Tutorial:

PROC TABULATE and PROC SGPLOT

Jaisreet Khaira

TABULATE PROCEDURE

The TABULATE process helps show facts about our data in a table. We can use different pieces of information from our data to make tables that are simple or very detailed.

PROC TABULATE is a tool that does similar math as other tools like MEANS, FREQ, and REPORT. It lets us be flexible with how we organize and group our data, making it easy to create simple tables. We can also make the tables look nice by adding labels and formatting. If we use the ACCESSIBLETABLE option, PROC TABULATE helps create tables that are easy to understand and allows us to add captions to them.

The dataset Math contains data about how students do in math at two schools in Portugal. The dataset contains information about student grades, demographics, social aspects, and school-related features. They got this data from school reports and asking students questions. We are going to be using this dataset to show some of the features of TABULATE Procedure.

To make a basic table using PROC TABULATE, we need at least two things:

1. VAR statement: This tells us which variables we want to analyze.
2. TABLE statement: This helps us decide which variables to include in the table and where to put them.

Finding the total sum:

In this section, we are going to find the total sum of final grades (G3) of all the students in both the schools.

```
proc tabulate data = math;
 var G3;
 table G3;
run;
```

This results in the following table:

G3
Sum
4114.00

Figure 1: Finding Total Sum

Looking at this result, PROC TABULATE defaults to showing the total sum for the analyzed variable, which is G3 in this example.

Exploring different statistics:

If you want to show a different number instead of the total sum, just change the TABLE statement. To choose the numbers you want, mention the variable name, put an asterisk (*) after it, and then say the number you're interested in. In PROC TABULATE, this asterisk (*) is like an operator telling SAS what calculations to do for each variable.

There are many numbers you can choose from in PROC TABULATE, and you can find a big list in the SAS documentation. This article will discuss some common numbers like the smallest (MIN), biggest (MAX), average (MEAN), count (N), and total sum (SUM).

```
proc tabulate data = math;
 var G3;
 table G3*mean;
run;
```

The result will show a table that gives the average G3 grades across both the schools in math dataset.

G3
Mean
10.42

Figure 2: Average grade

To find a different statistic, you can replace "mean" with the desired statistic, and you'll get the result you're looking for.

To show different statistics for a variable, you can list the variable, add an asterisk (*), and then mention each statistic you want in the TABLE statement. For instance, if you want the MEAN, MIN, and MAX, you will set up the TABLE statement like this:

```
proc tabulate data = math;
 var G3;
 table G3*mean G3*min G3*max;
run;
```

The result will show the mean, min and max of the final grade:

G3	G3	G3
Mean	Min	Max
10.42	0.00	20.00

Figure 3: Combining Statistics

Alternatively, this can also be written as:

```
proc tabulate data = math;
 var G3;
 table G3*(mean max min);
run;
```

As you can see in the table below, this way of writing also results in a cleaner output:

G3		
Mean	Max	Min
10.42	20.00	0.00

Figure 4: Combining Statistics - alternative method

Adding Classification Variables:

You can make your PROC TABULATE results even better by including groups or categories using class.

For instance, if you wish to find the average final grades (G3) for both schools in the dataset, you can include the school as a class in your PROC TABULATE statement to make that happen.

```
proc tabulate data = math;
 class school;
 var G3;
 table G3*mean*school;
run;
```

So, the resulting table will show the mean of G3 for both GP and MS schools as below:

G3	
Mean	
school	
GP	MS
10.49	9.85

Figure 5: Adding Classification variables

Adding Labels and Formatting Output:

Like with other SAS tools, you can make your results look better by adding formats and labels. Also, PROC TABULATE allows you to get rid of labels for rows or columns to make your results less crowded.

For instance, in the table mentioned earlier, if you don't want to show the mean row label and want to give a name to G3, you can do that by putting an equal sign (=) after the variable name in the TABLE statement, followed by the name you want in quotes. To remove a label for a row or column, just leave a space between the quotation marks. If you want the mean numbers to look like money with dollar signs, commas, and no decimal places, you can add the dollar8 format to the PROC TABULATE statement.

```
proc tabulate data = math;
 class school;
 var G3;
 table G3="Final Grades"*mean=""*school;
run;
```

With these changes, the output will now show "Final Grades" instead of "G3" and won't include the mean row label.

Final Grades	
school	
GP	MS
10.49	9.85

Figure 6: Formatting

To flip the results and make the school values appear as rows instead of columns, we use a comma (,) in the TABLE statement. The part before the comma becomes the rows, and the part after the comma becomes the columns.
In this case, school is listed before the comma, while G3 and MEAN are listed after the comma, like this:

```
proc tabulate data = math;
 class school;
 var G3;
 table school, G3*mean;
run;
```

This will result in School as the rows and G3*mean as the columns.

	G3
	Mean
school	
GP	10.49
MS	9.85

Figure 7: Switching rows and columns

Multiple Classification Variables:

You can make your tables more detailed by adding another category. For example, let's say you want to find the average of final grades for all the students just like before, but now you also want to see the average G3 grades based on sex of the student.
So first, we will add sex variable to our class statement and then to the table statement as follows:

```
proc tabulate data = math;
 class school sex;
 var G3;
 table school, G3*mean*sex;
run;
```

As you can see in the output below, we now have the desired result of average final grades based on school and gender.

	G3	
	Mean	
	sex	
	F	M
school		
GP	9.97	11.06
MS	9.92	9.76

Figure 8: Multiple Classification variables

Similarly, we can make our table more complex by adding more variables. Suppose we now want to include age in our previous example.

```
proc tabulate data = math;
 class school sex age;
```

```
 var G3;
 table school*sex, G3="Final Grades"*mean=""*age;
run;
```

This will result in the following table. The table below now shows the average grade for each age group, gender and school.
Note: Now, you might see that there are many empty spaces in the table. These empty spaces happen when there are no student that match a particular combination of categories in the table.

		Final Grades							
		age							
		15	16	17	18	19	20	21	22
school	sex								
GP	F	9.55	10.54	10.62	8.76	9.00	.	.	.
	M	12.73	11.56	9.69	9.57	9.17	18.00	.	8.00
MS	F	.	.	9.63	10.50	4.50	15.00	.	.
	M	.	.	12.50	10.36	6.25	9.00	7.00	.

Figure 9: combination of multiple variables

SGPLOT PROCEDURE

The SGPLOT procedure is versatile; it can generate one or multiple figures and layer them on a single axis. Additionally, you can utilize SGPLOT to make various graphics like histograms, scatter plots, and many other types of plots. It is one of the SG procedures that contain the ODS Statistical Graphics Package.

To create a simple PROC SGPLOT statement, we need:
1. One or more plot statements: These are like instructions for different types of plots, such as SCATTER, SERIES, VBOX, VBAR, HIGHLOW, and BUBBLE.
2. Optional statements: These are extra instructions to control specific plot features, like XAXIS, YAXIS, REFLINE, INSET, and KEYLEGEND.

Basic Scatterplot

Let's start with something simple—the SCATTER statement. It makes a scatter plot, and we just need to tell it which variables to use with X= and Y=.

```
proc sgplot data = math;
      title "Scatterplot";
      scatter x=G1 y=G3;
run;
```

This makes a graph where the x-axis is G3 and the y-axis is G1. It illustrates the connection between grades in period 1 and the final grades of math students. The graph indicates a positive link between the two variables—meaning if a student scores higher in Period 1, they are likely to also score higher in G3 (final grade).

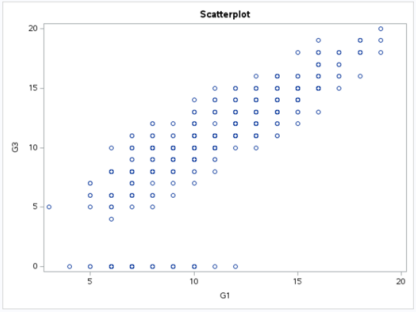

Figure 10: Scatterplot

We can also make the scatterplot more informative by including different groups of variables. For example, if we want to include the gender of the students along with the grades, we can do that by adding the group option. Please note it comes after a slash because it is optional.

```
proc sgplot data = math;
      scatter x=G1 y=G3/ group = sex;
run;
```

As you can see in the graph below, the data is now color coded based on the gender.

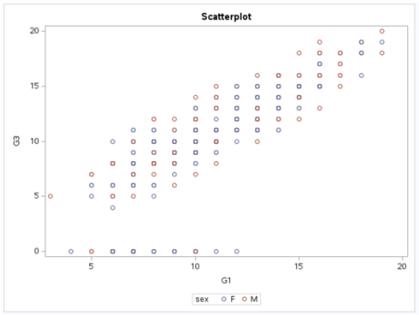

Figure 11: Scatterplot with grouping

To create different graphs, simply change the keyword "scatter" to other keywords like:

1. Bubble Plot: Shows the relationship between three variables. Use the bubble statement with x= variable_name1, y= variable_name2, and size= variable_name3.

```
proc sgplot data=your_dataset;
  bubble x=variablename1 y=variablename2 size=variablename3;
run;
```

2. Series Plot: Makes a line plot and only requires x= and y= values.

```
proc sgplot data=your_dataset;
  series x=variable_name1 y=variable_name2;
run;
```

3. High-low Plot: Connects high and low values with floating vertical or horizontal line segments. Specify HIGH= and LOW=, along with either X= or Y=. X= creates vertical line segments, and Y= creates horizontal ones. Optionally, use CLOSE= to specify a variable with closing tick marks.

```
proc sgplot data=your_dataset;
  highlow high=variable_name1 low=variable_name2 / x=variable_name3;
run;
```

4. Horizontal Box Plot: Provides information about the distribution of a continuous variable. HBOX creates a horizontal box plot, and VBOX creates a vertical one. The required argument is the name of a numeric variable. Include it directly after the HBOX or VBOX keyword. The category= option creates a separate box for each distinct value of a categorical variable. Combine it with group= to add groups within each category.

```
proc sgplot data=your_dataset;
  hbox variable_name;
run;
```

5. Vertical/Horizontal Bar Chart: A common way to summarize categorical data. Use HBAR for a horizontal bar chart and VBAR for a vertical one. All you need is a categorical variable. Optionally, use Group= option to create a separate bar for each distinct value of a grouping variable.

```
proc sgplot data=your_dataset;
  vbar variable_name / group=grouping_variable;
run;
```

6. Heat Map: Creates a grid of colored rectangles by grouping and plotting data values. Specify X= and Y= to choose the variables for grouping.

```
proc sgplot data=your_dataset;
  heatmap x=variable_name1 y=variable_name2 /
colorresponse=variable_name3;
run;
```

7. Vertical Line Chart: Plots statistics based on a categorical variable, similar to VBAR. Use RESPONSE= and STAT= options as with VBAR. VLINE doesn't include plot markers unless MARKERS is specified. For a horizontal line chart, use HLINE.

```
proc sgplot data=your_dataset;
  vline category=variable_name / response=variable_name2 stat=mean
markers;
run;
```

8. Regression Plot: Adds a regression line to a scatter plot, depicting the relationship between two variables. Specify the variables to plot using the required arguments X= and Y=.

```
proc sgplot data=your_dataset;
  scatter x=variable_name1 y=variable_name2;
  reg x=variable_name1 y=variable_name2;
run;
```

Combination Plots:

With SGPLOT, you can make combination plots by adding multiple plot requests. These plots layer on top of each other in the same graph space, sharing the same axis system. List the plot requests in the order you want them to appear, and each new request adds another layer to the graph.

```
proc sgplot data=your_data;
   scatter x=variable1 y=variable2;
   series x=variable3 y=variable4;
   /* Add more plot requests as needed */
run;
```

APPENDIX

REFERENCES

1. https://sascrunch.com/proc-tabulate/
2. https://support.sas.com/resources/papers/proceedings10/154-2010.pdf
3. https://www.pharmasug.org/proceedings/2022/QT/PharmaSUG-2022-QT-169.pdf
4. https://support.sas.com/resources/papers/proceedings10/154-2010.pdf
5. https://archive.ics.uci.edu/dataset/320/student+performance