

## SAS® Reporting 101: REPORT, TABULATE, ODS, and Microsoft Office

Lauren Haworth, Genentech, Inc., South San Francisco, CA

### ➤ ABSTRACT

So you've got some SAS data and your boss wants a fancy table to present to senior management. Don't panic, SAS makes it easy to quickly generate high quality reports. This paper is a whirlwind tour of your SAS reporting options. It covers the REPORT procedure, the TABULATE procedure, and which is best to use for what types of reports. After picking a procedure, the next step is to use the Output Delivery System to turn it into a web page, PDF file, Word document, PowerPoint slide, or Excel spreadsheet. This paper will show how to leverage ODS and Microsoft Office to deliver your results in the format most convenient for your customers.

This paper is designed for the beginning SAS programmer, and is based on SAS version 8.2.

### ➤ THE BASIC PROCEDURES: PRINT, MEANS, AND FREQ

When first learning SAS, the new user can easily learn how to use these three basic reporting procedures. The PRINT procedure lists out rows of data. For example, the following code will generate the output shown below.

```
proc print data=census;
  var sex educ income;
run;
```

Obs	sex	educ	income
1	Male	Not HS graduate	62528
2	Male	Not HS graduate	3138
3	Male	Not HS graduate	2500
4	Male	Not HS graduate	2671
5	Male	Not HS graduate	25700
6	Male	Not HS graduate	14560
7	Male	Not HS graduate	17842
8	Male	Not HS graduate	13730
9	Male	Not HS graduate	56392
10	Male	Not HS graduate	20160
11	Male	Not HS graduate	12185
12	Male	Not HS graduate	1500
13	Male	Not HS graduate	4164
14	Male	Not HS graduate	1000
15	Male	Not HS graduate	16300
16	Male	Not HS graduate	6400
17	Male	Not HS graduate	5000
18	Male	Not HS graduate	19000
19	Male	Not HS graduate	15000
20	Male	Not HS graduate	11372

This output isn't very exciting, but it does show the data. Another procedure most beginners quickly learn is the

MEANS procedure. For example, with the same data, the following code will generate the output shown below.

```
proc means data=census;
  var income;
run;
```

The MEANS Procedure				
Analysis Variable : income Adjusted gross income				
N	Mean	Std Dev	Minimum	Maximum
363	32487.93	26143.27	1000.00	99999.00

This procedure allows you to select various statistics to display for numeric variables. Again, the output isn't very exciting, but it gets the job done.

Rounding out this group of basic procedures is the FREQ procedure. The following example code and output illustrate the FREQ procedure in action.

```
proc freq data=census;
  tables sex educ;
run;
```

sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Male	242	66.67	242	66.67
Female	121	33.33	363	100.00

educ	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Not HS graduate	36	9.92	36	9.92
HS graduate	141	38.84	177	48.76
Some college	76	20.94	253	69.70
Associates degree	28	7.71	281	77.41
Bachelors degree	55	15.15	336	92.56
Post-graduate degree	27	7.44	363	100.00

This procedure provides frequency distributions on categorical character or numeric variables.

Using PRINT, MEANS, and FREQ, you can produce all sorts of statistics and displays of your data. However, you can't do much in the way of organizing that data into a meaningful summary report.

To move beyond basic data displays, you need to learn the procedures designed for producing reports. These procedures provide many of the same results as PRINT, MEANS, and FREQ, but they also allow you to combine different types of analyses, and to have control over how the results are displayed.

➤ **PROC REPORT**

To start off, we'll look at the REPORT procedure. This procedure has basic syntax that is very similar to the PRINT procedure. Instead of a VAR statement followed by a list of variables, it has a COLUMNS statement followed by a list of variables. The other item to note in the code below is the option NOWD. This causes REPORT to run in batch mode. The report procedure also has an interactive capability, where you can design your report on-screen. However, you will probably find that you get more control by learning the syntax and writing the code yourself.

If you look at the sample output below, you'll see that this basic PROC REPORT generates output that looks exactly like PROC PRINT.

```
proc report data=census nowd;
  columns educ sex income;
run;
```

Obs	sex	educ	income
1	Male	Not HS graduate	62528
2	Male	Not HS graduate	3138
3	Male	Not HS graduate	2500
4	Male	Not HS graduate	2671
5	Male	Not HS graduate	25700
6	Male	Not HS graduate	14560
7	Male	Not HS graduate	17842
8	Male	Not HS graduate	13730
9	Male	Not HS graduate	56392
10	Male	Not HS graduate	20160
11	Male	Not HS graduate	12185
12	Male	Not HS graduate	1500
13	Male	Not HS graduate	4164
14	Male	Not HS graduate	1000
15	Male	Not HS graduate	16300
16	Male	Not HS graduate	6400
17	Male	Not HS graduate	5000
18	Male	Not HS graduate	19000
19	Male	Not HS graduate	15000
20	Male	Not HS graduate	11372

To truly take advantage of the REPORT procedure, you need to use DEFINE statements to provide further instructions on how to display each variable. In this first example, a DEFINE statement is used to add a label for the column, and to define the variable educ as a grouping variable. This means that the results will be displayed in groups based on each category of education. The order=internal option is used to get the education categories to sort into proper order based on the data values, not the formatted values. Otherwise, the groups would be displayed in alphabetical order by formatted value.

```
proc report data=census nowd;
  columns educ sex income;
  define educ / 'Education'
  group
  order=internal;
run;
```

	Education	Sex	Adjusted gross income
	Post-graduate degree	Female	37000
		Female	99999
		Female	72313
		Female	33475
		Female	48481
		Female	23900
		Female	56000
		Female	1900
		Female	66400
		Female	8800
		Female	40725
		Female	41050
		Female	31257

This report is starting to look more interesting, but we need to define more variables. You can group your output by more than one variable. In this case, the variable sex will also be used as a group variable, and a label will be applied.

```
proc report data=census nowd;
  columns educ sex income;
  define educ / 'Education'
  group
  order=internal;
  define sex / 'Gender' group;
run;
```

	Education	Gender	Adjusted gross income
	Not HS graduate	Female	31140
		Male	438270
	HS graduate	Female	771729
		Male	3241629
	Some college	Female	659168
		Male	1431886
	Associates degree	Female	373868
		Male	691734
	Bachelors degree	Female	701787
		Male	1818369
	Post-graduate degree	Female	630000
		Male	1003537

Notice how the previous output now has subcategories of gender within categories of education. This grouping could be reversed to get subcategories of education within categories of gender. All you have to do is reverse the order of the two variables in the COLUMNS statement.

In this case, we'll leave the column order alone, and move on to defining the final column in the table. For this column, since we have subgroups of data, the REPORT procedure has guessed that we would like to see a sum of the

income values for each subgroup. We can change the default statistic to a mean by using the statistic keyword in a DEFINE statement.

In the code below, a label and format are applied to the income column, and the keyword mean is used to change the statistic.

```
proc report data=census nowd;
  columns educ sex income;
  define educ / 'Education'
    group
    order=internal;
  define sex / 'Gender' group;
  define income / 'Salary'
    mean
    format=dollar8.;
run;
```

	Education	Gender	Salary
Not HS graduate		Female	\$4,449
		Male	\$15,113
HS graduate		Female	\$17,539
		Male	\$33,419
Some college		Female	\$22,730
		Male	\$30,466
Associates degree		Female	\$33,988
		Male	\$40,690
Bachelors degree		Female	\$43,862
		Male	\$46,625
Post-graduate degree		Female	\$45,000
		Male	\$77,195

That completes the basic structure of our report. Now we can focus on making the table more attractive. In the example below, three new options are added to the PROC REPORT statement to improve the appearance. The HEADLINE option puts an underline between the column headings and the data. The HEADSKIP option puts a blank line between the table heading and the first row of data. The SPACING=2 option puts two spaces between each of the columns, making the table easier to read.

Further down in the code, there's another change. A BREAK statement is used to add a blank line after each of the education categories.

```
proc report data=census nowd
  headline headskip spacing=2;
  columns educ sex income;
  define educ / 'Education'
    group
    order=internal;
  define sex / 'Gender' group;
  define income / 'Salary'
    analysis mean
    format=dollar8.;
  break after educ / skip;
run;
```

	Education	Gender	Salary
Not HS graduate		Female	\$4,449
		Male	\$15,113
HS graduate		Female	\$17,539
		Male	\$33,419
Some college		Female	\$22,730
		Male	\$30,466
Associates degree		Female	\$33,988
		Male	\$40,690
Bachelors degree		Female	\$43,862
		Male	\$46,625
Post-graduate degree		Female	\$45,000
		Male	\$77,195

Looking at the results, you can see that this is quite an improvement on what you get from the basic PRINT, MEANS, and FREQ procedures.

You could get these same numbers from PROC MEANS with a BY or CLASS statement, but you could never make the output look this nice.

This is just an introduction to the REPORT procedure. To learn more, see the following papers in the SUGI 28 proceedings: "So you're still not using the REPORT Procedure. Why Not?" by Ray Pass, and "Getting Up to Speed with PROC REPORT," by Kimberly LeBouton.

➤ **PROC TABULATE**

To really understand TABULATE, you have to start very simply. The simplest possible table in TABULATE has to have three things: a PROC TABULATE statement, a TABLE statement, and a CLASS or VAR statement.

The CLASS and VAR statements are used to define your variables as either categorical (CLASS) or analysis (VAR). To use a variable to subgroup results, you list it in a CLASS statement. To use a variable to calculate a statistic, you list it in a VAR statement. In this example, we will use a VAR statement for income so we can compute statistics for income. Later examples will show the CLASS statement.

The TABLE statement is used to define the table. You use it to lay out the rows and columns that make up your table. In this first simple example, there is only one variable (income), and it is listed in the TABLE statement. The complete code and output are shown below.

```
proc tabulate data=census;
  var income;
  table income;
run;
```

Adjusted gross income
Sum
11793117.00

Okay, so this output isn't very exciting. To make things interesting, we need to add some categories. This is done with the CLASS statement. We also need to make the table two-dimensional, which we can do by adding a comma to the TABLE statement to divide the row definition from the column definition. An example procedure and output are shown below.

```
proc tabulate data=census;
  class educ;
  var income;
  table educ,
         income;
run;
```

	Adjusted gross income
	Sum
<b>educ</b>	
<b>Not HS graduate</b>	469410.00
<b>HS graduate</b>	4013358.00
<b>Some college</b>	2091054.00
<b>Associates degree</b>	1065602.00
<b>Bachelors degree</b>	2520156.00
<b>Post-graduate degree</b>	1633537.00

But we don't have to stop with one sub grouping. To make this a true cross-tabulation, we can add a second CLASS variable, and use it to define multiple columns for the table. This is done by adding the variable to the CLASS statement, and then adding it to the TABLE statement. To indicate that we want the columns of income broken down by gender, we use an asterisk to link the variable sex to the variable income. An example procedure and output are shown below.

```
proc tabulate data=census;
  class sex educ;
  var income;
  table educ,
         income*
         sex;
run;
```

	Adjusted gross income	
	Sum	
	Sex	
	Male	Female
<b>educ</b>		
<b>Not HS graduate</b>	438270.00	31140.00
<b>HS graduate</b>	3241629.00	771729.00
<b>Some college</b>	1431886.00	659168.00
<b>Associates degree</b>	691734.00	373868.00
<b>Bachelors degree</b>	1818369.00	701787.00
<b>Post-graduate degree</b>	1003537.00	630000.00

Now we have an interesting table structure for our report. However, the statistic isn't very useful. By default, the statistic generated by TABULATE for analysis variables

is a sum. To change this to a mean, we need to insert the statistic keyword mean. This statistic is added to the column definition with an asterisk.

```
proc tabulate data=census;
  class sex educ;
  var income;
  table educ,
         income*mean*
         sex;
run;
```

	Adjusted gross income	
	Mean	
	Sex	
	Male	Female
educ		
Not HS graduate	15112.76	4448.57
HS graduate	33418.86	17539.30
Some college	30465.66	22729.93
Associates degree	40690.24	33988.00
Bachelors degree	46624.85	43861.69
Post-graduate degree	77195.15	45000.00

Now we've got a nice report with the numbers we want, and we can turn to the cosmetic issues. By default, variables are labeled by TABULATE with their SAS label, if there is one. Otherwise the variable name is used. In this example, income has a label "Adjusted gross income" so that is used in the table. The variable educ does not have a label, so "educ" is used in the table.

You can apply new labels in the TABULATE procedure to customize the report. You can also remove labels altogether, if they are not needed. In the example procedure and output that follows, an equal sign is used to attach a text string to be used as the label to various items in the TABLE statement. Two labels are changed, and two unneeded labels are removed by setting them equal to a single space.

```
proc tabulate data=census;
  class sex educ;
  var income;
  table educ='Education',
         income='Average Salary'*
         mean=' '*
         sex=' ';
run;
```

	Average Salary	
	Male	Female
Education		
Not HS graduate	15112.76	4448.57
HS graduate	33418.86	17539.30
Some college	30465.66	22729.93
Associates degree	40690.24	33988.00
Bachelors degree	46624.85	43861.69
Post-graduate degree	77195.15	45000.00

Another nice TABULATE feature is the ability to easily add totals. There's a built-in variable called ALL that generates totals. We can use this to request average income by gender and overall, by adding ALL to the TABLE statement, as shown in the code below. The variable is added with a space as an operator, not an asterisk. This is because the total is a new column, not a sub grouping for existing columns. Also notice the parentheses that are used to apply the mean income to both the variable sex and the keyword all.

```
proc tabulate data=census;
  class sex educ;
  var income;
  table educ='Education',
         income='Average Salary'*
         mean=' '*
         (sex=' ' all);
run;
```

	Average Salary		
	Male	Female	All
Education			
Not HS graduate	15112.76	4448.57	13039.17
HS graduate	33418.86	17539.30	20463.53
Some college	30465.66	22729.93	27513.87
Associates degree	40690.24	33988.00	38057.21
Bachelors degree	46624.85	43861.69	45821.02
Post-graduate degree	77195.15	45000.00	60501.37

Finally, there's one last improvement we can make. We can apply a format to the table results. In this case the numbers are dollar amounts, so the format dollar makes sense. We can also give the format a width of 8. In TABULATE, the format width for the results controls the column width in the table. To squeeze more columns onto

a page, decrease the format width. A sample procedure and output are shown below.

```
proc tabulate data=census f=dollar8.;
  class sex educ;
  var income;
  table educ='Education',
         income='Average Salary'*
         mean=' '*
         (sex=' ' all);
run;
```

	Average Salary		
	Male	Female	All
Education			
Not HS graduate	\$15,113	\$4,449	\$13,039
HS graduate	\$33,419	\$17,539	\$28,464
Some college	\$30,466	\$22,730	\$27,514
Associates degree	\$40,690	\$33,988	\$38,057
Bachelors degree	\$46,625	\$43,862	\$45,821
Post-graduate degree	\$77,195	\$45,000	\$60,501

This was just a quick overview of the TABULATE procedure. To learn more:

In the SUGI 28 proceedings: “The Simplicity and Power of the TABULATE Procedure”, by Dan Bruns.

Online: “Anyone Can Learn PROC TABULATE”, by Lauren Haworth, [www2.sas.com/proceedings/sugi27/p060-27.pdf](http://www2.sas.com/proceedings/sugi27/p060-27.pdf)

### ➤ REPORT vs. TABULATE

You’ve now been introduced to two SAS reporting procedures. But which one should you use? For many reports, it doesn’t matter. Both procedures can produce the output that you need.

However, there are some types of reports where one procedure will be better than the other.

REPORT is generally best for:

- Reports with lots of text. The FLOW option allows you to wrap long labels within a column.

text1	text2
This is a very long text variable that is wrapped using PROC REPORT	This is a shorter text variable

- Reports with many row headings of varying width. Using the WIDTH option you can set the width of each column. The row headings in TABULATE have to be equal width.

text1	text2
This is a very long text variable that is wrapped using PROC REPORT	Short var.

- Reports that need customized summaries on each page, or each subsection. The COMPUTE block and the LINE statement allow you to calculate and display summary statistics and text messages.

Gender	Salary
Female	\$26,179
Ask for a raise!	
Male	\$35,642
Overpaid.	

The TABULATE procedure is generally best for:

- Complex cross-tabulations. REPORT can do limited cross-tabulations, but TABULATE has a lot more flexibility.

	Average Salary
All	\$32,488
Male	\$35,642
Female	\$26,179
Not HS graduate	
Male	\$15,113
Female	\$4,449
HS graduate	
Male	\$33,419
Female	\$17,539
Some college	
Male	\$30,466
Female	\$22,730
Associates degree	
Male	\$40,690
Female	\$33,988
Bachelors degree	
Male	\$46,625
Female	\$43,862
Post-graduate degree	
Male	\$77,195
Female	\$45,000

- Tables with multiple statistics displayed in the rows.

		Average Income		
		Sex		
		Male	Female	All
Education				
Not HS graduate	N	\$29	\$7	\$36
	Mean	\$15,113	\$4,449	\$13,039
HS graduate	N	\$97	\$44	\$141
	Mean	\$33,419	\$17,539	\$28,464
Some college	N	\$47	\$29	\$76
	Mean	\$30,466	\$22,730	\$27,514
Associates degree	N	\$17	\$11	\$28
	Mean	\$40,690	\$33,988	\$38,057
Bachelors degree	N	\$39	\$16	\$55
	Mean	\$46,625	\$43,862	\$45,821
Post-graduate degree	N	\$13	\$14	\$27
	Mean	\$77,195	\$45,000	\$60,501

For more information on the relative strengths of REPORT and TABULATE, see the following paper in the SUGI 28 proceedings: "Report vs. Tabulate - The Battle in Seattle!" by Ray Pass and Dan Bruns.

### ➤ SENDING RESULTS TO THE WEB

Creating web output from SAS is easy. As long as you are running version 8.0 or later, all you have to add is two lines of code. The first goes right before your reporting procedure:

```
ODS HTML FILE='myfilename.html';
```

After this line of code, you insert the code for your reporting procedure. Then, after the RUN statement that ends your procedure, you add the following line of code:

```
ODS HTML CLOSE;
```

The result is the output shown below. It is the same code created in the previous section on the TABULATE procedure.

	Average Salary		
	Male	Female	All
Education			
Not HS graduate	\$15,113	\$4,449	\$13,039
HS graduate	\$33,419	\$17,539	\$28,464
Some college	\$30,466	\$22,730	\$27,514
Associates degree	\$40,690	\$33,988	\$38,057
Bachelors degree	\$46,625	\$43,862	\$45,821
Post-graduate degree	\$77,195	\$45,000	\$60,501

If you don't like this look, you can change it by switching styles. The table above uses the default style, which is called "Default". For a different look try:

```
ODS HTML FILE='myfilename.html'
STYLE=BarrettsBlue;
* the TABULATE code goes here ;
ODS HTML CLOSE;
```

	Average Salary		
	Male	Female	All
Education			
Not HS graduate	\$15,113	\$4,449	\$13,039
HS graduate	\$33,419	\$17,539	\$28,464
Some college	\$30,466	\$22,730	\$27,514
Associates degree	\$40,690	\$33,988	\$38,057
Bachelors degree	\$46,625	\$43,862	\$45,821
Post-graduate degree	\$77,195	\$45,000	\$60,501

The output above shows the new look created by the style change from Default to BarrettsBlue. There are over a dozen styles for you to try. And, as you get more experi-

ence with ODS, you can even create you own custom style.

For more information on creating HTML output, see my paper “HTML for the SAS Programmer,” available at [http://www.laurenhaworth.com/pubs\\_current.htm](http://www.laurenhaworth.com/pubs_current.htm).

➤ **CREATING REPORTS TO IMPORT INTO WORD**

Just as you can use ODS HTML to create output destined for the web, you can use ODS RTF to generate a file in Rich Text Format, which can be opened in Word (or other word processors). Again, the syntax is to add an ODS statement with a FILE option before your reporting procedure, and an ODS CLOSE statement after the end of your procedure. In this case, both ODS statements specify RTF as the output destination, and the filename has an “.rtf” extension.

```
ODS RTF FILE='myfilename.rtf';
* the REPORT code goes here ;
ODS RTF CLOSE;
```

Education	Gender	Salary
Not HS graduate	Female	\$4,449
	Male	\$15,113
HS graduate	Female	\$17,539
	Male	\$33,419
Some college	Female	\$22,730
	Male	\$30,466
Associates degree	Female	\$33,988
	Male	\$40,690
Bachelors degree	Female	\$43,862
	Male	\$46,625
Post-graduate degree	Female	\$45,000
	Male	\$77,195

The output above shows what your SAS output looks like after it is opened in Word. The table of results is set up as a Word table, which means you can use all of the Word table formatting features to modify it if you like. To insert this table in another Word document, simple select the table and use copy/paste to drop it into another document.

There is one issue that comes up with ODS RTF output that you may need to address. By default, the RTF output has margins of .25” when you open it in Word. This is generally a lot smaller than most users would like. To fix this, you can reset the margins after you open the document in Word. Another option is to customize your ODS style to fix the margins. For instructions on how to do this, see the example on RTF margins in my paper “ODS

Tips & Tricks”, available at [http://www.laurenhaworth.com/pubs\\_current.htm](http://www.laurenhaworth.com/pubs_current.htm).

For more information about creating RTF output, check out the following papers:

“Now There Is an Easy Way to Get to Word, Just Use PROC TEMPLATE, PROC REPORT, and ODS RTF,” by Bob Hull, at [www2.sas.com/proceedings/sugi26/p163-26.pdf](http://www2.sas.com/proceedings/sugi26/p163-26.pdf).

“To ODS RTF and Beyond,” by David Shannon, at [www2.sas.com/proceedings/sugi27/p001-27.pdf](http://www2.sas.com/proceedings/sugi27/p001-27.pdf).

➤ **SENDING RESULTS TO PDF**

If you need to deliver printable reports via e-mail or the web, you may want to try out the PDF destination. This file format can be viewed using a free utility called Acrobat Reader, which you can download from [www.adobe.com](http://www.adobe.com).

The format can be viewed on many platforms, and its real strength is that your report will print out easily on many different printers, without any problems with margins and page breaks.

The code is quite simple. It’s just like the RTF code, except you call ODS PDF, and the filename gets a “.pdf”

```
ODS PDF FILE='myfilename.pdf';
* The REPORT code goes here ;
ODS PDF CLOSE;
```

Education	Gender	Salary
Not HS graduate	Female	\$4,449
	Male	\$15,113
HS graduate	Female	\$17,539
	Male	\$33,419
Some college	Female	\$22,730
	Male	\$30,466
Associates degree	Female	\$33,988
	Male	\$40,690
Bachelors degree	Female	\$43,862
	Male	\$46,625
Post-graduate degree	Female	\$45,000
	Male	\$77,195

You will notice that the output produced by ODS PDF looks a lot like the output produced by ODS RTF. That’s because the two ODS styles used by RTF and PDF are

closely related. You can create any look you like in either PDF or RTF by switching to another style, or even creating your own custom style.

One warning, if you are using SAS version 8.0 or 8.1, you will not be able to use the short “ODS PDF” syntax shown above. Instead, you will need to use ODS PRINTER with a PDF option, as shown below.

**ODS PRINTER PDF FILE='myfilename.pdf';**

PDF is part of the PRINTER family of destinations, and newer versions of SAS software include an alias for each of these types of printer output so you don't have to type the word PRINTER.

For more information on ODS PDF, check out the following paper in the SUGI 28 proceedings: “ODS PDF: It's not just for printing anymore!!” by Kevin Delaney.

➤ **SENDING RESULTS TO EXCEL**

There isn't an ODS EXCEL destination, or an ODS SPREADSHEET destination, but it's very easy to create an Excel file directly from SAS.

The way you do this is a cheat. Excel knows how to read HTML files, so you can open ODS HTML output directly from Excel. The tables, rows, and columns all transfer over correctly. You could always do a “Save As” to convert the file to Excel format, however there's a sneakier approach. In your ODS HTML statement, give the output file an “.xls” extension.

**ODS HTML FILE='myfile.xls';**  
**\* the TABULATE code goes here ;**  
**ODS HTML CLOSE;**

	A	B	C	D
1				
2				
3		<b>Average Salary</b>		
4		<b>Male</b>	<b>Female</b>	<b>All</b>
5	<b>Education</b>			
6	<b>Not HS graduate</b>	\$15,113	\$4,449	\$13,039
7	<b>HS graduate</b>	\$33,419	\$17,539	\$28,464
8	<b>Some college</b>	\$30,466	\$22,730	\$27,514
9	<b>Associates degree</b>	\$40,690	\$33,988	\$38,057
10	<b>Bachelors degree</b>	\$46,625	\$43,862	\$45,821
11	<b>Post-graduate degree</b>	\$77,195	\$45,000	\$60,501

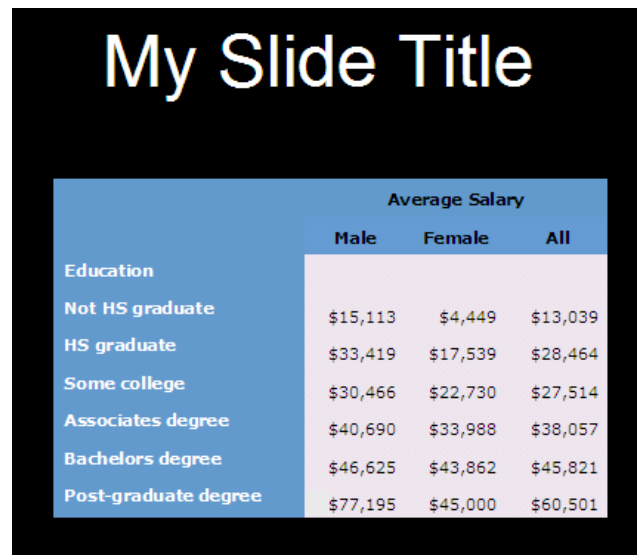
The above output shows what the file looks like when opened in Excel. Even though it's an HTML file, when you double-click on it to open the file, Excel automatically launches, and figures out how to handle the HTML. You can do a Save to ensure that the file is saved in true Excel format, but that's not necessary.

➤ **SENDING RESULTS TO POWERPOINT**

To get your SAS output into a PowerPoint slide, again ODS HTML is the best format to use. However, instead of opening the results directly from PowerPoint, you will be better off copying and pasting individual tables from your web browser into your PowerPoint slides.

This ensures that the tables are added in an appropriate size for your slide. If you open HTML directly from PowerPoint, it tends to convert the tables into extremely small tables on your slide, and resizing them is difficult.

The figure below shows a SAS table created by PROC TABULATE and output to HTML, as it looks when pasted onto a PowerPoint slide.



➤ **CONCLUSIONS**

This concludes our quick tour of your SAS reporting options. Hopefully you now have a basic idea of what options are available to you. In the future, when you need to design a new report, you have an idea what types of procedures and tools can generate that type of output.

Since this was only an introduction, as you start to use these techniques, be sure to explore the SAS documentation, as well as the SUGI papers referenced previously, for more in-depth discussion.

➤ **ACKNOWLEDGEMENTS**

SAS is a registered trademark or trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

➤ **CONTACTING THE AUTHOR**

Please direct any questions or feedback to the author at: [info@laurenhaworth.com](mailto:info@laurenhaworth.com)