In the Condition Selection, one can make the single condition selection or a From-To type selection. Two macros have been designed:

1) %CONM : will handle the single condition selection (e.g Department : L100).

2) %CONNECT : will handle the From-To type selection (e.g. from Department L100 to L600) and connect to the other conditions.

In the Output Selection, one can select output to the desired printer or save as a file. Macro %PRINT will handle this.

After all the selections have been made, the user then press <PF3> to execute. The desired tables will be generated automatically.

Conclusion

The idea of designing a menu-driven table generator comes from the needs to solve the great demands from the end users. Through this system, many versatile forms of tables selected by users can be generated automatically. After the implementation of this system, the responses of the end users show great favorable. The benefits we expected are realized gradually. Not only the users feel easy and friendly, but also the number of requests dropped dramatically as much as 85%.
Major Macro Programs

There are six major macros in the following:

1. %INTERVAL(BEGN, END, RANGE, VAR);
2. %RANGE(START, STOP, LOW, UP, VAR);
3. %FORMAT(START, STOP, LOW, UP, VAR, SPEC);
4. %TABLE(BG, START, STOP, V0, V1, V2, V3, V4, V5, V6, V7, V8, V9, V10);
5. %CONM(START, STOP, VAR, OP, PN, QT);
6. %CONNECT(A, B, C, D, E, F, G, H);

%MACRO INTERVAL(BEGN, END, RANGE, VAR);
%LET I=1;
%IF &&&BEGN,=0 & &&&END,=0 & &&&RANGE,=0 %THEN %DO;
%LET INTL=%EVAL((&&&END-&&&BEGN)/&&&RANGE);
%LET L=%EVAL(&&&BEGN+1); %LET U=%EVAL(&&&BEGN+&&&RANGE);
%DO %WHILE(&I<=&INTL);
%IF &I=1 %THEN PUT "IF &VAR<=&&&BEGN THEN &VAR.l='<=&&&BEGN
PUT "ELSE IF &L<=&VAR<=&U THEN &VAR.l='&L-&U';
%LET L=%EVAL(&L+&&&RANGE); %LET U=%EVAL(&U+&&&RANGE);
%LET I=%EVAL(&I+1);
%END; %LET U=%EVAL(&U-&&&RANGE+l); PUT "ELSE &VAR.l='&U-99';
%END;
%MEND INTERVAL;

%MACRO RANGE(START, STOP, LOW, UP, VAR);
%LET SW=Y; %LET I=&START;
%DO %WHILE(&I<=&STOP);
%IF &&&LOW&I,=0 & &&&UP&I,=0 %THEN %DO;
%IF &SW=Y %THEN %DO; %LET SW=N; PUT "IF &VAR<:'&&&LOW&I' THEN &VAR.l='<&&&LOW&I**';
%IF &SW=Y %THEN %DO; %LET SW=N; PUT "ELSE IF &VAR<:'&&&UP&I' THEN &VAR.l='&&&LOW&I-&&&UP&I';
%IF &&&UP&I,= %THEN %LET SS=&&&UP&I; %END; %LET I=%EVAL(&I+l);
%END; %IF &SW,=Y %THEN PUT" ELSE &VAR.l='&SS>**';
%MEND RANGE;

%MACRO FORMAT(START, STOP, LOW, UP, VAR, SPEC);
%LET SW=Y; %LET BL=%STR( ); %LET I=&START;
%DO %WHILE(&I<=&STOP);
%IF %LENGTH(&&&LOW&I),=0 & %LENGTH(&&&UP&I),=0 %THEN %DO;
%IF &SW=Y %THEN %DO; %LET SW=N; PUT "PROC FORMAT;
%END; PUT "VALUE $&VAR" I" '<&&&LOW&I****'='<=&&&LOW&I'';
%END; PUT @1 " '&LOW&I--&UP&I=' @16
" '&SPEC&I.1&BL&&&SPEC&I.2&BL&&&SPEC&I.3&BL&&&SPEC&I.4'';
%END; %LET I=%EVAL(&I+1);
%END;
%IF &SW,=Y %THEN %DO;
%LET FORMAT=%STR( &VAR.l $&VAR ..);PUT" OTHER='Other';
%MEND FORMAT;

%MACRO TABLE(BG, START, STOP, V0, V1, V2, V3, V4, V5, V6, V7, V8, V9, V10);
%LET J=1; %LET I=&STOP; %LET TABLE=; %LET RS=;
%IF &&&FIELD&BG,= %THEN %DO;
%DO %WHILE(&I>=&START);
%LET K=&&&FIELD&I;
%IF &&&FIELD&I,= %THEN %DO;

806
Illustrations and Sample Run

<table>
<thead>
<tr>
<th>Employee</th>
<th>Work</th>
<th>Job</th>
<th>Rate</th>
<th>Sex</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Wkyear</td>
<td>Educ</td>
<td>Exam</td>
<td>Native</td>
<td>Merit</td>
</tr>
<tr>
<td>Sex</td>
<td>Work</td>
<td>Job</td>
<td>Rate</td>
<td>Status</td>
<td>Age</td>
</tr>
<tr>
<td>Wkyear</td>
<td>Educ</td>
<td>Exam</td>
<td>Native</td>
<td>Merit</td>
<td></td>
</tr>
<tr>
<td>Native</td>
<td>Work</td>
<td>Job</td>
<td>Rate</td>
<td>Status</td>
<td>Age</td>
</tr>
<tr>
<td>Wkyear</td>
<td>Educ</td>
<td>Exam</td>
<td>Sex</td>
<td>Merit</td>
<td></td>
</tr>
<tr>
<td>Education</td>
<td>Work</td>
<td>Job</td>
<td>Rate</td>
<td>Status</td>
<td>Age</td>
</tr>
<tr>
<td>Wkyear</td>
<td>Sex</td>
<td>Exam</td>
<td>Native</td>
<td>Merit</td>
<td></td>
</tr>
</tbody>
</table>

X Age: From 20 to 50, Interval: 10. Average Age

Fig. 1 Table Generator —— Map 1

807
Work Year: From __ to __, Interval: ___. Average Wkyear
- Work _ Job _ Rate _ Status _ Age
- Sex _ Educ _ Exam _ Native _ Merit

Salary: Work _ Job _ Sex _ Status _ Age
Rate Wkyear _ Educ _ Exam _ Native _ Merit
- Low _ High ----Chinese Specification----

Examination: Work _ Job _ Rate _ Status _ Age
Wkyear _ Educ _ Sex _ Native _ Merit

Job Code: Work _ Sex _ Rate _ Status _ Age
Wkyear _ Educ _ Exam _ Native _ Merit

Fig. 2 Table Generator --- Map 2

Work _ Job _ Rate _ Status _ Sex _ Age
Class Wkyear _ Educ _ Exam _ Native _ Merit

Merit _ Merit Code _ Times

2. Row Selection _ Subtotal X Total

X Employee Work _ Job _ Rate _ Sex _ Age
Status Wkyear _ Educ _ Exam _ Native _ Merit

Sex _ Work _ Job _ Rate _ Status _ Age
Wkyear _ Educ _ Exam _ Native _ Merit

Native _ Work _ Job _ Rate _ Status _ Age
Wkyear _ Educ _ Exam _ Sex _ Merit

Education: Work _ Job _ Rate _ Status _ Age
Wkyear _ Sex _ Exam _ Native _ Merit

Age _ Work _ Job _ Rate _ Status _ Sex
Wkyear _ Educ _ Exam _ Native _ Merit

Work Year: Work _ Job _ Rate _ Status _ Age
Sex _ Educ _ Exam _ Native _ Merit

Fig. 3 Table Generator --- Map 3
3. Condition Selection:

a. Department:
   - Form A000 To R900, From IA00 To 9R00
   - Form _____ To _____, From _____ To _____

b. Employee:
   - Staff _ (E) Worker _ (C) Worker
   - Temp-Staff _ Other

c. Course:
   - ________

Fig. 4 Table Generator —— Map 4

4. Output Selection:

a. Printer:
   - 3820 B4 Form _ 3820 A4 Form(V) X 3820 A4 Form(H)
   - 3835 B4 Form _ 3835 A4 Form(V) _ 3835 A4 Form(H)
   - 3835 B4 (W:200, H:62)

b. File:
   - (Data Set Name: PA.PXXX.SASOUT.SAM)

5. Title:

6. Footnote:
   ————

After Selection press <PF3> to submit

Fig. 5 Table Generator —— Map 5
### Kaochung Refinery --- Statistics of Employee's Age

高雄煉油廠 --- 工作人員年齡統計表

<table>
<thead>
<tr>
<th>年齡</th>
<th>男性</th>
<th>女性</th>
<th>合計</th>
</tr>
</thead>
<tbody>
<tr>
<td>13-20</td>
<td>1,466</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21-30</td>
<td>5,049</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31-40</td>
<td>1,088</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41-50</td>
<td>1,075</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51-99</td>
<td>1,192</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The handwriting English Part is translated from Chinese for reading purposes.