

When and How the PROC TABULATE Can be Used

Minghui Yang
Project Manager and Research Specialist

0. Introduction

For many SAS users and programmers, PROC TABULATE might be one among the most difficult procedures in the Base SAS. PROC TABULATE can be a very useful tool for report writing and data analysis if used correctly. Instead of spending numerous time developing customer reports, PROC TABULATE can write many reports as good as customer ones. This article is trying to present when PROC TABULATE can be used and how to use it more efficiently. The key tips are:

- Always start from report request first to design the report format, then determine both class and analysis variables, then generate the SAS data;
- Understand the relationship between PROC TABULATE and the variables used - class variable(s) and analysis variable(s).
- Understand what kind of statistics can be generated directly by PROC TABULATE and what can not be generated directly. If not, how can we find a way to generate the statistics by using PROC TABULATE.
- Understand the syntax, formats and some tricks, especially when percentage is applied.

PROC TABULATE is a descriptive procedure in the base SAS. We call it a descriptive procedure because:

- It summarizes values for all values in a dataset;
- It prints the summaries in a table format;
- It provides statistics for class variables as well as analysis variables.

But PROC TABULATE can only provide certain statistics directly. Those statistics are similar to that generated by PROC MEANS or PROC FREQ, or PROC SUMMARY. The major statistics directly generated by PROC TABULATE are:

- N: count (frequencies, for class variable or analysis variable)
- SUM: sum (for analysis variable only)
- MEAN: average (for analysis variable only)
- MIN: minimum (for analysis variable only)
- MAX: maximum (for analysis variable only)
- PCTN: (percentage for N, for both class and analysis variables)
- PCTSUM (percentage for SUM, analysis variable only)

There are several other statistics can be generated by PROC TABULATE directly, but they're not commonly used. The above seven statistics are the most commonly used statistics directly generated by PROC TABULATE.

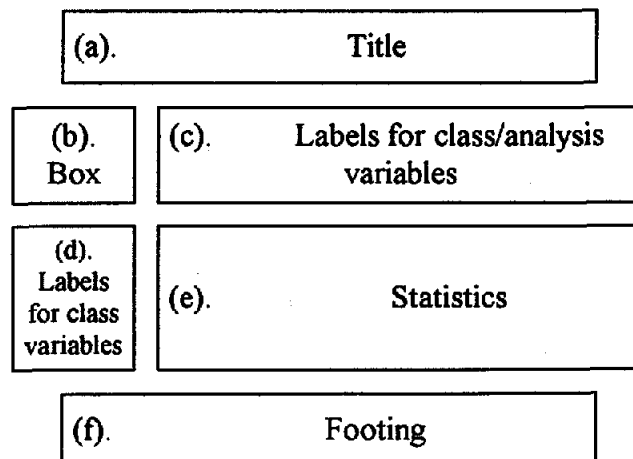
1. Report Design and Data Design

Some SAS programmers read raw data first, then sort, merge the raw data, then output a subset from the raw data. They then try to use such a dataset to write reports. When they can not get the report they want, they started to complain about PROC TABULATE. The correct procedure should be:

Design your report format first, especially when the report is complicated. Then generate your SAS dataset(s) according to the report format.

You need to design your report table first because it is not only PROC TABULATE is a 'difficult' procedure in the Base SAS, but also you need to understand what you want. If you want other statistics (not those seven mentioned above) in your report such as MEDIAN, Q3 (75 percentile), P90 (90 percentile) or P95 (95 percentile), you have to use PROC UNIVARIATE first to get those statistics. You may need to use DATA step and PROC UNIVARIATE and SORT several times to get some analysis variables such as the average within a group, percentage, sum or count. Those statistics generally become analysis variables in your dataset. Finally, you merge all datasets together to generate one dataset which can use PROC TABULATE directly. If any statistics can not be generated by PROC TABULATE directly, you must use other SAS procedures to design and generate your final dataset for PROC TABULATE. If all statistics you want can be generated by PROC TABULATE directly, it is much easier for you to use PROC TABULATE to generate report tables.

A report written by PROC TABULATE can have the following formats:



Where Footing (f) is seldom used, Box (b) can be omitted in many cases, Title (a) is useful, but (c), (d) and (e) are the key components in the PROC TABULATE. When we talk about design, we are mainly talking about (c), (d), and (e).

Because (c) and (d) are related to the class variable and analysis variable in PROC TABULATE, we need some explanations here.

A Class Variable should be

- Character or
- Numerical and
- Discrete (only few levels)

An Analysis Variables should be

- Numerical only and
- Continuous (infinite levels) or discrete (few levels)

If you use PROC SORT, the sorted variables are generally class variables. We will talk more about class variables later.

2. The Types of Reports Generated by PROC TABULATE

Generally speaking, there are three types of reports:

1. No CLASS variables. The table contains statistics for analysis variables only. We use VAR and TABLE in PROC TABULATE (a rare case);
 2. Frequencies within CLASS variables only. We use CLASS and TABLE in PROC TABULATE (N and PCTN are the major statistics in such a case, common);
 3. Frequencies and direct statistics within CLASS variables. We use all CLASS, VAR, and TABLE in PROC TABULATE (very common, all seven statistics can be used).
- An option 'by' can be added in all the three basic types.

Among the three basic types, the Type 2 and Type 3 are used often and need a good understanding. Since Type 1 is not used very often, you do not need to remember it. Now, we use an example to illustrate how to work at the Type 2 and Type 3.

[Example] The Legislature would like to know the number of criminal felony cases for juveniles (under 18) in 1995 in a county. In addition, they would like to know the number by gender, ethnic group, and age. How can you generate a quick report by using PROC TABULATE?

[Solution]: Because the number (count) is wanted, and the gender, ethnic group, and age are with finite levels, this is the Type 2 case. The class variables are *gender*, *ethnic*, and *age*. If the data is read from a large database where age is not available, but the birthday and the charge date are available, you should know how to use SAS time functions to calculate the age. Also, if you prefer certain order to display your table, say male first and female second; or in ethnic group, you prefer the display order as White,

Black, and Other, you can use numerical variables for all three class variables and use PROC FORMAT to generate the order you want:

```
proc format;
value gender
1='Male'
2='Female'; /* in this case, character may work by descending sort */

value ethnic
1='White'
2='Black'
3='Other'; /* This can guarantee the right order for your table display */

value age;
0-10='0-10'
11-13='11-13'
14-15='14-15'
16='16'
17-18='17'; /* use format to control different age group */

data test; /* input data for the PROC TABULATE */
input gender race age charge;
cards;
1 1 11 1
1 2 10 3
1 3 14 2
1 2 17 1
1 1 16 1
2 1 12 2
1 2 15 1
(continued)
;
proc tabulate data=test; /* Line 1 */
class gender race age; /* Line 2 */
format gender gender. race ethnic. age age.; /* Line 3 */
table (gender=' ' all='Total')*(race=' ' all='Total'),
(age='Age' all='All')*N=' '*f=5.0/Box='Sample Table' RTS=27; /* Line 4 */
title 'Juvenile Felony Cases by Gender, Ethnic, and Age in 1995'; /* Line 5 */
```

The table will look like (This is not a real table rather than illustrations)

Juvenile Felony Cases by Gender, Ethnic, and Age in 1995

Sample Table		Age					All
		0-10	11-13	14-15	16	17	
Male	White						
	Black						
	Other						
	Total						
Female	Similar to above						
Total	Similar to above						

The PROC TABULATE looks simple, but there are several things you may need to know:

- **Separation lines:** if you add 'noseps' option at Line 1, the separation lines between 'White', 'Black', 'Others', 'Total' will disappear.;
- **Data sort:** You don't need to sort this data if you only want statistics on N or PCTN (sorting does not hurt). You can still generate the table;
- **'LABEL' and 'KEYLABEL' statement:** 'LABEL' is for class/analysis variable names and 'KEYLABEL' is for statistics such as 'N', 'MIN', 'SUM' etc. You don't need to use them in PROC TABULATE. You may use the free format instead. The advantage of the free format is the flexibility. If you want the 'all' for gender labeled 'Total' and the 'all' for age labeled 'All', you can do so by the free format method.
- **Free format:** (age='Age' all='All')*N=' '*f=5.0 in Line 4 is a typical free format method which is very flexible. N=' ' mean the label for N will not be printed.
- **Depth:** the depth of the class/analysis variables (including analysis variables, statistics, and formats) has a default value 10, for example the statement of $a*b*c*N=' '*f=9.0$ has the depth 5 (there are 4 '*'s) where a , b , c are either all class variables or a and b are class variables while c is an analysis variables.
- **'By' Option:** If you have several years data (not only 1995), *year* can be a class variable. But you could use the 'by' option to generate several tables (the data needs to be sorted by *year*) so each year has a table.
- **RTS:** RTS=27 will control the width of Box (b). Since the width will be divided by 2 after subtracting 3 (3 lines for 2 class variables) from 27, so each class variable of *gender*race* in our example will have a width 12. You need to consider the width for the whole table, as well as the label part for the class variables. You might change RTS= statement to achieve the best width for your table display.
- **Comma:** You can have at most two commas: the 1st comma is for page, between the 1st and 2nd comma is for rows, after 2nd comma is for columns. It means you can

have at most three dimensions for your table. If you put one comma, by default, it is for rows and columns. Please see Line 4 for its usage.

3. Percentage (PCTN<>)

Percentage part might be the most headache part for many SAS programmers. The only tips here are 'trying'. Now, we change the Line 4 in above program as:

```
(age='Age' all)*(N='#'*f=2.0
PCTN<age all>='%'*f=4.1)/Box='Sample Table' RTS=19;
```

There are several changes:

1. *RTS=19* instead of 27 which changes the width of Box (b)
2. *N='#'*f=2.0* instead of *N=' '*f=5.0* which changes the width of the column
3. Add *PCTN<age all>='%'*f=4.1* to print the percentage for the age group
4. Add () after *(age='Age' all)**

The following is the actual output:

Juvenile Felony cases by Gender, Ethnic, and Age in 1995

Sample Table		Age											
		0-10		11-13		14-15		16		17		ALL	
		#	%	#	%	#	%	#	%	#	%	#	%
Male	White	3	11.1	3	11.1	6	22.2	9	33.3	6	22.2	27	100
	Black	3	20.0	*	*	6	40.0	3	20.0	3	20.0	15	100
	Other	*	*	3	33.3	6	66.7	*	*	*	*	9	100
	Total	6	11.8	6	11.8	18	35.3	12	23.5	9	17.6	51	100
Female	White	2	11.8	6	35.3	4	23.5	2	11.8	3	17.6	17	100
	Black	*	*	3	42.9	3	42.9	1	14.3	*	*	7	100
	Other	*	*	1	16.7	1	16.7	4	66.7	*	*	6	100
	Total	2	6.7	10	33.3	8	26.7	7	23.3	3	10.0	30	100
Total	White	5	11.4	9	20.5	10	22.7	11	25.0	9	20.5	44	100
	Black	3	13.6	3	13.6	9	40.9	4	18.2	3	13.6	22	100
	Other	*	*	4	26.7	7	46.7	4	26.7	*	*	15	100
	Total	8	9.9	16	19.8	26	32.1	19	23.5	12	14.8	81	100

There are five things we need to pay a close attention:

1. Percentage can only be generated directly within class variable(s). If there are two analysis variables, you can not generate percentage between them directly;
2. You need to determine the column variable(s) as well as the row variable(s) first. The column variable(s) and the row variable(s) are separated by a comma (,). For example, in our TABLE statement of *(gender all)*(race all), (age all)*, we know *(age all)* is for columns and *(gender all)*(race all)* is for rows.
3. If you want column percentage, you put the full column variable descriptions into $\diamond$ of your PCTN $\diamond$ statement. Similarly, if you want row statistics, you put the full row variable descriptions into $\diamond$. For example, *PCTN<age all>* will produce percentage cross columns. Similarly, *PCTN<gender all>* or *PCTN<race all>* produces row percentage because *gender all* and *race all* are for rows. But the output for A*B might not look nice sometimes. For example, *(age all)*PCTN<gender all>* in our case will generate a table which is hard to be read. But *(age all)*PCTN<race all>* in our case will generate a table which is easier to be read because *(race all)* is within *(gender all)*;
4. The full description implies if you use 'all' to those variable(s) in your table statement, 'all' becomes a non-separable part of the variable(s) descriptions used for your percentage generation. You can NOT eliminate 'all' in you PCTN $\diamond$ if 'all' is part of your statistics for a class variable. For example *(age all)*PCTN<gender>* will not work in our example because 'all' is part of the statistics for *gender*. Similarly, you can not use *PCTN<gender*race>*;
5. You can NOT put () into PCTN $\diamond$. *PCTN<(gender all)*(race all)>* will give you an error message. Since () is not allowed in $\diamond$, you may not have form like (A ALL)*(B ALL) or A*(B ALL) in your percentage expression. However, you can have A ALL, or B ALL, or A*B ALL expressions in $\diamond$. In our example, we can print following percentage: *(age all)*PCTN<age all>*, or *(age all)*PCTN<gender all>* or *(age all)*PCTN<race all>*. Although the statement of A*B ALL is syntax correct, the percentage generated by A*B ALL is not the same as (A ALL)*(B ALL).

Now, we show you how to generate both row and column percentage by above rules. We still use the same data. The PROC TABULATE used here is:

```
proc tabulate data=test;
class gender race age;
format gender gender. race ethnic. age age.;
table (gender=' ' all='Total'),
(age='Age' all)*(N='#'*f=2.0
PCTN<age all>='C %'*f=4.1
PCTN<gender all>='R %'*f=4.1)/Box='Sample Table' RTS=10;
title 'Percentage for Columns and Rows';
run;
```

It generates a table below. Here, we only show you part of the output table where C % means column percentage and R % means row percentage.

Percentage for Columns and Rows

Sample Table	Age											
	0-10			11-13			14-15			16		
	#	C %	R %	#	C %	R %	#	C %	R %	#	C %	R %
Male	6	11.8	75.0	6	11.8	37.5	18	35.3	69.2	12	23.5	63.2
Female	2	6.7	25.0	10	33.3	62.5	8	26.7	30.8	7	23.3	36.8
Total	8	9.9	100	16	19.8	100	26	32.1	100	19	23.5	100

4. Analysis Variables and PCTSUM

In many cases analysis variable statistics are needed within class variables defined in PROC TABULATE. In our example, what if we want the average age by gender and ethnic group. It is true the variable age is a numerical variable. But after it is defined as a class variable, you can NOT define it as an analysis variable. If you try to define like:

```
class age;
var age;
```

You will receive an error message. You may have to use a data step to generate a new variable, such as *ages=age*, then uses *ages* as an analysis variable for your PROC TABULATE.

When you use an analysis variable, you must know the difference between "N" and "SUM", PCTN and PCTSUM

- 'N' counts numbers while 'SUM' adds numbers;
- If the variable is an 1 or 0 variable => you can get 'N' when use 'SUM';
- PCTSUM generates the percentage for the sum of an analysis with a class. Suppose there are 2 boys with ages of 4 and 6 respectively and two girls with ages of 5 and 10 respectively. If we assume *gender* is class variable and *ages* is an analysis variable, the statement of *gender*ages*PCTN<gender>* will print 50% and 50% while the statement of *gender*ages*PCTSUM<gender>* will print 40% and 60%.

You also must know that for an analysis variable, PROC TABULATE can only generate its percentage directly within a class variable or class variables. If there are two analysis variables, you can NOT generate percentage between them directly. For example, if F, M, and S are the numbers of the female, male, and total juveniles respectively and they are analysis variables, we can not use PROC TABULATE to generate the percentage for the

female among the total. This is important for you to design your SAS data. You may need to convert some of your analysis variables into class variables. When percentage is wanted in your table, please do not have too many analysis variables. Otherwise, you have to calculate those percentages in SAS step before using the PROC TABULATE.

The following example show you how to generate the Type 3 report:

```
data test;set test1;ages=age; /* generate a new dataset with ages=age */
proc tabulate data=test;
class gender race age;
var ages; /* ages: an analysis variable */
format gender gender. race ethnic. age age.;
table (gender=' ' all='Total')*(race=' ' all='Total'),
ages='Age'*MEAN='Ave.'*f=4.1 /* a single column */
(age='Age' all)*(ages=' '*MEAN='Mean Age'*f=4.1
ages=' '*PCTSUM<age all>='% of Age Sum'*f=4.1
ages=' '*N='#'*f=2.0
PCTN<gender all>='% '*f=4.1)/Box='Sample Table' RTS=19;
title 'Percentage of PCTN and PCTSUM';
run;
```

A proportion of the output report is showed below:

Sample Table		Age											
		0-10				11-13				14-			
		Age	% of			Age	% of			Age	% of		
		Mean	Age			Mean	Age			Mean	Age		
		Ave.	Age	Sum	#	%	Age	Sum	#	%	Age	Sum	
Male	White	14.8	10.0	7.5	3	60.0	11.0	8.3	3	33.3	15.0	22.6	
	Black	14.6	10.0	13.7	3	100	*	*	*	*	15.0	41.1	
	Other	13.0	*	*	*	*	11.0	28.2	3	75.0	14.0	71.8	
	Total	14.4	10.0	8.2	6	75.0	11.0	9.0	6	37.5	14.7	35.9	
Female	White	13.8	10.0	8.5	2	40.0	11.8	30.3	6	66.7	15.0	25.6	

5. Self-Checking

The following questions are designed for your self-checking to see if you understand PROC TABULATE. Suppose a, b, c, d, e, and x are either class variables or analysis variables:

- In the statement of *'table x, a*b*c, d*e'*, how many dimensions do we have for the table? Which one is for pages-expression, row-expression, or column expressions?
- Please draw a graph for (a ALL)*(b ALL)*(c ALL) vs (a*b*c ALL). You may assume several values for a, or b, or c.
- If year is 1992 to 1996, and the dataset has no missing value for each year, how many tables can be generated when you use *'by year'* statement in PROC TABULATE?
- When you use *KEYLABEL SUM=' ' N=' '* what will be printed for the label *SUM* and *N*?
- What will be printed by *a='Salary'*N='Count'*f=5.2*? What is the depth in this expression? How wide each column is?
- When a macro variable is used in PROC TABULATE in the title statement, which one in the following statements will replace the macro variable:
title '&x County 1996 Report';
title "&x County 1996 Report";
- If a, b, c are three class variables with 2, 3, 4 levels respectively, how many rows and columns are we going to generate in the following statement (draw the picture please):
a, b c
a*b, c
a, b*c
a*b*c
- What does *BOX=' ' print?*
- If *RTS= 17* for a*b (two class variables), how wide should each label be?
- Please explain the following:
proc tabulate;
data=test format=6.1 missing noseps formchar='*';
- Please explain the following:
CONDENSE in a TABLE statement
- Please explain *FREQ* (number of observations, one variable) in PROC TABULATE.
- Show one example for the following requirements:
How to use *PCTN* (count)?
How to use *PCTSUM* (sum)?
How to determine the denominator for the row percentage or column percentage?