“A Hands-On Introduction to PROC TABULATE”

Thomas J. Winn, Jr.
Austin, Texas

Abstract

PROC TABULATE is used to build tabular reports containing descriptive statistical information, including hierarchical relationships among variables. The code that is used to invoke PROC TABULATE is complicated, and much of it looks quite different from other SAS procedures. Nevertheless, it is well worth the necessary investment of time and effort for learning the intricacies and subtleties of its syntax. Besides providing a simplified, step-by-step approach for coding PROC TABULATE, the hands-on workshop that accompanies this paper also will provide some practical experiences for participants.

Introduction

PROC TABULATE is used to build tabular reports containing descriptive statistical information, including hierarchical relationships among variables. PROC TABULATE is the SAS® System’s implementation of TPL (Table Producing Language), which was developed at the U.S. Bureau of Labor Statistics during the 1970s, for generating tabular reports of descriptive statistics involving employment data.

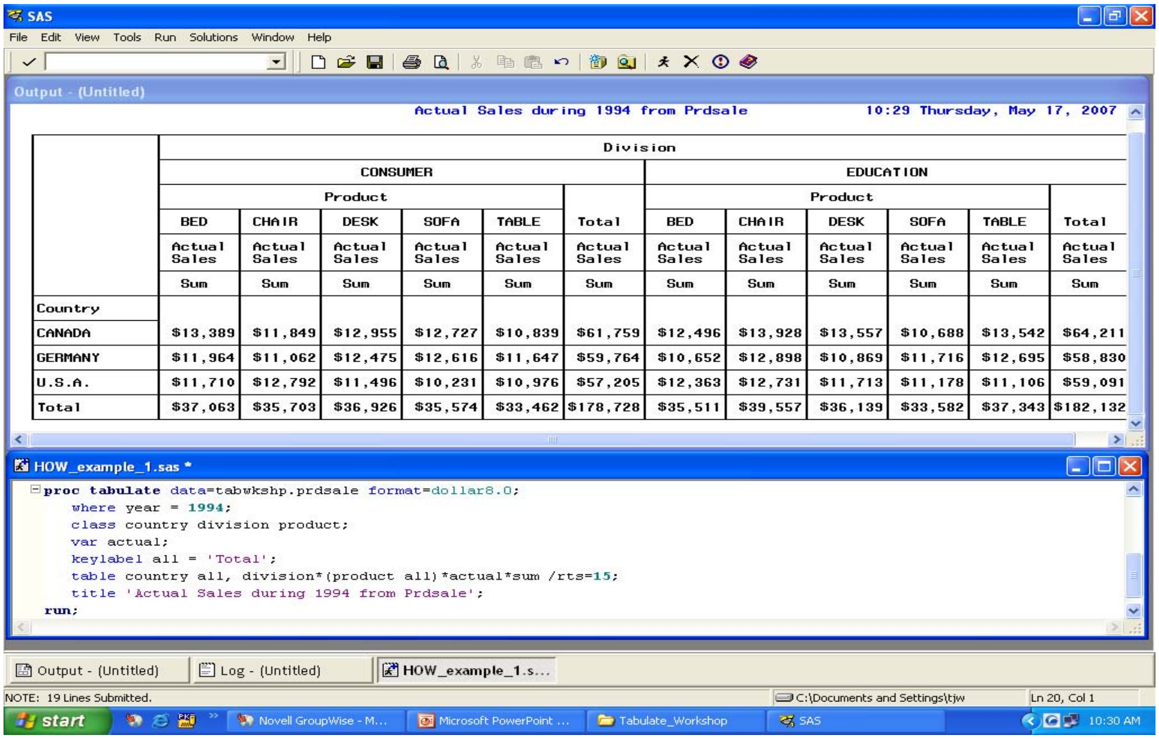
PROC TABULATE is more powerful for producing tabulations than PROC FREQ, and it is a more flexible statistical report writer than PROC MEANS. Although PROC TABULATE and PROC REPORT are both capable of generating similar tabular reports in many situations, each of these procedures has strengths and weaknesses. PROC TABULATE seems to be better for displaying hierarchical relationships. The syntax used to invoke PROC TABULATE and PROC REPORT differ from one another, and both are complicated. However, it is well worth the necessary investment of time and effort for learning the intricacies and subtleties of coding both procedures. This workshop will cover the fundamentals of coding PROC TABULATE.

The examples and exercises in this workshop make use of the SAS**®** data files, *prdsale* (from the SAS 9.1 *sashelp* data library), and *empldata* (an inner join of the *empinfo, jobcodes,* and *salary* SAS® data sets, from the SAS 9.1 *sample* data library).

Getting Started With PROC TABULATE

The first thing the would-be PROC TABULATE programmer needs to do, *before* beginning to write any SAS code, is to decide what the final report should look like. What is the intended design for the pages, rows, and columns of the report?

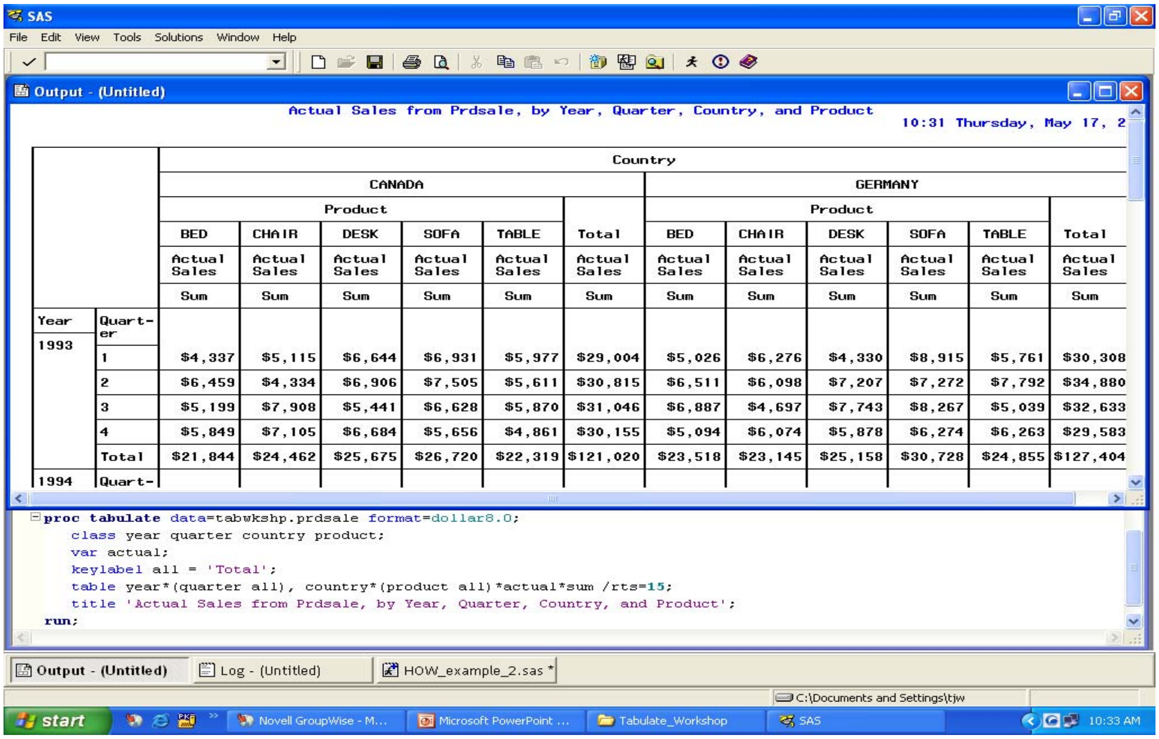
Here are a few examples of uses of PROC TABULATE:



```
                               Actual Sales during 1994 from Prdsale          10:29 Thursday, May 17, 2007

            |                                              Division
            |              CONSUMER                               |                     EDUCATION
            |              Product                      |         |              Product                      |
            |  BED    | CHAIR   | DESK    | SOFA    | TABLE   |  Total  |  BED    | CHAIR   | DESK    | SOFA    | TABLE   |  Total
            | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual
            | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales
            |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum
Country     |         |         |         |         |         |         |         |         |         |         |         |
CANADA      | $13,389 | $11,849 | $12,955 | $12,727 | $10,839 | $61,759 | $12,496 | $13,928 | $13,557 | $10,688 | $13,542 | $64,211
GERMANY     | $11,964 | $11,062 | $12,475 | $12,616 | $11,647 | $59,764 | $10,652 | $12,898 | $10,869 | $11,716 | $12,695 | $58,830
U.S.A.      | $11,710 | $12,792 | $11,496 | $10,231 | $10,976 | $57,205 | $12,363 | $12,731 | $11,713 | $11,178 | $11,106 | $59,091
Total       | $37,063 | $35,703 | $36,926 | $35,574 | $33,462 |$178,728 | $35,511 | $39,557 | $36,139 | $33,582 | $37,343 |$182,132
```

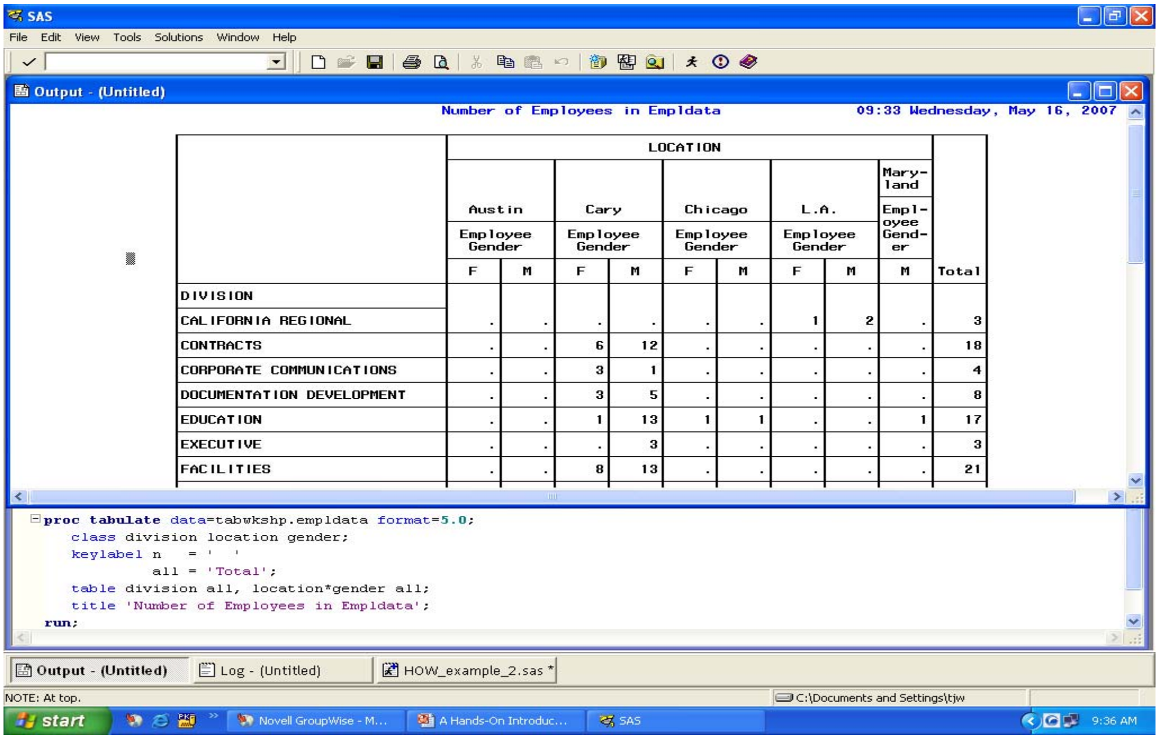
```
proc tabulate data=tabwkshp.prdsale format=dollar8.0;
    where year = 1994;
    class country division product;
    var actual;
    keylabel all = 'Total';
    table country all, division*(product all)*actual*sum /rts=15;
    title 'Actual Sales during 1994 from Prdsale';
run;
```

NOTE: 19 Lines Submitted.



```
                       Actual Sales from Prdsale, by Year, Quarter, Country, and Product
                                                                         10:31 Thursday, May 17, 2

            |                                               Country
            |              CANADA                                 |                     GERMANY
            |              Product                      |         |              Product                      |
            |  BED    | CHAIR   | DESK    | SOFA    | TABLE   |  Total  |  BED    | CHAIR   | DESK    | SOFA    | TABLE   |  Total
            | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual  | Actual
            | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales   | Sales
            |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum    |  Sum
Year |Quart-|         |         |         |         |         |         |         |         |         |         |         |
     |er    |         |         |         |         |         |         |         |         |         |         |         |
1993 |1     |  $4,337 |  $5,115 |  $6,644 |  $6,931 |  $5,977 | $29,004 |  $5,026 |  $6,276 |  $4,330 |  $8,915 |  $5,761 | $30,308
     |2     |  $6,459 |  $4,334 |  $6,906 |  $7,505 |  $5,611 | $30,815 |  $6,511 |  $6,098 |  $7,207 |  $7,272 |  $7,792 | $34,880
     |3     |  $5,199 |  $7,908 |  $5,441 |  $6,628 |  $5,870 | $31,046 |  $6,887 |  $4,697 |  $7,743 |  $8,267 |  $5,039 | $32,633
     |4     |  $5,849 |  $7,105 |  $6,684 |  $5,656 |  $4,861 | $30,155 |  $5,094 |  $6,074 |  $5,878 |  $6,274 |  $6,263 | $29,583
     |Total | $21,844 | $24,462 | $25,675 | $26,720 | $22,319 |$121,020 | $23,518 | $23,145 | $25,158 | $30,728 | $24,855 |$127,404
1994 |Quart-|         |         |         |         |         |         |         |         |         |         |         |
```

```
proc tabulate data=tabwkshp.prdsale format=dollar8.0;
    class year quarter country product;
    var actual;
    keylabel all = 'Total';
    table year*(quarter all), country*(product all)*actual*sum /rts=15;
    title 'Actual Sales from Prdsale, by Year, Quarter, Country, and Product';
run;
```



```
                         Number of Employees in Empldata        09:33 Wednesday, May 16, 2007

 ------------------------------------------------------------------------------------------------
 |                                |                       LOCATION                   |       |
 |                                |--------------------------------------------------|       |
 |                                |          |          |          |          |Mary- |       |
 |                                |          |          |          |          |land  |       |
 |                                |          |          |          |          |------|       |
 |                                |  Austin  |   Cary   | Chicago  |   L.A.   |Empl- |       |
 |                                |----------+----------+----------+----------|oyee  |       |
 |                                | Employee | Employee | Employee | Employee |Gend- |       |
 |                                |  Gender  |  Gender  |  Gender  |  Gender  | er   |       |
 |                                |----------+----------+----------+----------+------|       |
 |                                |  F  |  M |  F  |  M |  F  |  M |  F  |  M |  M   | Total |
 |--------------------------------+-----+----+-----+----+-----+----+-----+----+------+-------|
 |DIVISION                        |     |    |     |    |     |    |     |    |      |       |
 |--------------------------------|     |    |     |    |     |    |     |    |      |       |
 |CALIFORNIA REGIONAL             |    .|   .|    .|   .|    .|   .|    1|   2|     .|      3|
 |--------------------------------+-----+----+-----+----+-----+----+-----+----+------+-------|
 |CONTRACTS                       |    .|   .|    6|  12|    .|   .|    .|   .|     .|     18|
 |--------------------------------+-----+----+-----+----+-----+----+-----+----+------+-------|
 |CORPORATE COMMUNICATIONS        |    .|   .|    3|   1|    .|   .|    .|   .|     .|      4|
 |--------------------------------+-----+----+-----+----+-----+----+-----+----+------+-------|
 |DOCUMENTATION DEVELOPMENT       |    .|   .|    3|   5|    .|   .|    .|   .|     .|      8|
 |--------------------------------+-----+----+-----+----+-----+----+-----+----+------+-------|
 |EDUCATION                       |    .|   .|    1|  13|    1|   1|    .|   .|     1|     17|
 |--------------------------------+-----+----+-----+----+-----+----+-----+----+------+-------|
 |EXECUTIVE                       |    .|   .|    .|   3|    .|   .|    .|   .|     .|      3|
 |--------------------------------+-----+----+-----+----+-----+----+-----+----+------+-------|
 |FACILITIES                      |    .|   .|    8|  13|    .|   .|    .|   .|     .|     21|
 |--------------------------------+-----+----+-----+----+-----+----+-----+----+------+-------|
```

```
proc tabulate data=tabwkshp.empldata format=5.0;
    class division location gender;
    keylabel n   = '  '
           all = 'Total';
    table division all, location*gender all;
    title 'Number of Employees in Empldata';
run;
```

Unfortunately, a lot of the code that is used to invoke PROC TABULATE looks quite different from the code that is used for other SAS procedures.   Here is the basic syntax for coding PROC TABULATE:

```
PROC TABULATE <option-list>;
   BY <NOTSORTED> <DESCENDING> variable-1
       < … <DESCENDING> variable-n>;
   CLASS class-variable-list;
   CLASSLEV class-variable-list / STYLE=<style-element-name>
                  <[style-attribute-specification(s)]>;
   VAR analysis-variable-list;
   TABLE <<page-expression,>  row-expression,>
            column-expression </ table-option-list>;
   FORMAT variable-list-1 format-1
               < … variable-list-n format-n>;
   FREQ variable;
   KEYLABEL keyword-1='description-1'
                  < … keyword-n="description-n'>;
   KEYWORD keyword(s) / STYLE=<style-element-name>
                  <[style-attribute-specification(s)]>;
   LABEL variable-1='label-1' <…variable-n='label-n'>;
   WEIGHT variable;
   TITLE 'text';
```

Here are some options that are frequently used in PROC TABULATE statements:

DATA=SAS-dataset-name
FORMAT=formatname
MISSING
NOSEPS

The *DATA=* option specifies the SAS dataset to be used.
The *FORMAT* option specifies a default format for each table cell. The default format is overridden by any format specified in a subsequent TABLE statement.
The *MISSING* option requests that missing values be regarded as valid levels for classification variables. Unless the MISSING option is specified, observations with missing values for class variables will not be included in the analysis.
The *NOSEPS* option removes the interior horizontal lines from the printed report.

Steps for Coding PROC TABULATE

The simplest way to approach coding for PROC TABULATE may be expressed in terms of the following five steps:

(1) Specification of classification variables,
(2) Specification of analysis variables,
(3) Definition of dimensions of the table,
(4) Identification of desired statistics, and
(5) Labeling and formatting the table.

What are Classification and Analysis Variables?

Classification variables are used to identify categorical groups on which calculations are performed. They are the variables that make up the rows and columns of the table. They may be either character or numeric. If numeric, generally only a small number of distinct values are permitted.

Analysis variables are numeric variables that are used to compute statistics that are reported in the body of the table.

Step 1 – Specification of Classification Variables

The *CLASS* statement is used to specify any categorical variables that will be used for grouping purposes in the analysis. The *CLASS* statement is required.

```
PROC TABULATE DATA=tabwkshp.prdsale;
     CLASS country division product;     . . .
```

Step 2 – Specification of Analysis Variables

The *VAR* statement is used to list any variables that will be used for computing the statistics that are to appear in the body of the table. Frequency counts can be computed without an analysis variable. However, under most conditions, the *VAR* statement is required.

```
PROC TABULATE DATA=tabwkshop.prdsale;
     CLASS country division product;
     VAR actual;   . . .
```

Step 3 – Definition of the Table's Dimensions

The *TABLE statement* is used to define both the arrangement of the rows and columns of the table, as well as the requests for any summary statistics. It is the tricky aspect of using PROC TABULATE. So let's consider this complex statement one piece at a time.

In a *TABLE* statement, the comma is a very important symbol, because it separates the dimensions of the table.

- If two commas were specified, then the table would have three dimensions, and the order would be *pages, rows*, and *columns.*
- If only one comma was specified, then the table would have two dimensions, and the order would be *rows, columns*.
- No comma would be interpreted to mean that the table's *only* dimension would be the *column* dimension. The table would only have one row.

So, *TABLE* statements look like the following:

```
TABLE page-expression, row-expression, column-expression . . .;
TABLE row-expression, column-expression . . .;     or
TABLE column-expression . . .;
```

In this context, an *expression* can consist of variables, statistics, operators, format specifications, and label assignments.

Because our immediate concern is only to define the table's *dimensions*, we are pleased to discover that only three operators are needed to specify the page, row, and column headings that identify the structure of a table.

- An asterisk **_(*)_** can be used to *cross* the classification variables; that is, to arrange them in a nested manner, according to the order listed (top, middle, and lower).
- A blank space is used to *concatenate* two classification variables (which will appear in the table: top-to-bottom for row headings, left-to-right for column headings).
- Parentheses ( ) are used to *group* the elements of an expression, and to associate an adjacent operator with each concatenated element inside the parentheses.

Here are a few simple examples:

> *TABLE var1, var2****;***
> The preceding would generate a two-dimensional table in which the row dimension would be the values of *var1*, and the column dimension would be the values of *var2*.
>
> *TABLE var1, var2 var3;*
> This would result in a two-dimensional table in which the row dimension would be the values of *var1*, and the column dimension would be comprised of values resulting from the side-by-side concatenation of *var2* and *var3*.
>
> *TABLE var1, var2*var3;*
> This statement would generate a two-dimensional table in which the row dimension would be the values of *var1*, and the column dimension would be a hierarchical arrangement of the values of *var2* and *var3***,** with the values of *var2* comprising the top columns, and for each of these values as columns, the values of *var3* as the lower columns.

Here are three additional examples:

> *TABLE var1, var2 * (var3 var4) var5 ;*
> The preceding would generate a two-dimensional table in which the row dimension would be the values of *var1*, and the column dimension would be two side-by-side components, the first of which would be a hierarchical nesting of the values of *var2* over a side-by-side arrangement of the values of *var3* and *var4***,** and the second component would be the values of *var5*.
>
> *TABLE var1, (var2 var3) * var4 ;*
> This would result in a two-dimensional table in which the row dimension would be the values of *var1*, and the column dimension would be two side-by-side hierarchical components, the first of which would be the values of *var2* over the values of *var4***,** and the second would be the values of *var3* over the values of *var4*.
>
> *TABLE var1 var2, var3 * var4 ;*
> This code would generate a two-dimensional table in which the row dimension would be the concatenated values of *var1* and *var2*, and the column dimension would be a hierarchical nesting of the values of *var3* over the values of *var4*.

Step 4 – Identification of Desired Statistics

We have been considering incomplete *TABLE* statements (because up to this point we have only focused on the *dimensions* of the tables).

Besides specifying the dimensions, the *TABLE* statement also identifies which <u>summary statistics</u> should be produced, and pertaining to which analysis variables. Each statistic is identified by a *keyword*.

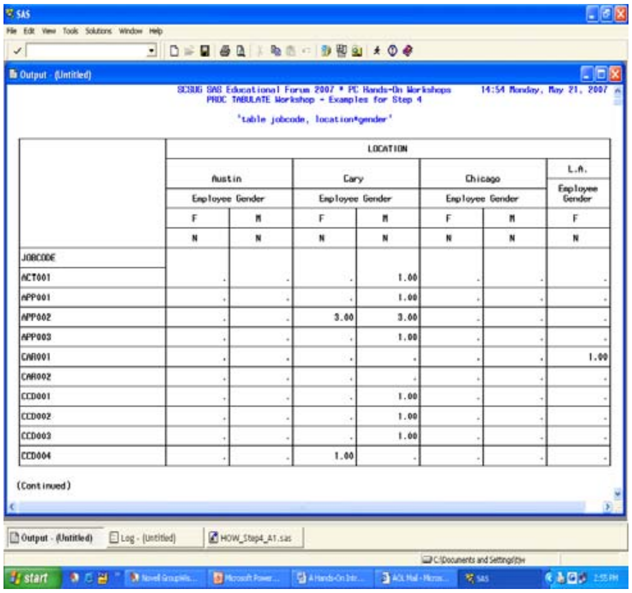
N = the number of observations, the frequency count
MIN = the smallest value
MAX = the largest value
MEAN = the arithmetic mean, or the average value
STD = the standard deviation
VAR = the variance
MEDIAN = the middle (50th percentile) value
SKEWNESS = a measure of the asymmetry of the distribution of values
SUM = the sum of the values
PCTN = the percentage that one frequency is of another frequency
PCTSUM = the percentage that one sum is of another sum
. . . *(and other descriptive statistics)*.

Whenever you cross a variable with a keyword for a statistic, you are identifying the statistic to be applied to that variable (which tells PROC TABULATE what type of calculation to perform). You can cross classification variables only with the *N* or *PCTN* statistics. By default, if the *TABLE* statement does not include an analysis variable or a statistic, then PROC TABULATE automatically crosses the *N* statistic with the indicated class variables. Analysis variables can be crossed with any statistic. By default, if the *TABLE* statement includes an analysis variable but without crossing it with any statistic, PROC TABULATE automatically crosses it with *SUM*.

Here are some examples:

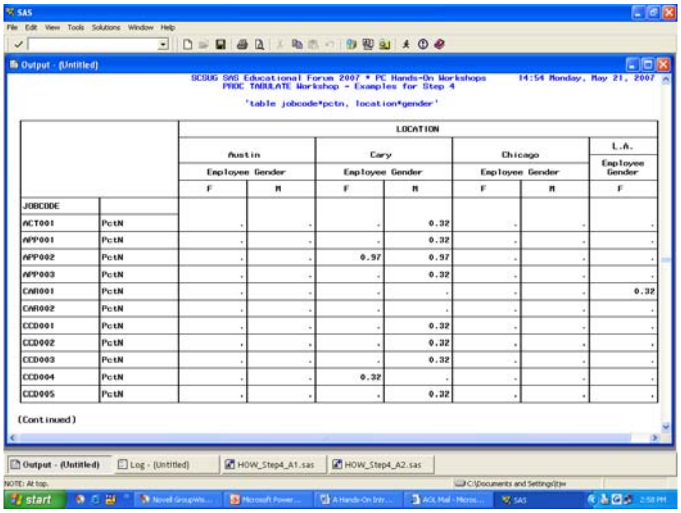
```
PROC TABULATE  DATA=tabwkshp.empldata;
      CLASS  jobcode location gender;
     TABLE  jobcode,  location*gender;
     TABLE  jobcode*PCTN,  location*gender;
     TABLE  jobcode,  location*gender*PCTN;
     TABLE  jobcode PCTN,  location* gender;
```

The first *TABLE* statement would generate a hierarchical breakdown of frequency counts in the data set, according to values of *jobcode* (the rows) and the nested values of *location* and *gender* (the columns).



```
                  SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops     14:54 Monday, May 21, 2007
                       PROC TABULATE Workshop - Examples for Step 4

                               'table jobcode, location*gender'

 -----------------------------------------------------------------------------------
 |              |                            LOCATION                              |
 |              |------------------------------------------------------------------|
 |              |                 |                 |                 |   L.A.   |
 |              |     Austin      |      Cary       |     Chicago     |----------|
 |              |-----------------+-----------------+-----------------| Employee |
 |              | Employee Gender | Employee Gender | Employee Gender |  Gender  |
 |              |-----------------+-----------------+-----------------+----------|
 |              |    F   |   M    |    F   |   M    |    F   |   M    |    F     |
 |              |--------+--------+--------+--------+--------+--------+----------|
 |              |    N   |   N    |    N   |   N    |    N   |   N    |    N     |
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |JOBCODE       |        |        |        |        |        |        |          |
 |--------------|        |        |        |        |        |        |          |
 |ACT001        |       .|       .|       .|    1.00|       .|       .|         .|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |APP001        |       .|       .|       .|    1.00|       .|       .|         .|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |APP002        |       .|       .|    3.00|    3.00|       .|       .|         .|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |APP003        |       .|       .|       .|    1.00|       .|       .|         .|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |CAR001        |       .|       .|       .|       .|       .|       .|      1.00|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |CAR002        |       .|       .|       .|       .|       .|       .|         .|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |CCD001        |       .|       .|       .|    1.00|       .|       .|         .|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |CCD002        |       .|       .|       .|    1.00|       .|       .|         .|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |CCD003        |       .|       .|       .|    1.00|       .|       .|         .|
 |--------------+--------+--------+--------+--------+--------+--------+----------|
 |CCD004        |       .|       .|    1.00|       .|       .|       .|         .|
 -----------------------------------------------------------------------------------

(Continued)
```

The second and third *TABLE* statements would generate a hierarchical breakdown of percentages represented in each cell, according to values of *jobcode* and the nested values of *location* and *gender*.

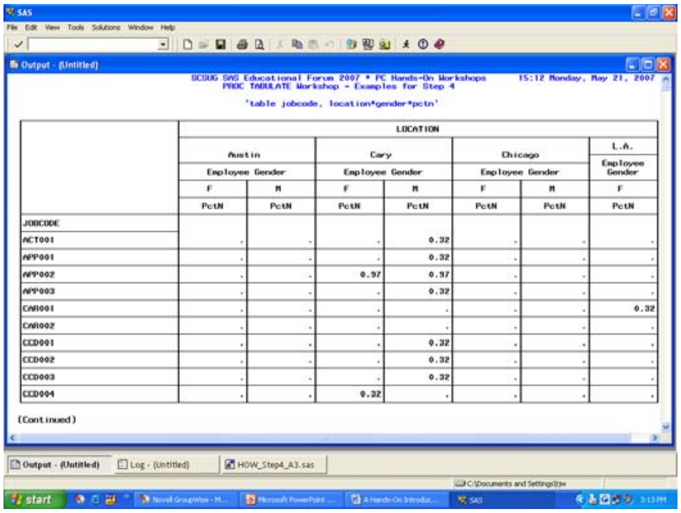


SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 14:54 Monday, May 21, 2007
PROC TABULATE Workshop - Examples for Step 4

'table jobcode*pctn, location*gender'

		LOCATION						
		Austin		Cary		Chicago		L.A.
		Employee Gender		Employee Gender		Employee Gender		Employee Gender
		F	M	F	M	F	M	F
JOBCODE								
ACT001	PctN	.	.	.	0.32	.	.	.
APP001	PctN	.	.	.	0.32	.	.	.
APP002	PctN	.	.	0.97	0.97	.	.	.
APP003	PctN	.	.	.	0.32	.	.	.
CAR001	PctN	.	.	.	.	.	.	0.32
CAR002	PctN	.	.	.	.	.	.	.
CCD001	PctN	.	.	.	0.32	.	.	.
CCD002	PctN	.	.	.	0.32	.	.	.
CCD003	PctN	.	.	.	0.32	.	.	.
CCD004	PctN	.	.	0.32	.	.	.	.
CCD005	PctN	.	.	.	0.32	.	.	.

(Continued)



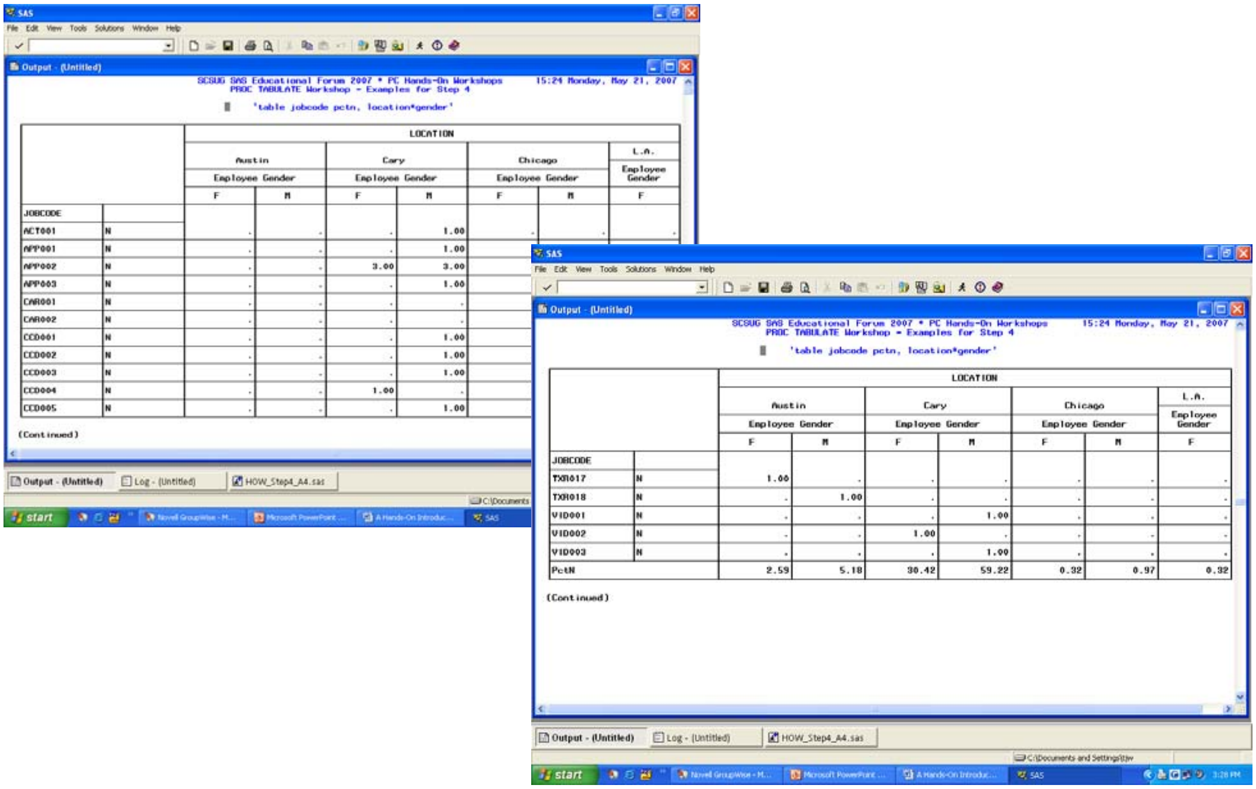
SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 15:12 Monday, May 21, 2007
PROC TABULATE Workshop - Examples for Step 4

'table jobcode, location*gender*pctn'

	LOCATION						
	Austin		Cary		Chicago		L.A.
	Employee Gender		Employee Gender		Employee Gender		Employee Gender
	F	M	F	M	F	M	F
	PctN	PctN	PctN	PctN	PctN	PctN	PctN
JOBCODE							
ACT001	.	.	.	0.32	.	.	.
APP001	.	.	.	0.32	.	.	.
APP002	.	.	0.97	0.97	.	.	.
APP003	.	.	.	0.32	.	.	.
CAR001	.	.	.	.	.	.	0.32
CAR002	.	.	.	.	.	.	.
CCD001	.	.	.	0.32	.	.	.
CCD002	.	.	.	0.32	.	.	.
CCD003	.	.	.	0.32	.	.	.
CCD004	.	.	0.32	.	.	.	.

(Continued)

The fourth *TABLE* statement would generate a hierarchical breakdown of frequency counts represented in each cell, according to values of *jobcode* and the nested values of *location* and *gender*. It would include an additional row that would represent the percentage of the total population of the data set included in each column.



SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 15:24 Monday, May 21, 2007
PROC TABULATE Workshop - Examples for Step 4

'table jobcode pctn, location*gender'

		LOCATION						
		Austin		Cary		Chicago		L.A.
		Employee Gender		Employee Gender		Employee Gender		Employee Gender
		F	M	F	M	F	M	F
JOBCODE								
ACT001	N	.	.	.	1.00	.	.	.
APP001	N	.	.	.	1.00			
APP002	N	.	.	3.00	3.00			
APP003	N	.	.	.	1.00			
CAR001	N	.	.	.	.			
CAR002	N	.	.	.	.			
CCD001	N	.	.	.	1.00			
CCD002	N	.	.	.	1.00			
CCD003	N	.	.	.	1.00			
CCD004	N	.	.	1.00	.			
CCD005	N	.	.	.	1.00			

(Continued)

SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 15:24 Monday, May 21, 2007
PROC TABULATE Workshop - Examples for Step 4

'table jobcode pctn, location*gender'

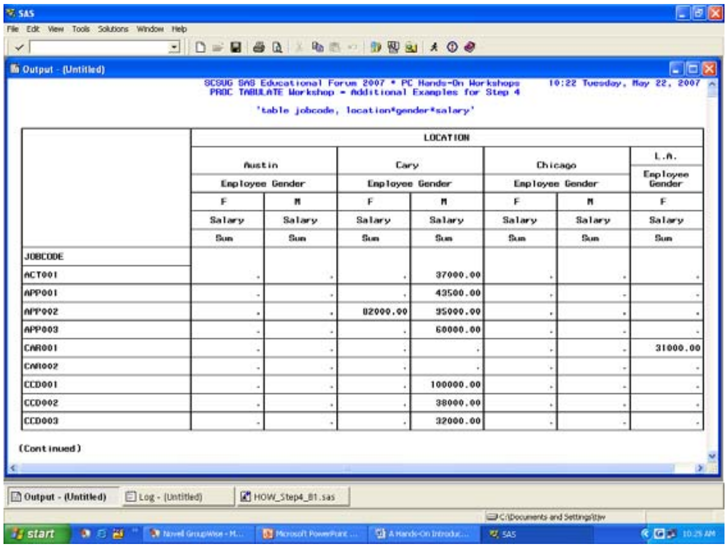
		LOCATION						
		Austin		Cary		Chicago		L.A.
		Employee Gender		Employee Gender		Employee Gender		Employee Gender
		F	M	F	M	F	M	F
JOBCODE								
TXR017	N	1.00	.	.	.	.	.	.
TXR018	N	.	1.00	.	.	.	.	.
VID001	N	.	.	.	1.00	.	.	.
VID002	N	.	.	1.00	.	.	.	.
VID003	N	.	.	.	1.00	.	.	.
PctN		2.59	5.18	30.42	59.22	0.32	0.97	0.32

(Continued)

Here are a few more examples:

```
PROC TABULATE  DATA=tabwkshp.empldata;
      CLASS jobcode location gender;
      VAR    salary;
      TABLE  jobcode, location*gender*salary;
      TABLE  jobcode, location*gender*salary*PCTSUM;
      TABLE  jobcode, location*gender*salary*MEAN;
```

The first *TABLE* statement would generate a hierarchical breakdown of the sum of the salary amounts represented in each cell, according to values of *jobcode* and the nested values of *location* and *gender*.

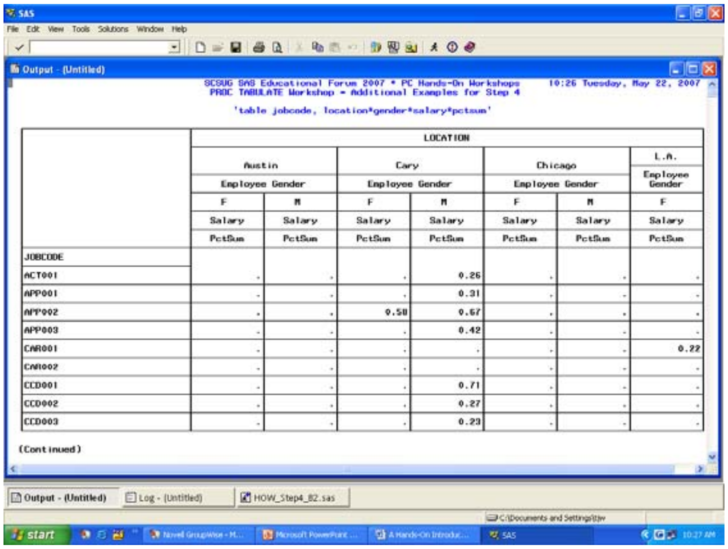


```
            SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops      10:22 Tuesday, May 22, 2007
             PROC TABULATE Workshop - Additional Examples for Step 4

                         'table jobcode, location*gender*salary'
```

	LOCATION						
	Austin		Cary		Chicago		L.A.
	Employee Gender		Employee Gender		Employee Gender		Employee Gender
	F	M	F	M	F	M	F
	Salary	Salary	Salary	Salary	Salary	Salary	Salary
	Sum	Sum	Sum	Sum	Sum	Sum	Sum
JOBCODE							
ACT001	.	.	.	37000.00	.	.	.
APP001	.	.	.	43500.00	.	.	.
APP002	.	.	82000.00	95000.00	.	.	.
APP003	.	.	.	60000.00	.	.	.
CAR001	.	.	.	.	.	.	31000.00
CAR002	.	.	.	.	.	.	.
CCD001	.	.	.	100000.00	.	.	.
CCD002	.	.	.	38000.00	.	.	.
CCD003	.	.	.	32000.00	.	.	.

(Continued)

The second *TABLE* statement would generate a hierarchical breakdown of the percentage of the total of salary amounts represented in each cell, according to values of *jobcode* and the nested values of *location* and *gender*.

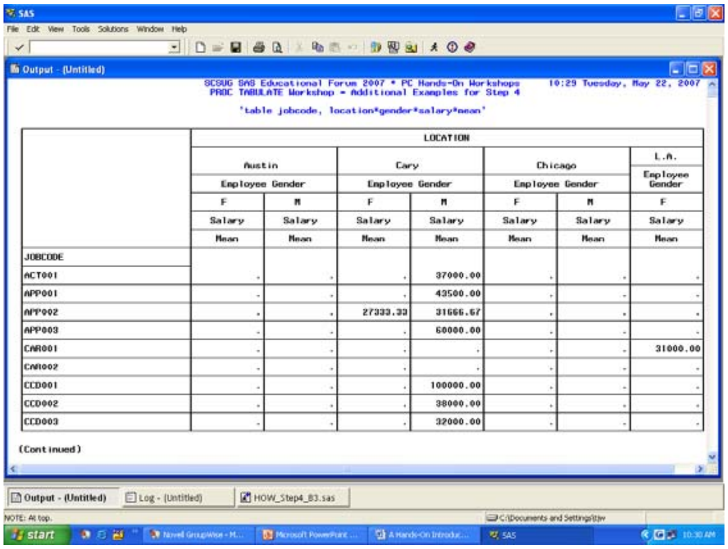


```
            SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops      10:26 Tuesday, May 22, 2007
             PROC TABULATE Workshop - Additional Examples for Step 4

                     'table jobcode, location*gender*salary*pctsum'
```

	LOCATION						
	Austin		Cary		Chicago		L.A.
	Employee Gender		Employee Gender		Employee Gender		Employee Gender
	F	M	F	M	F	M	F
	Salary	Salary	Salary	Salary	Salary	Salary	Salary
	PctSum	PctSum	PctSum	PctSum	PctSum	PctSum	PctSum
JOBCODE							
ACT001	.	.	.	0.26	.	.	.
APP001	.	.	.	0.31	.	.	.
APP002	.	.	0.58	0.67	.	.	.
APP003	.	.	.	0.42	.	.	.
CAR001	.	.	.	.	.	.	0.22
CAR002	.	.	.	.	.	.	.
CCD001	.	.	.	0.71	.	.	.
CCD002	.	.	.	0.27	.	.	.
CCD003	.	.	.	0.23	.	.	.

(Continued)

The third *TABLE* statement would generate a hierarchical breakdown of the average salary represented in each cell, according to values of *jobcode* and the nested values of *location* and *gender*.



```
SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops     10:29 Tuesday, May 22, 2007
PROC TABULATE Workshop - Additional Examples for Step 4

'table jobcode, location*gender*salary*mean'
```

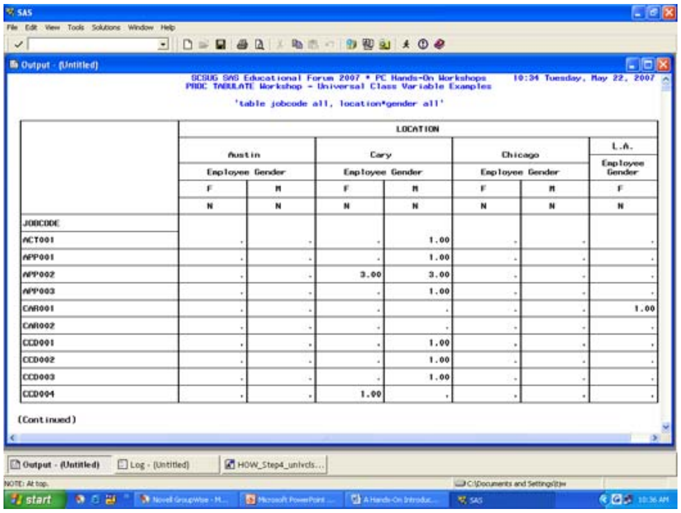
	LOCATION						
	Austin		Cary		Chicago		L.A.
	Employee Gender		Employee Gender		Employee Gender		Employee Gender
	F	M	F	M	F	M	F
	Salary	Salary	Salary	Salary	Salary	Salary	Salary
	Mean	Mean	Mean	Mean	Mean	Mean	Mean
JOBCODE							
ACT001	.	.	.	37000.00	.	.	.
APP001	.	.	.	43500.00	.	.	.
APP002	.	.	27333.33	31666.67	.	.	.
APP003	.	.	.	60000.00	.	.	.
CAR001	.	.	.	.	.	.	31000.00
CAR002	.	.	.	.	.	.	.
CCD001	.	.	.	100000.00	.	.	.
CCD002	.	.	.	38000.00	.	.	.
CCD003	.	.	.	32000.00	.	.	.

(Continued)

PROC TABULATE has a <u>*universal class variable*</u>, *ALL*, which can be used to generate totals for any specified class variable. Just concatenate the keyword *ALL* into the row or column expression of a *TABLE* statement.

```
PROC TABULATE  DATA=tabwkshp.empldata;
        CLASS jobcode location gender;
       TABLE jobcode ALL, location*gender  ALL;
       TABLE jobcode ALL, (location  ALL)*gender;
```

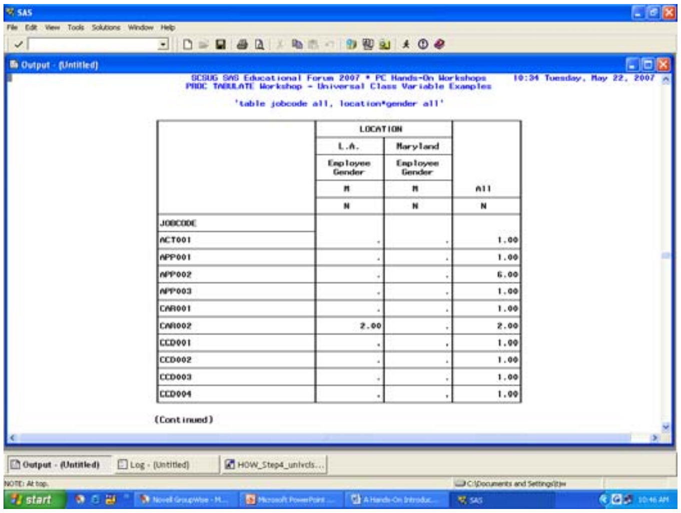
What is the difference between the tables produced by these two *TABLE* statements? Here are the results from the first *TABLE* statement:



```
SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops     10:34 Tuesday, May 22, 2007
PROC TABULATE Workshop - Universal Class Variable Examples

'table jobcode all, location*gender all'
```

	LOCATION						
	Austin		Cary		Chicago		L.A.
	Employee Gender		Employee Gender		Employee Gender		Employee Gender
	F	M	F	M	F	M	F
	N	N	N	N	N	N	N
JOBCODE							
ACT001	.	.	.	1.00	.	.	.
APP001	.	.	.	1.00	.	.	.
APP002	.	.	3.00	3.00	.	.	.
APP003	.	.	.	1.00	.	.	.
CAR001	.	.	.	.	.	.	1.00
CAR002	.	.	.	.	.	.	.
CCD001	.	.	.	1.00	.	.	.
CCD002	.	.	.	1.00	.	.	.
CCD003	.	.	.	1.00	.	.	.
CCD004	.	.	1.00	.	.	.	.

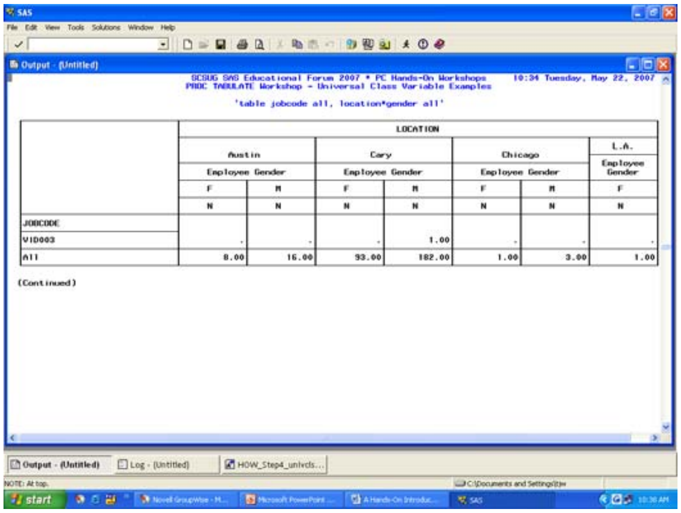
(Continued)



```
SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops     10:34 Tuesday, May 22, 2007
PROC TABULATE Workshop - Universal Class Variable Examples

'table jobcode all, location*gender all'
```

	LOCATION		
	L.A.	Maryland	
	Employee Gender	Employee Gender	
	M	M	All
	N	N	N
JOBCODE			
ACT001	.	.	1.00
APP001	.	.	1.00
APP002	.	.	6.00
APP003	.	.	1.00
CAR001	.	.	1.00
CAR002	2.00	.	2.00
CCD001	.	.	1.00
CCD002	.	.	1.00
CCD003	.	.	1.00
CCD004	.	.	1.00

(Continued)

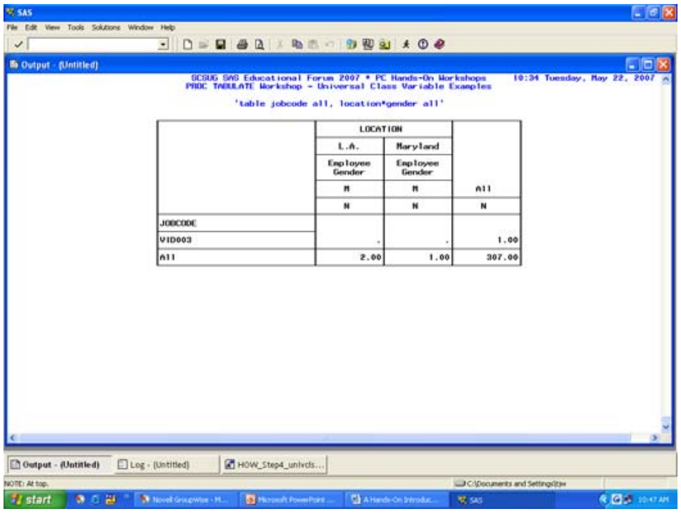


SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 10:34 Tuesday, May 22, 2007
PROC TABULATE Workshop - Universal Class Variable Examples

'table jobcode all, location*gender all'

	LOCATION						
	Austin		Cary		Chicago		L.A.
	Employee Gender		Employee Gender		Employee Gender		Employee Gender
	F	M	F	M	F	M	F
	N	N	N	N	N	N	N
JOBCODE							
VID003	.	.	.	1.00	.	.	.
All	8.00	16.00	93.00	182.00	1.00	3.00	1.00

(Continued)

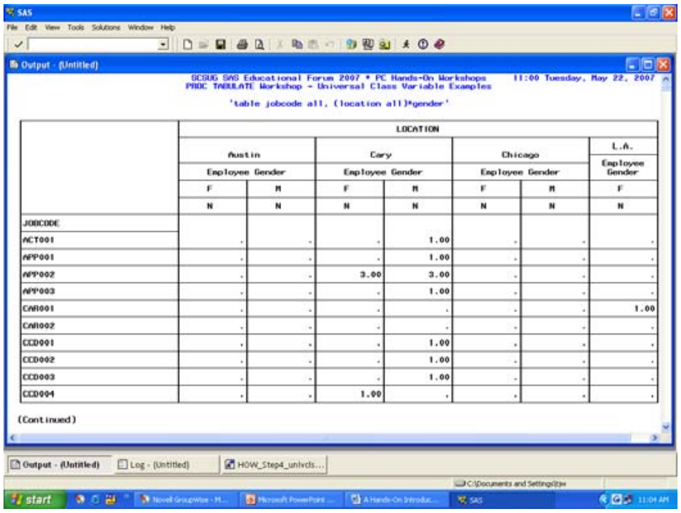


SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 10:34 Tuesday, May 22, 2007
PROC TABULATE Workshop - Universal Class Variable Examples

'table jobcode all, location*gender all'

	LOCATION		
	L.A.	Maryland	
	Employee Gender	Employee Gender	
	M	M	All
	N	N	N
JOBCODE			
VID003	.	.	1.00
All	2.00	1.00	307.00

And here are the results from the second TABLE statement:

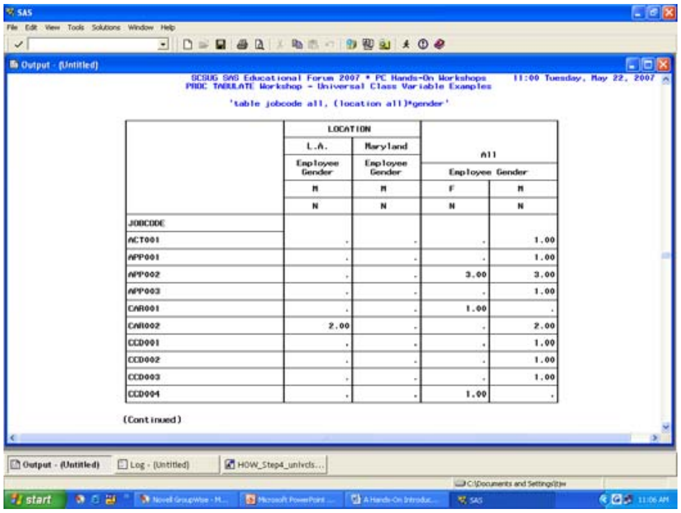


SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 11:00 Tuesday, May 22, 2007
PROC TABULATE Workshop - Universal Class Variable Examples

'table jobcode all, (location all)*gender'

	LOCATION						
	Austin		Cary		Chicago		L.A.
	Employee Gender		Employee Gender		Employee Gender		Employee Gender
	F	M	F	M	F	M	F
	N	N	N	N	N	N	N
JOBCODE							
ACT001	.	.	.	1.00	.	.	.
APP001	.	.	.	1.00	.	.	.
APP002	.	.	3.00	3.00	.	.	.
APP003	.	.	.	1.00	.	.	.
CAR001	.	.	.	.	.	.	1.00
CAR002	.	.	.	.	.	.	.
CCD001	.	.	.	1.00	.	.	.
CCD002	.	.	.	1.00	.	.	.
CCD003	.	.	.	1.00	.	.	.
CCD004	.	.	1.00	.	.	.	.

(Continued)

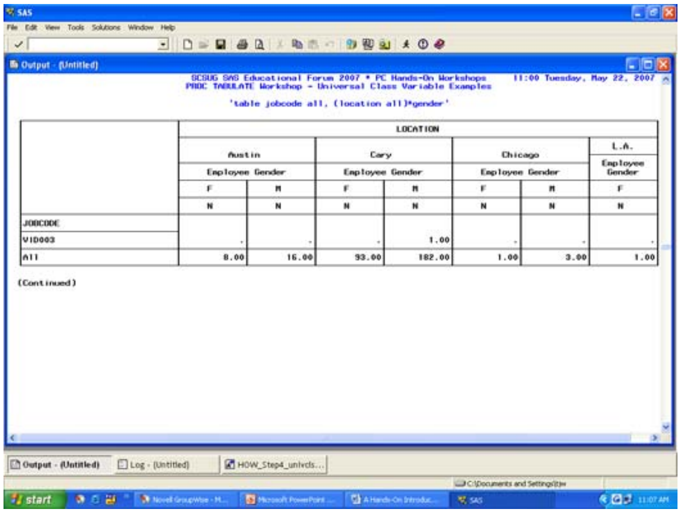


SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 11:00 Tuesday, May 22, 2007
PROC TABULATE Workshop - Universal Class Variable Examples

'table jobcode all, (location all)*gender'

	LOCATION			
	L.A.	Maryland	All	
	Employee Gender	Employee Gender	Employee Gender	
	M	M	F	M
	N	N	N	N
JOBCODE				
ACT001	.	.	.	1.00
APP001	.	.	.	1.00
APP002	.	.	3.00	3.00
APP003	.	.	.	1.00
CAR001	.	.	1.00	.
CAR002	2.00	.	.	2.00
CCD001	.	.	.	1.00
CCD002	.	.	.	1.00
CCD003	.	.	.	1.00
CCD004	.	.	1.00	.

(Continued)

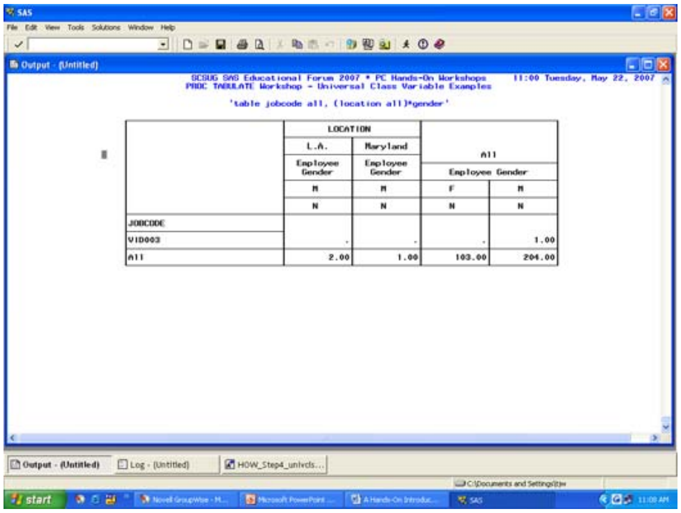


SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 11:00 Tuesday, May 22, 2007
PROC TABULATE Workshop - Universal Class Variable Examples

'table jobcode all, (location all)*gender'

	LOCATION						
	Austin		Cary		Chicago		L.A.
	Employee Gender		Employee Gender		Employee Gender		Employee Gender
	F	M	F	M	F	M	F
	N	N	N	N	N	N	N
JOBCODE							
VID003	.	.	.	1.00	.	.	.
All	8.00	16.00	93.00	182.00	1.00	3.00	1.00

(Continued)



SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 11:00 Tuesday, May 22, 2007
PROC TABULATE Workshop - Universal Class Variable Examples

'table jobcode all, (location all)*gender'

	LOCATION			
	L.A.	Maryland	All	
	Employee Gender	Employee Gender	Employee Gender	
	M	M	F	M
	N	N	N	N
JOBCODE				
VID003	.	.	.	1.00
All	2.00	1.00	103.00	204.00

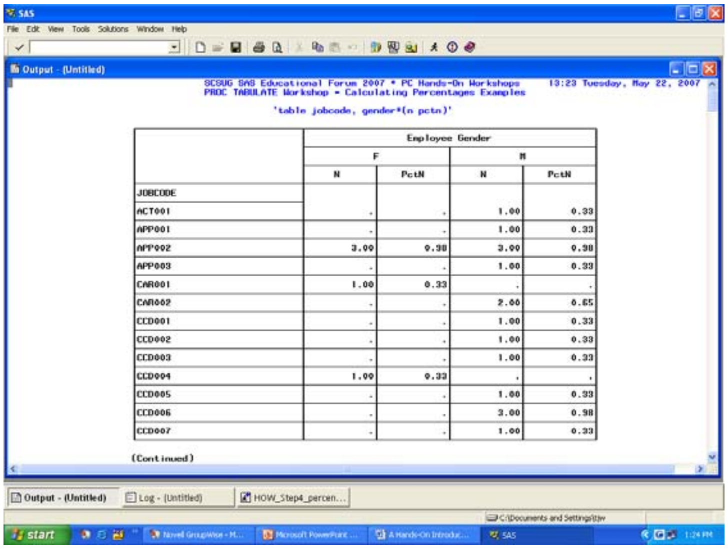
Calculating Percentages with PROC TABULATE:

PROC TABULATE has the capability of determining the percentage of the value in one cell to the value in another cell, or to the total of a group of cells, through the *PCTN* and *PCTSUM* statistics, respectively. Whenever you invoke PROC TABULATE to calculate percentages you, explicitly or implicitly, define the value that is to be used as the denominator. If you do not specify a particular denominator then, by default, PROC TABULATE uses the sum of the values in the *N* cells for the *PCTN* denominator, and the sum of the values in the *SUM* cells for the *PCTSUM* denominator.

Brackets < > are used to explicitly specify the denominator that is to be used in the calculation of percentages.

```
PROC TABULATE  DATA=tabwkshp.empldata;
      CLASS  jobcode  gender;
      TABLE  jobcode, gender*(N PCTN) ;
      TABLE  jobcode, gender*(N PCTN<gender>) ;
```

The first *TABLE* statement specifies that the display should include the number of instances of values for *gender* occurring with each value of *jobcode*, and the percentages of those numbers to the total across all combinations of values of *jobcode* and *gender*.



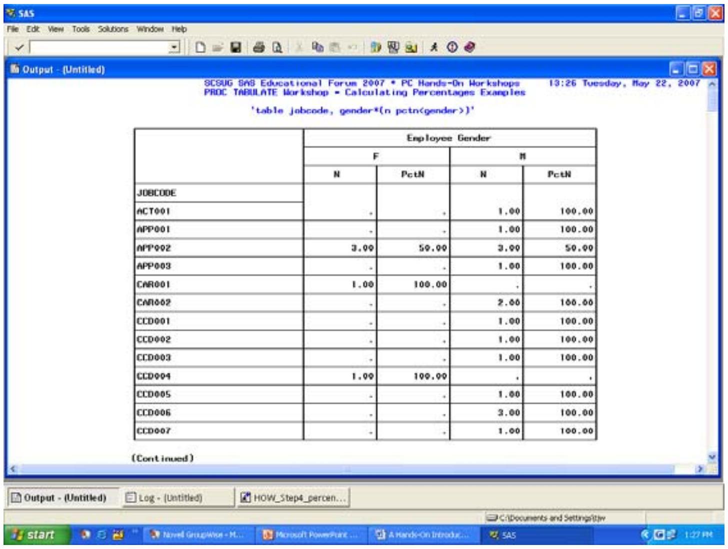
SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops 13:23 Tuesday, May 22, 2007
PROC TABULATE Workshop - Calculating Percentages Examples

'table jobcode, gender*(n pctn)'

	Employee Gender			
	F		M	
	N	PctN	N	PctN
JOBCODE				
ACT001	.	.	1.00	0.33
APP001	.	.	1.00	0.33
APP002	3.00	0.98	3.00	0.98
APP003	.	.	1.00	0.33
CAR001	1.00	0.33	.	.
CAR002	.	.	2.00	0.65
CCD001	.	.	1.00	0.33
CCD002	.	.	1.00	0.33
CCD003	.	.	1.00	0.33
CCD004	1.00	0.33	.	.
CCD005	.	.	1.00	0.33
CCD006	.	.	3.00	0.98
CCD007	.	.	1.00	0.33

(Continued)

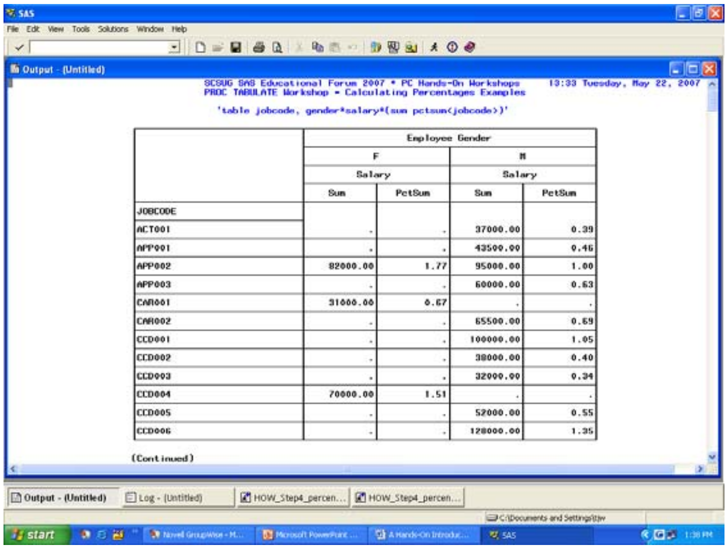
The second *TABLE* statement specifies that the display should include the number of occurrences of values for *gender*, and the percentage of that number to the total for *all values* of *gender* in each *jobcode* (that is, a row-percentage).



```
                 SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops     13:26 Tuesday, May 22, 2007
                 PROC TABULATE Workshop - Calculating Percentages Examples

                       'table jobcode, gender*(n pctn<gender>)'

         ------------------------------------------------------
         |                  |         Employee Gender         |
         |                  |---------------------------------|
         |                  |        F       |        M       |
         |                  |----------------+----------------|
         |                  |   N    |  PctN |   N    |  PctN |
         |------------------+--------+-------+--------+-------|
         |JOBCODE           |        |       |        |       |
         |------------------|        |       |        |       |
         |ACT001            |       .|      .|    1.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |APP001            |       .|      .|    1.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |APP002            |    3.00|  50.00|    3.00|  50.00|
         |------------------+--------+-------+--------+-------|
         |APP003            |       .|      .|    1.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |CAR001            |    1.00| 100.00|       .|      .|
         |------------------+--------+-------+--------+-------|
         |CAR002            |       .|      .|    2.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |CCD001            |       .|      .|    1.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |CCD002            |       .|      .|    1.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |CCD003            |       .|      .|    1.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |CCD004            |    1.00| 100.00|       .|      .|
         |------------------+--------+-------+--------+-------|
         |CCD005            |       .|      .|    1.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |CCD006            |       .|      .|    3.00| 100.00|
         |------------------+--------+-------+--------+-------|
         |CCD007            |       .|      .|    1.00| 100.00|
         ------------------------------------------------------

         (Continued)
```

Here is another example of calculating percentages:

```
PROC TABULATE  DATA=tabwkshp.empldata;
     CLASS  jobcode gender;
     VAR     salary;
     TABLE  jobcode, gender*salary*(SUM PCTSUM<jobcode>) ;
```



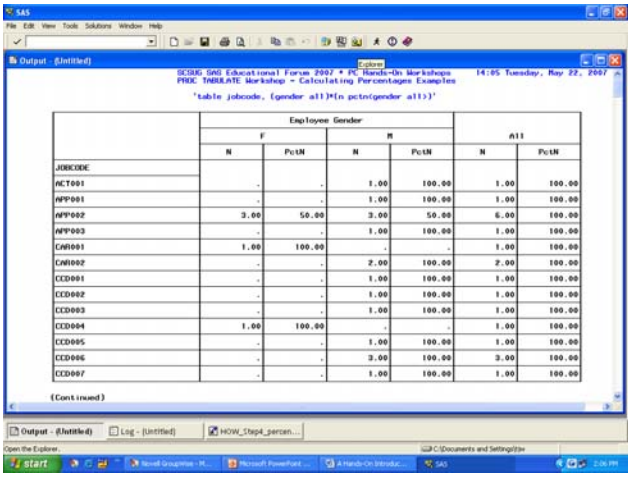
```
                 SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops     13:33 Tuesday, May 22, 2007
                 PROC TABULATE Workshop - Calculating Percentages Examples

                    'table jobcode, gender*salary*(sum pctsum<jobcode>)'

         ----------------------------------------------------------
         |                  |           Employee Gender           |
         |                  |-------------------------------------|
         |                  |        F         |        M         |
         |                  |------------------+------------------|
         |                  |      Salary      |      Salary      |
         |                  |------------------+------------------|
         |                  |    Sum   | PctSum|    Sum    |PctSum|
         |------------------+----------+-------+-----------+------|
         |JOBCODE           |          |       |           |      |
         |------------------|          |       |           |      |
         |ACT001            |         .|      .|   37000.00|  0.39|
         |------------------+----------+-------+-----------+------|
         |APP001            |         .|      .|   43500.00|  0.46|
         |------------------+----------+-------+-----------+------|
         |APP002            |  82000.00|   1.77|   95000.00|  1.00|
         |------------------+----------+-------+-----------+------|
         |APP003            |         .|      .|   60000.00|  0.63|
         |------------------+----------+-------+-----------+------|
         |CAR001            |  31000.00|   0.67|          .|     .|
         |------------------+----------+-------+-----------+------|
         |CAR002            |         .|      .|   65500.00|  0.69|
         |------------------+----------+-------+-----------+------|
         |CCD001            |         .|      .|  100000.00|  1.05|
         |------------------+----------+-------+-----------+------|
         |CCD002            |         .|      .|   38000.00|  0.40|
         |------------------+----------+-------+-----------+------|
         |CCD003            |         .|      .|   32000.00|  0.34|
         |------------------+----------+-------+-----------+------|
         |CCD004            |  70000.00|   1.51|          .|     .|
         |------------------+----------+-------+-----------+------|
         |CCD005            |         .|      .|   52000.00|  0.55|
         |------------------+----------+-------+-----------+------|
         |CCD006            |         .|      .|  128000.00|  1.35|
         ----------------------------------------------------------

         (Continued)
```

This *TABLE* statement specifies that the display should include breakdowns of the total *salary* amounts, and the associated percentages, for each combination of values of *jobcode* and *gender*, where the percentages are calculated in a column-wise manner.

Observe that, to obtain percentages by row, we use the column-expression in the "denominator definition"; and to obtain percentages by column, we use the row-expression in the "denominator definition."

And here are two more examples involving percentages, but these examples include the universal class variable, *ALL.*

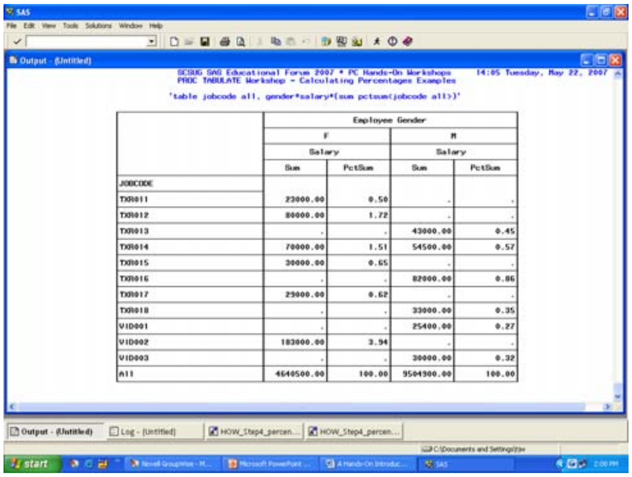
```
        PROC TABULATE  DATA=tabwkshp.empldata;
            CLASS  jobcode gender;
            VAR      salary;
            TABLE  jobcode, (gender  ALL)*(N PCTN<gender ALL>) ;
            TABLE  jobcode  ALL,
                        gender*salary*(SUM PCTSUM<jobcode  ALL>) ;
```


SCSUG SAS Educational Forum 2007 • PC Hands-On Workshops   14:05 Tuesday, May 22, 2007
PROC TABULATE Workshop - Calculating Percentages Examples

'table jobcode, (gender all)*(n pctn(gender all))'

	Employee Gender					
	F		M		All	
	N	PctN	N	PctN	N	PctN
JOBCODE						
ACT001	.	.	1.00	100.00	1.00	100.00
APP001	.	.	1.00	100.00	1.00	100.00
APP002	3.00	50.00	3.00	50.00	6.00	100.00
APP003	.	.	1.00	100.00	1.00	100.00
CAR001	1.00	100.00	.	.	1.00	100.00
CAR002	.	.	2.00	100.00	2.00	100.00
CCD001	.	.	1.00	100.00	1.00	100.00
CCD002	.	.	1.00	100.00	1.00	100.00
CCD003	.	.	1.00	100.00	1.00	100.00
CCD004	1.00	100.00	.	.	1.00	100.00
CCD005	.	.	1.00	100.00	1.00	100.00
CCD006	.	.	3.00	100.00	3.00	100.00
CCD007	.	.	1.00	100.00	1.00	100.00

(Continued)


SCSUG SAS Educational Forum 2007 • PC Hands-On Workshops   14:05 Tuesday, May 22, 2007
PROC TABULATE Workshop - Calculating Percentages Examples

'table jobcode all, gender*salary*(sum pctsum(jobcode all))'

	Employee Gender			
	F		M	
	Salary		Salary	
	Sum	PctSum	Sum	PctSum
JOBCODE				
TXR011	23000.00	0.50	.	.
TXR012	80000.00	1.72	.	.
TXR013	.	.	43000.00	0.45
TXR014	70000.00	1.51	54500.00	0.57
TXR015	30000.00	0.65	.	.
TXR016	.	.	82000.00	0.86
TXR017	29000.00	0.62	.	.
TXR018	.	.	33000.00	0.35
VID001	.	.	25400.00	0.27
VID002	183000.00	3.94	.	.
VID003	.	.	30000.00	0.32
All	4640500.00	100.00	9504900.00	100.00

Notice that whenever row- or column-percentages are to be produced for a column- or row-expression that includes the *ALL* universal class variable, then *ALL* also must be included in the "denominator definition."

Step 5 – Labeling and Formatting the Table

Now that we know how to define the basic structure of the tables we will generate, we would like to be able to make the tables more self-explanatory; that is, easier to read and interpret.

As in many other SAS procedures, you can use a *LABEL* statement to replace variable names with more descriptive headings for your class variables. There also is a way to specify temporary labels in a *TABLE* statement.
Similarly, *TITLE* and *FOOTNOTE* statements also can be used to enhance the tabular reports generated by PROC TABULATE.

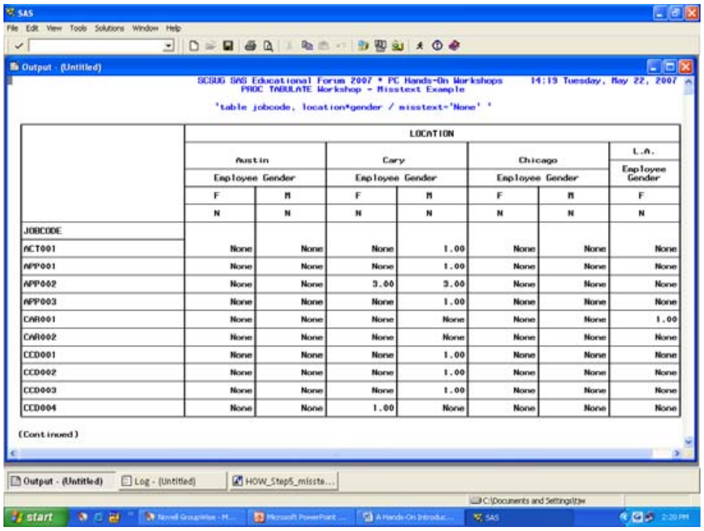
To assign labels to procedure-generated statistics and the universal class variable, we use the *KEYLABEL* statement.

```
KEYLABEL  N = ‘Count‘
          ALL = ‘Total’
          PCTN = ‘Percent’;
```

As in other SAS procedures, formats can be used to substitute labels for values of the classification variables.   Formats also can be used to combine many values of the classification variables into a much smaller number of values to be printed in the report.   We create custom formats by using PROC FORMAT, and we invoke those formats in PROC TABULATE either through a *FORMAT* statement, or by crossing *F=format-name.* in the *TABLE* statement with the particular variable.

The default for displaying cells with missing numeric values is a period. You can change the way missing values are displayed by using the *MISSTEXT=* option to define up to twenty characters of text that will print in the table cells whenever a particular combination of class variable values is not found in the input data set.

```
PROC TABULATE  DATA=tabwkshp.empldata;
      CLASS jobcode location gender;
      TABLE  jobcode, location*gender / MISSTEXT=‘None’;
```



```
                SCSUG SAS Educational Forum 2007 * PC Hands-On Workshops     14:19 Tuesday, May 22, 2007
                         PROC TABULATE Workshop - Misstext Example

                   'table jobcode, location*gender / misstext='None' '

--------------------------------------------------------------------------------------------
|                  |                                 LOCATION                              |
|                  |-----------------------------------------------------------------------|
|                  |     Austin      |      Cary       |     Chicago     |     L.A.        |
|                  |-----------------+-----------------+-----------------+-----------------|
|                  | Employee Gender | Employee Gender | Employee Gender |    Employee     |
|                  |                 |                 |                 |     Gender      |
|                  |-----------------+-----------------+-----------------+-----------------|
|                  |    F   |   M    |    F   |   M    |    F   |   M    |        F        |
|                  |--------+--------+--------+--------+--------+--------+-----------------|
|                  |    N   |   N    |    N   |   N    |    N   |   N    |        N        |
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|JOBCODE           |        |        |        |        |        |        |                 |
|------------------|        |        |        |        |        |        |                 |
|ACT001            |    None|    None|    None|    1.00|    None|    None|             None|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|APP001            |    None|    None|    None|    1.00|    None|    None|             None|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|APP002            |    None|    None|    3.00|    3.00|    None|    None|             None|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|APP003            |    None|    None|    None|    1.00|    None|    None|             None|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|CAR001            |    None|    None|    None|    None|    None|    None|             1.00|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|CAR002            |    None|    None|    None|    None|    None|    None|             None|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|CCD001            |    None|    None|    None|    1.00|    None|    None|             None|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|CCD002            |    None|    None|    None|    1.00|    None|    None|             None|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|CCD003            |    None|    None|    None|    1.00|    None|    None|             None|
|------------------+--------+--------+--------+--------+--------+--------+-----------------|
|CCD004            |    None|    None|    1.00|    None|    None|    None|             None|
--------------------------------------------------------------------------------------------
(Continued)
```

Here are a couple of useful *TABLE* statement options that can be used for customizing the appearance of tables:

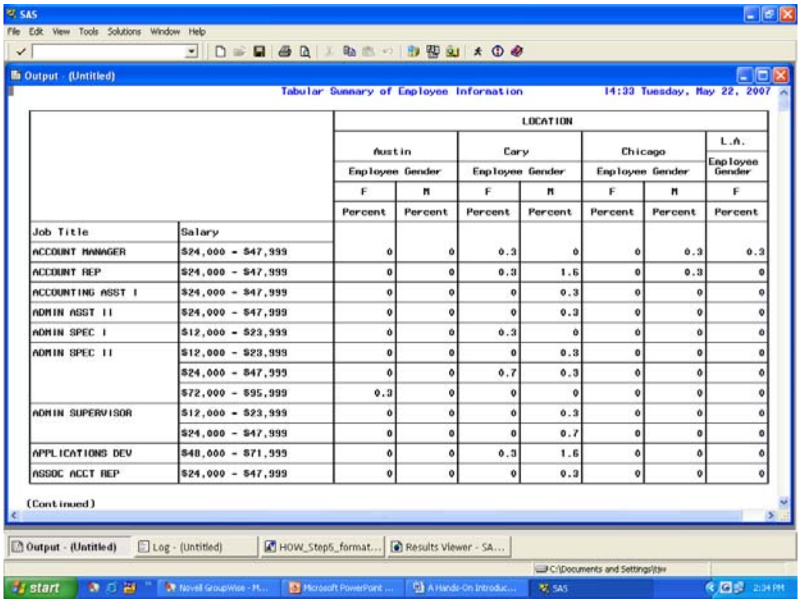
- The *RTSPACE=* (or *RTS=*) option defines the total amount of space for the row headings. If there are several levels of headings for rows, then the space is divided equally among the levels, after subtracting the spaces that are needed for the vertical lines.

- Whenever a table produced by PROC TABULATE is too wide to fit on a single page, the procedure automatically splits the table, to span as many separate pages as are necessary for printing. For short, wide tables, the *CONDENSE* option could be specified on the *TABLE* statement, in order to print as many logical pages as possible on a single page, one below the other.

Some people think that traditional SAS output is ugly. Beginning with Version 7, the SAS System provided an ability to deliver procedure output in a flexible variety of file types and formats, through the SAS <u>Output Delivery System (ODS)</u>. Under SAS 9.1.3, ODS can be used to generate results as SAS data sets, output listings, PostScript, HTML, RTF, PDF, PCL, XML, Excel, and other output file types. ODS can be used to enhance tabular reports, by wrapping the PROC TABULATE code in ODS destinations, by changing fonts, colors, and other style attributes, and by adding graphics. For further information about ODS and TABULATE, consult the software documentation provided by SAS Institute.

Here is a final example, which illustrates several of the “Labeling & Formatting” techniques that I have described:

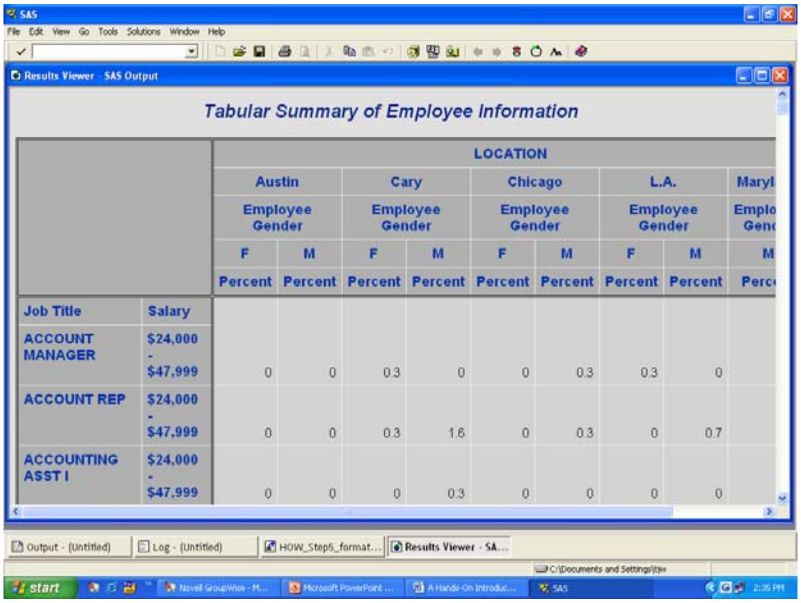
```
PROC FORMAT;
            VALUE    salfmt    low-<12000 = ‘Less than $12,000’
                               12000-<24000 = ‘$12,000 - $23,999’
                               24000-<48000 = ‘$24,000 - $47,999’
                               48000-<72000 = ‘$48,000 - $71,999’
                               72000-<96000 = ‘$72,000 - $95,999’
                              96000-<120000 = ‘$96,000 - $119,999’
                                120000-high = ‘$120,000 or more’;
 RUN;

 ODS HTML BODY=‘tables.htm’;

 PROC TABULATE  DATA=tabwkshp.empldata MISSING FORMAT=9.1;
    CLASS  title location gender salary;
    FORMAT salary salfmt.;
    LABEL title = ‘Job Title’;
    KEYLABEL  PCTN = ‘Percent’    ALL = ‘Total’;
    TABLE title*salary  ALL, (location*gender ALL)*PCTN
                / RTS=50  MISSTEXT=‘0’;
    TITLE ‘Tabular Summary of Employee Information’;
 RUN;
 ODS HTML CLOSE;
 RUN;
```



Tabular Summary of Employee Information 14:33 Tuesday, May 22, 2007

		LOCATION						
		Austin		Cary		Chicago		L.A.
		Employee Gender		Employee Gender		Employee Gender		Employee Gender
		F	M	F	M	F	M	F
		Percent	Percent	Percent	Percent	Percent	Percent	Percent
Job Title	Salary							
ACCOUNT MANAGER	$24,000 - $47,999	0	0	0.3	0	0	0.3	0.3
ACCOUNT REP	$24,000 - $47,999	0	0	0.3	1.6	0	0.3	0
ACCOUNTING ASST I	$24,000 - $47,999	0	0	0	0.3	0	0	0
ADMIN ASST II	$24,000 - $47,999	0	0	0	0.3	0	0	0
ADMIN SPEC I	$12,000 - $23,999	0	0	0.3	0	0	0	0
ADMIN SPEC II	$12,000 - $23,999	0	0	0	0.3	0	0	0
	$24,000 - $47,999	0	0	0.7	0.3	0	0	0
	$72,000 - $95,999	0.3	0	0	0	0	0	0
ADMIN SUPERVISOR	$12,000 - $23,999	0	0	0	0.3	0	0	0
	$24,000 - $47,999	0	0	0	0.7	0	0	0
APPLICATIONS DEV	$48,000 - $71,999	0	0	0.3	1.6	0	0	0
ASSOC ACCT REP	$24,000 - $47,999	0	0	0	0.3	0	0	0

(Continued)



Tabular Summary of Employee Information

		LOCATION								
		Austin		Cary		Chicago		L.A.		Maryl
		Employee Gender		Employee Gender		Employee Gender		Employee Gender		Emplo Genc
		F	M	F	M	F	M	F	M	M
		Percent	Percent	Percent	Percent	Percent	Percent	Percent	Percent	Perc
Job Title	Salary									
ACCOUNT MANAGER	$24,000 - $47,999	0	0	0.3	0	0	0.3	0.3	0	
ACCOUNT REP	$24,000 - $47,999	0	0	0.3	1.6	0	0.3	0	0.7	
ACCOUNTING ASST I	$24,000 - $47,999	0	0	0	0.3	0	0	0	0	

Conclusion

PROC TABULATE is a very useful and very powerful procedure for constructing tabular reports containing descriptive statistical information, including hierarchical relationships among variables. It is well worth the necessary investment of time and effort for learning the intricacies and subtleties of its syntax. Concentrating on the five steps makes it much easier to learn how to code PROC TABULATE. We have only just “scratched the surface” of this wonderful procedure! But now you know enough to continue learning about it on your own.

Happy tabulating!

Resources for Additional Reading

Jeffery M. Abolafia & Stephen M. Noga, “The TABULATE Procedure: One Step Beyond the Final Chapter,”

- SESUG ‘97 Proceedings (1997), pp. 293-300; and
- Proceedings of the Twenty-Third Annual SAS Users Group International Conference (1998), pp. 839-844.

Jonas V. Bilenas, “Making Sense of PROC TABULATE (Updated for SAS9®),” Proceedings of the SAS Global Forum 2007, Paper #230-2007.

Dan Bruns, “The Utter Simplicity of the TABULATE Procedure,”

- Proceedings of the Sixteenth Annual SAS Users Group International Conference (1991), pp. 365-371; and
- Proceedings of the Twentieth Annual SAS Users Group International Conference (1995), pp. 363-368; and
- Proceedings of the Seventh Annual South-Central SAS Users’ Conference (1997), pp. 54-60.

Dan Bruns, “The Utter ‘Simplicity?’ of the TABULATE Procedure,” Proceedings of the Seventeenth Annual SAS Users Group International Conference (1992), pp. 216-220.

Dan Bruns, “Advanced Features of PROC TABULATE -- or -- The Utter Simplicity of the TABULATE Procedure - The Sequel,” Proceedings of the Twenty-First Annual SAS Users Group International Conference (1996), pp. 242-247.

Dan Bruns, “The Utter ‘Simplicity?’ of the TABULATE Procedure -- The Final Chapter,”

- Proceedings of the Twenty-Second Annual SAS Users Group International Conference (1997), pp. 251-256 ; and
- Proceedings of the Seventh Annual South-Central SAS Users’ Conference (1997), pp. 61-66; and
- Proceedings of the Twenty-Ninth Annual SAS Users Group International Conference (2004), Paper #241-29.

Dan Bruns, “The Power and Simplicity of the TABULATE Procedure,”

- Proceedings of the Twenty-Fourth Annual SAS Users Group International Conference (1999), Paper #152; and
- Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference (2000), Paper #152-25; and
- Proceedings of the Twenty-Sixth Annual SAS Users Group International Conference (2001), Paper #148.

Dan Bruns, “The Simplicity and Power of the TABULATE Procedure,” Proceedings of the Twenty-Eighth Annual SAS Users Group International Conference (2003), Paper #197.

Dan Bruns & Ray Pass, “Battle of the Titans: REPORT vs. TABULATE,” Proceedings of the Twenty-Seventh Annual SAS Users Group International Conference (2002), Paper #133-27.

Dan Bruns & Ray Pass, “To REPORT or to TABULATE?: That is the Question!,” Proceedings of the Twenty-Ninth Annual SAS Users Group International Conference (2004), Paper #122-29.

Diane Louise Rhodes, “Speaking Klingon: A Translator’s Guide to PROC TABULATE,” Proceedings of the Thirtieth Annual SAS Users Group International Conference (2005), Paper #258-30.

SAS Institute Inc., “The TABULATE Procedure,” in SAS® 9.1.3 Help and Documentation, an HTML application that is installed as a component of the SAS System and is accessible from the SAS main menu by clicking “Help.”

SAS Institute Inc., “The TABULATE Procedure,” Chapter 52 of Base SAS® 9.1.3 Procedures Guide (2006), available online at *http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_913/ base_proc_8977_new.pdf* .

SAS Institute Inc., SAS Guide to TABULATE Processing, Second Edition (Cary, NC: SAS Institute Inc., 1990).

Bob Virgile, “The Right Approach to Learning PROC TABULATE,” SESUG ‘97 Proceedings (1997), pp. 189-195.

Tom Winn, “Introduction to Using PROC TABULATE,”

- Proceedings of the Eighth Annual South-Central SAS Users’ Conference (1998), pp. 316-326,
- Proceedings of the New Mexico SAS Users Conference (2002),
- Proceedings of the Arkansas SAS Day (2005), pp. 14-33,
- Proceedings of the 2006 Wisconsin-Illinois SAS Users Conference .

Tom Winn, “Advanced Features of PROC TABULATE,”

- Proceedings of the Twenty-Fourth Annual SAS Users Group International Conference (1999), Paper #153,
- Proceedings of the Ninth Annual South-Central SAS Users’ Conference (1999), pp. 243-248.

Author Information

Tom Winn, Ph.D.
10906 Rustic Manor Lane
Austin, TX 78750

Telephone: 512 / 250-8610
E-Mail: TWIN504@aol.com