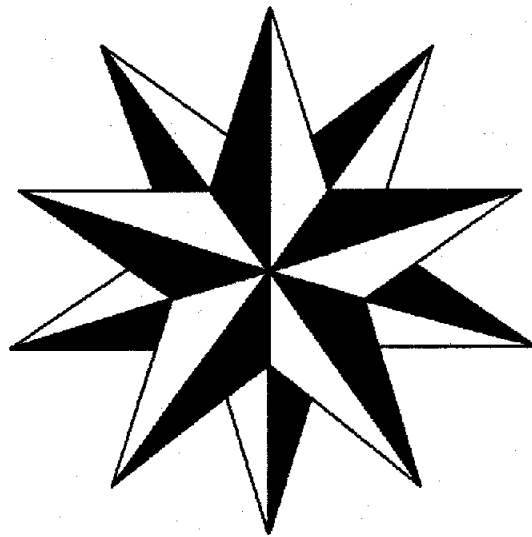


**PLEASING THE PICKY MANAGER
WITH THE TABULATE PROCEDURE**



**Stephen Harper
InterTechnologies Group
Minnesota Department of Administration**

PLEASEING THE PICKY MANAGER WITH THE TABULATE PROCEDURE
Stephen A. Harper, InterTechnologies Group

Introduction

PROC TABULATE can be used to create cross-tabulation tables in up to three dimensions: page, column, and row. Within each of these dimensions, the data can be further subdivided through the use of crossing (or nesting) operations. This tutorial will present a series of examples, building from the very simple, single-column report, all the way up to a complex, multi-column, multi-row report with customized formatting. This paper does not purport to cover all of the very extensive capabilities available in the procedure, but does present some of the most useful ones.

Overview of TABULATE statements

Tabulate makes use of the following statements:

PROC TABULATE: This is used to start the procedure, and set some general defaults (e.g. the general format of the cells).

TABLE or TABLES: This is the most complex of the statements, and the key to using Tabulate. Through correct use of the TABLE statement, you can construct a wide variety of tables.

CLASS: The CLASS statement is used to name the classification (or grouping) variables that define the cells in the table. These can be either numeric or character, but should contain a limited number of discrete values. If you are using a continuous, numeric variable, the values can be grouped with PROC FORMAT.

VAR: Names the analysis variables which constitute the values displayed in the table cells. Analysis variables must be numeric, but do not necessarily have to be discrete.

FORMAT: Formats only the classification variables. To format analysis variables, the variable name must be crossed with the desired format. This is illustrated in Examples 7 and 8 below.

LABEL: Attach labels of up to 40 characters to classification or analysis variables. These are then used as page, row, or column headings.

KEYLABEL: Attach labels to key words used in Tabulate (e.g. ALL, SUM).

BY: Obtain separate tables for each BY group. (Similar to the page dimension.)

FREQ: Name a variable whose value represents the frequency of the observation.

WEIGHT: Name a variable whose value represents the weight of the observation.

One important difference between Tabulate and some other procedures in the SAS* system is that all of the variables being used by Tabulate must be named on one of the statements above. In general, this means including the variable on either the CLASS or VAR statement.

Overview of Tabulate keywords

Tabulate makes use of a number of keywords for calculating statistics and totals.

ALL: Special classification variable used to calculate a total for whatever variables it is crossed or concatenated with. This is illustrated in examples 9-16 below.

PCTSUM: Used to calculate the percentage that one summed value represents of another summed value. This is illustrated in examples 11-16 below.

PCTN: Used to calculate the percentage that one frequency value represents of another frequency value.

Other available statistics include:
N NMISS MIN MAX RANGE SUM SUMWGT MEAN
USS CSS VAR STD STDERR CV T PRT. These are all discussed extensively in the manuals.

Introduction to the TABLE statement

The TABLE statement is used to specify the desired table through the combination of variables and table operators. The major operations that can be done are:

Creating dimensions: The three dimensions of the table are separated by commas and are defined in the order: page, row, column. If only two dimensions are specified, they are: row, column. If only a single dimension is specified, it defines columns.

Concatenation: Adding two tables together, end to end. This is accomplished by listing variables separated by a blank.

Crossing or nesting: Nesting the values of one variable based on the values of another variable. Through nesting, you can also produce subtotals. The nesting operator is an asterisk.

Grouping: Variables can be grouped with parentheses so that all nestings do not need to be spelled out completely.

Percentages: The PCTSUM and PCTN keywords discussed above utilize brackets to enclose the denominator specification.

Labelling: Specific variables can be labelled with the use of an equals sign.

The data being used

```
data;
  input name $1-19 sex $20 dept $22-27
  salary 29-32 benefits 34-36;
cards;
JOHN BROWN      M ADMIN  2000 150
STEPHANIE JONES F SALES   2500 150
VIRGINIA HART   F ADMIN   1500 150
ROBERT JOHNSON  M SALES   2500 150
MAUREEN ADAMS   F ADMIN   3000 150
JOANNA JOHNSTON F ADMIN   3000 150
TERRY MASTERS   M ADMIN   3000 150
JAMES JOYCE     M SALES   2050 150
ALFRED WHITEHEAD M SALES   2200 150
JOHN MUIR       M ADMIN   2800 150
;
```

EXAMPLE 1: Concatenating two class variables

Concatenating two class variables produces two distinct tables. Because only a single dimension is being specified, both of the tables are columnar. Also note that the statistic being produced is the default N (frequency count) statistic in the BEST format. This is the only statistic available when two class variables are used.

```
PROC TABULATE;
  CLASS DEPT SEX;
  TABLES DEPT SEX;
  RUN;
```

DEPT		SEX	
ADMIN	SALES	F	M
N	N	N	N
6.00	4.00	4.00	6.00

EXAMPLE 2: Crossing two class variables

Crossing two class variables produces a single, columnar table with the interaction of the two variables. Again, only the default N statistic can be produced.

```
proc tabulate ;
  class dept sex;
  tables DEPT*SEX;
  run;
```

DEPT			
ADMIN		SALES	
SEX		SEX	
F	M	F	M
N	N	N	N
3.00	3.00	1.00	3.00

EXAMPLE 3: Crossing class and analysis variables

Crossing a class and an analysis variable produces a different default statistic (the sum of the analysis variable). Note that the N statistic has been eliminated. It can be specifically requested as demonstrated later in the paper.

```
proc tabulate ;
  class dept;
  VAR SALARY;
  tables DEPT*SALARY;
  run;
```

DEPT	
ADMIN	SALES
SALARY	SALARY
SUM	SUM
15300.00	9250.00

EXAMPLE 4: Adding a row dimension

A row dimension is added by separating the variables with a comma. This forces the salary variable into the column dimension.

```
proc tabulate ;
  class dept ;
  var salary;
  tables DEPT,SALARY;
  run;
```

	SALARY	
	SUM	
DEPT		
ADMIN	15300.00	
SALES	9250.00	

EXAMPLE 5: Crossing in the row dimension

Crossing two class variables in the row dimension produces subdivisions of the data.

```
proc tabulate ;
  class dept sex;
  var salary;
  tables DEPT*SEX,
         SALARY;
run;
```

		SALARY	
		SUM	
DEPT	SEX		
ADMIN	F	7500.00	
	M	7800.00	
SALES	F	2500.00	
	M	6750.00	

EXAMPLE 6: Adding other statistics

To produce additional statistics, the analysis variable is crossed with the requested statistic(s).

```
proc tabulate ;
  class dept sex;
  var salary;
  tables dept*sex,
         salary*(SUM N MEAN);
run;
```

		SALARY		
		SUM	N	MEAN
DEPT	SEX			
ADMIN	F	7500.00	3.00	2500.00
	M	7800.00	3.00	2600.00
SALES	F	2500.00	1.00	2500.00
	M	6750.00	3.00	2250.00

EXAMPLE 7: Changing the default format

When the majority of the cells in a table share the same format, the default format can be changed through an option on the PROC TABULATE statement. This format affects only the analysis variables. Formatting class variables is illustrated in Example 16.

```
proc tabulate FORMAT=DOLLAR8.;
  class dept sex;
  var salary;
  tables dept*sex,
         salary*(SUM N MEAN);
run;
```

		SALARY		
		SUM	N	MEAN
DEPT	SEX			
ADMIN	F	\$7,500	3	\$2,500
	M	\$7,800	3	\$2,600
SALES	F	\$2,500	1	\$2,500
	M	\$6,750	3	\$2,250

EXAMPLE 8: Formatting an individual variable

When one column of cells differs in format from the remainder, that column can be formatted individually by crossing the analysis variable with the desired format. Note that the format specified also controls the width of the column. If the column heading exceeds that width, it is hyphenated and printed on multiple lines.

```
proc tabulate format=dollar8. ;
  class dept sex;
  var salary;
  tables dept*sex,
         salary*(sum N*F=3. mean);
run;
```

		SALARY		
		SUM	N	MEAN
DEPT	SEX			
ADMIN	F	\$7,500	3	\$2,500
	M	\$7,800	3	\$2,600
SALES	F	\$2,500	1	\$2,500
	M	\$6,750	3	\$2,250

EXAMPLE 9: Calculating grand totals

Grand totals can be produced by concatenating the special class variable ALL at the end of any dimension.

```
proc tabulate format=dollar8. ;
  class dept sex;
  var salary;
  tables DEPT*SEX ALL,
    salary*(sum n*f=3. mean);
run;.
```

		SALARY		
		SUM	N	MEAN
DEPT	SEX			
ADMIN	F	\$7,500	3	\$2,500
	M	\$7,800	3	\$2,600
SALES	F	\$2,500	1	\$2,500
	M	\$6,750	3	\$2,250
ALL		\$24,550	10	\$2,455

EXAMPLE 10: Calculating column subtotals

Subtotals can be produced by crossing the special class variable ALL with the class variable for which subtotals are needed.

```
proc tabulate format=dollar8.;
  class dept sex;
  var salary;
  tables DEPT*(SEX ALL) ALL,
    salary*(sum n*f=3. mean);
run;
```

		SALARY		
		SUM	N	MEAN
DEPT	SEX			
ADMIN	F	\$7,500	3	\$2,500
	M	\$7,800	3	\$2,600
	ALL	\$15,300	6	\$2,550
SALES	F	\$2,500	1	\$2,500
	M	\$6,750	3	\$2,250
	ALL	\$9,250	4	\$2,313
ALL		\$24,550	10	\$2,455

EXAMPLE 11: Calculating percentage of subtotals

Various percentages can be calculated through the naming of a denominator within a particular crossing. This denominator must already exist in the table as the result of some crossing or concatenation.

The default labels on keywords can also be changed with the KEYLABEL statement.

```
proc tabulate format=dollar8. ;
  class dept sex;
  var salary;
  tables dept*(sex all) all,
    salary*(sum PCTSUM<SEX ALL>*F=7.2
      n*f=5. mean);
  KEYLABEL ALL = 'TOTAL'
    N = 'COUNT'
    SUM = 'TOTAL'
    PCTSUM = '% OF TOTAL';
run;
```

		SALARY			
		TOTAL	% OF TOTAL	COUNT	MEAN
DEPT	SEX				
ADMIN	F	\$7,500	49.02	3	\$2,500
	M	\$7,800	50.98	3	\$2,600
	TOTAL	\$15,300	100.00	6	\$2,550
SALES	F	\$2,500	27.03	1	\$2,500
	M	\$6,750	72.97	3	\$2,250
	TOTAL	\$9,250	100.00	4	\$2,313
TOTAL		\$24,550	100.00	10	\$2,455

EXAMPLE 12: Calculating percentage of grand total: Part I

The percentages obtained are dependent on which total is used as the denominator in the PCTSUM calculation.

```
proc tabulate format=dollar8. ;
  class dept sex;
  var salary;
  tables dept*(sex all) all,
    salary*(sum PCTSUM<DEPT ALL>*f=7.2
      n*f=5. mean);
  keylabel all = 'TOTAL'
    sum = 'TOTAL'
    n = 'COUNT'
    pctsum = '% OF TOTAL';
run;
```

		SALARY			
		TOTAL	% OF TOTAL	COUNT	MEAN
DEPT	SEX				
ADMIN	F	\$7,500	75.00	3	\$2,500
	M	\$7,800	53.61	3	\$2,600
	TOTAL	\$15,300	62.32	6	\$2,550
SALES	F	\$2,500	25.00	1	\$2,500
	M	\$6,750	46.39	3	\$2,250
	TOTAL	\$9,250	37.68	4	\$2,313
TOTAL		\$24,550	100.00	10	\$2,455

EXAMPLE 14: Calculating percentage of grand total: Part III

Grouping parentheses can not be used within a denominator definition, so all crossings and concatenations must be spelled out completely.

```
proc tabulate format=dollar8. ;
  class dept sex;
  var salary;
  tables dept*(sex all) all,
    salary*(sum PCTSUM<DEPT*SEX
      DEPT*ALL ALL>*f=7.2 n*f=5. mean);
  keylabel all = 'TOTAL'
    n = 'COUNT'
    sum = 'TOTAL'
    pctsum = '% OF TOTAL';
run;
```

EXAMPLE 13: Calculating percentage of grand total: Part II

Both concatenations and crossings can be used for naming the desired denominator.

```
proc tabulate format=dollar8. ;
  class dept sex;
  var salary;
  tables dept*(sex all) all,
    salary*(sum PCTSUM<DEPT*SEX
      ALL>*f=7.2 n*f=5. mean);
  keylabel all = 'TOTAL'
    n = 'COUNT'
    sum = 'TOTAL'
    pctsum = '% OF TOTAL';
run;
```

		SALARY			
		TOTAL	% OF TOTAL	COUNT	MEAN
DEPT	SEX				
ADMIN	F	\$7,500	30.55	3	\$2,500
	M	\$7,800	31.77	3	\$2,600
	TOTAL	\$15,300	62.32	6	\$2,550
SALES	F	\$2,500	10.18	1	\$2,500
	M	\$6,750	27.49	3	\$2,250
	TOTAL	\$9,250	37.68	4	\$2,313
TOTAL		\$24,550	100.00	10	\$2,455

		SALARY			
		TOTAL	% OF TOTAL	COUNT	MEAN
DEPT	SEX				
ADMIN	F	\$7,500	30.55	3	\$2,500
	M	\$7,800	31.77	3	\$2,600
	TOTAL	\$15,300	100.00	6	\$2,550
SALES	F	\$2,500	10.18	1	\$2,500
	M	\$6,750	27.49	3	\$2,250
	TOTAL	\$9,250	100.00	4	\$2,313
TOTAL		\$24,550	100.00	10	\$2,455

EXAMPLE 15: Dividing one variable by another

The denominator specified can also be another analysis variable.

```
proc tabulate format=dollar8. ;
  class dept sex;
  var salary BENEFITS;
  tables dept*(sex all) all,
    salary*SUM BENEFITS*(SUM
      PCTSUM<SALARY>*F=6.2);
  keylabel all = 'TOTAL'
    sum = 'TOTAL'
    pctsum = '% OF TOTAL';
run;
```

		SALARY	BENEFITS	
		TOTAL	TOTAL	% OF TOTAL
DEPT	SEX			
ADMIN	F	\$7,500	\$450	6.00
	M	\$7,800	\$450	5.77
	TOTAL	\$15,300	\$900	5.88
SALES	F	\$2,500	\$150	6.00
	M	\$6,750	\$450	6.67
	TOTAL	\$9,250	\$600	6.49
TOTAL		\$24,550	\$1,500	6.11

EXAMPLE 16: Putting it all together

A column heading (SEX) can be eliminated completely. This must be done within the TABLES statement itself, rather than on a LABEL statement.

Class variables can be formatted with the use of the FORMAT statement.

The RTS option can be used to increase the amount of space given to the row headings. The default is 1/4 of the total line size required by the table, which can result in the hyphenation of labels.

```
PROC FORMAT;
  VALUE $SEX 'F' = 'FEMALE'
            'M' = 'MALE'
            ;
RUN;

proc tabulate format=dollar8.;
  class dept sex;
  var salary benefits;
  tables dept*(SEX=' ' all) all,
         n*f=5,
         salary*(sum pctsum<dept*sex
         dept*all all>*f=6.2)
         benefits*(sum
         (pctsum<salary>=
         '% OF SALARY')*f=6.2)
         /RTS=25;

  keylabel all = 'TOTAL'
           sum = 'TOTAL'
           n   = 'COUNT'
           pctsum = '% OF TOTAL';
  LABEL DEPT = 'DEPARTMENT';
  FORMAT SEX $SEX.;
run;
```

DEPARTMENT	SALARY		BENEFITS	
	COUNT	TOTAL	% OF TOTAL	SALARY
ADMIN				
	3	\$7,500	30.55	\$450
	3	\$7,800	31.77	\$450
SALES				
	6	\$15,300	62.32	\$900
	1	\$2,500	10.18	\$150
TOTAL				
	3	\$6,750	27.49	\$450
	4	\$9,250	37.68	\$600
TOTAL	10	\$24,550	100.00	\$1,500

* SAS is a registered trademark of SAS Institute, Cary, NC, USA.

For further information contact:

Stephen A. Harper
InterTechnologies Group
State of Minnesota
658 Cedar Street
St. Paul, MN 55155
(612) 297-4544