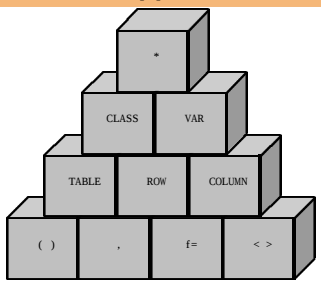
The Building Blocks of PROC TABULATE

Ron Coleman
Links Analytical, Inc.



There are a number of basic concepts that need to be understood before you can expect to work effectively with TABULATE. Without a thorough knowledge of the keywords and operators TABULATE can be very difficult. A little time and a building block approach will make this complex SAS® procedure easier to use and understand.

First, our example data set.

Our data is from a fictional company, Educational Demos, for 1995 and 1996. The company has clients in 6 states broken into North and South regions. The North region includes the states Connecticut, Maine, and Massachusetts. The South region includes Alabama, Florida, and Georgia.

Here is a partial PROC CONTENTS listing of the TABDATA data set.

```
Alphabetic List of Variables and Attributes-----

   #    Variable    Type    Len    Pos
   ffffffffffffffffffffffffffff
   5    DATE        Num       8     23
   4    MON         Num       8     15
   6    MONNAME     Char      3     31
   9    NEWCUSTS    Num       8     43
   7    QUARTER     Char      1     34
   1    REGION      Char      5      0
   8    SALES       Num       8     35
   2    STATE       Char      2      5
   3    YEAR        Num       8      7
```

The PROC TABULATE statement begins the TABULATE step.

The general form of the TABULATE statement is:

```
PROC TABULATE DATA=SAS-data-set <option(s)>;
```

DATA= is an option but I always stress the importance in using it to identify the source of the data coming into the procedure.

There are many options that can be used with the TABULATE procedure and I will address some of the more common ones later in this presentation.

CLASS and VAR Statements

All variables can be categorized into one or both; *classification* or *analysis* variables. In order to define to PROC TABULATE how each variable is to be used we must first define the difference between classification and analysis variables.

– *Classification* – variables contain values by which you want to summarize or group your data. Classification variables can be either character, (N, S, E, W or Alabama, Arkansas, etc.), or numeric (months 1 – 12, area codes, etc). Classification variables are best when they have discrete values.

– *Analysis* – variables are always numeric and are used to compute statistics and arithmetic sums. Analysis variables tend to be continuous.

A variable can only be used as either a classification or analysis variable within a PROC TABULATE step. Therefore, a variable may appear only once on either a CLASS or VAR statement.

The CLASS Statement

The CLASS statement is used to list all classification variables to be used within PROC TABULATE. Only one CLASS statement may be used in a step.

The general form of the CLASS statement is:

```
CLASS variable(s);
```

The VAR Statement

The VAR statement is used to list all analysis variables to be used within PROC TABULATE. Only one VAR statement may be used in a step.

The general form of a VAR statement is:

```
VAR variable(s);
```

The TABLE Statement

The TABLE statement is the workhorse of PROC TABULATE. This is where you define how you want your table to appear; which variables are in the table, row, and column headings, which analysis variables appear where and what statistics are displayed, what labels are used for the variables and/or statistics or is there to be no label printed.

The general form of the TABLE statement is:

```
TABLE <<table-expression,> row-expression,> column-expression /
<option(s)>;
```

table-expression describes the CLASS variables whose values will be used to create separate tables

row-expression describes the CLASS and/or VAR variables that will be used to define the rows of the table

column-expression describes the CLASS and/or VAR variables that will be used to define the columns of the table

Commas separate the table/row/column dimension expressions. The dimensions always start from the right and build left. In other words, if only one dimension is given it is the column dimension. Two dimensions are the row, column. Etc.

Table Expressions

These *expressions* consist of variables, statistics, and operators. The variables come from the input data set. The available statistics for PROC TABULATE are listed in the table below.

N	NMISS	MEAN
STD	MIN	MAX
RANGE	SUM	USS
CSS	STDERR	CV
T	PRT	VAR
SUMWGT	PCTN	PCTSUM

The operators of PROC TABULATE are the most difficult topic to grasp. The operators used are: **,** (comma), ***** (asterisk or star), (**space** or **blank**), **()** (parentheses), **‘ ‘** (quotations), **=** (equal sign), **<>** (denominator), and **f=** (formatting).

The Comma

The first operator, mentioned on the previous page, is the comma. The Comma separates the table dimension-expressions for the table, row, and column dimensions. You must use a comma for any cross-tab type tables.

All of the remaining operators are used within the dimension-expressions and can be used in all dimensions.

Enough words, let’s get to some tables!

Exploring Dimensions

Begin with one CLASS variable. The default statistic when only CLASS variables are used in the N or number of observations in each category.

```
proc tabulate data=tabdata;
   class region;
   table region;
run;
```

REGION	
North	South
N	N
1080.00	984.00

Add another CLASS variable and another dimension to the table.

```
proc tabulate data=tabdata;
   class region year;
   table region, year;
run;
```

	YEAR	
	1995	1996
	N	N
REGION		
North	624.00	456.00
South	480.00	504.00

Add a third CLASS variable and a third dimension to the table.

```
proc tabulate data=tabdata;
   class region year state;
   table region, state, year;
run;
```

REGION North

	YEAR	
	1995	1996
	N	N
STATE		
CT	216.00	120.00
MA	192.00	120.00
ME	216.00	216.00

REGION South

	YEAR	
	1995	1996
	N	N
STATE		
AL	144.00	180.00
FL	180.00	120.00
GA	156.00	204.00

Exploring Operators

The ALL keyword gives summaries for the class variables it is associated with.

```
proc tabulate data=tabdata;
   class region year;
   table region all, year all;
run;
```

	YEAR		
	1995	1996	ALL
	N	N	N
REGION			
North	624.00	456.00	1080.00
South	480.00	504.00	984.00
ALL	1104.00	960.00	2064.00

The above example also illustrates the *space* operator. A *space* allows you to concatenate, or have appear next to on another, CLASS variables.

```
proc tabulate data=tabdata;
         class region year;
   table region year;
run;
```

REGION		YEAR	
North	South	1995	1996
N	N	N	N
1080.00	984.00	1104.00	960.00

```
table region all year all;
```

REGION		ALL	YEAR		ALL
North	South		1995	1996	
N	N	N	N	N	N
1080.00	984.00	2064.00	1104.00	960.00	2064.00

The next operator to learn in the '*' or asterisk. The asterisk causes a 'crossing' of CLASS variables. In the example below I specify `year*quarter`, that means for each value of `year`, give me the statistics for each value of `quarter`.

```
proc tabulate data=tabdata format=3.;
   class region year quarter;
   table region, year*quarter;
run;
```

	YEAR							
	1995				1996			
	QUARTER				QUARTER			
	I	II	III	IV	I	II	III	IV
	N	N	N	N	N	N	N	N
REGION								
North	156	156	156	156	114	114	114	114
South	120	120	120	120	126	126	126	126

I also added the format= option on the PROC TABULATE statement. This option sets the default format for all cells in the table. The standard default format is 12.2. You can also set specific formats for each column, We'll see how to do this later.

We can do crossings in any dimension of the table.

```
proc tabulate data=tabdata format=3.;
   class region year quarter state;
   table region*state, year*quarter;
run;
```

The Building Blocks of PROC TABULATE

		YEAR							
		1995				1996			
		QUARTER				QUARTER			
		I	II	III	IV	I	II	III	IV
		N	N	N	N	N	N	N	N
REGION	STATE								
North	CT	54	54	54	54	30	30	30	30
	MA	48	48	48	48	30	30	30	30
	ME	54	54	54	54	54	54	54	54
South	AL	36	36	36	36	45	45	45	45
	FL	45	45	45	45	30	30	30	30
	GA	39	39	39	39	51	51	51	51

Adding Analysis Variables

We can have tables with only analysis variables:

```
proc tabulate data=tabdata format=dollar15.2;
   var sales;
   table sales;
run;
```

SALES
SUM
$154,672,967.00

The default statistic for analysis variables is the SUM. Analysis variables can only appear in <u>one</u> dimension of the table.

Most often we need to see the analysis variables in terms of categories so we need to mix CLASS and VAR variables to create more useful tables.

```
proc tabulate data=tabdata format=dollar11.;
   class year;
   var sales;
   table year*sales;
run;
```

YEAR	
1995	1996
SALES	SALES
SUM	SUM
$82,999,668	$71,673,299

Again we use the * to do a crossing. Remember, the * means – for each value of the first variable, `year`, give me the values of the second. Since the second variable, `sales`, is an analysis variable we get a statistic.

Now let's add our row dimension again.

```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales;
   table region, year*sales;
run;
```

	YEAR	
	1995	1996
	SALES	SALES
	SUM	SUM
REGION		
North	$46,870,009	$34,038,985
South	$36,129,659	$37,634,314

Now we will specify that we want a different statistic, the mean:

```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales;
   table region, year*sales*mean;
run;
```

	YEAR	
	1995	1996
	SALES	SALES
	MEAN	MEAN
REGION		
North	$75,112	$74,647
South	$75,270	$74,671

Adding Statistics

Now let's do two statistics. To do this we will need a new operator (), parentheses, to group items within the dimension-expression.

```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales;
   table region, year*sales*(sum mean);
run;
```

	YEAR			
	1995		1996	
	SALES		SALES	
	SUM	MEAN	SUM	MEAN
REGION				
North	$46,870,009	$75,112	$34,038,985	$74,647
South	$36,129,659	$75,270	$37,634,314	$74,671

Specify a smaller format for the mean:

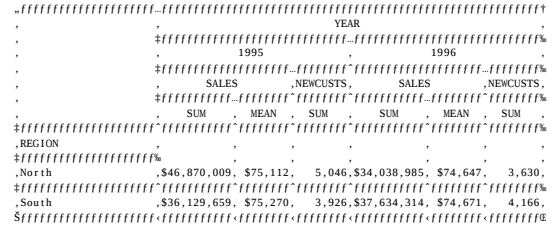
```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales;
   table region, year*sales*(sum mean*f=dollar8.);
run;
```

	YEAR			
	1995		1996	
	SALES		SALES	
	SUM	MEAN	SUM	MEAN
REGION				
North	$46,870,009	$75,112	$34,038,985	$74,647
South	$36,129,659	$75,270	$37,634,314	$74,671

One more level, multiple analysis and multiple statistics.

```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales newcusts;
   table region, year*(sales*(sum mean*f=dollar8.)
      newcusts*sum*f= comma8.);
run;
```

The Building Blocks of PROC TABULATE



Whoa! What did I just do?

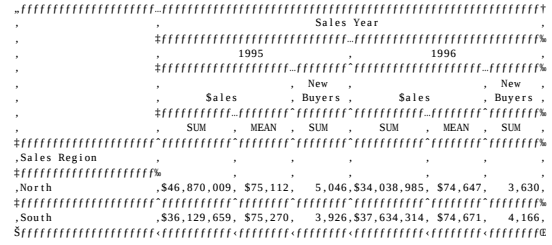
Let's put the column dimension-expression into words:

For each value of the variable `year`, show the analysis variables `sales` and `newcusts`. For `sales` show the `sum`, formatted with the default format, and the `mean`, formatted using the `dollar8.` format. For `newcusts` show the `sum`, formatted using the `comma8.` format.

Building Tables

Now that we know how to present our data we need to dress it up! We can begin with a simple LABEL statement. NOTE: PROC TABULATE does not recognize the split= option of PROC PRINT.

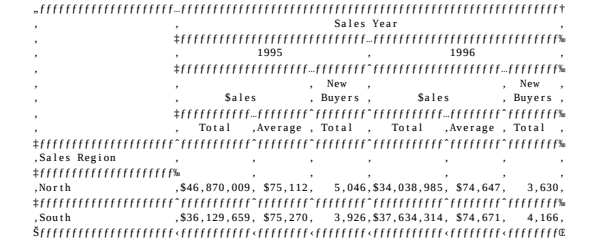
```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales newcusts;
   table region, year*(sales*(sum mean*f=dollar8.)
      newcusts*sum*f= comma8.);
   label year     = 'Sales Year'
         region   = 'Sales Region'
         sales    = '$ales'
         newcusts = 'New Buyers';
run;
```



Note how PROC TABULATE wraps the titles automatically as with 'New Buyers'.

You can also apply labels for the statistics using the KEYLABEL statement.

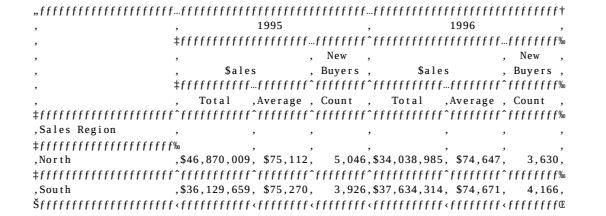
```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales newcusts;
   table region, year*(sales*(sum mean*f=dollar8.)
      newcusts*sum*f= comma8.);
   label year     = 'Sales Year'
         region   = 'Sales Region'
         sales    = '$ales'
         newcusts = 'New Buyers';
   keylabel sum    = 'Total'
            mean   = 'Average';
run;
```



LABEL and KEYLABEL statements are fine for setting the default labels but like with formats, TABULATE allows you to specify different labels within the TABLE statement.

This is also a means of eliminating a label from the table while all of the data is still displayed.

```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales newcusts;
   table region, year=' '*(
                    sales*(sum mean*f=dollar8.)
                    newcusts*sum='Count'*f= comma8.
                    );
   label year     = 'Sales Year'
         region   = 'Sales Region'
         sales    = '$ales'
         newcusts = 'New Buyers';
   keylabel sum    = 'Total'
            mean   = 'Average';
run;
```



When we specify the format for each column we say that the data will be displayed in *n* places within each cell. That defines the width of the cell and the labels are automatically wrapped or hyphenated to fit in the column.

You also have control over the space used for the row titles, called the row title space, through the RTS= table option. The number of spaces given in the RTS= option **includes** the vertical lines within the row titles.

```
proc tabulate data=tabdata format=dollar11.;
   class year region;
   var sales newcusts;
   table region, year=' '*(
                    sales*(sum mean*f=dollar8.)
                    newcusts*sum='Count'*f= comma8.
                    )
      / rts=8;
   label year     = 'Sales Year'
         region   = 'Sales Region'
         sales    = '$ales'
         newcusts = 'New Buyers';
   keylabel sum    = 'Total'
            mean   = 'Average';
run;
```

The Building Blocks of PROC TABULATE

```
„ffffff…fffffffffffffffffffffffffffff…ffffffffffffffffffffffffffff†
,      ,          1995          ,          1996          ,
,      ‡fffffffffffffffffff…fffffff^fffffffffffffffffff…fffffff%
,      ,                ,  New  ,                ,  New  ,
,      ,      $sales    , Buyers ,      $sales    , Buyers ,
,      ‡fffffffffff…fffffff^fffffff^fffffffffff…fffffff^fffffff%
,      ,  Total  ,Average , Count  ,  Total  ,Average , Count  ,
‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,Sales ,         ,        ,        ,         ,        ,        ,
,Region,         ,        ,        ,         ,        ,        ,
‡ffffff%         ,        ,        ,         ,        ,        ,
,North ,$46,870,009, $75,112,   5,046,$34,038,985, $74,647,   3,630,
‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,South ,$36,129,659, $75,270,   3,926,$37,634,314, $74,671,   4,166,
Šffffff‹fffffffffff‹fffffff‹fffffff‹fffffffffff‹fffffff‹fffffffŒ
```

If you have more than one variable in the row-expression then you need to calculate the RTS= amount carefully:

```
proc tabulate data=tabdata format=dollar11.;
   class year region state;
   var sales newcusts;
   table region*(state all), year=' '*(
                    sales*(sum mean*f=dollar8.)
                    newcusts*sum='Count'*f= comma8.
                    )
      / rts=15;
   label year     = 'Sales Year'
         region   = 'Sales Region'
         state    = 'State'
         sales    = '$ales'
         newcusts = 'New Buyers';
   keylabel sum    = 'Total'
            mean   = 'Average';
run;
```

```
„fffffffffffff…ffffffffffffffffffffffffffff…ffffffffffffffffffffffffffff†
,           ,          1995          ,          1996          ,
,           ‡fffffffffffffffffff…fffffff^fffffffffffffffffff…fffffff%
,           ,                ,  New  ,                ,  New  ,
,           ,      $sales    , Buyers ,      $sales    , Buyers ,
,           ‡fffffffffff…fffffff^fffffff^fffffffffff…fffffff^fffffff%
,           ,  Total  ,Average , Count  ,  Total  ,Average , Count  ,
‡ffffff…fffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,Sales ,State ,          ,        ,        ,         ,        ,        ,
,Region,      ,          ,        ,        ,         ,        ,        ,
‡ffffff^fffff%          ,        ,        ,         ,        ,        ,
,North ,CT    ,$16,251,251, $75,237,   1,830, $8,956,063, $74,634,     951,
,     ‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,     ,MA    ,$14,404,460, $75,023,   1,513, $8,958,733, $74,656,     960,
,     ‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,     ,ME    ,$16,214,298, $75,066,   1,703,$16,124,189, $74,649,   1,719,
,     ‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,     ,ALL   ,$46,870,009, $75,112,   5,046,$34,038,985, $74,647,   3,630,
‡ffffff^fffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,South ,State ,          ,        ,        ,         ,        ,        ,
,     ‡ffffff%          ,        ,        ,         ,        ,        ,
,     ,AL    ,$10,985,641, $76,289,   1,157,$13,494,644, $74,970,   1,503,
,     ‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,     ,FL    ,$13,526,146, $75,145,   1,467, $8,846,066, $73,717,   1,004,
,     ‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,     ,GA    ,$11,617,872, $74,474,   1,302,$15,293,604, $74,969,   1,659,
,     ‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,     ,ALL   ,$36,129,659, $75,270,   3,926,$37,634,314, $74,671,   4,166,
Šffffff‹fffff‹fffffffffff‹fffffff‹fffffff‹fffffffffff‹fffffff‹fffffffŒ
```

TABULATE automatically shares the row title space equally with each crossing variable.

SAS Versions 6.10 and beyond, sorry not is 6.09E, has a useful table option INDENT=. This allows the use of many crossings without increasing the required row title space significantly.

```
proc tabulate data=tabdata format=dollar11.;
   class year region state;
   var sales newcusts;
   table region*(state all), year=' '*(
                    sales*(sum mean*f=dollar8.)
                    newcusts*sum='Count'*f= comma8.
                    )
      / rts=8 indent=3;
   label year     = 'Sales Year'
         region   = 'Sales Region'
         state    = 'State'
         sales    = '$ales'
         newcusts = 'New Buyers';
   keylabel sum    = 'Total'
            mean   = 'Average';
run;
```

```
„ffffff…ffffffffffffffffffffffffffff…ffffffffffffffffffffffffffff†
,      ,          1995          ,          1996          ,
,      ‡fffffffffffffffffff…fffffff^fffffffffffffffffff…fffffff%
,      ,                ,  New  ,                ,  New  ,
,      ,      $sales    , Buyers ,      $sales    , Buyers ,
,      ‡fffffffffff…fffffff^fffffff^fffffffffff…fffffff^fffffff%
,      ,  Total  ,Average , Count  ,  Total  ,Average , Count  ,
‡ffffff^fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,North ,         ,        ,        ,         ,        ,        ,
‡ffffff,         ,        ,        ,         ,        ,        ,
,   CT ,$16,251,251, $75,237,   1,830, $8,956,063, $74,634,     951,
‡ffffff‡fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,   MA ,$14,404,460, $75,023,   1,513, $8,958,733, $74,656,     960,
‡ffffff‡fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,   ME ,$16,214,298, $75,066,   1,703,$16,124,189, $74,649,   1,719,
‡ffffff‡fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,   ALL,$46,870,009, $75,112,   5,046,$34,038,985, $74,647,   3,630,
‡ffffff‡fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,South ,         ,        ,        ,         ,        ,        ,
‡ffffff,         ,        ,        ,         ,        ,        ,
,   AL ,$10,985,641, $76,289,   1,157,$13,494,644, $74,970,   1,503,
‡ffffff‡fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,   FL ,$13,526,146, $75,145,   1,467, $8,846,066, $73,717,   1,004,
‡ffffff‡fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,   GA ,$11,617,872, $74,474,   1,302,$15,293,604, $74,969,   1,659,
‡ffffff‡fffffffffff^fffffff^fffffff^fffffffffff^fffffff^fffffff%
,   ALL,$36,129,659, $75,270,   3,926,$37,634,314, $74,671,   4,166,504,
Šffffff‹fffffffffff‹fffffff‹fffffff‹fffffffffff‹fffffff‹fffffffŒ
```

Percentages in PROC TABULATE

Each observation on our input data represents a salesman so we can use the N statistic to see how many salesmen were associated with the results. The statistic PCTN will also show us how the salesmen are distributed throughout the table. By default, the PCTN statistic gives percentages of the entire table.

```
proc tabulate data=tabdata format=dollar11.;
   class year region state;
   var sales newcusts;
   table region*(state all), year=' '*(
                    sales*(sum mean*f=dollar8.)
                    newcusts*sum='Count'*f= comma8.
                    n='# S/P'*f=3. pctn='%'*f=5.1
                    )
      / rts=8 indent=3;
   label year     = 'Sales Year'
         region   = 'Sales Region'
         state    = 'State'
         sales    = '$ales'
         newcusts = 'New Buyers';
   keylabel sum    = 'Total'
            mean   = 'Average';
run;
```

```
„ffffff…fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff†
,      ,          1995          ,          1996          ,
,      ‡fffffffffffffffffff…fffffff…fffff^fffffffffffffffffff…fffffff…fff…fffff%
,      ,                ,  New  ,  ,  ,                ,  New  ,  ,  ,
,      ,      $sales    , Buyers ,  ,  ,      $sales    , Buyers ,  ,  ,
,      ‡fffffffffff…fffffff^fffffff% # ,    ‡fffffffffff…fffffff^fffffff% # ,  ,
,      ,  Total  ,Average , Count  ,S/P, %  ,  Total  ,Average , Count  ,S/P, %  ,
‡ffffff^fffffffffff^fffffff^fffffff^fff^fffff^fffffffffff^fffffff^fffffff^fff^fffff%
,North ,         ,        ,        ,  ,  ,         ,        ,        ,  ,  ,
‡ffffff,         ,        ,        ,  ,  ,         ,        ,        ,  ,  ,
,   CT ,$16,251,251, $75,237,   1,830,216, 10.5, $8,956,063, $74,634,     951,120,  5.8,
‡ffffff‡fffffffffff^fffffff^fffffff^fff^fffff^fffffffffff^fffffff^fffffff^fff^fffff%
,   MA ,$14,404,460, $75,023,   1,513,192,  9.3, $8,958,733, $74,656,     960,120,  5.8,
‡ffffff‡fffffffffff^fffffff^fffffff^fff^fffff^fffffffffff^fffffff^fffffff^fff^fffff%
,   ME ,$16,214,298, $75,066,   1,703,216, 10.5,$16,124,189, $74,649,   1,719,216, 10.5,
‡ffffff‡fffffffffff^fffffff^fffffff^fff^fffff^fffffffffff^fffffff^fffffff^fff^fffff%
,   ALL,$46,870,009, $75,112,   5,046,624, 30.2,$34,038,985, $74,647,   3,630,456, 22.1,
‡ffffff‡fffffffffff^fffffff^fffffff^fff^fffff^fffffffffff^fffffff^fffffff^fff^fffff%
,South ,         ,        ,        ,  ,  ,         ,        ,        ,  ,  ,
‡ffffff,         ,        ,        ,  ,  ,         ,        ,        ,  ,  ,
,   AL ,$10,985,641, $76,289,   1,157,144,  7.0,$13,494,644, $74,970,   1,503,180,  8.7,
‡ffffff‡fffffffffff^fffffff^fffffff^fff^fffff^fffffffffff^fffffff^fffffff^fff^fffff%
,   FL ,$13,526,146, $75,145,   1,467,180,  8.7, $8,846,066, $73,717,   1,004,120,  5.8,
‡ffffff‡fffffffffff^fffffff^fffffff^fff^fffff^fffffffffff^fffffff^fffffff^fff^fffff%
,   GA ,$11,617,872, $74,474,   1,302,156,  7.6,$15,293,604, $74,969,   1,659,204,  9.9,
‡ffffff‡fffffffffff^fffffff^fffffff^fff^fffff^fffffffffff^fffffff^fffffff^fff^fffff%
,   ALL,$36,129,659, $75,270,   3,926,480, 23.3,$37,634,314, $74,671,   4,166,504, 24.4,
Šffffff‹fffffffffff‹fffffff‹fffffff‹fff‹fffff‹fffffffffff‹fffffff‹fffffff‹fff‹fffffŒ
```

We can specify the denominator for the percentages by using the angle bracket operators ‘<>’. We specify the dimension-expression piece we want to represent 100%.

The Building Blocks of PROC TABULATE

```
proc tabulate data=tabdata format=dollar11.;
   class year region state;
   var sales newcusts;
   table region*(state all), year=' '*(
                     sales*(sum mean*f=dollar8.)
                     newcusts*sum='Count'*f= comma8.
                     n='# S/P'*f=3. pctn<state all>='%'*f=5.1
                     )
      / rts=8 indent=3;
   label year     = 'Sales Year'
         region   = 'Sales Region'
         state    = 'State'
         sales    = '$ales'
         newcusts = 'New Buyers';
   keylabel sum    = 'Total'
            mean   = 'Average';
run;
```

		1995					1996				
		$ales		New Buyers	#		$ales		New Buyers	#	
		Total	Average	Count	S/P	%	Total	Average	Count	S/P	%
North											
	CT	$16,251,251	$75,237	1,830,216	34.6		$8,956,063	$74,634	951,120	26.3	
	MA	$14,404,460	$75,023	1,513,192	30.8		$8,958,733	$74,656	960,120	26.3	
	ME	$16,214,298	$75,066	1,703,216	34.6		$16,124,189	$74,649	1,719,216	47.4	
	ALL	$46,870,009	$75,112	5,046,624	100.0		$34,038,985	$74,647	3,630,456	100.0	
South											
	AL	$10,985,641	$76,289	1,157,144	30.0		$13,494,644	$74,970	1,503,180	35.7	
	FL	$13,526,146	$75,145	1,467,180	37.5		$8,846,066	$73,717	1,004,120	23.8	
	GA	$11,617,872	$74,474	1,302,156	32.5		$15,293,604	$74,969	1,659,204	40.5	
	ALL	$36,129,659	$75,270	3,926,480	100.0		$37,634,314	$74,671	4,166,504	100.0	

The other kind of percentage you can use is the PCTSUM statistic. In the example below we use PCTSUM with year as the denominator to find the relationship of the Total Sales in years 1995 and 1996.

```
proc tabulate data=tabdata format=dollar11.;
   class year region state;
   var sales newcusts;
   table region*(state all), year=' '*(
                     sales*(sum pctsum<year>='%'*f=5.1)
                     newcusts*sum='Count'*f= comma8.
                     n='# S/P'*f=3. pctn<state all>='%'*f=5.1
                     )
      / rts=8 indent=3;
   label year     = 'Sales Year'
         region   = 'Sales Region'
         state    = 'State'
         sales    = '$ales'
         newcusts = 'New Buyers';
   keylabel sum    = 'Total'
            mean   = 'Average';
run;
```

		1995					1996				
		$ales		New Buyers	#		$ales		New Buyers	#	
		Total	%	Count	S/P	%	Total	%	Count	S/P	%
North											
	CT	$16,251,251	64.5	1,830,216	34.6		$8,956,063	35.5	951,120	26.3	
	MA	$14,404,460	61.7	1,513,192	30.8		$8,958,733	38.3	960,120	26.3	
	ME	$16,214,298	50.1	1,703,216	34.6		$16,124,189	49.9	1,719,216	47.4	
	ALL	$46,870,009	57.9	5,046,624	100.0		$34,038,985	42.1	3,630,456	100.0	
South											
	AL	$10,985,641	44.9	1,157,144	30.0		$13,494,644	55.1	1,503,180	35.7	
	FL	$13,526,146	60.5	1,467,180	37.5		$8,846,066	39.5	1,004,120	23.8	
	GA	$11,617,872	43.2	1,302,156	32.5		$15,293,604	56.8	1,659,204	40.5	
	ALL	$36,129,659	49.0	3,926,480	100.0		$37,634,314	51.0	4,166,504	100.0	

Next Steps

Here are some more powerful options that I don't have time to cover in detail but you will find a need for some time in the future.

TABULATE statement options:

ORDER= Defines how your categories will be displayed, in formatted order, in internal (actual value) order, or first encountered – second encountered – third, etc.

FORMCHAR= Defines the 11 characters that are used to draw the lines for the box.

NOSEPS Turns off the horizontal lines within the data portion of the table.

TABLE statement options:

BOX= Allows you to specify text to be put into the big empty in the upper left corner of the table.

MISSTEXT= Allows you to specify the character or text to print in empty cells instead of the '.' that SAS uses for missing numeric values.

Author contact information:

LINKS ANALYTICAL, INC.
Ron Coleman
3545-1 St Johns Bluff Rd., Suite 300
Jacksonville FL 32224
(904) 641-4766
rcoleman@linksanalytical.com