Paper BB-12

PROC SUMMARY Options Beyond the Basics

Susmita Pattnaik, PPD Inc, Morrisville, NC

ABSTRACT

PROC SUMMARY is used for summarizing the data across all observations and is familiar to most SAS® users. This paper will discuss some of the options less commonly used that could add values and can save some programming time when used in full potential.

INTRODUCTION

This paper outlined the options of the summary procedure in two places. PROC SUMMARY statement options and OUTPUT statement options. This paper discusses less commonly used, but very powerful options throughout a series of examples.

The basic syntax of the PROC SUMMARY is shown below in the table 1. For the full syntax and list of keywords, please refer to the SAS documentation.

The core statements used in this paper:

CLASS: Identify the variables that will be sub-grouped for the analysis.

VAR: Identify the variable that analysis will be done and the order of the results in the output dataset.

OUTPUT: Outputs to a dataset. Also, this is the place where the type of statistics can be selected and named. The default statistics will be produced N MEAN STD MIN MAX.

This paper used the CLASS dataset from the SASHELP throughout the examples.

```
/* Table 1: Basic Summary */

proc summary data = sashelp.class(where=(age in (12,14,15)));
  class sex age;
  var height;
  output out=mysumm1 n=hgt_n mean=hgt_mean std=hgt_std;
run;
```

Table 1: Basic Summary

Obs	Sex	Age	_TYPE_	_FREQ_	hgt_n	hgt_mean	hgt_std
1		.	0	13	13	63.0231	3.93259
2		12	1	5	5	59.4400	3.29742
3		14	1	4	4	64.9000	2.80119
4		15	1	4	4	65.6250	2.09662
5	F	.	2	6	6	62.0333	3.57081
6	M	.	2	7	7	63.8714	4.29950
7	F	12	3	2	2	58.0500	2.47487
8	F	14	3	2	2	63.5500	1.06066
9	F	15	3	2	2	64.5000	2.82843
10	M	12	3	3	3	60.3667	3.93234
11	M	14	3	2	2	66.2500	3.88909
12	M	15	3	2	2	66.7500	0.35355

NWAY Option

Usually PROC SUMMARY outputs statistics for the combinations that are available in the input dataset as well as the grand total. If we just want to see only the summarization for both SEX and AGE combinations then we can use NWAY option that is equivalent to the highest value of _TYPE_ variable which is 3 as displayed in the Table 2.

```
/*Table 2: NWAY option displays the output of only the rows that described in
the class statement - highest value of _TYPE_ variable*/

proc summary data = sashelp.class nway;
  class sex age;
  var height;
  output out=mysumm2 n=hgt_n mean=hgt_mean std=hgt_std;
run;
```

Table 2: USING NWAY OPTION

Obs	Sex	Age	_TYPE_	_FREQ_	hgt_n	hgt_mean	hgt_std
1	F	11	3	1	1	51.3000	.
2	F	12	3	2	2	58.0500	2.47487
3	F	13	3	2	2	60.9000	6.22254
4	F	14	3	2	2	63.5500	1.06066
5	F	15	3	2	2	64.5000	2.82843
6	M	11	3	1	1	57.5000	.
7	M	12	3	3	3	60.3667	3.93234
8	M	13	3	1	1	62.5000	.
9	M	14	3	2	2	66.2500	3.88909
10	M	15	3	2	2	66.7500	0.35355
11	M	16	3	1	1	72.0000	.

CHARTYPE Option:

The _TYPE_ is an integer variable directly related to CLASS statement. Without the NWAY option, PROC SUMMARY will calculate the subtotal for every combination of classification variables, including the grand total, where _TYPE_ = 0. Other values of _TYPE_ may be interpreted by imagining the binary translation of the _TYPE_ number. To avoid the mental binary conversion confusion, there is an option to simplify it. Adding option CHARTYPE results in table 3 containing the same 20 rows, but _TYPE_ column is now a bit string (the binary value of _TYPE_ integer in the previous step).

Examine the CLASS statement again: since there are 2 classification variables SEX and AGE in our examples, the bit string values contain 2 digits. There are 4 different _TYPE_ values, counting the 0 total. Each _TYPE_ includes one or more of the CLASS variables. The bit string will contain a 1 for each variable is involved in this row, 0 if omitted. Total is the exception, being equal to 00. Remember NWAY, the combination of all two variables? It wrote out only the rows where _TYPE_ = 11 = 3. Let's examine a few more _TYPE_ values. What is the _TYPE_ for the combination of SEX and ETHNIC if the class statement has SEX AGE ETHNIC? In bit string, that would be 101 = 5. We can use CHARTYPE if we want to see the bit string in _TYPE_ for ease of understanding and avoiding mental binary contortions.

```
/* Table 3: CHARTYPE option specifies that the _TYPE_ variable in the output
dataset is a character representation of the binary value of _TYPE_*/

proc summary data = sashelp.class chartype;
  class sex age ethnic;
  var height;
  output out=mysumm3 n=hgt_n mean=hgt_mean std=hgt_std;
run;
```

Table 3: USING CHARTYPE OPTION

Obs	Sex	Age	_TYPE_	_FREQ_	hgt_n	hgt_mean	hgt_std
1		.	00	19	19	62.3368	5.12708
2		11	01	2	2	54.4000	4.38406
3		12	01	5	5	59.4400	3.29742
4		13	01	3	3	61.4333	4.49592
5		14	01	4	4	64.9000	2.80119
6		15	01	4	4	65.6250	2.09662
7		16	01	1	1	72.0000	.
8	F	.	10	9	9	60.5889	5.01833
9	M	.	10	10	10	63.9100	4.93794
10	F	11	11	1	1	51.3000	.
11	F	12	11	2	2	58.0500	2.47487
12	F	13	11	2	2	60.9000	6.22254
13	F	14	11	2	2	63.5500	1.06066
14	F	15	11	2	2	64.5000	2.82843
15	M	11	11	1	1	57.5000	.
16	M	12	11	3	3	60.3667	3.93234
17	M	13	11	1	1	62.5000	.
18	M	14	11	2	2	66.2500	3.88909
19	M	15	11	2	2	66.7500	0.35355
20	M	16	11	1	1	72.0000	.

COMPLETETYPES Option

In the table 4, we see that there are some combinations of classification variables that have not produced a row, since that combination of data values did not occur in our raw data table 4. There is no row of _TYPE_ = 011 with AGE = 11 and RACE = ‘African American’ or RACE = ‘Asian’, for example. Suppose we want rows for all possible combinations. Adding option COMPLETETYPES takes all possible combinations of class variables including those with 0 counts. Table 5 shows all 3 possible rows with zero counts.

```
/* Table 4: */

data mytbl;
  set sashelp.class;
  if 11<= age <= 13 then race = 'Native American ';
  else if 13 < age <= 15 then race = 'African American';
  else race = 'Asian';
run;

proc summary data = mytbl_1(where=(age in (11,12))) chartype;
  class sex age race;
  var height;
  output out=mysumm4 n=hgt_n mean=hgt_mean std=hgt_std;
run;
```

Table 4: Without_ Completetypes

Obs	Sex	Age	race	_TYPE_	_FREQ_	hgt_n	hgt_mean	hgt_std
1		.		000	7	7	58.0000	4.06202
2		.	Native American	001	7	7	58.0000	4.06202
3		11		010	2	2	54.4000	4.38406
4		12		010	5	5	59.4400	3.29742
5		11	Native American	011	2	2	54.4000	4.38406
6		12	Native American	011	5	5	59.4400	3.29742
7	F	.		100	3	3	55.8000	4.27200
8	M	.		100	4	4	59.6500	3.51615
9	F	.	Native American	101	3	3	55.8000	4.27200
10	M	.	Native American	101	4	4	59.6500	3.51615
11	F	11		110	1	1	51.3000	.
12	F	12		110	2	2	58.0500	2.47487
13	M	11		110	1	1	57.5000	.
14	M	12		110	3	3	60.3667	3.93234
15	F	11	Native American	111	1	1	51.3000	.
16	F	12	Native American	111	2	2	58.0500	2.47487
17	M	11	Native American	111	1	1	57.5000	.
18	M	12	Native American	111	3	3	60.3667	3.93234

```
 /* Table 5: COMPLETETYPE option takes all possible combinations of class
variables even if the combination does not occur in the input dataset */

proc summary data = mytbl CHARTYPE COMPLETETYPES;
  class sex age race;
  var height;
  output out=mysumm4_2 n=hgt_n mean=hgt_mean std=hgt_std;
run;
```

Table 5: With_Completetypes

Obs	Sex	Age	race	_TYPE_	_FREQ_	hgt_n	hgt_mean	hgt_std
1		.		000	18	18	61.9667	5.00764
2		.	African American	001	7	7	64.7286	1.90675
3		.	Asian	001	1	1	72.0000	.
4		.	Native American	001	10	10	59.0300	4.27110
5		11		010	2	2	54.4000	4.38406
6		12		010	5	5	59.4400	3.29742
7		13		010	3	3	61.4333	4.49592
8		14		010	3	3	63.5333	0.75056
9		15		010	4	4	65.6250	2.09662
10		16		010	1	1	72.0000	.
11		11	African American	011	0	0	.	.
12		11	Asian	011	0	0	.	.
13		11	Native American	011	2	2	54.4000	4.38406

PRINTIDVARS Option

The Table 6 shows the output of the ID variable HEIGHT in the displayed format. By default, it displayed the highest values by the sub-group SEX. 72.0 is the highest value of HEIGHT in the male group and 66.5 is the from the female group. Use this option in the PROC statement to include the value of the ID variable in the displayed output.

```
/*Table 6: PRINTIDVARS: Displays the values of the ID variables in printed
output*/
```

```
proc summary data = sashelp.class printidvars;
  class sex;
  var age;
  id height;
  output out=mysumm6;
run;
```

Table 6: USING PRINTIDVARS OPTION

Obs	Sex	Height	_TYPE_	_FREQ_	hgt_n	hgt_mean
1		72.0	0	19	19	13.3158
2	F	66.5	1	9	9	13.2222
3	M	72.0	1	10	10	13.4000

The SUMMARY Procedure

Analysis Variable: Age

Sex	Height	N Obs	N	Mean	Std Dev	Minimum
F	66.5	9	9	13.2222222	1.3944334	11.0000000
M	72	10	10	13.4000000	1.6465452	11.0000000

MAXID and MINID

In table 7, I want to find out the names that have the highest and lowest heights. Using the MAXID and MINID options on the OUTPUT statement, name the variable HEIGHT that contains the value I want to see to identify the highest/ lowest values for that value of the CLASS variable: You will notice two new variables “maxhtname” and “minhtname” in the OUTPUT statement have been used to capture the values of the ID variable for the maximum and minimum values. Philip and Joyce have the respective 72.0 and 51.3
Note that if there are ties either in the extreme and least values, then the id names (observation) will be pulled alphabetically and will have only one record.

```
/* Table 7: MAXID and MINID: are the options in the OUTPUT statement to find
out the variable that contains highest and lowest values by identifying the
observation that contains the extreme and least values*/

proc summary data = sashelp.class;
  class sex age;
  var height;
  output out=mysumm5
       mean=hgt_mean
       max=hgt_max
       min=hgt_min
       maxid(height(name))=maxhtname
         minid(height(name))=minhtname;
run;
```

Table 7: USING MAXID/MINID OPTIONS

Obs	Sex	Age	_TYPE_	_FREQ_	hgt_mean	hgt_max	hgt_min	maxhtname	minhtname
1		.	0	19	62.3368	72.0	51.3	Philip	Joyce
2		11	1	2	54.4000	57.5	51.3	Thomas	Joyce
3		12	1	5	59.4400	64.8	56.3	Robert	Louise
4		13	1	3	61.4333	65.3	56.5	Barbara	Alice
5		14	1	4	64.9000	69.0	62.8	Alfred	Carol
6		15	1	4	65.6250	67.0	62.5	Ronald	Janet
7		16	1	1	72.0000	72.0	72.0	Philip	Philip
8	F	.	2	9	60.5889	66.5	51.3	Mary	Joyce
9	M	.	2	10	63.9100	72.0	57.3	Philip	James
10	F	11	3	1	51.3000	51.3	51.3	Joyce	Joyce
11	F	12	3	2	58.0500	59.8	56.3	Jane	Louise
12	F	13	3	2	60.9000	65.3	56.5	Barbara	Alice
13	F	14	3	2	63.5500	64.3	62.8	Judy	Carol
14	F	15	3	2	64.5000	66.5	62.5	Mary	Janet
15	M	11	3	1	57.5000	57.5	57.5	Thomas	Thomas
16	M	12	3	3	60.3667	64.8	57.3	Robert	James
17	M	13	3	1	62.5000	62.5	62.5	Jeffrey	Jeffrey
18	M	14	3	2	66.2500	69.0	63.5	Alfred	Henry
19	M	15	3	2	66.7500	67.0	66.5	Ronald	William
20	M	16	3	1	72.0000	72.0	72.0	Philip	Philip

AUTONAME Option

Suppose we need more than the just one statistic for each of our analytical variable, like SUM and MEAN. To labels the column easily, use AUTONAME option. Table 8 shows our 20 rows with two columns of averages “MEAN =” added to the two columns of totals “SUM =”. AUTONAME labels each column with *varname_statistic* for us. Otherwise, we could code each column name as we wished.

```
/* Table 8: AUTONAME: */

proc summary data = sashelp.class;
  class sex age;
  var height weight;
  output out = mysumm7
  sum=MEAN=/AUTONAME;
run;
```

Table 8: USING AUTONAME OPTION

Obs	Sex	Age	_TYPE_	_FREQ_	Height_Sum	Weight_Sum	Height_Mean	Weight_Mean
1		.	0	19	1184.4	1900.5	62.3368	100.026
2		11	1	2	108.8	135.5	54.4000	67.750
3		12	1	5	297.2	472.0	59.4400	94.400
4		13	1	3	184.3	266.0	61.4333	88.667
5		14	1	4	259.6	407.5	64.9000	101.875
6		15	1	4	262.5	469.5	65.6250	117.375
7		16	1	1	72.0	150.0	72.0000	150.000
8	F	.	2	9	545.3	811.0	60.5889	90.111
9	M	.	2	10	639.1	1089.5	63.9100	108.950
10	F	11	3	1	51.3	50.5	51.3000	50.500
11	F	12	3	2	116.1	161.5	58.0500	80.750
12	F	13	3	2	121.8	182.0	60.9000	91.000
13	F	14	3	2	127.1	192.5	63.5500	96.250
14	F	15	3	2	129.0	224.5	64.5000	112.250
15	M	11	3	1	57.5	85.0	57.5000	85.000
16	M	12	3	3	181.1	310.5	60.3667	103.500
17	M	13	3	1	62.5	84.0	62.5000	84.000
18	M	14	3	2	132.5	215.0	66.2500	107.500
19	M	15	3	2	133.5	245.0	66.7500	122.500
20	M	16	3	1	72.0	150.0	72.0000	150.000

LEVELS and WAYS Options

Using these options will add more flexibility to the look and feel of the summarization. These two options create two variables _LEVEL_ and _WAY_ in the output dataset.
LEVEL variable contains a value from 1 to n which specifies a unique combination of the values of CLASS (_TYPE_) variables.
WAY variable contains a value from 1 to the number of variables present in the CLASS statement.
You can see in the table 9 which created these two variables. _LEVEL_ counted to 1 when _TYPE_ = 0, counted to 6 when _TYPE_ = 1, and so on. _WAY_ counted up to 2 since we have two variables in the CLASS statement.

```
/* Table 9: WAYS and LEVELS Options */

proc summary data = sashelp.class;
  class sex age;
  var height;
  output out = mysumm9 mean=hgt_mean /levels ways;
run;
```

Table 9: USING LEVELS and WAYS OPTION

Obs	Sex	Age	_WAY_	_TYPE_	_LEVEL_	_FREQ_	hgt_mean
1		.	0	0	1	19	62.3368
2		11	1	1	1	2	54.4000
3		12	1	1	2	5	59.4400
4		13	1	1	3	3	61.4333
5		14	1	1	4	4	64.9000
6		15	1	1	5	4	65.6250
7		16	1	1	6	1	72.0000
8	F	.	1	2	1	9	60.5889
9	M	.	1	2	2	10	63.9100
10	F	11	2	3	1	1	51.3000
11	F	12	2	3	2	2	58.0500
12	F	13	2	3	3	2	60.9000
13	F	14	2	3	4	2	63.5500
14	F	15	2	3	5	2	64.5000
15	M	11	2	3	6	1	57.5000

16	M	12	2	3	7	3	60.3667
17	M	13	2	3	8	1	62.5000
18	M	14	2	3	9	2	66.2500
19	M	15	2	3	10	2	66.7500
20	M	16	2	3	11	1	72.0000

CONCLUSIONS

PROC SUMMARY is a very powerful SAS/BASE procedure. With the use of numeric variables, options, statements, and formats, this procedure can perform very complex multidimensional data summarizations using one-way, two-way, all-way via the automatic _TYPE_ variable. This paper has presented a few options to refresh the readers. Understanding PROC SUMMARY and its use can make it easier when summarizing the large data sets.

REFERENCES

ACKNOWLEDGEMENTS

I would like to thank the following people.

Ken Borowiak, Kim Sturgen for encouragement, review and inputs on this paper.
Jin Liang and Joshua Hunter for their reviews and inputs on this paper.
Harry Droogendyk and SESUG for allowing me to present it.

DISCLAIMER

The content of this paper are the work of the author and do not necessarily represent the opinions, recommendations, or practices of PPD, Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged

Susmita Pattnaik
PPD, Inc.
3900 N Paramount Parkway
Susmita.Pattnaik@ppdi.com
Susmitapattnaik123@gmail.com