

PROC Tabulate Basics

Prepared by



Global Business Intelligence
Consulting and Training

Destiny Corporation
100 Great Meadow Rd Suite 601
Wethersfield, CT 06109-2379
Phone: (860) 721-1684 1-800-7TRAINING Fax:
(860) 721-9784
Email: info@destinycorp.com
Web: www.destinycorp.com
Copyright 2003

Introduction

Proc Tabulate provides a different method for presenting data reports.

Instead of a listing structure familiar with Proc Print, Proc SQL, and Proc Report, Proc Tabulate organizes the data summarization in a table structure.

Proc Tabulate is one of the most powerful of SAS procedures. Its syntax allows the programmer access to a wide variety of simple and complex tables.

Basic Proc Tabulate Syntax

Proc Tabulate requires the following two statements in addition to the Proc and Run statements:

- Class statement – identifies which variables are used to classify the data
- Table statement – tells Proc Tabulate how to construct the tabular output

Submit the following code.

```
Command ==>
00001 proc tabulate data=saved.demog;
00002   class status;
00003   table status;
00004 run;
```

- The Class statement defines status as a Class variable.
- The Table statement constructs the table.
- The variable referenced in the Table statement must be defined as a Class variable (see above) or Var variable (introduced later).

STATUS				
D	H	P	S	U
N	N	N	N	N
6.00	27.00	4.00	10.00	2.00

The simple syntax creates a default table and statistic (N).

```
Command ==>
00001 proc tabulate data=saved.demog;
00002   class status gender grade;
00003   table status;
00004   table gender;
00005   table grade;
00006 run;
```

STATUS				
D	H	P	S	U
N	N	N	N	N
6.00	27.00	4.00	10.00	2.00

GENDER	
F	M
N	N
22.00	27.00

GRADE	
high	low
N	N
23.00	26.00

The Blank to Assign Table Structure

In the Table statement, the Blank is used to designate that tables should be concatenated at the side.

```
Command ==>
00001 proc tabulate data=saved.demog;
00002   class gender grade;
00003   table gender grade;
00004 run;
```

Visualizing the output, two tables – Gender and Grade – are simply joined at the sides.

GENDER		GRADE	
F	M	high	low
N	N	N	N
22.00	27.00	23.00	26.00

Additional examples show the concatenation.

```
Command ==>
00001 proc tabulate data=saved.demog;
00002   class gender grade status;
00003   table gender grade;
00004   table status grade;
00005   table grade status;
00006 run;
```

GENDER		GRADE		STATUS	
F	M	high	low	D	H
N	N	N	N	N	N
22.00	27.00	23.00	26.00	6.00	27.00

STATUS						GRADE	
D	N	P	S	M		high	low
8.00	27.00	4.00	10.00	2.00		23.00	26.00

GRADE		STATUS					
high	low	D	N	P	S	M	
23.00	26.00	8.00	27.00	4.00	10.00	2.00	

The Comma to Assign Table Structure

In contrast to the blank, the Comma is used to denote a crossing of variables, down the side and across the top.

```
Command ==>
00001 proc tabulate data=saved.demog;
00002   class gender grade status;
00003   table gender,grade;
00004   table status,grade;
00005   table grade,status;
00006 run;
00007
```

	GRADE	
	high	low
GENDER	N	N
F	8.00	10.00
M	14.00	12.00

	GRADE	
	high	low
STATUS	N	N
D	2.00	4.00
N	10.00	11.00
P	1.00	3.00
S	3.00	7.00
M	1.00	1.00

	STATUS				
	D	N	P	S	M
GRADE	N	N	N	N	N
high	2.00	10.00	1.00	3.00	1.00
low	4.00	11.00	3.00	7.00	1.00

Three Dimensional Reports

Three-dimensional reports can be created in two ways:

- Referencing three crossing variables in the Table statement
- Using a By statement

The following example references three crossing variables in the Table statements.

```
Command ==>
00001 proc tabulate data=saved.demog;
00002   class gender grade status;
00003   table gender,grade,status;
00004   table grade,status,gender;
00005 run;
00006
```

GENDER F		STATUS				
GRADE		D	N	P	S	M
		N	N	N	N	N
high		8.00				1.00
low		8.00	2.00	2.00		1.00

GRADE high		GENDER	
STATUS		F	M
		N	N
D			2.00
N		8.00	0.00
P			1.00
S			3.00
M		1.00	

Three-dimensional output can be created with sorted data, as displayed below.

```
Command ==>
00001 proc sort data=saved.demog out=work.demog;
00002   by grade;
00003 run;
00004
00005 proc tabulate data=work.demog;
00006   class gender grade status;
00007   table gender,status;
00008   table status,gender;
00009 run;
00010
```

List output is displayed below.

GRADE-high		STATUS				
GENDER		D	N	P	S	M
		N	N	N	N	N
F			8.00			1.00
M		2.00	0.00	1.00	3.00	

GRADE-low		GENDER	
STATUS		F	M
		N	N
D			2.00
N		8.00	0.00
P			1.00
S			3.00
M			

GRADE=low					
	STATUS				
	D	N	P	S	M
GENDER	N	N	N	N	N
F		8.00	2.00	2.00	1.00
M	1.00	2.00	1.00	5.00	

GRADE=low		
STATUS	GENDER	
	F	N
D		4.00
N	8.00	2.00
P	2.00	1.00
S	2.00	5.00
M	1.00	

All in One

All three structures – concatenation with a blank, crossing with a comma, and hierarchical with an asterisk – can be found in a single table structure.

```
Command ===>
00001 proc tabulate data=saved.demog;
00002   class gender grade status;
00003   table gender,grade*status;
00004 run;
00005
```

GRADE									
low					high				
STATUS					STATUS				
D	N	P	S	M	D	N	P	S	M
F	8	2	2	1	F	8	2	2	1
M	2	1	5		M	2	1	5	

```
Command ===>
00001 proc tabulate data=saved.demog;
00002   class gender grade status;
00003   table gender;
00004   table status*grade,gender;
00005 run;
00006
```

GENDER	
F	N
N	N
22.00	27.00

STATUS	GRADE	GENDER	
		F	N
D	high		2.00
D	low		4.00
N	high	8.00	8.00
N	low	8.00	2.00
P	high		1.00
P	low	2.00	1.00
S	high		5.00
S	low	2.00	5.00
M	high	1.00	
M	low	1.00	

The Asterisk to Assign Table Structure

The third method of structuring a table is to use the Asterisk (cf., Blank, Comma). The Asterisk creates a hierarchical structure for the variables.

Submit the code below.

```
Command ===>
00001 proc tabulate data=saved.demog;
00002   class gender grade status;
00003   table gender*grade;
00004   table status*grade;
00005   table grade*status;
00006 run;
00007
```

List output is displayed below.

GENDER			
F		N	
GRADE		GRADE	
high	low	high	low
N	N	N	N
5.00	12.00	14.00	19.00

STATUS							
D		N		P		S	
GRADE		GRADE		GRADE		GRADE	
high	low	high	low	high	low	high	low
N	N	N	N	N	N	N	N
2.00	4.00	15.00	11.00	1.00	3.00	5.00	

(Cont Insead)

STATUS			
D		N	
GRADE		GRADE	
low	high	low	high
N	N	N	N

```

Command ==>
00001 proc tabulate data=saved.demog;
00002 class gender grade;
00003 var children;
00004 table gender*grade=children;
00005 run;
00006

```

GENDER			
F		M	
GRADE		GRADE	
high	low	high	low
CHILDREN	CHILDREN	CHILDREN	CHILDREN
Sum	Sum	Sum	Sum
28.00	14.00	25.00	29.00

Using All in the Table Statement

The All option is used in the Table statement to generate overall table values. The values can be placed, for instance, to the left or at the bottom of the Proc Tabulate table.

Some examples of placing the All column at the sides are shown with the following program.

```

Command ==>
00001 options ls=100 center;
00002 proc tabulate data=saved.demog;
00003 class gender status grade;
00004 var salary age children;
00005 table status,gender*salary all;
00006 table gender*grade*age all*age all*salary;
00007 table all*age grade*gender=children;
00008 run;

```

STATUS	GENDER				All	
	F		M			
	SALARY		SALARY			
	Sum	Sum	Sum	Sum		
D			189882.00		5.00	
N	317468.00		238110.00		27.00	
P	25024.00		32916.00		4.00	
B	27830.00		158755.00		10.00	
M	58040.00				2.00	

GRADE	GENDER				All	All		
	F		M					
	AGE		AGE					
	Sum	Sum	Sum	Sum	Sum	Sum		
high	465.00	472.00	565.00	575.00	1077.00	358125.00		
low								

All	GRADE			
	high		low	
	GENDER		GENDER	
	F	M	F	M
AGE	CHILDREN	CHILDREN	CHILDREN	CHILDREN
Sum	Sum	Sum	Sum	Sum
1077.00	28.00	29.00	14.00	18.00

Additional examples of placing the All row at the top or bottom is displayed in the syntax below.

```

Command ==>
00001 options ls=100 center;
00002 proc tabulate data=saved.demog;
00003 class gender status grade;
00004 var salary age children;
00005 table status all, grade*children;
00006 table grade all, gender*salary;
00007 table grade all, gender*age;
00008 run;

```

STATUS	GRADE	
	high	low
	CHILDREN	CHILDREN
	Sum	Sum
D	5.00	0.00
N	47.00	19.00
P	2.00	4.00
B	6.00	0.00
M	3.00	1.00
All	67.00	32.00

GRADE	GENDER	
	F	M
	SALARY	SALARY
	Sum	Sum
high	252816.00	330016.00
low	185740.00	141747.00
All	428356.00	509763.00

GRADE	GENDER	
	F	M
	AGE	AGE
	Sum	Sum
high	465.00	565.00
low	472.00	375.00
All	937.00	940

Command ==>

	GENDER		
	F		
	SALARY	AGE	CHILDREN
	Mean	Max	Min
GRADE			
high	28179.56	61.00	2.00
low	12749.60	66.00	0.00

Command ==>

	GENDER		
	F		
	SALARY	AGE	CHILDREN
	Mean	Mean	Mean
GRADE			
high	28179.56	61.67	2.11
low	12749.60	66.31	1.00

Command ==>

	GENDER			
	F			
	SALARY		CHILDREN	
	Min	Max	Min	Max
GRADE				
high	15840.00	47520.00	2.00	5.00
low	8870.00	14890.00	0.00	2.00

Subsetting with Proc Tabulate

Proc Tabulate can use standard SAS subsetting syntax:

- Where option
- Drop/Keep option
- Where statement
- Firstobs and/or Obs option

```

Command ==>
00001 options ls=100 center;
00002 proc tabulate data=sasued.demog;
00003   class gender status grade;
00004   var salary age children;
00005   where status in ("D", "W", "S");
00006   table status all, grade=children;
00007   table grade all, gender=salary;
00008   table grade all, gender=age;
00009 run;

```