

Paper 77-27

What's Your `_TYPE_`? How to find the CLASS You Want in PROC SUMMARY

Frank Ferriola, Highlands Ranch, CO

ABSTRACT

When using the CLASS statement in your PROC SUMMARY, SAS® will calculate every possible combination of the CLASS variables. By understanding the way SAS® summarizes, you can find a simple way to determine which summaries you really want to keep and eliminating the others.

INTRODUCTION

I have always been fascinated by number patterns. When I was about 8 years old, my father showed me a "Magic Square", a 3 x 3 or odd numbered square in which you could place the numbers 1 through 9 and have all rows, across, down and diagonally equal the same number. It's easy to memorize the answer for a 3x3 square:

8	1	6
3	5	7
4	9	2

But how do you memorize a 5x5, 7x7 or even a 99x99 square? There are many people who can spout complicated algorithms and formulas to compute the answer, however, my father showed me a pattern with simple rules to fill out the square under any circumstances. This makes the solution easy if not impressive. I'd tell you the pattern but then you'd probably stop reading the paper and start trying larger squares.

In the same way, SAS® uses a simple binary pattern to summarize the CLASS statement in PROC SUMMARY and other PROCs. By knowing this pattern, you can simplify your output to just the summaries you need cutting down on paper and processing time.

If the user is looking only for the summary with all the CLASS variables this can be done using the NWAY option in the PROC SUMMARY statement:

```
PROC SUMMARY NWAY;
  CLASS A B C D;
  VAR TOTAL;
  OUTPUT OUT =NWAYSUMM
  SUM=SUMTOTAL;
RUN;
```

You will also get the same result with the BY statement in a summary preceded by a PROC SORT;

```
PROC SORT;
  BY A B C D;
RUN;

PROC SUMMARY;
  BY A B C D;
  VAR TOTAL;
  OUTPUT OUT =BYSUMM
  SUM=SUMTOTAL;
RUN;
```

Many times you need more than one summary. On these occasions, you can easily do one PROC SUMMARY and retain only the summaries you need from that one procedure.

UNDERSTANDING `_TYPE_`

To understand how `_TYPE_` works in SAS® let's use a simple PROC SUMMARY:

```
PROC SUMMARY;
  CLASS A;
  VAR TOTAL;
  OUTPUT OUT = ONEVAR
  SUM=SUMTOTAL;
```

```
RUN;
```

This summary will produce 2 summaries and assign `_TYPE_` as follows:

```
_TYPE_ = 0 All records
_TYPE_ = 1 Records summarized by A.
```

Now we can expand it by one variable.

```
PROC SUMMARY;
  CLASS A B;
  VAR TOTAL;
  OUTPUT OUT = TWOVAR
  SUM=SUMTOTAL;
RUN;
```

In this way we double the number of `_TYPE_'s`:

```
_TYPE_ = 0 All records
_TYPE_ = 1 Records summarized by B only.
_TYPE_ = 2 Records summarized by A only.
_TYPE_ = 3 Records summarized by both A and B.
```

Now add a 3rd Variable:

```
PROC SUMMARY;
  CLASS A B C;
  VAR TOTAL;
  OUTPUT OUT = THREEVAR
  SUM=SUMTOTAL;
RUN;
```

Once again we double the number of `_TYPE_'s`:

```
_TYPE_ = 0 All records
_TYPE_ = 1 Records summarized by C only.
_TYPE_ = 2 Records summarized by B only.
_TYPE_ = 3 Records summarized by both B and C.
_TYPE_ = 4 Records summarized by A only.
_TYPE_ = 5 Records summarized by A and C only.
_TYPE_ = 6 Records summarized by A and B only.
_TYPE_ = 7 Records summarized by A and B and C.
```

If you are following along you should start seeing the pattern.

Let's add one more variable:

```
PROC SUMMARY;
  CLASS A B C D;
  VAR TOTAL;
  OUTPUT OUT = FOURVAR
  SUM=SUMTOTAL;
```

```
RUN;
```

Once again we double the number of `_TYPE_'s`:

```
_TYPE_ = 0 All records
_TYPE_ = 1 Records summarized by D only.
_TYPE_ = 2 Records summarized by C only.
_TYPE_ = 3 Records summarized by both C and D.
_TYPE_ = 4 Records summarized by B only.
_TYPE_ = 5 Records summarized by B and D only.
_TYPE_ = 6 Records summarized by B and C only.
_TYPE_ = 7 Records summarized by B, C and D.
_TYPE_ = 8 Records summarized by A only.
_TYPE_ = 9 Records summarized by A and D.
_TYPE_ = 10 Records summarized by A and C.
_TYPE_ = 11 Records summarized by A, C, and D.
_TYPE_ = 12 Records summarized by A and B.
_TYPE_ = 13 Records summarized by A, B, and D.
_TYPE_ = 14 Records summarized by A, B and C.
_TYPE_ = 15 Records summarized by A, B, C and D.
```

THE BASIC FORMULA

From this pattern we can determine a formula:

- 1 Variable = 2 Summaries = 2^1
 - 2 Variables = 4 Summaries = 2^2
 - 3 Variables = 8 Summaries = 2^3
 - 4 Variables = 16 Summaries = 2^4
 - 5 Variables = 32 Summaries = 2^5
 - 6 Variables = 64 Summaries = 2^6
- or $S=2^n$

where S = # of Summaries and n = number of variables in the CLASS Statement.

```
PROC SUMMARY;
* _TYPE_ 32 16 8 4 2 1;
CLASS A B C D E F;
VAR TOTAL;
OUTPUT OUT = WHICHSUM
SUM=SUMTOTAL;
RUN;
```

Remember PROC SUMMARY will summarize CLASS from right to left, so we assign the value of the variable by itself F=1, E=2, D=4, C=8, B=16 A=32. Also remember that using no variables (total of data) is always 0. Using all variables will be 2 times the left most variable - 1. In this case 63.

Figure 1

Obs	Year	Month	Category	TYP	_TYPE_	_FREQ_	net
1	.	.			0	1499	2,003.39
2	.	.		ATM	1	63	-7,736.00
3	.	.		BILLPAY	1	321	-81,704.52
4	.	.		CREDIT	1	203	1,403.50
5	.	.		DEBIT	1	381	-14,002.10
6	.	.		DEPOSIT	1	332	109,900.38
7	.	.		EFT	1	27	-10,346.56
8	.	.		OTHER	1	161	3,104.99
9	.	.		XFER	1	11	1,383.70
10	.	.	A/R	Expense	2	46	-7,102.16
11	.	.	Income	Income	2	1397	-83,906.89
12	.	.	Savings	Savings	2	50	97,012.44
13	.	.	A/R	Expense	2	6	-4,000.00
14	.	.	BILLPAY	BILLPAY	3	2	-1,097.94
15	.	.	CREDIT	CREDIT	3	6	-275.07
16	.	.	DEBIT	DEBIT	3	3	-140.60
17	.	.	DEPOSIT	DEPOSIT	3	29	8,960.61
18	.	.	OTHER	OTHER	3	5	-15,549.16
19	.	.	XFER	XFER	3	1	1,000.00
20	.	.	ATM	Expense	3	63	-7,736.00
21	.	.	BILLPAY	Expense	3	319	-80,606.58
22	.	.	CREDIT	Expense	3	197	1,678.57
23	.	.	DEBIT	Expense	3	378	-13,861.50
24	.	.	DEPOSIT	Expense	3	253	3,927.33
25	.	.	EFT	Expense	3	27	-10,346.56
26	.	.	OTHER	Expense	3	155	17,654.15
27	.	.	XFER	Expense	3	5	5,383.70
28	.	.	DEPOSIT	Income	3	50	97,012.44
29	.	.	OTHER	Savings	3	1	1,000.00
30	.	.	XFER	Savings	3	5	-5,000.00
31	.	1			4	163	1,429.69
32	.	2			4	128	1,211.79
33	.	3			4	175	-3,226.03
34	.	4			4	139	3,506.50
35	.	5			4	183	-502.71
36	.	6			4	135	-3,333.46
37	.	7			4	152	-723.04
38	.	8			4	149	4,343.47
39	.	9			4	101	-3,914.27

We also find that PROC SUMMARY generates the summary variables from right to left in the CLASS statement.

So, how do we tell which summaries we want?

Let's look at the following example:

```
PROC SUMMARY;
CLASS A B C D E F;
VAR TOTAL;
OUTPUT OUT = WHICHSUM
SUM=SUMTOTAL;
RUN;
```

Now we want to find the summary _TYPE_ number for variables B, D & F. We can use the formula above to insert a comment line to help us:

So how do we determine the summary for B D & F. From the comment line we take

$(B=16) + (D=4) + (F=1)$

or $16 + 4 + 1 = 21$

So _TYPE_ = 21 would be summarized by B,D&F.

Similarly, A,C,E,F would be equal $32 + 16 + 2 + 1 = 51$.

Once you decide which _TYPE_'s you want you can print the shortened list with a where clause.

```
PROC PRINT DATA=WHICHSUM;
Where _TYPE_ in (0,21,51,63);
RUN;
```

A PRACTICAL EXAMPLE

The first table (Figure 1) shows an example of PROC SUMMARY output. The example is data from a checkbook program and is the first page of a longer summary. Looking at the `_TYPE_` column the first observation shows a 0, which is the total of all records in the file. Each subsequent `_TYPE_` will have all its observations add up to this `_FREQ_` and `NET`. For example, the 8 records for `_TYPE_=1` (sorted by `TYP`), will add back to 1,499 records and \$2,003.39.

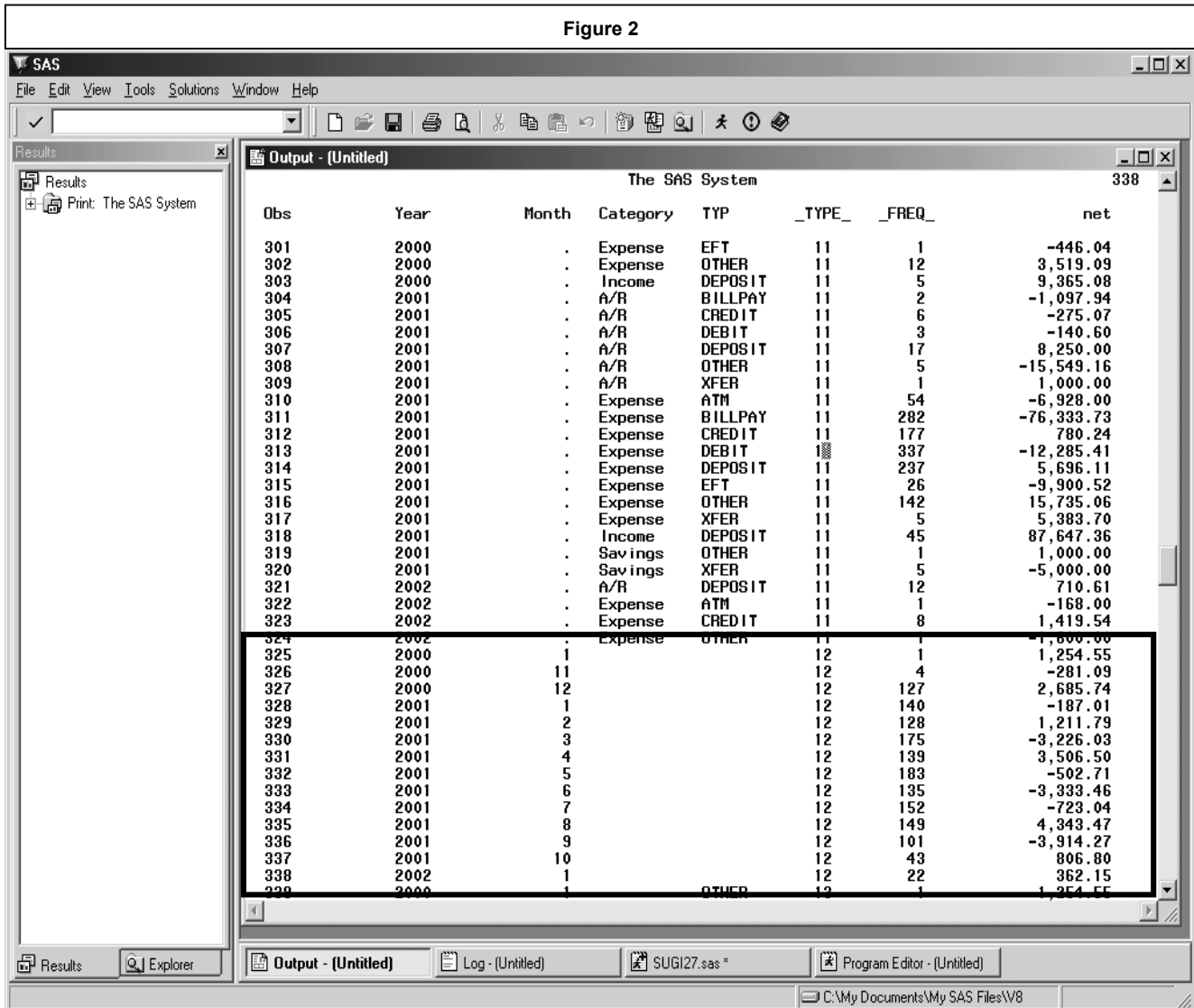
You will also note how the table goes from right to left, as the `_TYPE_` increases the table will "move" to the left by `CLASS` variable.

CONCLUSION

Understanding PROC SUMMARY and how it works can make it easier when summarizing large data sets. By knowing the number of summaries your PROC SUMMARY creates you can determine which summaries you want to keep and which ones will be of no use to you.

The formula for total summaries is $S=2^n$ where S = number of summaries and n =number of class variables.

This method will also work with PROC MEANS and other SAS Procedures, which use the `CLASS` Statement.



The table was created with the following code:

```
proc summary;
  * _TYPE_ = 8 4 2 1 ;
  class year month category typ;
  var net;
  output out=sumcheck sum=;
run;
```

In order to find the total number of summaries we would use the formula $S=2^n$ or $s=2^4$ or 16. To find a particular summary such as Year and Month we would add $8 + 4$ and see that the Summary is 12. In Figure 2 you will see the output for `_TYPE_=12` in the highlighted area.

ACKNOWLEDGMENTS

I would like to thank the following people:

Doug Felton for encouraging me to write this paper;

Nikki Carroll and Denver SAS Users Group for allowing me to present it;

Joe Leoni of Blue Cross Blue Shield of Delaware who saw the potential of SAS® in me in 1984; and

my father, Frank Ferriola, Sr. who encouraged me as a child in my fascination with numbers and logic and who now encourages it in my children.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author at:

Frank Ferriola

3110 W. Deer Creek Dr.

Highlands Ranch, CO 80129

Phone: (303) 683-7340

Email: fafferriola@yahoo.com

Website and Magic Square Solution:

www.geocities.com/fafferriola

SAS is and all SAS Institute, Inc. product or service names are registered trademarks or trademarks of SAS Institute, Inc in the United States or other countries. ® indicates USA registration.