

## Descriptive Statistics – Categorical Variables



<b>Introduction.....</b>	<b>41</b>
<b>Computing Frequency Counts and Percentages.....</b>	<b>42</b>
<b>Computing Frequencies on a Continuous Variable .....</b>	<b>44</b>
<b>Using Formats to Group Observations.....</b>	<b>45</b>
<b>Histograms and Bar charts.....</b>	<b>48</b>
<b>Creating a Bar Chart Using PROC SGPLOT.....</b>	<b>49</b>
<b>Using ODS to Send Output to Alternate Destinations.....</b>	<b>50</b>
<b>Creating a Cross-Tabulation Table .....</b>	<b>52</b>
<b>Changing the Order of Values in a Frequency Table .....</b>	<b>53</b>
<b>Conclusions.....</b>	<b>55</b>

---

### Introduction

This chapter continues with methods of examining categorical variables. You will learn how to produce frequencies for single variables and then extend the process to create cross-tabulation tables. You will also learn several graphical approaches that are used with categorical variables. Finally, you will learn how to use SAS to group continuous variables into categories using a variety of techniques. Let's get started.

## Computing Frequency Counts and Percentages

You can use PROC FREQ to count frequencies and calculate percentages for categorical variables. This procedure can count unique values for either character or numeric variables. Let's start by computing frequencies for Gender and Drug in the Blood\_Pressure data set used in the previous chapter.

### Program 3.1: Computing Frequencies and Percentages Using PROC FREQ

```
title "Computing Frequencies and Percentages Using PROC FREQ";
proc freq data=example.Blood_Pressure;
  tables Gender Drug;
run;
```

PROC FREQ uses a TABLES statement to identify which variables you want to process. This program selects Gender and Drug. Here is the output:

#### Computing Frequencies and Percentages Using PROC FREQ

##### The FREQ Procedure

Gender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	28	48.28	28	48.28
M	30	51.72	58	100.00

Frequency Missing = 2

Drug	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Drug A	20	33.33	20	33.33
Drug B	20	33.33	40	66.67
Placebo	20	33.33	60	100.00

By default, PROC FREQ computes frequencies, percentages, cumulative frequencies, and cumulative percentages. In addition, it reports the frequency of missing values. If you do not want all of these values, you can add options to the TABLES statement and specify what statistics you want or do not want. For example, if you want only frequencies and percentages, you can use the TABLES option NOCUM (no cumulative statistics) to remove them from the output, like this:

**Program 3.2: Demonstrating the NOCUM Tables Option**

```

title "Demonstrating the NOCUM Tables Option";
proc freq data=example.Blood_Pressure;
    tables Gender Drug / nocum;
run;

```

Because NOCUM is a statement option, it follows the usual SAS rule: it follows a slash. The following output shows the effect of the NOCUM option:

**Demonstrating the NOCUM Tables Option****The FREQ Procedure**

Gender	Frequency	Percent
F	28	48.28
M	30	51.72

Frequency Missing = 2

Drug	Frequency	Percent
Drug A	20	33.33
Drug B	20	33.33
Placebo	20	33.33

As you can see, the output now contains only frequencies and percents.

One TABLES option, MISSING, deserves special attention. This option tells PROC FREQ to treat missing values as a valid category and to include them in the body of the table. Program 3.3 shows the effect of including the MISSING option:

**Program 3.3: Demonstrating the Effect of the MISSING Option with PROC FREQ**

```

title "Demonstrating the effect of the MISSING Option";
proc freq data=example.Blood_Pressure;
    tables Gender Drug / nocum missing;
run;

```

## 44 SAS Statistics by Example

Here is the output:

**Demonstrating the effect of the MISSING Option**

The FREQ Procedure

Gender	Frequency	Percent
	2	3.33
F	28	46.67
M	30	50.00

Drug	Frequency	Percent
Drug A	20	33.33
Drug B	20	33.33
Placebo	20	33.33

Notice that the two subjects with missing values for Gender are now included in the body of the table. Even more important, the percentages for females and males have changed. When you use the MISSING option, SAS treats missing values as a valid category and includes the missing values when it computes percentages. To summarize, without the MISSING option, percentages are computed as the percent of all nonmissing values; with the MISSING option, percentages are computed as the percent of all observations, missing and nonmissing.

---

## Computing Frequencies on a Continuous Variable

What happens if you compute frequencies on a continuous numeric variable such as SBP (systolic blood pressure)? Program 3.4 shows what happens when you try to compute frequencies on a continuous numeric variable:

### Program 3.4: Computing Frequencies on a Continuous Variable

```
title "Computing Frequencies on a Continuous Variable";
proc freq data=example.Blood_Pressure;
  tables SBP / nocum;
run;
```

This is the output:

#### Computing Frequencies on a Continuous Variable

The FREQ Procedure

SBP	Frequency	Percent
108	1	1.79
110	1	1.79
112	2	3.57
114	2	3.57
118	3	5.36
120	5	8.93
122	1	1.79
124	4	7.14
126	2	3.57
128	1	1.79
130	4	7.14
134	6	10.71
136	4	7.14
138	4	7.14
140	5	8.93
142	2	3.57
144	2	3.57
146	1	1.79
148	1	1.79
150	1	1.79
156	1	1.79

Frequency Missing = 4

Each unique value of SBP is considered a category. Now let's see how to group continuous values into categories.

---

## Using Formats to Group Observations

SAS can apply formats to character or numeric variables. What is a format? Suppose you have been using M for males and F for females but you want to see the labels Male and Female in your output. You can create a format that associates any text (Male, for

example) to one or more values. To demonstrate, let's start by making a format for Gender, SBP, and DBP, and using these formats with PROC FREQ.

**Program 3.5: Writing a Format for Gender, SBP, and DBP**

```
proc format;
  value $gender 'M' = 'Male'
               'F' = 'Female';
  value sbpgroup low-140 = 'Normal'
            141-high      = 'High';
  value dbpgroup low-80 = 'Normal'
            81-high      = 'High';
run;

proc freq data=example.Blood_Pressure;
  tables Gender SBP DBP / nocum;
  format Gender $gender.
         SBP sbpgroup.
         DBP dbpgroup.;
run;
```

You use PROC FORMAT to create formats—labels associated with values. If you are planning to create formats for character variables, the format names must start with a dollar sign. Formats to be used with numeric variables cannot start with a dollar sign. In addition, format names cannot end with a number. All format names are limited to a maximum of 32 characters, including the initial dollar sign for character format names. Finally, format names can contain letters, digits, and the underscore character. You name each format on a VALUE statement. This statement lets you specify unique values, groups of values, or ranges of values on the left side of the equal sign, and labels that you want to associate with these values on the right side of the equal sign.

The first format in Program 3.5 is \$gender. This name is a good choice because this format will be used later with the variable Gender (a character variable). All the format names are, however, completely arbitrary: you could have called this format \$xyz if you wanted to. The \$gender format associates the text "Male" with M and "Female" with F. You can use either single or double quotation marks when you create formats—just be sure to use double quotation marks if the format that you are creating contains an apostrophe (which is rendered as a single quotation mark).

The next two formats are to be used with the two variables SBP and DBP. For the SBPGROUP format, the range of values associated with the text "Normal" is from the lowest nonmissing value to 190. You can use the keywords LOW and HIGH when you are defining format ranges.

PROC FORMAT creates formats, but it does not associate any of these formats with SAS variables (even if you are clever and name them so that it is clear which format will go with which variable). To associate a format with one or more SAS variables, you use a FORMAT statement. You can place this statement in either a DATA step or a PROC step. If you place a FORMAT statement in a PROC step (as in Program 3.5), the format will be associated with the variables only for the duration of that procedure. If you place a FORMAT statement in a DATA step, the formats will be permanently assigned to the variables.

In a FORMAT statement, you start with the keyword FORMAT, followed by one or more variables names, followed by the format you want to associate with the variables you listed. On a FORMAT statement, you *must* follow each format name with a period. If you omit the period, SAS will think that you are writing a variable name and not a format. It is slightly confusing—when you create the format with a VALUE statement, you do not end the name with a period (SAS knows this is a format name). When you write a FORMAT statement, you must end the format name with a period.

Let's see what happens when you run Program 3.5—here is the output:

### Computing Frequencies on a Continuous Variable

#### The FREQ Procedure

Gender	Frequency	Percent
Female	28	48.28
Male	30	51.72

Frequency Missing = 2

SBP	Frequency	Percent
Normal	48	85.71
High	8	14.29

Frequency Missing = 4

DBP	Frequency	Percent
Normal	24	42.86
High	32	57.14

Frequency Missing = 4

Instead of F's and M's you now see Female and Male. Instead of frequencies for individual values of SBP and DBP, you see only two categories, Normal and High.

---

## Histograms and Bar Charts

Sometimes it is useful to show frequencies in a graphical display. With SAS, you have several options: First, there is an older SAS procedure called GCHART, which is part of the SAS/GRAPH collection of procedures. A newer procedure, PROC SGPLOT, can produce a wide variety of plots and charts.

The first example of a bar chart uses PROC GCHART to display the frequencies of a variable called Region (region of the country) from a data set called `store`. You can skip this section and go right to the next section, which shows you how to create a bar chart using PROC SGPLOT. However, at some point you might need to run or modify an older SAS program that uses PROC GCHART. Here is the code:

### Program 3.6: Generating a Bar Chart Using PROC GCHART

```
options reset=all;
pattern value = solid color = blue;
title "Generating a Bar Chart - Using PROC GCHART";
proc gchart data=store;
    vbar Region;
run;
quit;
```

The first statement (GOPTIONS, which stands for graphic options), is not mandatory, but if you have been using any SAS/GRAPH procedures during your SAS session, it is a good idea to reset all the options to their default values. Why? Because when you set any graphic option such as color or plotting symbol, these options remain in effect until you change them. This behavior is similar to TITLE statements, which persist unless you change them or omit the titles completely.

The PATTERN statement enables you to select the type of bar (SOLID in this case) and the color of the bars. A useful hint is to set VALUE=EMPTY if you are sending the output to a dot matrix printer. The EMPTY option displays only the outline of the box and keeps you from rushing out to the office supply store to buy more ink cartridges.

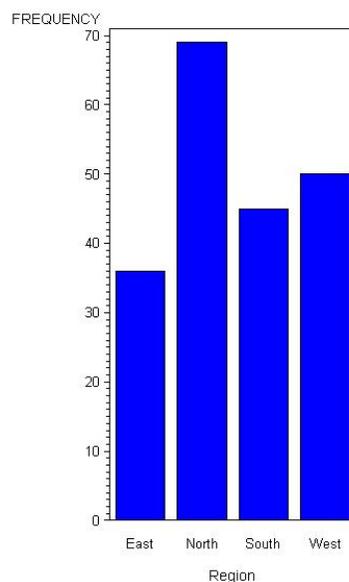
The VBAR (vertical bar) statement lets you list the variables for which you want to generate bar charts. If you prefer a horizontal bar chart, use the HBAR statement instead.

Notice the QUIT statement in this program. Certain procedures in SAS such as PROC GCHART have something called RUN-group processing. This kind of processing keeps

the procedure in memory, even after it encounters a RUN statement. Because the procedure is still in memory, you can request additional charts or, in the case of other procedures, new models, etc. The QUIT statement ends the procedure. If you omit a QUIT statement, the procedure ends when the next DATA or PROC step executes.

Here is the output from Program 3.6:

### Generating a Bar Chart - Using PROC GCHART



---

### Creating a Bar Chart Using PROC SGPLOT

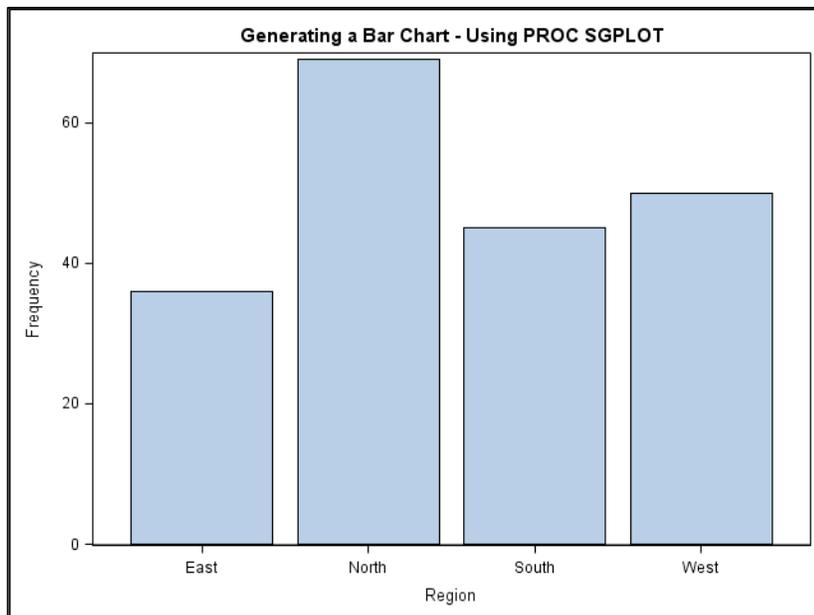
You can create a similar bar chart using PROC SGPLOT. A number of built-in styles make it very easy to customize your output. For example, a style called JOURNAL produces black and white output, suitable for publication in a journal. A style called STATISTICAL gives you output that is designed for statistical purposes.

Program 3.7 shows how to produce a chart similar to the one produced in Program 3.6:

**Program 3.7: Generating a Bar Chart Using PROC SGPLOT**

```
title "Generating a Bar Chart - Using PROC SGPLOT";  
proc sgplot data=store;  
  vbar Region;  
run;
```

The syntax is almost identical to PROC GCHART. You enter the keyword VBAR, followed by one or more variables for which you want to create a bar chart. Here is the output:



---

## Using ODS to Send Output to Alternate Destinations

To demonstrate the flexibility of the SGPLOT procedure, the next example shows you how to use a built-in style to send the same chart to a PDF file.

**Program 3.8: Using ODS to Create PDF Output**

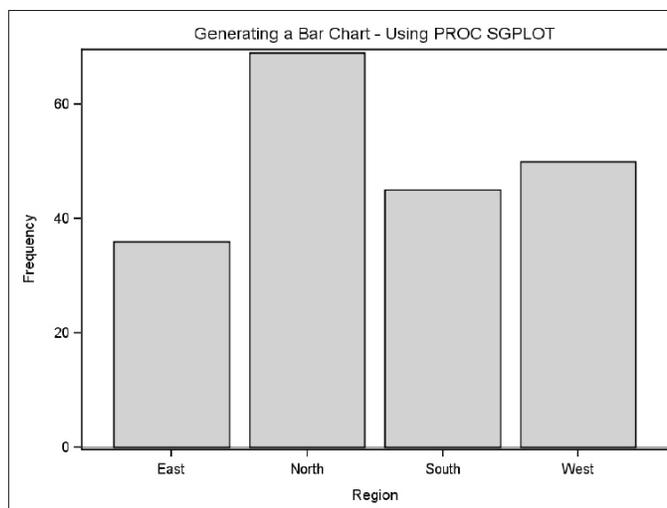
```
ods listing close;
ods pdf file='c:\books\statistics by example\bar.pdf'
    style=journal;
title "Generating a Bar Chart - Using PROC SGPLOT";
proc sgplot data=store;
    vbar Region;
run;
quit;
ods pdf close;
ods listing;
```

This program is identical to the previous one, except you place an ODS statement before the procedure that tells SAS two things: 1) you want to produce a PDF file and 2) you want to use the built-in style called JOURNAL. Following the procedure, you close the destination using another ODS statement.

You should close all your ODS destinations before you exit your SAS session. It is also a good idea to include the ODS LISTING CLOSE statement before the procedure so that you don't get two outputs—one sent to the PDF file and the other sent to the normal SAS output location. Remember that you need to reopen the listing file using the ODS LISTING statement following the procedure.

The PDF file that was created by Program 3.8 looks like this:

*Generating a Bar Chart - Using PROC SGPLOT*



This PDF file can be read by Adobe and used in any application that can work with PDF files.

## Creating a Cross-Tabulation Table

You can use PROC FREQ to create a cross-tabulation table. You start out with the keyword TABLES. Following this, you specify the two variables of interest, separated by an asterisk. For example, the `store` data set contains the variables `Region` and `Gender`. If you want to see the distribution of `Gender` across all values of `Region`, you proceed with Program 3.9:

### Program 3.9: Creating a Cross-Tabulation Table Using PROC FREQ

```
title "Demonstrating a Cross-Tabulation Table using PROC FREQ";
proc freq data=store;
  tables Gender * Region;
run;
```

This program requests a table of `Gender` by `Region`. In this example, `Gender` will form the rows of the table, and `Region` will form the columns.

The general form of a cross-tabulation request is:

```
tables row-variable * column-variable;
```

Here is the output from Program 3.9:

#### Demonstrating a Cross-Tabulation Table using PROC FREQ

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Gender by Region					
	Gender	Region				Total
		East	North	South	West	
Female	22	39	23	26	110	
	11.00	19.50	11.50	13.00	55.00	
	20.00	35.45	20.91	23.64		
	61.11	56.52	51.11	52.00		
Male	14	30	22	24	90	
	7.00	15.00	11.00	12.00	45.00	
	15.56	33.33	24.44	26.67		
	38.89	43.48	48.89	48.00		
Total	36	69	45	50	200	
	18.00	34.50	22.50	25.00	100.00	

Each box in the table contains four values; the meaning of these values is found in the key in the upper-left corner of the table. As you can see, the top number in each box is the number of observations. For example, there are 22 females in the Eastern region. The next number is a percentage. In this example, 11% of all observations are females in the Eastern region. The third number in each box is a row percentage—20% of the females were in the Eastern region. Finally, the fourth number in each box is a column percentage; 61.11% of the observations from the Eastern region are female.

Notice the order of the rows and columns in the output. By default, SAS orders the rows and columns in a table (or for a single variable) by the internal value of the variable—alphabetically for character variables and numerically for numeric variables. This is why the rows in the previous table were ordered Female→Male and the order of the columns was East→North→South→West.

---

## Changing the Order of Values in a Frequency Table

Whether you have a one-way or a two-way table, you might want to control the order that SAS uses for the rows, the columns, or both. In the previous example, maybe you want the regions to be ordered North→East→South→West. Or you might be computing an odds ratio in a 2x2 table and want the first column to be labeled Yes and the second column to be labeled No.

You can accomplish these goals in several ways. One is to create a new variable from the existing variable, where the internal values are in the desired order. Another, easier, method is to associate formats that are in the target order and associate that format with your variable. You can then use a PROC FREQ option called ORDER=FORMATTED to tell SAS to order the rows, columns, or both by their formatted values, rather than by their internal values.

The example that follows uses this method to force the order of the regions to be North, East, South, and West. First the program, then the explanation.

**Program 3.10: Changing the Order of Values in a PROC FREQ Table By Using Formats**

```
proc format;
  value $region 'North' = '1 North'
                'East'  = '2 East'
                'South' = '3 South'
                'West'  = '4 West';
run;

title "Change the Order in a PROC FREQ Output";
proc freq data=store order=formatted;
  tables Gender * Region;
  format Region region.;
run;
```

The four formatted values created by the \$region format are in the desired order alphabetically. (Note that the digits 1, 2, 3, and 4 are part of the format labels, and 1 comes before 2 alphabetically, etc.) Including a FORMAT statement in PROC FREQ associates the \$region format with the variable called Region. (Remember that the association is made by the FORMAT statement, not because the name of the format is similar to the name of the variable.) Finally, to tell SAS to order the table by the formatted values rather than by the internal values, you must include the PROC FREQ option ORDER=FORMATTED.

Here is the output from Program 3.10:

### Change the Order in a PROC FREQ Output

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Gender by Region					
	Gender	Region				Total
		1 North	2 East	3 South	4 West	
Female	39	22	23	26	110	
	19.50	11.00	11.50	13.00	55.00	
	35.45	20.00	20.91	23.64		
	56.52	61.11	51.11	52.00		
Male	30	14	22	24	90	
	15.00	7.00	11.00	12.00	45.00	
	33.33	15.56	24.44	26.67		
	43.48	38.89	48.89	48.00		
Total	69	36	45	50	200	
	34.50	18.00	22.50	25.00	100.00	

The regions are now ordered 1 North, 2 East, 3 South, and 4 West. Because female comes before male alphabetically, the order is Female, then Male.

---

## Conclusions

In this chapter, you learned how to display values of categorical variables, both in tabular and graphical form. Although you can use several methods to change the order of rows and columns in a table, using formats might be the simplest.

You also learned how to use the built-in styles to create attractive output with a minimum of effort. And you saw how to use ODS to send this output to a variety of destinations, such as HTML, PDF, and RTF.

The next chapter finishes up our discussion of descriptive statistics by showing you how to produce numerical and graphical displays for bivariate relationships.

**56** *SAS Statistics by Example*