Advanced Features of PROC TABULATE

** or **

The Utter Simplicity of the TABULATE Procedure - The Sequel

Dan Bruns, Tennessee Valley Authority, Chattanooga, TN

Not for Beginners

I had not given much thought to doing another TABULATE tutorial until a friend of mine (I won't mention any names but his initials are Frank D'Eloro) asked when I was going to do "The Son of PROC TABULATE?" Well, I got to looking and found some topics that I had not really addressed in either my beginning or advanced tutorials. So this "intermediate" tutorial picks up where my beginning tutorial ("The Utter Simplicity of the TABULATE Procedure") left off, and fills a gap between the advanced tutorial. We will discover how to do subtotalling, do some rather interesting formatting, titling, and labeling, see how some of the options work and interact, and last, but of course not least, BRIEFLY discuss how the PCTN and PCTSUM with the denominator specification works. This stuff is not for the weak-of-heart, nor for beginners, so just so you know a little about the dataset we will be working with.

In the above example you see a complete PROC step with a column for the count(N) and mean of SCORE for each location.

**TABLE ORG*LOC, SCORE*(N MEAN MAX PCTN);**

Note that the location field was simply moved from the column expression to the row expression.

**TABLE ORG LOC, SCORE*(N*F=5.0 MEAN MAX PCTN);**

Double your productivity with two tables in one: the N, MEAN, MAX, and PCTN statistics in the column expression allows you to use the row expression to see a summary by two different variables (ORG and LOC) in one table.
TABLE ORG=ALL
(LOC ALL)^SCORE=(N=5.0 MEAN)
/RTS=12;

You may not be able to tell from these examples, but TABULATE computes true statistics (i.e. MEAN above). That means it does not add up the means from the tables and then divide by the number of tables entries; it accumulates each observation value and divides by the number of observations with non-missing values.

**Titles and Labels**

You can see that to have TABULATE put descriptive titles or labels for the variables you simply need to assign meaningful labels to them. You can either do this in earlier steps that create the dataset with a LABEL statement in the PROC step. But what about the statistics and ALL? Simply attach a descriptive label to any variable or statistic right in the TABULATE statement. Follow it with an equals sign (=) and a quoted label (This is a label 'just like you do in a LABEL statement. Or if you want to use a certain label for every use of the statistic, use the KEYLABEL statement which looks exactly like the LABEL statement except you use the statistic's name instead of a variable name. Following is an example of doing both.

**KEYLABEL**
N='Count'
MEAN='Avg'
ALL='Total';

**TABLE ORG='Department' ALL,
(LOC ALL)^Score=(N=5.0 MEAN)
/RTS=12 BOX='SESUG 95';

The above example has another table option specified (BOX=) that specifies what to put in the upper-left corner box of the table. You can specify a quoted string as above or a variable as in the next example.

In the following example we added the MISSING and NOSEPS options to the PROC statement to have TABULATE treat missing values as a valid category (which it does not by default) and remove the separation lines between the rows. I also specified some table options: BOX=SCORE to label the upper-left box with the SCORE variable's label, and MISSTEXT='None' to label missing values in the tables with the text 'None' instead of the standard period.

**PROC TABULATE DATA='CLASS MISSING NOSEPS';
CLASS ORG LOC DATE;
VAR SCORE;
LABEL ORG='Department'
SCORE='Final Exam Averages';
KEYLABEL N='Count'
MEAN='Avg'
ALL='Total';

**TABLE ORG ALL=Total
(LOC ALL)^Score=(N=5.0 MEAN)
/RTS=12 BOX=SCORE ROW=FLOAT MISSTEXT='None';

Notice that since the MEAN label was blank and the ROW=FLOAT was specified, that no space is wasted for it. Basically ROW=FLOAT removes rows in the table that are blank and "floats" the rest of the lines up. Also notice the additional row and column for the "blank" (missing) organization and location that has appeared because of the MISSING option on
the PROC statement. Even though the MISSING options created the column the “None” indicates the scores are missing.

The label “Final Exam Averages” appears several times and seems unnecessary, so let’s get rid of it out of the column label and make a table that doesn’t look like a table.

```
PROC TABULATE DATA = CLASS MISSING NOSEPS FORMCHAR=' ';
CLASS ORG LOC DATE;
VAR SCORE;
LABEL ORG = "Department";
SCORE = "Final Exam Averages";
KEYLABEL N = "Count"
MEAN = "Avg"
ALL = "Total";
TABLE ORG ALL = " Totals ",
(LOC ALL = " Dept Totals ")
*(SCORE = "MEAN" = F= 6.1)
/ BOX = SCORE ROW = FLOAT MISSTEXT = "None";
```

<table>
<thead>
<tr>
<th>Final Exam Averages</th>
<th>Location</th>
<th>Department</th>
<th>Energy</th>
<th>Hgt S</th>
<th>Power</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>None</td>
<td>85.4</td>
<td>75.4</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>None</td>
<td>73.4</td>
<td>79.0</td>
<td>81.9</td>
<td>84.8</td>
</tr>
</tbody>
</table>

By simply adding the FORMCHAR= option to the PROC statement and specifying 16 blanks, you change all the characters that TABULATE uses to “draw” the box or frame to blanks. If you have access to a laser printer you can also use characters that form “solid” lines around your table. And like in any other SAS procedure, you can use user-defined formats to enhance the output.

```
PROC FORMAT;
  value Snums
  "A" = "Addams"  "B" = "Rettejuice"
  "C" = "Carmen"  "D" = "None"

PROC TABULATE DATA = CLASS MISSING NOSEPS FORMCHAR = TABFACCBBCBEFBCABCB
  BB4E7E4F60AFE04C6E401X; /* for IBM 3820 */
CLASS ORG LOC DATE;
VAR SCORE;
LABEL ORG = "Department";
SCORE = "Final Exam Averages";
KEYLABEL N = "Count"
MEAN = "Avg"
ALL = "Total";
TABLE ORG ALL = " Totals ",
(LOC ALL = " Dept Totals ")
*(SCORE = "MEAN" = F= 6.1)
/ BOX = SCORE ROW = FLOAT MISSTEXT = "None";
```

Can you tell what the three lines for each date represent? Only the label in the box gives us a clue.

```
TABLE SCORE = "(MEAN"
DATE = "Final Exam: Avg/Max/Min"
/ RTS = 15 ROW = FLOAT ;
```

This one is rather interesting, but I haven’t found a way to get rid of the row titles. But you can “hide” them by using the FORMCHAR = ’ ’ and NOSEPS options.

```
PROC TABULATE DATA = CLASS MISSING FORMAT = 6.1;
CLASS ORG LOC DATE;
VAR SCORE;
LABEL ORG = "Department";
SCORE = "Final Exam Averages";
KEYLABEL N = "Count"
MEAN = "Avg"
ALL = "Total";
TABLE DATE SCORE = "(MEAN"
LOC
/ RTS = 15 ROW = FLOAT
BOX = "Final Exam: Avg/Max/Min" ;
```

<table>
<thead>
<tr>
<th>Final Exam: Avg/Max/Min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>07APR 03MAY 22JUN 12OCT</td>
</tr>
<tr>
<td>85.41 90.31 81.01 90.81</td>
</tr>
<tr>
<td>99.41 95.51 98.31 100.01</td>
</tr>
<tr>
<td>51.31 85.41 64.61 47.31</td>
</tr>
</tbody>
</table>

The combination of these options has proved to be a real challenge, but can produce some very interesting reports.
Or you can just use a label in the row title.

**TABLE**

```
TABLE SCORE = "(MEAN MAX MIN)*F = 5.1, DATE = "LOC = " / RTS = 12 ROW = FLOAT BOX = "Final Exam Scores" ;
```

### TABLE ORG (LOC DATE)

```
Above we see the nested ORG value.
```

Here is a table that I wasn't sure could be done!

**TABLE**

```
TABLE ORG (LOC DATE ALL = "Subtotal") ALL = "All Departments", SCORE = "MEAN = 6.1" / RTS = 25 BOX = "SCORE"
```

### Subtotalling

The only real trick to doing subtotalling is the nesting of ALL in the row expression.

**TABLE**

```
TABLE ORG (LOC ALL = "Subtotal") ALL = "Dept Total", SCORE = "MEAN = 6.1" / RTS = 25 BOX = "SCORE"
```

```
ROW = FLOAT MISSTEXT = "None";
```

### Formatting

Just about anywhere you can specify a format (FORMAT = on the PROC statement or F = in the table definitions) you can use a SAS output format, i.e. DOLLAR12.2, COMMA10.0, etc., or a user-defined format.

**PROC FORMAT**

```
VALUE GRADE
LOW -60 = "F";
60 -70 = "D";
70 -80 = "C";
80 -90 = "B";
90 -HIGH = "A";
```

**PROC TABULATE DATA = CLASS FORMAT = 2.9:**

**CLASS** ORG LOC DATE SCORE;

**FORMAT** SCORE GRADE. ;

**KEYLABEL N=""**

```
TABLE ORG ALL (LOC ALL) SCORE = "Grade" / RTS = 10 ROW = FLOAT;
```

The same table as earlier, but we have added DATE to the grouping for subtotalling. The computation of the subtotal is not affected by the additional rows for DATE since all it has to total is the MEAN of SCORE.

### Advanced Tutorials

#### Advanced Tutorials

Here is a table that I wasn't sure could be done!

**TABLE**

```
TABLE ORG (LOC DATE ALL = "Subtotal") ALL = "All Departments", SCORE = "MEAN = 6.1" / RTS = 25 BOX = "SCORE"
```

```
ROW = FLOAT MISSTEXT = "None";
```

### Advanced Tutorials

#### Advanced Tutorials
The only real problem is the grades are backwards. To tell TABULATE to use the formatted values in ordering the columns, use the ORDER=FORMATTED option.

PROC TABULATE DATA=CLASS FORMAT=2.0 ORDER=FORMATTED;
CLASS ORG LOC DATE SCORE;
FORMAT SCORE GRDE;
KEYLABEL N=''
TABLE ORG ALL (LOC ALL)*SCORE="Grade"
/RTS=10 ROW=FLOAT;

TABLE ORG DATE ALL='-' Totals=","(LOC='ALL=') SCORE="Mean='F'=grade;")
/ BOX=SCORE ROW=FLOAT MISSTEXT="None"

This question always seems to come up: "How do I know where to put the format specification? In a FORMAT statement or a F= option?"

Here is a handy rule-of-thumb:
Use a FORMAT statement to format a CLASS variable; and use a format option (F=) to format an ANALYSIS variable.

Percentages
In its simplest form the PCTN or PCTSUM is just another statistic like N or MEAN you can request.

PROC TABULATE DATA=CLASS FORMAT=6.1;
CLASS ORG LOC DATE SCORE;
VAR SCORE;
TABLE ORG,(LOC ALL)*(N*F=3.0 PCTN);

Unless otherwise specified, the percentage is computed based on all the observations in the dataset. Notice that the PCTN under the ALL column does not add up to 100 due to rounding.

To specify how the percentage is computed you simply attach a denominator specification to PCTN or PCTSUM using the inequality signs less-than (<) and greater-than (>). The real trick to understanding how the denominator specification works is to remember you are telling TABULATE what the denominator is to divide into the N or SUM value.

<table>
<thead>
<tr>
<th>Final Exam</th>
<th>Location</th>
<th>BEST</th>
<th>DEPT</th>
<th>ADD</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>None</td>
<td>None</td>
<td>B</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Mat S</td>
<td>None</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>None</td>
<td>B</td>
<td>C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>--- Totals ---</td>
<td>None</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

But this looks rather ridiculous, so simply give the format some width and "POOP", its fixed.

<table>
<thead>
<tr>
<th>Final Exam Averages Location</th>
<th>BEST</th>
<th>DEPT</th>
<th>ADD</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>None</td>
<td>None</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Mat S</td>
<td>None</td>
<td>C</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td>None</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>--- Totals ---</td>
<td>None</td>
<td>B</td>
<td>A</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Department</th>
<th>N</th>
<th>PCTN</th>
<th>N</th>
<th>PCTN</th>
<th>N</th>
<th>PCTN</th>
<th>N</th>
<th>PCTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>6</td>
<td>22.21</td>
<td>7</td>
<td>7.41</td>
<td>6</td>
<td>22.21</td>
<td>7</td>
<td>7.41</td>
</tr>
<tr>
<td>Mat S</td>
<td>6</td>
<td>30.00</td>
<td>7</td>
<td>7.41</td>
<td>6</td>
<td>30.00</td>
<td>7</td>
<td>7.41</td>
</tr>
<tr>
<td>Power</td>
<td>6</td>
<td>30.00</td>
<td>7</td>
<td>7.41</td>
<td>6</td>
<td>30.00</td>
<td>7</td>
<td>7.41</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Department</th>
<th>N</th>
<th>PCTN</th>
<th>N</th>
<th>PCTN</th>
<th>N</th>
<th>PCTN</th>
<th>N</th>
<th>PCTN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>6</td>
<td>22.21</td>
<td>7</td>
<td>7.41</td>
<td>6</td>
<td>22.21</td>
<td>7</td>
<td>7.41</td>
</tr>
<tr>
<td>Mat S</td>
<td>6</td>
<td>30.00</td>
<td>7</td>
<td>7.41</td>
<td>6</td>
<td>30.00</td>
<td>7</td>
<td>7.41</td>
</tr>
<tr>
<td>Power</td>
<td>6</td>
<td>30.00</td>
<td>7</td>
<td>7.41</td>
<td>6</td>
<td>30.00</td>
<td>7</td>
<td>7.41</td>
</tr>
</tbody>
</table>
The above example shows the row expression in the denominator specification. Notice that none of the counts(N) have changed but the PCTN values have because the denominator has changed from the entire dataset (27 observations) to all the observations for ORG within that columns(LOC) value. Notice that since PCTN is nested in LOC that the denominator specification is saying to divide each cell under that location by the total number of observations that are in that location. So why do you specify the row expression? Because that is simply telling TABULATE which number of observations to total. So, in the above example, we see that location A cells are divided by 12, the total of all the ORG observations in that location. For location B we see each cell is divided by 8, the total of all the ORG observations in that location. And for location C we see each cell is divided by 10, the total of all the ORG observations in that location. And for the ALL column we see each cell is divided by the total of all the ORG observations in all the locations.

Here is a handy rule-of-thumb:

To get percentages by column, use the row expression; and to get percentages by row, use the column expression.

```
TABLE ORG, (LOC ALL) \N*F=3.0 PCTN < LOC ALL>

<table>
<thead>
<tr>
<th>Location</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>6</td>
<td>75.0</td>
<td>2</td>
<td>25.0</td>
</tr>
<tr>
<td>Mgt S</td>
<td>4</td>
<td>28.0</td>
<td>11</td>
<td>10.0</td>
</tr>
<tr>
<td>Power</td>
<td>4</td>
<td>44.4</td>
<td>22</td>
<td>22.2</td>
</tr>
</tbody>
</table>
```

Notice in the above example that the entire column expression is coded as the denominator specification. If you don’t, strange results or even errors can occur. As before, you are simply telling TABULATE which number of observations to total. So, in the above example, we see that organization ‘Energy’ cells are divided by 8, the total of all the LOC observations in that organization. For organization ‘Mgt S’ we see each cell is divided by 10, the total of all the LOC observations in that organization. For organization ‘Power’ we see each cell is divided by 9, the total of all the LOC observations in that organization. And for the ALL column we see each cell is divided by the total of all the LOC observations in that organization, thus the 100 percent.

I’ll leave you with the same example as above except using PCTSUM instead of PCTN, just so you can see it works the same way.

```
TABLE ORG, (LOC ALL) \N*F=3.0 PCTSUM < LOC ALL>

<table>
<thead>
<tr>
<th>Location</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>357.6</td>
<td>65.7</td>
<td>119.2</td>
<td>36.3</td>
</tr>
<tr>
<td>Mgt S</td>
<td>146.7</td>
<td>17.8</td>
<td>85.4</td>
<td>10.5</td>
</tr>
<tr>
<td>Power</td>
<td>356.2</td>
<td>48.5</td>
<td>141.5</td>
<td>19.2</td>
</tr>
</tbody>
</table>
```

This is hardly BRIEF as promised, but hopefully enough to get you started using percentages. The more complicated your “crossings”, as the SAS manuals refer to them, the tougher it is going to be to determine the denominator specification.

### In Summary

This paper is not intended to be a cure for all your TABULATE problems. Every use of TABULATE is unique in some ways. All I have attempted to do is give you a good starting point or foundation to better understand how to get TABULATE to give you what you want.

So good luck and happy tabulating!!

### Acknowledgements


SAS is a registered trademark of the SAS Institute, Inc., Cary, NC.

### Author

If you have any questions or comments, please write or call:

Dan Bruns
Tennessee Valley Authority
1101 Market Street(MP 5A)
Chattanooga, TN 37402
(423) 751-6436 FAX: (423) 751-3163
Email: spdeb@mhs-tva.attnmail.com