SAS does the dirty work, how do I make it look pretty?
Sarah Bobiak, PhD National Comprehensive Cancer Network, Fort Washington, PA

ABSTRACT
This paper describes how to incorporate elements from PROC TABULATE with ODS style statements to create a customized table in Microsoft word. Starting with a brief introduction to the basic code to develop a table using PROC TABULATE, detailed instructions will be provided to walk through the process of generating and enhancing output in a rich text file using ODS styles. A brief discussion of PROC TABULATE will be followed by a more detailed description of ODS style options in each of the dimension expressions available in PROC TABULATE in order to customize the look of the table in Microsoft word.

INTRODUCTION
Proc tabulate is a highly versatile procedure that allows one to generate tables of summary statistics that could otherwise be produced using PROC PRINT, MEANS, or FREQ, however the presentation of these tables in PROC TABULATE is more flexible, especially when combined with ODS style options (Delwiche, 2004).

PROC TABULATE, THE BASICS
Proc tabulate is made up of 4 basic commands as follows:

    proc tabulate data=XXX;
    class variable list;
    var variable list;
    table page-dimension, row-dimension, column-dimension;
    run;

TABULATE:
The *tabulate* statement details the procedure and the data set used for analysis. By default, SAS excludes missing data among class variables from the table. To include missing data for all variables simply add the *missing* option to the procedure statement.

CLASS:
The *class* statement informs SAS of the categorical variables to be used to stratify analysis. By default, simple counts of observations are generated for each class variable in the table statement. If it is preferred to have only some missing values included, incorporate the *missing* option into the class statement. Multiple class statements can be used to specify different options for different sets of categorical variables.

Example 1a: Missing data excluded

    proc tabulate data=rawd.patient_characteristics;
    class racecomp;
    table racecomp, all;
    run;

<table>
<thead>
<tr>
<th>Race/Ethnicity</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caucasian</td>
<td>692</td>
</tr>
<tr>
<td>African American</td>
<td>58</td>
</tr>
<tr>
<td>Asian/ Pacific Islander</td>
<td>29</td>
</tr>
<tr>
<td>American Indian, Aleutian, Eskimo</td>
<td>1</td>
</tr>
<tr>
<td>Unknown</td>
<td>5</td>
</tr>
<tr>
<td>Other</td>
<td>13</td>
</tr>
<tr>
<td>Other/Hispanic</td>
<td>7</td>
</tr>
</tbody>
</table>

Example 1b. Missing data included.

    proc tabulate data=rawd.patient_characteristics;
    class racecomp / missing;
    table racecomp, all;
    run;
VAR:
The VAR statement informs SAS of the continuous variables for which statistics can be computed. To incorporate these analysis variables into the table, separate the class variable and the var variable of interest with an asterisk (*) followed by the particular statistics of interest. By default sums are computed. Among the additional options are N, mean, median, quantiles, column percent, row percent, total percent, minimum, maximum and range.

Example 2. Statistics with continuous variables.

```
proc tabulate data=rawd.patient_characteristics;
var agedx;
table agedx*(median n);
run;
```

TABLE:
The TABLE statement informs SAS about the organization of the output table including what to compute and include in the table (i.e., percents, means, etc). Multiple tables can be created by incorporating multiple table statements within one tabulate procedure. One important note, either a class or var statement or both can appear, however in order to be included in the table, the variable must be included in either the class or var statement. Table statements include the dimensions of the output table. The table can be as many as 3 dimensions. If one dimension is specified, that becomes the column dimension. If 2 dimensions are specified, that becomes the row and column dimensions. If 3 dimensions are specified, the page, row and column dimensions are provided.

Example 3a. One dimensional table statement.

```
proc tabulate data=rawd.patient_characteristics;
class gender;
table gender;
run;
```

Example 3b. Two dimensional table statement

```sas
proc tabulate data=rawd.patient_characteristics;
  class gender vitalstat;
  table gender, vitalstat;
run;
```

<table>
<thead>
<tr>
<th></th>
<th>Vitalstat</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alive</td>
<td>Dead</td>
</tr>
<tr>
<td>gender</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Male</td>
<td>234</td>
<td>137</td>
</tr>
<tr>
<td>Female</td>
<td>268</td>
<td>155</td>
</tr>
</tbody>
</table>

**Concatenate, cross, and group**
When specifying the dimensions, variables can be concatenated by listing them separately with a space dividing them; cross variables by using an asterisk to separate variables; and to group the variables enclose the variables in parentheses.

Example 3c. Concatenate, cross, group variables in table statement.

```sas
proc tabulate data=patient_characteristics;
  class racecomp yeardx_3cat vitalstat;
  var agedx;
  table racecomp yeardx_3cat*agedx*(mean stddev max), vitalstat;
run;
```

<table>
<thead>
<tr>
<th></th>
<th>Vitalstat</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alive</td>
<td>Dead</td>
</tr>
<tr>
<td>gender</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Male</td>
<td>234</td>
<td>137</td>
</tr>
<tr>
<td>Female</td>
<td>268</td>
<td>155</td>
</tr>
</tbody>
</table>

**Row and column labels:**
The given variable names are the default headings listed in the output table. Generally, more formal labels for the column and row headings are desired. This can be achieved by incorporating the title in single quotes following the variable name and an equal sign. This rule extends to requested statistics as well, where the statistics option is followed by an equal sign and the requested title in single quotes. If missing values are requested with the missing statement either in the tabulate procedure or the class statement, the misstext option can be specified as well with the equals sign and the value to be displayed for missing values. To hide the current label simply put a space between two single quotes for that variable.

Example 3d. Labeling column and row headings.
**proc tabulate data=patient_characteristics;**
*class* gender racecomp yeardx_3cat vitalstat;
*var* agedx;
*table* gender='Gender' yeardx_3cat*agedx*(mean='Average' stddev='SD' max='Max age'), vitalstat='Vital status';
*run;*

*F for formatting:
Further formatting the numbers in the table can be done by using the *F= expression followed by standard formats recognized by SAS (i.e., dollar4.; 6.2, etc.). This will allow rounding of percents or means and insertion of currency nomenclature.

Example 3e. Formatting using *F= expression.

**proc tabulate data=patient_characteristics;**
*class* gender racecomp yeardx_3cat vitalstat;
*var* agedx;
*table* gender='Gender' yeardx_3cat*agedx*(mean='Average' *f=2.0' stddev='SD' max='Max age' *f=2.0'), vitalstat='Vital status';
*run;*
PROC TABULATE, the options
CLASSLEV:
For each variable listed in the class statement there are a number of categories or values that define that class variable, for example gender would include male and female. The classlev statement allows control of the style elements for each class level value. Discussion of how to incorporate style elements into the class levels are addressed in the style section.

BOX:
All tables have a box above the row headings. Text can be inserted by using the box command followed by an equal sign with the text requested in single quotes.

```sas
proc tabulate data=patient_characteristics;
  class gender racecomp yeardx_3cat vitalstat hist_4cat;
  var agedx;
  table gender='Gender' hist_4cat='Histology at diagnosis'
         yeardx_3cat*agedx*(mean='Average' *f=2.0 stddev='SD' max='Max age' *f=2.0),
         vitalstat='Vital status'
   /box='Patient characteristics';
run;
```

ODS STYLE, the options:
Now that the content of the table has been generated, the focus will turn to customizing the table with available style options in ODS. Instructions describing appropriate placement of style statements in the procedure, class, classlev, var, table and box statements as well as the elements affected by those statements (i.e., variable names, headings, cells values, etc.) follow.

<table>
<thead>
<tr>
<th>Proc tabulate command:</th>
<th>Element affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>procedure</td>
<td>Entire table</td>
</tr>
<tr>
<td>class</td>
<td>All class labels</td>
</tr>
<tr>
<td>classlev</td>
<td>All class level values</td>
</tr>
<tr>
<td>var</td>
<td>All continuous variables</td>
</tr>
<tr>
<td>table</td>
<td>Specified output</td>
</tr>
<tr>
<td>box</td>
<td>Box at the top right</td>
</tr>
</tbody>
</table>
Style statements included in the tabulate command will affect the output of the table, however appropriate placement of the style commands within each of the commands outlined above will affect only that portion of the table allowing for greater control over how the final table is formatted. To begin, some of the available ODS style elements and how they can modify the look of the table are described.

**Foreground/ Background:**
To modify the foreground/background of various cells in the table use the following statements in the appropriate proc tabulate statement:

- `S=[Background=color name]`
- `S=[Foreground=color name]`

**Example 4a. Formatting the foreground/background**

```proc tabulate data=patient_characteristics style={foreground=blue};
class gender racecomp yeardx_3cat vitalstat hist_4cat;
classlev gender racecomp yeardx_3cat hist_4cat / s=[background=white];
classlev vitalstat / s=[background=white];
var agedx / s=[background=white];
table gender=(label='Gender')*(n=(s=[background=white]))
    hist_4cat=(label='Histology at diagnosis')*n=(s=[background=white])
    yeardx_3cat*agedx*(mean=(label='Average') s=[background=white])*f=2.0
    stddev=(label='SD' s=[background=white])
    max=(label='Max age' s=[background=white]) *f=2.0,
    vitalstat='Vital status' / box=(label='Patient characteristics' s=[foreground=blue background=white]);
run;```

**Text style:**
To modify the style of the text in the report, there are a number of options including font, size, style, weight, and width. Using the following code will make these adjustments in the applicable proc tabulate statements.

- `S=[font face="font style (i.e., Helvetica)"]`
- `S=[font size=#]`: note: this is relative to your browser, experiment to optimize font size.
- `S=[font style=style (i.e., italic, roman, slant)]`
- `S=[font weight=weight (i.e., medium, bold, demi_bold, light)]`
S=[font_width=width (i.e., normal, compressed, extra_compressed, narrow, wide)]

Example 4b. Formatting the text style.

```
proc tabulate data=patient_characteristics style={foreground=blue};
class gender racecomp yeardx_3cat vitalstat hist_4cat / s=[font_style=italic font_weight=bold];
classlev gender racecomp yeardx_3cat hist_4cat / s=[background=white font_face="helvetica" font_weight=light];
classlev vitalstat / s=[background=white font_weight=light];
var agedx / s=[background=white font_face="helvetica" font_weight=light];
table gender={label='Gender'}*(n={s=[background=white font_weight=light]})
hist_4cat={label='Histology at diagnosis'}*(n={s=[background=white font_weight=light]}) yeardx_3cat*agedx*(mean={label='Average'} s={background=white font_weight=light])*f=2.0 stddev={label='SD' s={background=white font_weight=light}}
max={label='Max age' s={background=white font_weight=light}}*f=2.0), vitalstat='Vital status' / box={label='Patient characteristics' s=[foreground=blue background=white font_size=10 font_width=wide]};
run;
```

### Vertical and horizontal text alignment:

To further modify the style of the text in the report, adjustments can be made to the alignment of variable labels and output values by using the following code in the appropriate proc tabulate statements. It is important to note these options are not available in the table statement.

S=[vjust (i.e., center, top, bottom)]
S=[just (i.e., center, left, right)]
Example 4c. Vertical and horizontal text alignment.

```proc tabulate data=patient_characteristics style={foreground=blue
just=center};
class gender racecomp yeardx_3cat vitalstat hist_4cat / s={font_style=italic
font_weight=bold};
classlev gender racecomp yeardx_3cat hist 4cat / s={[background=white
font_face="helvetica" font_weight=light vjust=center]};
classlev vitalstat / s={[background=white font_weight=light]};
var agedx / s={[background=white font_face="helvetica" font_weight=light
vjust=center]};
table gender={label='Gender'}*(n={s={[background=white font_weight=light]}})
hist_4cat={label='Histology at diagnosis'}*(n={s={[background=white
font_weight=light]}})
yeardx_3cat*agedx*{mean={label='Average' s={[background=white
font_weight=light]}}*f=2.0
stddev={label='SD' s={[background=white font_weight=light]}}
max={label='Max age' s={[background=white font_weight=light]}}*f=2.0),
vitalstat='Vital status'
/box={label='Patient characteristics' s={[foreground=blue background=white
font_size=10 font_width=wide]});
run;
```

**Patient characteristics**

<table>
<thead>
<tr>
<th>Gender</th>
<th>Alive</th>
<th>Dead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>227</td>
<td>134</td>
</tr>
<tr>
<td>Female</td>
<td>256</td>
<td>150</td>
</tr>
</tbody>
</table>

**Histology at diagnosis**

<table>
<thead>
<tr>
<th>Type</th>
<th>Alive</th>
<th>Dead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carcinoid</td>
<td>235</td>
<td>95</td>
</tr>
<tr>
<td>pNET</td>
<td>97</td>
<td>49</td>
</tr>
<tr>
<td>NET, NOS</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>Other</td>
<td>48</td>
<td>37</td>
</tr>
</tbody>
</table>

**Year of diagnosis**

<table>
<thead>
<tr>
<th>Year</th>
<th>Average</th>
<th>SD</th>
<th>Max age</th>
<th>Max age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior to 2000</td>
<td>Average</td>
<td>SD</td>
<td>Max age</td>
<td>Max age</td>
</tr>
<tr>
<td>Age at diagnosis</td>
<td>40</td>
<td>15.71</td>
<td>73</td>
<td>78</td>
</tr>
<tr>
<td>2000 to 2003</td>
<td>Average</td>
<td>SD</td>
<td>Max age</td>
<td>Max age</td>
</tr>
<tr>
<td>Age at diagnosis</td>
<td>53</td>
<td>14.04</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>2004 to 2011</td>
<td>Average</td>
<td>SD</td>
<td>Max age</td>
<td>Max age</td>
</tr>
<tr>
<td>Age at diagnosis</td>
<td>54</td>
<td>13.45</td>
<td>85</td>
<td>88</td>
</tr>
</tbody>
</table>

**Cell dimensions and white space**

Perhaps the design of the table requires more white space to ease interpretation. ODS style options allow adjustments to the white space surrounding text in the cell by adjusting the height and width of the cells. The following code is inserted in the appropriate proc tabulate statement to make adjustments as desired.

S=[cellwidth=#]
S=[cellheight=#]
Example 4d. Cell dimensions and white space

``` SAS
proc tabulate data=patient_characteristics style={foreground=blue just=center};
class gender racecomp yeardx_3cat vitalstat hist_4cat / s=[font_style=italic font_weight=bold cellheight=75];
classlev gender racecomp yeardx_3cat hist 4cat / s=[background=white font_weight=light vjust=center];
classlev vitalstat / s=[background=white font_weight=light cellwidth=200];
var agedx / s=[background=white font_face="helvetica" font_weight=light vjust=center];
table gender=[label='Gender']*(n=[background=white font_weight=light])
hist_4cat=[label='Histology at diagnosis']*(n=[background=white font_weight=light])
yeardx_3cat*agedx*(mean=[label='Average' s=[background=white font_weight=light]]*f=2.0)
stddev=[label='SD' s=[background=white font_weight=light]]
max=[label='Max age' s=[background=white font_weight=light]]*f=2.0),vitalstat='Vital status' /box=[label='Patient characteristics' s=[foreground=blue background=white font_size=10 font_width=wide]];
run;
```

### Patient characteristics

<table>
<thead>
<tr>
<th>Gender</th>
<th>Alive</th>
<th>Dead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>227</td>
<td>134</td>
</tr>
<tr>
<td>Female</td>
<td>256</td>
<td>150</td>
</tr>
</tbody>
</table>

### Histology at diagnosis

<table>
<thead>
<tr>
<th>Histology at diagnosis</th>
<th>Alive</th>
<th>Dead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carcinoid</td>
<td>235</td>
<td>95</td>
</tr>
<tr>
<td>pNET</td>
<td>97</td>
<td>49</td>
</tr>
<tr>
<td>NET, NOS</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>Other</td>
<td>48</td>
<td>37</td>
</tr>
</tbody>
</table>

### Year of diagnosis

#### Prior to 2000

- **Age at diagnosis**
  - Average: 40
  - SD: 15.71
  - Max age: 73

#### 2000 to 2003

- **Age at diagnosis**
  - Average: 53
  - SD: 14.04
  - Max age: 80

#### 2004 to 2011

- **Age at diagnosis**
  - Average: 54
  - SD: 13.45
  - Max age: 85

---

**Conditional formatting**

There may be some instances when highlighting cells with a threshold value or specific value is important to the ease of interpretation of the data. In such circumstances, it is possible to define under what conditions conditional

---
formatting is desired. In order to achieve this, a *proc format* is needed to define the parameters of the conditional formatting. Within the proc tabulate procedure the format previously defined is called to identify the conditional formatting. The code below is provided as an example.

```plaintext
*title 'Sample of conditional formatting';
proc format;
  value colorcode
      0 - 50 = 'red'
     50-5000 = 'black';
run;
proc tabulate data=patient_characteristics style={foreground=blue just=center foreground=colorcode};
class gender racecomp yeardx_3cat vitalstat hist_4cat /
     s=[font_style=italic font_weight=bold cellheight=75];
classlev gender racecomp yeardx_3cat hist_4cat /
     s=[background=white font_weight=light vjust=center];
classlev vitalstat / s=[background=white font_weight=light cellwidth=200];
var agedx / s=[background=white foreground=black font_face="helvetica"
     font_weight=light vjust=center];
table gender=[label='Gender']*(n=[background=white font_weight=light])
       hist_4cat=[label='Histology at diagnosis']*(n=[background=white
     font_weight=light])
       yeardx_3cat*agedx*(mean=[label='Average' s=[background=white
     font_weight=light])*f=2.0 stddev=[label='SD'
     s=[background=white font_weight=light])
       max=[label='Max age' s=[background=white font_weight=light])*f=2.0),
     vitalstat='Vital status'
     /box=[label='Patient characteristics' s=[foreground=blue background=white
     font_size=10 font_width=wide]];
run;
```

### Patient characteristics

<table>
<thead>
<tr>
<th></th>
<th>Alive</th>
<th>Dead</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gender</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>227</td>
<td>134</td>
</tr>
<tr>
<td>Female</td>
<td>256</td>
<td>150</td>
</tr>
<tr>
<td><strong>Histology at diagnosis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carcinoid</td>
<td>235</td>
<td>95</td>
</tr>
<tr>
<td>pNET</td>
<td>97</td>
<td>49</td>
</tr>
<tr>
<td>NET, NOS</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>Other</td>
<td>48</td>
<td>37</td>
</tr>
<tr>
<td><strong>Year of diagnosis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prior to 2000</td>
<td>Average 40</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>SD 15.71</td>
<td>13.34</td>
</tr>
<tr>
<td></td>
<td>Max age 73</td>
<td>78</td>
</tr>
<tr>
<td>2000 to 2003</td>
<td>Average 53</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>SD 14.04</td>
<td>12.87</td>
</tr>
<tr>
<td></td>
<td>Max age 80</td>
<td>80</td>
</tr>
<tr>
<td>2004 to 2011</td>
<td>Average 54</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>SD 13.45</td>
<td>12.97</td>
</tr>
<tr>
<td></td>
<td>Max age 85</td>
<td>88</td>
</tr>
</tbody>
</table>

All values 0-50 have been highlighted in red.
Document formatting

ODS style commands will not affect the formatting of the output table when viewed in SAS. However the benefit of ODS is that the output tables can be generated in a rich text file where the formatting and style options are applicable. Now that the placement and purpose of a number of style elements has been explained, it is important to understand that all of these statements must be contained within the ODS commands as shown below. In addition, there are other capabilities embedded in ODS to help control how many tables appear on each page, page orientation, as well as others. In the following example, the command orientation=landscape/portrait allows control over the orientation of the tables in the final document. For each proc tabulate command a new page is generated automatically in the rich text file. To modify this, the command ods startpage=no/yes is inserted and tells SAS when to generate a new page for the proc tabulate command and when to keep them on the same page. Sample code is provided below for reference.

```sas
options nodate orientation=landscape;
ods rtf file='z:\NESUG\2011\Tables\tables.rtf';
ods startpage=no;
ods rtf close;
```

COMMANDS AS OUTLINED ABOVE ARE EMBEDDED HERE IN ORDER TO OUTPUT THE TABLES TO A RICH TEXT FILE AND OBTAIN THE BENEFITS OF THE ODS FORMATTING REQUESTS

CONCLUSION

SAS can provide table outputs using a number of different procedure statements. The purpose of the report ultimately determines which SAS procedure to use. If the primary purpose for a report is a custom report with specific formatting, PROC TABULATE with ODS style options is one way to achieve this. This paper has provided the reader with a basic introduction to the capabilities of PROC TABULATE and a larger discussion of how to incorporate ODS style options to customize the exported table.

REFERENCES


CONTACT INFORMATION

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

Name: Sarah Bobiak
Enterprise: National Comprehensive Cancer Network
Address: 275 Commerce Dr.
City, State ZIP: Fort Washington, PA
Work Phone: 215.690.0569
Fax: 215.690.0280
E-mail: Bobiak@nccn.org
Web: www.nccn.org

The author would like to acknowledge the NCCN and its financial and material support in the development of the NCCN Outcomes Database.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.