Paper B2B11

Going from Zero to Report Ready with PROC TABULATE

Rachel Straney, University of Central Florida

ABSTRACT

The TABULATE procedure in SAS® can be used to summarize your data into clean and organized tables. This procedure can calculate many of the descriptive statistics that the MEANS, FREQ, and REPORT procedures do, but with the flexibility to display them in a customized tabulated format. At first, the syntax may seem difficult and overwhelming but with practice and some basic examples you can go from zero to report ready in no time. This paper will discuss the benefits of using PROC TABULATE and identify the kinds of reports that this procedure is best suited. An example will be used to illustrate the syntax and statements needed to generate a complex table. The table will include multiple classification variables as well as more than one numeric variable for various computed statistics. Readers will learn the functions of the CLASS, VAR and TABLE statements and how to include subtotals and totals with the keyword ALL. To make the finished table 'report ready,' examples of how to apply formats, labels and styles will also be shared.

INTRODUCTION

Although there are alternative ways of obtaining the same information using PROC MEANS, FREQ or REPORT, the benefit of using TABULATE is that this procedure is ideal for nesting multiple variables and creating output with hierarchical levels. As an analyst, it is quite common to get a request from a supervisor, colleague or client that begins small and eventually takes on a life of its own. What initiated as a simple question then becomes an exercise to densely pack as much information as possible into a compact and easily readable table. This paper will use a scenario such as this to illustrate how to build a complex table using PROC TABULATE. The table will then be built step by step to show readers how to generate the table and apply formats, labels and styles so that it is aesthetically pleasing and ready for dissemination.

Our story begins at an institution of higher education. Although the setting takes place in the education field, this example is general enough to be applied across many disciplines. The analyst is a SAS programmer and is maintaining a data set which contains various pieces of information on students enrolled at the institution. The data set is named STUDENTS. Table 1 shows some of the output from the CONTENTS procedure:

Alphabetic List of Variables and Attributes			
#	**Variable**	**Type**	**Len**
3	ETHNICITY	Char	1
4	FIN_AID	Num	8
2	GENDER	Char	1
6	GPA	Num	8
5	SAT	Num	8
1	STUDENTID	Num	8

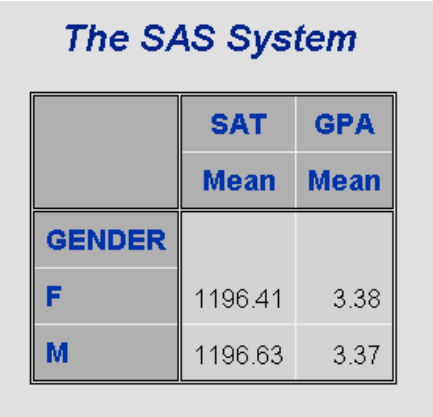
Output 1. Output from CONTENTS Procedure: Variables and Attributes from the STUDENTS data set

THE FIRST REQUEST

The analyst receives an initial request regarding enrolled students at the institution and their last semester GPA.

Initially the associate was interested student academic performance compared across gender. Academic performance was defined by SAT scores and student GPA in the last semester. Because this was an internal request, the analyst decided to run a quick report in SAS using PROC TABULATE and provide it to the client.

Shown below is the output that the analyst provided and the SAS syntax used to fulfill this request.



The SAS System

	SAT	GPA
	Mean	Mean
GENDER		
F	1196.41	3.38
M	1196.63	3.37

```
PROC TABULATE data=students;
CLASS gender ;
VAR sat gpa ;
TABLE gender,(sat gpa)*mean;
RUN;
```

Output 2. PROC TABULATE output for the first request

CLASS STATEMENT

The CLASS statement is reserved for one or more categorical variables that are used in the table. Note that since gender is a categorical variable it is listed in the CLASS statement. If the analyst was not using a categorical variable it would not be needed in this program. If there were more than one categorical variable used in the table then these would have been listed in the CLASS statement as well.

VAR STATEMENT

The VAR statement is reserved for one or more analysis or numeric variables that are used in the table to display descriptive statistics. Similar to the CLASS statement, all numeric variables need to be listed in the statement and if there were no descriptive statistics to compute and display then the VAR statement would not be needed.

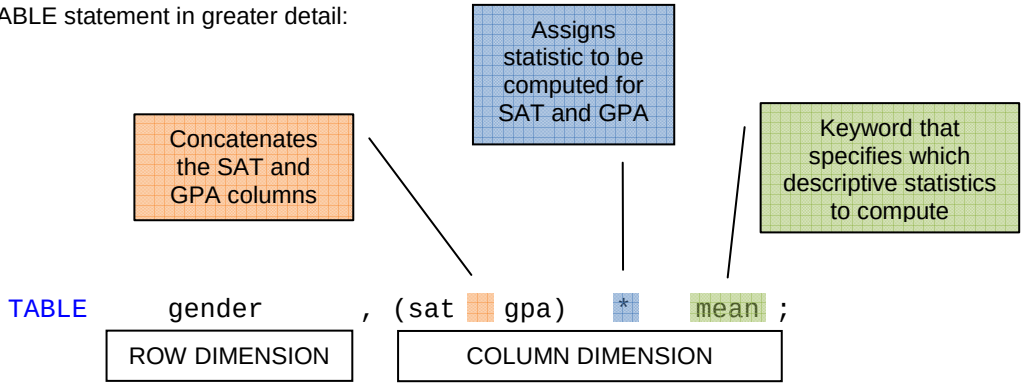
TABLE STATEMENT

The TABLE statement is used to define the structure of the table. There are a few special operators that the TABLE statement uses to structure the table. These are described in the table below.

Operators Available in the TABULATE Procedure	
Operator	**Function**
Comma - “,”	Separates the variables in the row dimension from the variables in the column dimension
Asterisk – “*”	Nests or cross tabulates categorical variables with one another or assigns a statistic to an analysis variable
Blank space – “ “	Concatenates variables one after the other
Parentheses – “( )”	Groups variables and other components together to apply various operations
Angle brackets - ”<>”	Specifies the denominator in calculations for percentages

Table 1. Operators used in the TABLE statement of the TABULATE procedure

Let us look at the TABLE statement in greater detail:



Here, the mean is displayed for both SAT and GPA since these variables are grouped by parentheses. By removing the parentheses the mean would only be displayed for GPA and by default the sum would be displayed for SAT. There are many descriptive statistics that can be summarized in the TABULATE procedure including sum, mode, standard deviation, min, max and quantile statistics. An exhaustive list of statistics available in PROC TABULATE is listed in Table 2 and was obtained from the Base SAS 9.2 Procedures Guide.

Statistics Available in PROC TABULATE	
Descriptive statistic keywords	
COLPCTN	PCTSUM
COLPCTSUM	RANGE
CSS	REPPCTN
CV	REPPCTSUM
KURTOSIS | KURT	ROWPCTN
LCLM	ROWPCTSUM
MAX	SKEWNESS | SKEW
MEAN	STDDEV | STD
MIN	STDERR
MODE	SUM
N	SUMWGT
NMISS	UCLM
PAGEPCTN	USS
PAGEPCTSUM	VAR
PCTN	
Quantile statistic keywords	
MEDIAN | P50	Q3 | P75
P1	P90
P5	P95
P10	P99
Q1|P25	QRANGE
Hypothesis testing keywords	
PROBT | PRT	T

Table 2. Statistics available in the TABULATE procedure

THE SECOND REQUEST

The following day the associate comes into the analyst's office and asks, "Is there a way we can add ethnicity to this table? And as a side note, how many students are we talking about here?" The analyst adapts the original SAS program to include this additional information in the report.

Shown below is the output that the analyst provided and the SAS syntax used to fulfill the second request.

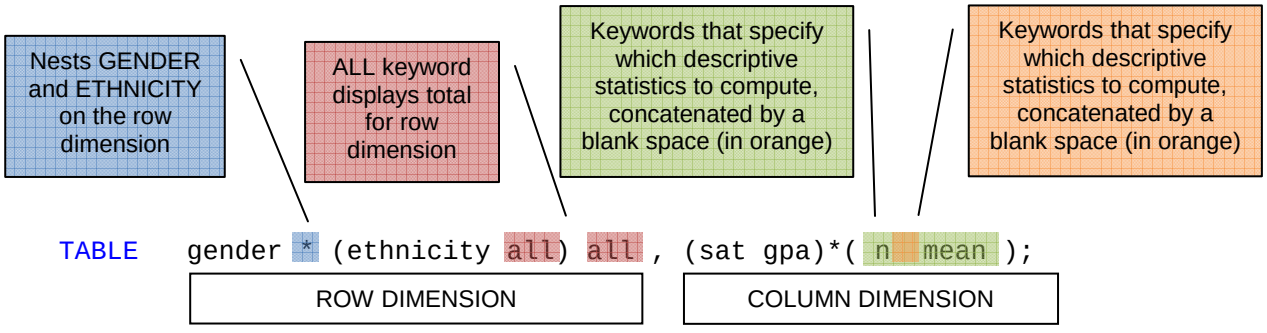
The SAS System

		SAT		GPA	
		N	Mean	N	Mean
GENDER	ETHNICITY				
F	B	82	1201.59	99	3.37
	H	69	1185.94	89	3.39
	O	10	1228.00	17	3.40
	W	299	1196.35	355	3.38
	All	460	1196.41	560	3.38
M	ETHNICITY				
	B	64	1214.53	73	3.41
	H	67	1180.45	81	3.36
	O	22	1219.55	23	3.23
	W	224	1194.11	263	3.37
	All	377	1196.63	440	3.37
All		837	1196.51	1000	3.37

```
PROC TABULATE data=students;
CLASS gender ethnicity;
VAR sat gpa ;
TABLE gender*(ethnicity all) all,
      (sat gpa)*(n mean);
RUN;
```

Output 3. PROC TABULATE output for the second request

Let us look at the TABLE statement in greater detail:



The variable ETHNICITY has been added to the row dimension (before the comma) and has been nested with the other categorical variable GENDER (using the asterisk). Note that ETHNICITY has been listed in the CLASS statement. Because the client was interested in how many students the table was summarizing, the analyst added the keyword N which computes the descriptive statistic count for SAT and GPA. Changing the order of N and MEAN in the syntax will change the order in which they are displayed in the table. Additionally, the keyword ALL was added in two places, the first is grouped with the categorical variable ETHNICITY to generate a subtotal row and the second is listed at the end of the row dimension to compute a grand total.

THE THIRD REQUEST

There is a third follow up between the analyst and the associate. The associate would like information on student financial aid, specifically how much aid is awarded and how many students are receiving these funds. Again, the analyst adapts the SAS program.

Shown below is the output that the analyst provided and the SAS syntax used to fulfill the third request.

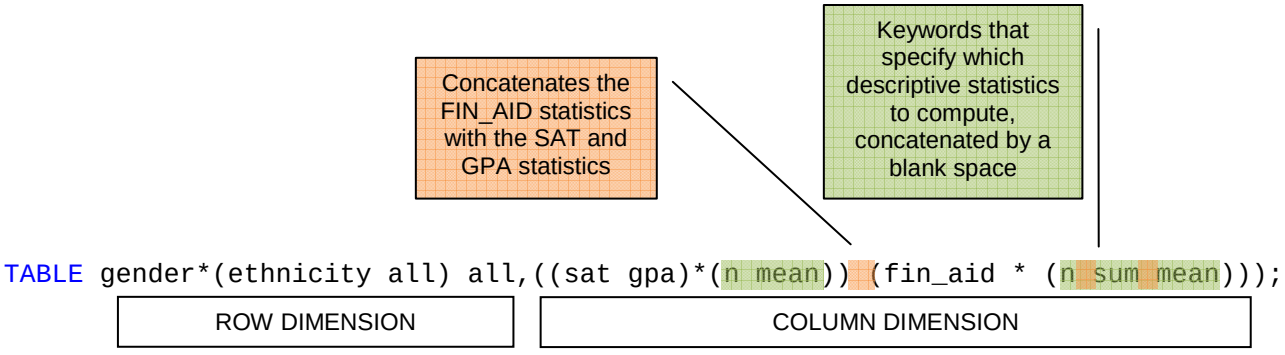
The SAS System

		SAT		GPA		FIN_AID		
		N	Mean	N	Mean	N	Sum	Mean
GENDER	ETHNICITY							
F	B	82	1201.59	99	3.37	81	119503.08	1475.35
	H	69	1185.94	89	3.39	73	111901.44	1532.90
	O	10	1228.00	17	3.40	15	20478.73	1365.25
	W	299	1196.35	355	3.38	282	434545.06	1540.94
	All	460	1196.41	560	3.38	451	686428.32	1522.01
M	ETHNICITY							
	B	64	1214.53	73	3.41	63	95300.11	1512.70
	H	67	1180.45	81	3.36	64	95953.90	1499.28
	O	22	1219.55	23	3.23	17	23010.95	1353.59
	W	224	1194.11	263	3.37	211	314167.94	1488.95
	All	377	1196.63	440	3.37	355	528432.89	1488.54
All		837	1196.51	1000	3.37	806	1214861.21	1507.27

```
PROC TABULATE data=students;
CLASS gender ethnicity;
VAR sat gpa fin_aid;
TABLE gender*(ethnicity all) all,
      ((sat gpa)*(n mean))
      (fin_aid*(n sum mean));
RUN;
```

Output 4. PROC TABULATE output for the third request

Let us look at the TABLE statement in greater detail:



Parentheses are used to group the SAT and GPA variables so that the N and MEAN keywords are applied to both. The financial aid variable is also grouped such that the three descriptive statistics (N, SUM and MEAN) are computed and displayed. Financial aid information is appended to the end of the table by the blank space character.

THE FOURTH REQUEST

A week passes and there is one more final request. The table structure was exactly what was needed to understand some key characteristics about the current student population. It was so informative that the associate asked to have the table "spruced up a bit" so it could be disseminated to other constituents.

Thus far we have walked through examples of building the table structure. The fourth request entails formatting, labeling and stylizing the report to make a final product. We will walk through each of these tasks step by step using the table from the third request.

LABELS

1. Labeling variables within the TABLE statement

The TABLE statement in the SAS code below has each of the variables labeled by using an '=' and then defining the label in single quotes. The benefits of this method are the ability to control each header individually and the capability to hide variable labels altogether. GENDER and ETHNICITY have their labels hidden in the example below.

The SAS System

		Incoming SAT		First Term GPA		Financial Aid		
		Number of Students	Average Score	Number of Students	Average Score	Number Receiving Aid	Aid Awarded (Sum)	Aid Awarded (Average)
F	B	82	1201.59	99	3.37	81	119503.08	1475.35
	H	69	1185.94	89	3.39	73	111901.44	1532.90
	O	10	1228.00	17	3.40	15	20478.73	1365.25
	W	299	1196.35	355	3.38	282	434545.06	1540.94
	Subtotal	460	1196.41	560	3.38	451	686428.32	1522.01
M	B	64	1214.53	73	3.41	63	95300.11	1512.70
	H	67	1180.45	81	3.36	64	95953.90	1499.28
	O	22	1219.55	23	3.23	17	23010.95	1353.59
	W	224	1194.11	263	3.37	211	314167.94	1488.95
	Subtotal	377	1196.63	440	3.37	355	528432.89	1488.54
Total		837	1196.51	1000	3.37	806	1214861.21	1507.27

Output 5. PROC TABULATE output for the third request with labels inside the TABLE statement

```
PROC TABULATE data=students;
CLASS gender ethnicity;
VAR sat gpa fin_aid;
TABLE gender=''*(ethnicity='' all='Subtotal') all='Total',
      ((sat='Incoming SAT' gpa='First Term GPA')*
      (n='Number of Students' mean='Average Score'))
      (fin_aid='Financial Aid'*
      (n='Number Receiving Aid' sum='Aid Awarded (Sum)' mean='Aid Awarded
      (Average)'));
RUN;
```

2. Labeling variables using LABEL and KEYLABEL statements

The labels for the variables in the next example are defined using the LABEL statement. When this strategy is used the variable labels cannot be hidden as was done in the previous example with GENDER and ETHNICITY. The KEYLABEL statement can be used to label the descriptive statistics in the table. The KEYLABEL statement will control the label for all statistics of the same type, so all counts using the keyword N will be labeled the same. Although there are a few limitations using these methods, one of the benefits is that the SAS code is less cluttered and it is easier read.

The SAS System

		Incoming SAT		First Term GPA		Financial Aid		
		Count	Average	Count	Average	Count	Sum	Average
Gender	Ethnicity							
F	B	82	1201.59	99	3.37	81	119503.08	1475.35
	H	69	1185.94	89	3.39	73	111901.44	1532.90
	O	10	1228.00	17	3.40	15	20478.73	1365.25
	W	299	1196.35	355	3.38	282	434545.06	1540.94
	Total	460	1196.41	560	3.38	451	686428.32	1522.01
M	Ethnicity							
	B	64	1214.53	73	3.41	63	95300.11	1512.70
	H	67	1180.45	81	3.36	64	95953.90	1499.28
	O	22	1219.55	23	3.23	17	23010.95	1353.59
	W	224	1194.11	263	3.37	211	314167.94	1488.95
	Total	377	1196.63	440	3.37	355	528432.89	1488.54
Total		837	1196.51	1000	3.37	806	1214861.21	1507.27

Output 6. PROC TABULATE output for the third request with labels using the LABEL statement

```
PROC TABULATE data=students;
CLASS gender ethnicity;
VAR sat gpa fin_aid;
LABEL gender = 'Gender'
      ethnicity='Ethnicity'
      sat='Incoming SAT'
      gpa='First Term GPA'
      fin_aid='Financial Aid'
      ;
KEYLABEL all = 'Total'
         mean = 'Average'
         n = 'Count'
         sum = 'Sum';
TABLE gender*(ethnicity all) all,((sat gpa)*(n mean))(fin_aid*(n sum mean));
RUN;
```

FORMATS AND ORDER

Formats can be applied to the categorical variable levels by using the FORMAT procedure. To alter the order in which these levels are displayed, the ORDER option is available in the CLASS statement. As for formatting the values within the table, the TABLE statement allows the syntax '*f=' followed by the desired SAS format to be attached to any of the descriptive statistics. Examples of all these techniques are shown below.

The SAS System

		SAT		GPA		FIN_AID		
		N	Mean	N	Mean	N	Sum	Mean
GENDER	ETHNICITY							
Female	White	299	1196.4	355	3.4	282	$434,545.06	$1,540.94
	Black	82	1201.6	99	3.4	81	$119,503.08	$1,475.35
	Hispanic	69	1185.9	89	3.4	73	$111,901.44	$1,532.90
	Other	10	1228.0	17	3.4	15	$20,478.73	$1,365.25
	All	460	1196.4	560	3.4	451	$686,428.32	$1,522.01
Male	ETHNICITY							
	White	224	1194.1	263	3.4	211	$314,167.94	$1,488.95
	Black	64	1214.5	73	3.4	63	$95,300.11	$1,512.70
	Hispanic	67	1180.4	81	3.4	64	$95,953.90	$1,499.28
	Other	22	1219.5	23	3.2	17	$23,010.95	$1,353.59
	All	377	1196.6	440	3.4	355	$528,432.89	$1,488.54
All		837	1196.5	1,000	3.4	806	$1,214,861.21	$1,507.27

Output 7. PROC TABULATE output for the third request with formats and order

```
PROC FORMAT;
VALUE $f_gender
        'F' = 'Female'
        'M' = 'Male'
        ;
VALUE $f_ethnicity
        'W' = 'White'
        'B' = 'Black'
        'H' = 'Hispanic'
        'O' = 'Other'
        ;

PROC TABULATE data=students;
CLASS gender ethnicity / order=data;
VAR sat gpa fin_aid;
TABLE gender*(ethnicity all) all,
      ((sat gpa)*(n*f=comma6. mean*f=6.1))
      (fin_aid*(n*f=comma6. sum*f=dollar13.2 mean*f=dollar13.2));
format gender $f_gender. ethnicity $f_ethnicity.;
RUN;
```

STYLIZING THE TABLE

1. Defining style elements within PROC TABULATE

The first option to stylize the table requires that the style elements are defined within multiple statements of the TABULATE procedure. There are some new statements in the SAS syntax for this example, namely CLASSLEV and KEYWORD. These statements control the style elements for categorical levels and keywords like descriptive statistic respectively. The BOX option in the TABLE statement controls the top left cell of the table.

The SAS System

		Incoming SAT		First Term GPA		Financial Aid		
		Number of Students	Average Score	Number of Students	Average Score	Number Receiving Aid	Aid Awarded (Sum)	Aid Awarded (Average)
Female	White	299	1196.4	355	3.4	282	$434,545.06	$1,540.94
	Black	82	1201.6	99	3.4	81	$119,503.08	$1,475.35
	Hispanic	69	1185.9	89	3.4	73	$111,901.44	$1,532.90
	Other	10	1228.0	17	3.4	15	$20,478.73	$1,365.25
	Subtotal	460	1196.4	560	3.4	451	$686,428.32	$1,522.01
Male	White	224	1194.1	263	3.4	211	$314,167.94	$1,488.95
	Black	64	1214.5	73	3.4	63	$95,300.11	$1,512.70
	Hispanic	67	1180.4	81	3.4	64	$95,953.90	$1,499.28
	Other	22	1219.5	23	3.2	17	$23,010.95	$1,353.59
	Subtotal	377	1196.6	440	3.4	355	$528,432.89	$1,488.54
Total		837	1196.5	1,000	3.4	806	$1,214,861.21	$1,507.27

Output 8. PROC TABULATE output for the third request controlling style elements within the procedure

```
PROC TABULATE data=students style=[background=white font_face=calibri
                                  foreground=black font_size=3];
CLASS gender ethnicity / order=freq;
CLASSLEV gender / style=[background=#E3DFCA font_face=cambria
                         foreground=black font_size=4 outputwidth=1in];
CLASSLEV ethnicity / style=[background=#E3DFCA font_face=cambria
                            foreground=black font_size=4 outputwidth=1.2in];
VAR sat gpa fin_aid / style=[background=#E3DFCA font_face=cambria
                             foreground=black font_size=4 ];
KEYWORD n sum mean all / style=[background=#E3DFCA font_face=cambria
                                foreground=black font_size=4 outputwidth=1in];
TABLE gender=''*(ethnicity='' all='Subtotal') all='Total',
      ((sat='Incoming SAT' gpa='First Term GPA')*
      (n='Number of Students'*f=comma6. mean='Average Score'*f=6.1))
      (fin_aid='Financial Aid'*(n='Number Receiving Aid'*f=comma6.
      sum='Aid Awarded (Sum)'*f=dollar13.2
      mean='Aid Awarded (Average)'*f=dollar13.2)) /
      box=[style=[background=#E3DFCA]] ;
FORMAT gender $f_gender. ethnicity $f_ethnicity.;
RUN;
```

The style is controlled in multiple places in the procedure using the keyword STYLE.

Table 3 lists which statements in the procedure the STYLE keyword can be used and where in the table output these options are applied.

Style Attributes in the TABULATE Procedure	
Statement	**Where Style Attributes are Applied**
PROC TABULATE	Inside the table for the data cells
CLASS	Categorical variable header cells
CLASSLEV	Categorical level headings
VAR	Analysis variable header cells
KEYWORD	Headings for keywords such as ALL, N, SUM, etc.
TABLE	Typically reserved for table borders, cell spacing and other rules that are not defined elsewhere

Table 3. Style Attributes and their functions when used in the PROC TABULATE statements

2. Defining style elements using the Output Delivery System (ODS)

In the previous example, the syntax was quite long and for complex tables such as these defining style elements within the procedure itself can be time consuming and cumbersome. The second way to control the style of the table is by utilizing ODS and SAS predefined styles. This is the strategy that the analyst used to fulfill the fourth and final request. ODS can be used to output SAS results in the following file formats: HTML, PDF, RTF, and PS (PostScript). In the example below, the table is generated in an HTML format. Information on ODS and the available destination options can be found at:
http://support.sas.com/documentation/cdl/en/odsug/61723/PDF/default/odsug.pdf

		Incoming SAT		First Term GPA		Financial Aid		
		Number of Students	Average Score	Number of Students	Average Score	Number Receiving Aid	Aid Awarded (Sum)	Aid Awarded (Average)
Female	White	299	1196.4	355	3.4	282	$434,545.06	$1,540.94
	Black	82	1201.6	99	3.4	81	$119,503.08	$1,475.35
	Hispanic	69	1185.9	89	3.4	73	$111,901.44	$1,532.90
	Other	10	1228.0	17	3.4	15	$20,478.73	$1,365.25
	Subtotal	460	1196.4	560	3.4	451	$686,428.32	$1,522.01
Male	White	224	1194.1	263	3.4	211	$314,167.94	$1,488.95
	Black	64	1214.5	73	3.4	63	$95,300.11	$1,512.70
	Hispanic	67	1180.4	81	3.4	64	$95,953.90	$1,499.28
	Other	22	1219.5	23	3.2	17	$23,010.95	$1,353.59
	Subtotal	377	1196.6	440	3.4	355	$528,432.89	$1,488.54
Total		837	1196.5	1,000	3.4	806	$1,214,861.21	$1,507.27

Output 9. PROC TABULATE output for the third request controlling style elements using ODS

The OUTPUTWIDTH style attribute was defined in this example to exhibit how ODS and defining style within the procedure can work together. The width of the columns can also be controlled using ODS by altering or customizing the SAS parent style through PROC TEMPLATE.

```
ods listing close;
ods html style=styles.analysis;

PROC TABULATE data=students;
CLASS gender ethnicity / order=freq;
CLASSLEV gender / style=[outputwidth=1in];
CLASSLEV ethnicity / style=[outputwidth=1.2in];
VAR sat gpa fin_aid;
TABLE gender=''*(ethnicity='' all='Subtotal') all='Total',
      ((sat='Incoming SAT' gpa='First Term GPA')*
      (n='Number of Students'*f=comma6.*[style=[outputwidth=1in]]
      mean='Average Score'*f=6.1*[style=[outputwidth=1in]]))
      (fin_aid='Financial Aid'*
      (n='Number Receiving Aid'*f=comma6.*[style=[outputwidth=1in]]
      sum='Aid Awarded (Sum)'*f=dollar13.2*[style=[outputwidth=1in]]
      mean='Aid Awarded (Average)'*f=dollar13.2*[style=[outputwidth=1in]]));
FORMAT gender $f_gender. ethnicity $f_ethnicity.;
RUN;

ods html close;
ods listing;
```

When using ODS to apply styles to the TABULATE output, the statements highlighted in yellow must be included before and after the procedure. This not only applies to PROC TABUALTE, but any SAS procedure or data step. Note that the style applied is called ‘styles.analysis’ which is a predefined style in the SAS style library that is included in the Base SAS package. There are a number of predefined styles available which can be viewed by running the following SAS code:

```
proc template;
     list styles;
run;
```

As was mentioned above, various styles can be customized using PROC TEMPLATE. There are many resources available on the TEMPLATE procedure that can assist in customizing a style for SAS output.

CONCLUSION

Our analyst was successful in producing a report quickly that answered the needs of his associate by relying on the TABULATE procedure in SAS. This procedure is well suited for complex tables, especially in situations where multiple variables must be cross tabulated to display information. It is important to remember there is more than one way to show information in a data set. Similarly, there is more than one way to get a TABULATE procedure to build and structure a table. These examples were shared to introduce some of the fundamental strategies and syntax one needs to run the procedure properly for a professional report. PROC TABULATE is a powerful and flexible tool and there is much more it can do to assist with reporting of data. SAS has many resources such technical papers and documentation on this procedure and it is highly recommended to use them.

REFERENCES

Bobiak, Sarah (2011), “SAS does the dirty work, how do I make it look pretty?,” Proceedings of the 2011 North East SAS Users Group.

Bruns, Dan (2004), “To REPORT or to TABULATE? – That is the Question!,” Proceedings of the 29th SAS User’s Group International.

Gupta, Sunil K. (2004), “Using Styles and Templates to Customize SAS® ODS Output,” Proceedings of the 29th SAS User’s Group International.

SAS Institute Inc. (2008), Using Style Elements in the REPORT and TABULATE Procedures. Cary, NC. Available at support.sas.com/resources/papers/stylesinprocs.pdf.

SAS Institute Inc. 2009. Base SAS® 9.2 Procedures Guide. Cary, NC. Available at support.sas.com/documentation/cdl/en/proc/61895/PDF/default/proc.pdf.

SAS Institute Inc. 2009. SAS ® 9.2 Output Delivery System: User’s Guide. Cary, NC. Available at support.sas.com/documentation/cdl/en/odsug/61723/PDF/default/odsug.pdf.

Wright, Wendi L. (2008), “PROC TABULATE® and the Neat Things You Can Do With It,” Proceedings of the 2008 SAS Global Forum.

AUTHOR CONTACT

Your comments and questions are valued and encouraged. Contact the author at:

Rachel Straney
University of Central Florida
12424 Research Parkway, Suite 225
Orlando, FL 32826
407-882-0280
rstraney@ucf.edu