

Learning to Create Bar Charts in SAS: A Practical Guide with Examples

Authored by
Mohammed loot

October 31, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Create Bar Charts in SAS: A Practical Guide with Examples*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=7424>

Introduction to Creating Bar Charts in SAS

Bar charts are among the most fundamental and effective tools in statistical visualization, providing a clear graphical representation of categorical data. They are particularly useful for displaying frequencies, counts, or sums across different groups. The [SAS](#) System offers robust procedures for generating these graphics, primarily through the use of the powerful **SGPLOT Procedure**. This procedure is designed for creating high-quality statistical graphs efficiently and with minimal coding complexity.

Mastering bar chart creation requires understanding just a few key commands within the [PROC SGPLOT](#) framework. Whether you need a simple frequency plot, a comparative stacked view, or a detailed clustered analysis, the underlying syntax remains consistent and intuitive. This guide explores three essential methods for generating different types of [Bar Chart](#) visualizations, detailing the required SAS syntax for each application. These methods allow analysts to move beyond simple data tables and communicate complex distributions effectively to any audience.

Below are the fundamental syntax patterns we will be exploring. These examples utilize the **VBAR** statement, which is central to vertical bar chart generation within **PROC SGPLOT**, along with crucial options for handling grouped and hierarchical data structures.

Method 1: Creating a Single Bar Chart (Frequency)

```
proc sgplot data = my_data;  
vbar variable1;  
run;
```

Method 2: Creating a Stacked Bar Chart (Part-to-Whole Comparison)

```
proc sgplot data = my_data;  
vbar variable1 / group = variable2;  
run;
```

Method 3: Creating a Clustered Bar Chart (Side-by-Side Comparison)

```
proc sgplot data = my_data;  
vbar variable1 / group = variable2 groupdisplay = cluster;  
run;
```

Preparation: Defining the Sample Data Set

To demonstrate these three visualization techniques effectively, we will utilize a sample data set named `my_data`. This data set simulates performance metrics for different sports teams and player positions. Understanding the structure of this data is essential, as the effectiveness of a bar chart relies heavily on the correct classification of categorical variables. In this example, `team` and `position` are the key categorical variables, while `points` represents a continuous measure, although for the basic bar charts, we will primarily focus on frequency counts of the categorical variables.

The following SAS Data Step code creates and populates the `my_data` set. It defines three variables: `team` and `position` (both character variables denoted by the dollar sign \$), and `points` (a numeric variable). This structure allows us to explore how frequencies vary both across teams and within specific positions.

```
/*create dataset*/  
data my_data;  
input team $ position $ points;  
datalines;  
A Guard 8  
A Guard 6  
A Guard 6  
A Forward 9  
A Forward 14  
A Forward 11  
B Guard 10  
B Guard 9  
B Guard 5  
B Forward 7  
C Guard 10  
C Forward 6  
C Forward 8  
;  
run;  
  
/*view dataset*/  
proc print data=my_data;
```

Executing the `PROC PRINT` command confirms the successful creation and structure of our working data set, which serves as the foundation for the subsequent visualizations. Analysts must always

ensure their data is clean and structured appropriately before proceeding to graphical analysis, as improper data input can lead to misleading visualizations.

Obs	team	position	points
1	A	Guard	8
2	A	Guard	6
3	A	Guard	6
4	A	Forward	9
5	A	Forward	14
6	A	Forward	11
7	B	Guard	10
8	B	Guard	9
9	B	Guard	5
10	B	Forward	7
11	C	Guard	10
12	C	Forward	6
13	C	Forward	8

Example 1: Generating a Simple Frequency Bar Chart

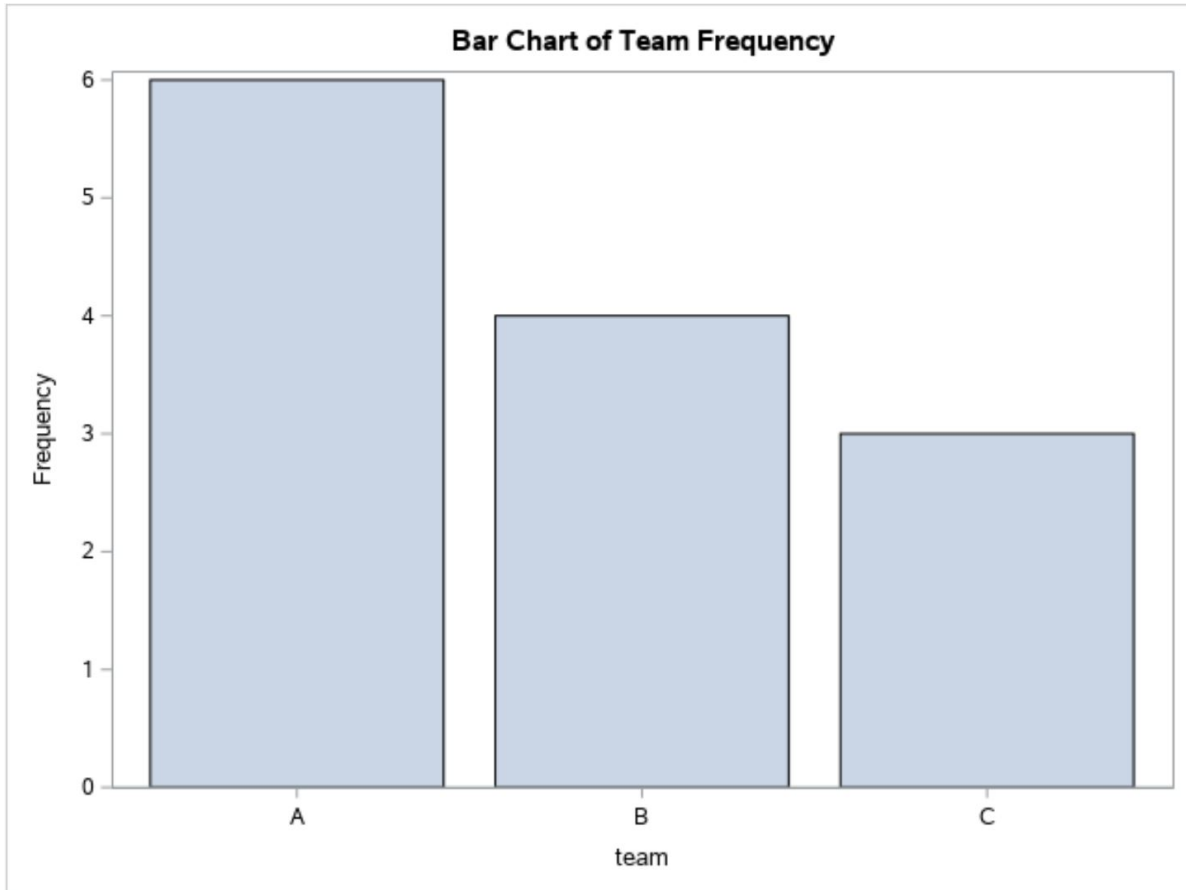
The most basic application of the bar chart in SAS is visualizing the frequency distribution of a single categorical variable. This process involves using the **VBAR** statement within [PROC SGPLOT](#), specifying only the variable whose counts you wish to display. When no response variable or specific statistical option is provided, the procedure defaults to calculating the number of observations (frequency) for each unique category found in the specified variable. This is a crucial first step in exploratory data analysis.

In this first example, we utilize the `team` variable to visualize how many observations (or rows) are associated with Team A, Team B, and Team C. This provides an immediate visual comparison of the sample size or representation of each team in our data. We also use the `TITLE` statement to provide essential context for the resulting graphic, adhering to best practices in data visualization documentation. The **VBAR** command automatically sets the team categories along the X-axis and the calculated frequency on the Y-axis.

```
/*create bar chart to visualize frequency of teams*/  
title "Bar Chart of Team Frequency";  
proc sgplot data = my_data;  
vbar team;
```

```
run;
```

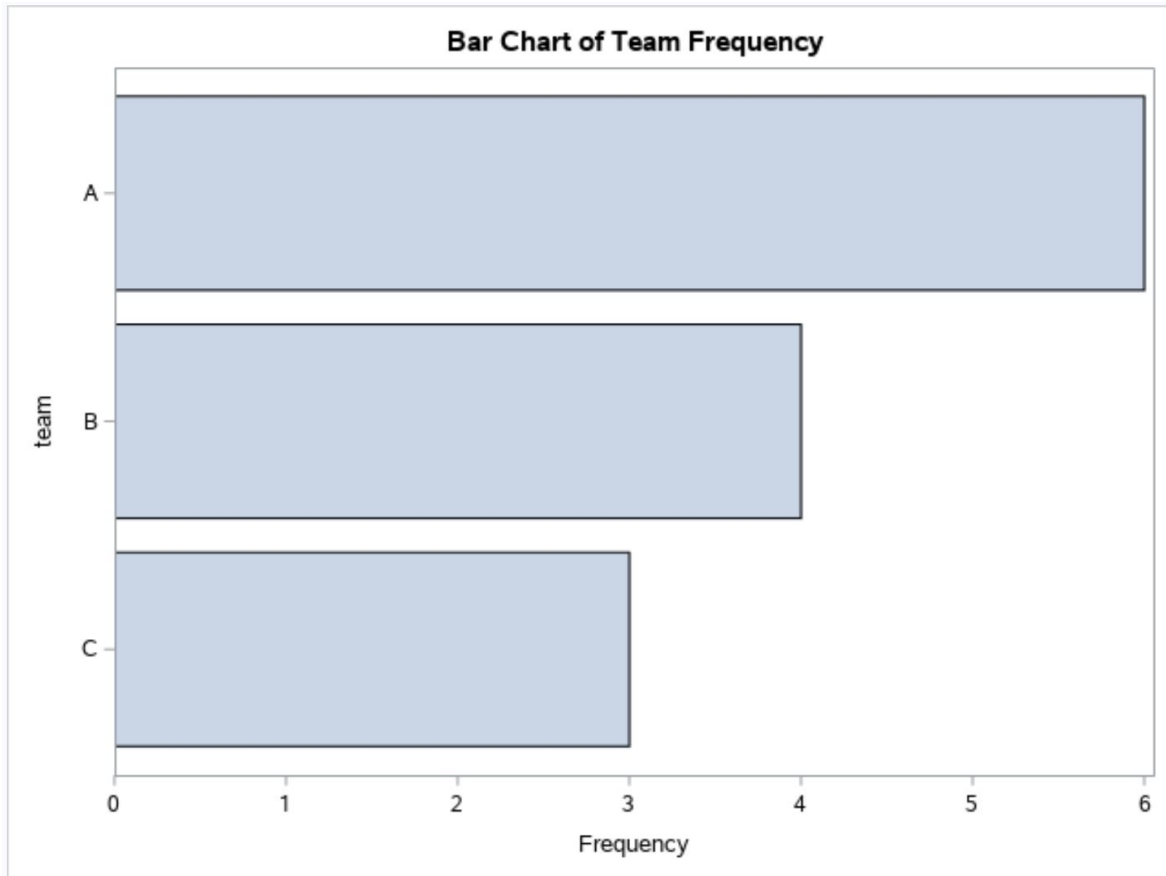
The resulting output is a clear, vertical bar chart summarizing the frequency counts:



For situations where the category labels are long or numerous, a horizontal orientation often improves readability. To achieve this, we simply replace the **VBAR** command with the **HBAR** command. The syntax remains structurally identical, but the orientation of the resulting bars is rotated, placing the categories on the Y-axis and the frequencies on the X-axis. This flexibility ensures that the chart is optimized for clear communication.

```
/*create horizontal bar chart to visualize frequency of teams*/  
title "Bar Chart of Team Frequency";  
proc sgplot data = my_data;  
hbar team;  
run;
```

The horizontal representation provides the same statistical information but changes the visual presentation:



Example 2: Visualizing Subgroups with Stacked Bar Charts

When analyzing data, it is frequently necessary to visualize the distribution of one variable while simultaneously observing how a second categorical variable influences that distribution. This is where the **Stacked Bar Chart** becomes invaluable, allowing for an efficient "part-to-whole" comparison. In SAS, this is achieved by adding the `GROUP` option to the **VBAR** statement.

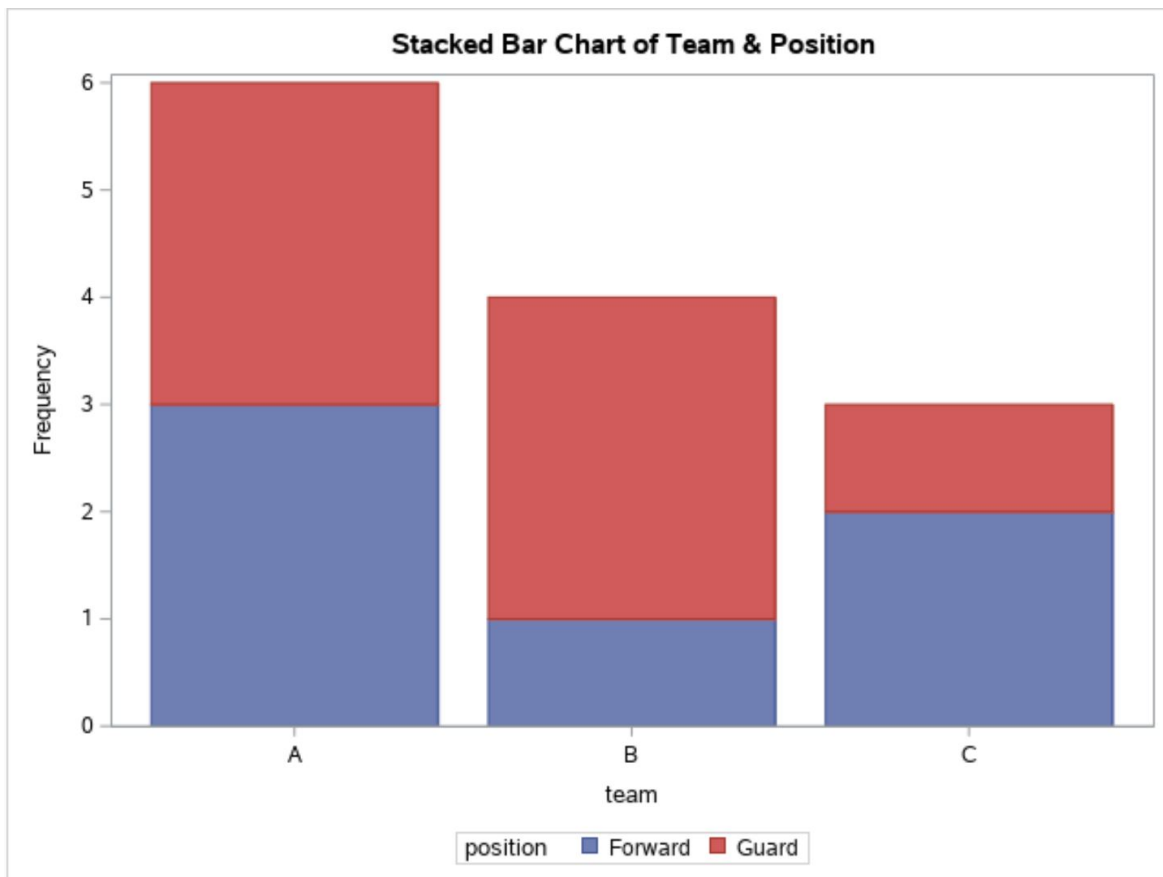
A stacked bar chart divides each main bar (defined by the primary variable, `team`) into segments corresponding to the categories of the secondary variable (`position`). The height of the total bar represents the overall frequency of the team, while the colored segments show the proportional breakdown of positions within that team. This structure is ideal for showing composition and relative contributions of subgroups.

In the following code, we instruct `VBAR` to plot the `team` variable and then use the `GROUP = position` option to segment the bars based on the player position. SAS automatically assigns different colors to the group variable, enhancing visual differentiation between the subgroups.

```
/*create stacked bar chart*/
title "Stacked Bar Chart of Team & Position";
```

```
proc sgplot data = my_data;  
vbar team / group = position;  
run;
```

The resulting visualization clearly illustrates not only the overall frequency of each team but also the exact frequency of Guards versus Forwards within each team's total count. This allows analysts to quickly assess the positional composition across the different teams.



This chart allows us to visualize the frequency of each team along with the frequency of positions within each team. Specifically, we can see that while Team A has the highest total frequency, the distribution between Guard and Forward positions is relatively balanced, whereas other teams might show a heavier reliance on one position over the other.

Example 3: Comparing Subgroups with Clustered Bar Charts

While stacked bar charts are excellent for part-to-whole comparisons, they can sometimes make it difficult to compare the absolute totals of subgroups across different main categories. For instance, comparing the total count of "Guards" in Team A versus Team B is visually challenging in a

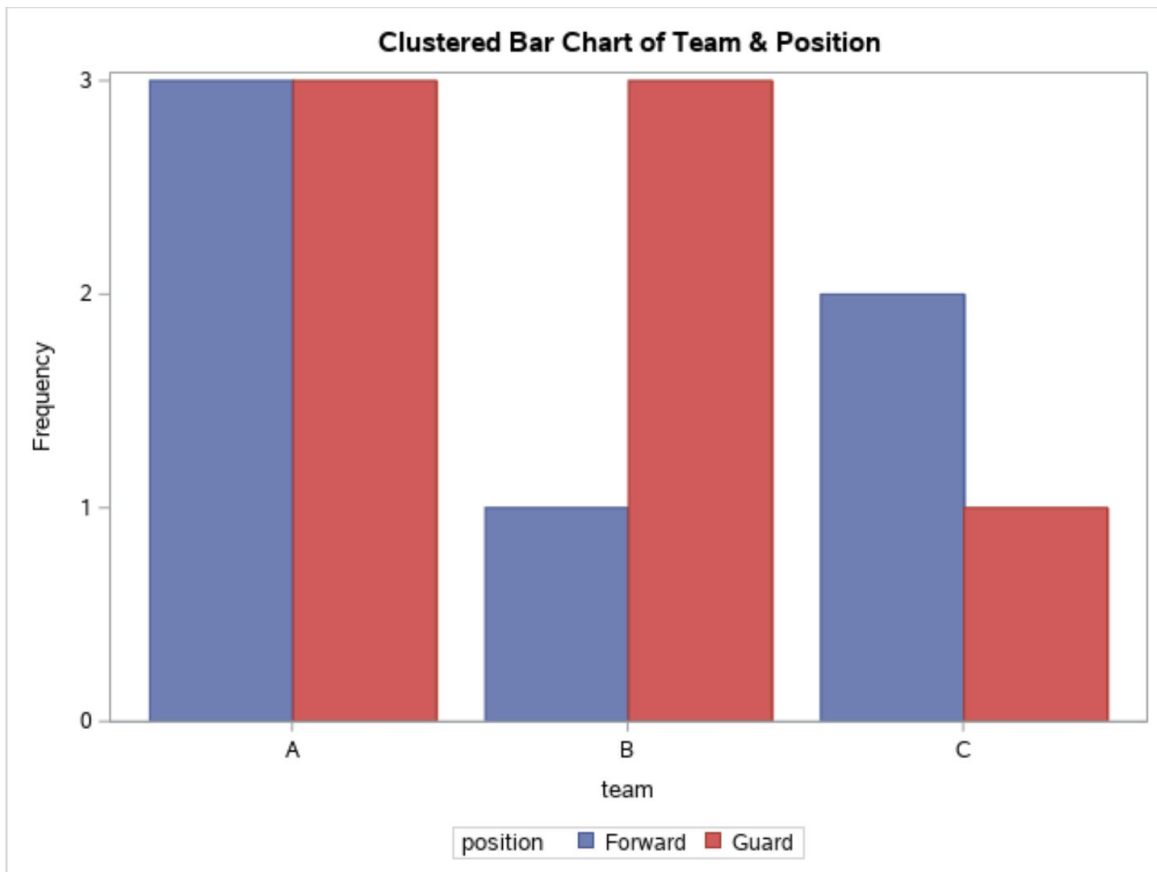
stacked chart because the segments do not share a common baseline. The **Clustered Bar Chart** solves this issue by placing the subgroup bars side-by-side, allowing for direct, baseline-aligned comparisons.

To create a clustered bar chart in SAS using [PROC SGPLOT](#), we again use the `GROUP` option, but we add the critical `GROUPDISPLAY = CLUSTER` sub-option. This tells SAS to render the bars for the group variable (`position`) adjacent to each other within each main category (`team`), rather than stacking them vertically.

The syntax below demonstrates the inclusion of this specific display option. This method is highly favored when the primary goal of the visualization is to contrast the performance or frequency of specific subgroups across multiple higher-level categories.

```
/*create clustered bar chart*/  
title "Clustered Bar Chart of Team & Position";  
proc sgplot data = my_data;  
vbar team / group = position groupdisplay = cluster;  
run;
```

This bar chart displays the same underlying data as the previous example, but the bars are "clustered" together instead of stacked on top of each other. The benefit is immediate: we can easily see that the frequency of Guards is very similar across all three teams (A, B, and C), while the frequency of Forwards varies more significantly.



Conclusion and Visualization Best Practices

The ability to generate vertical, horizontal, stacked, and clustered bar charts using the [PROC SGPLOT](#) procedure in SAS provides analysts with immense flexibility in visualizing categorical data. By mastering the core commands--**VBAR** (or [HBAR](#)) and the `GROUP` and `GROUPDISPLAY` options--users can tailor their graphics to best suit the analytical question at hand, whether it involves simple frequency counts, proportional breakdowns, or direct subgroup comparisons.

When choosing which type of bar chart to use, consider the objective:

If the goal is to show the overall distribution of a single variable, a **simple bar chart** (using **VBAR** or **HBAR**) is sufficient.

If the goal is to show how subgroups contribute to the total count (part-to-whole analysis), the **stacked bar chart** is the preferred method.

If the goal is to compare the magnitude of subgroups across different main categories, the **clustered bar chart** offers the clearest visual comparison.

Effective data visualization relies not only on accurate coding but also on thoughtful design

choices. Always ensure that axes are clearly labeled, titles are descriptive, and the chosen chart type maximizes the clarity of the underlying data story. These three examples provide a solid foundation for generating professional-grade bar charts within the SAS environment.

Additional Resources for SAS Graphics

The following tutorials explain how to create other advanced charts and visualizations in SAS, extending the functionality demonstrated here with [PROC SGPLOT](#) and other powerful SAS procedures:

[Creating Detailed Line Plots for Time Series Analysis.](#)

[Generating Histograms and Density Plots for Continuous Data Distribution.](#)

[Customizing Scatter Plots for Correlation Analysis.](#)

[Using the GCHART procedure for legacy graphics options.](#)