How to Easily Create Stunning Bar Charts in SAS

Authored by

stats writer

December 1, 2025

The creation of effective data visualizations is a cornerstone of statistical analysis, and the SAS (Statistical Analysis System) software suite provides robust tools for this purpose. Among the most fundamental visualizations is the bar chart, used primarily to display the frequencies, counts, or sums of categorical data. In SAS, generating these charts is highly streamlined, utilizing the powerful ODS Statistical Graphics procedures.

The core utility employed for generating these graphics is the PROC SGPLOT procedure. This routine offers exceptional flexibility, allowing users to define specific chart types through simple statements like `VBAR` for vertical bars or `HBAR` for horizontal bars. Beyond the basic structure, PROC SGPLOT supports extensive customization options, including controlling colors, managing legends, adjusting axis labels, and creating complex multivariate charts such as stacked and clustered displays. This comprehensive guide details three essential methods for generating bar charts in SAS, providing clear examples and the associated code necessary for replication.

Understanding the fundamental structure of the bar chart statements within the PROC SGPLOT procedure is crucial for efficient data visualization. Whether you are generating a simple frequency plot or a complex clustered visualization, the syntax remains intuitive. The examples below illustrate the three primary configurations used to plot categorical data using this procedure:

Method 1: Create One Bar Chart - This standard approach uses either the VBAR Statement (vertical) or the HBAR Statement (horizontal) to plot the count or sum of a single categorical variable.
Method 2: Create Stacked Bar Chart - This method introduces the `GROUP` option, which segments the bars based on a second categorical variable, allowing for visualization of sub-frequencies within the main categories.
Method 3: Create Clustered Bar Chart - Similar to the stacked chart, this method uses the `GROUP` option but adds the `GROUPDISPLAY = CLUSTER` option, placing the sub-segments side-by-side for easier comparison of individual group values.

These methods are built upon a clean and concise syntax structure within SAS, making them readily adaptable to various analytical needs. We will now review the specific code required for each of these three chart types, followed by detailed examples using a sample dataset.

Syntax for Basic, Stacked, and Clustered Bar Charts

While the full code includes data preparation and title specification, the core syntax for rendering the graphical output remains brief and powerful. The following blocks show the essential commands for each method, assuming your data resides in a dataset named `my_data`.

Method 1: Create One Bar Chart (Vertical)

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

Method 2: Create Stacked Bar Chart

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

Method 3: Create Clustered Bar Chart

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

Preparing the Sample Dataset for Visualization

To demonstrate these visualization techniques effectively, we will utilize a small, realistic dataset tracking player statistics across different teams and positions. This dataset, which we name `my_data`, contains three variables: `team` (categorical), `position` (categorical), and `points` (numeric). Creating the data step-by-step ensures that the subsequent code examples run correctly and produce predictable output.

The following SAS code block initializes the dataset using the `DATALINES` statement, which allows us to input data directly within the program environment. Following the data creation, we use the `PROC PRINT` procedure to display the resulting dataset, ensuring data integrity before proceeding to the graphical analysis. This step is a crucial best practice in any SAS programming workflow.

```
/*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;
run;

Once this code runs, the `PROC PRINT` output confirms that the dataset contains 13 observations, distributed across three teams (A, B, C) and two positions (Guard, Forward), setting the stage for our visualizations. The image below shows the typical tabular output generated by the `PROC PRINT` command, confirming the structure of `my_data`.

The resulting dataset, `my_data`, is now ready for graphical analysis. Notice the clear structure: the `team` and `position` variables are character types (indicated by the `$` in the `input` statement), and `points` is numeric. The following examples will utilize the counts of these categorical variables to generate the bar chart 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 Basic Vertical Bar Chart

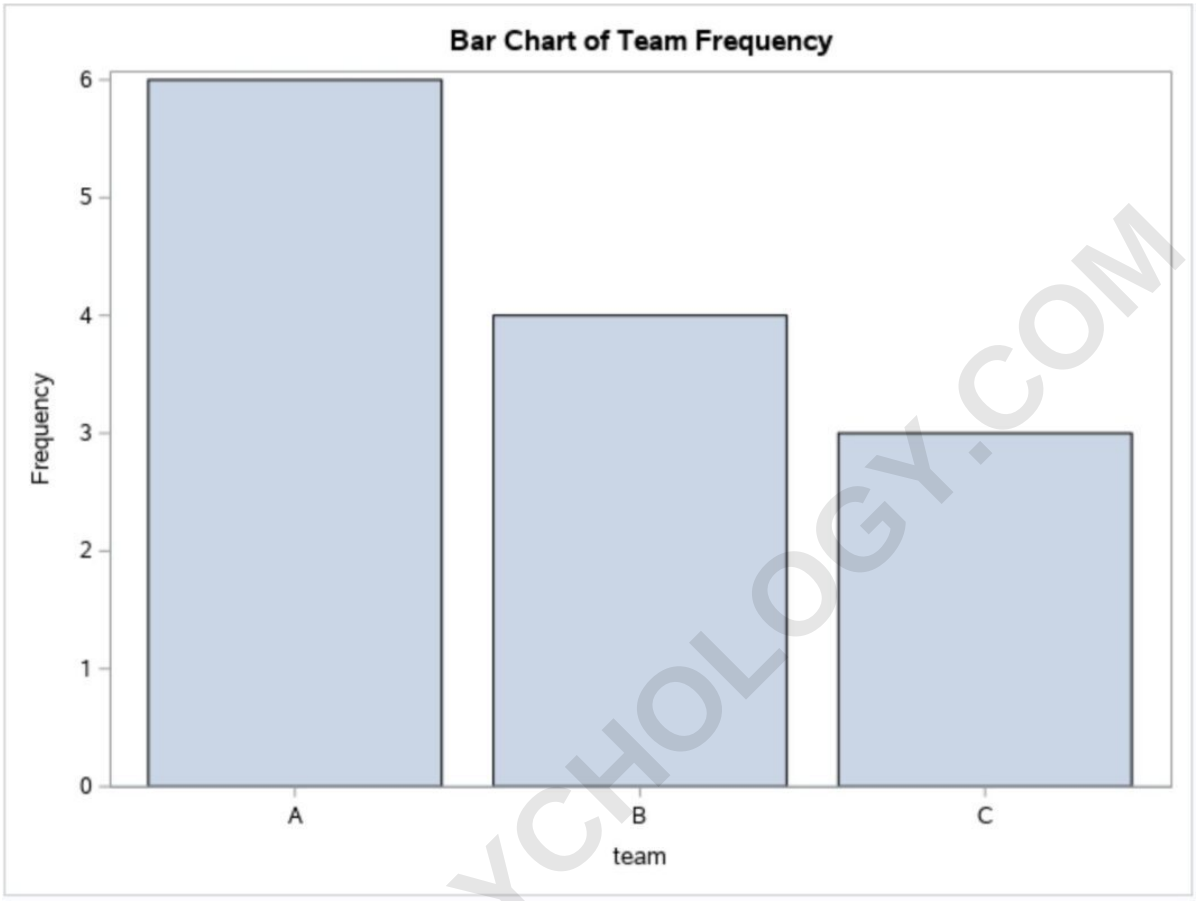
The most common application of a bar chart is to visualize the frequency of observations across a single categorical variable. In this first example, we use the VBAR Statement within PROC SGPLOT to count and plot the occurrences of each team (A, B, and C) in our dataset. By default, the VBAR Statement automatically calculates the frequency (count) of the variable specified on the vertical axis.

The code below defines a descriptive title for the output and then executes PROC SGPLOT, directing it to use `my_data`. The simple command `VBAR team;` instructs SAS to render a bar for each unique value of the `team` variable, with the height of the bar proportional to its frequency in the dataset. This visualization quickly reveals the relative distribution of observations across the teams.

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

Executing this code produces a vertical bar chart. As shown in the resulting image, Team A has 6 observations, Team B has 4, and Team C has 3. This type of visualization is essential for initial

data exploration and quality checks, ensuring a balanced distribution or highlighting unexpected concentrations within the data.



Creating Horizontal Bar Charts with the HBAR Statement

While vertical bar charts are standard, horizontal bar charts often offer superior readability, especially when dealing with variables that have long category names. The flexibility of PROC SGPLOT allows us to switch from a vertical orientation to a horizontal one simply by replacing the VBAR statement with the HBAR Statement.

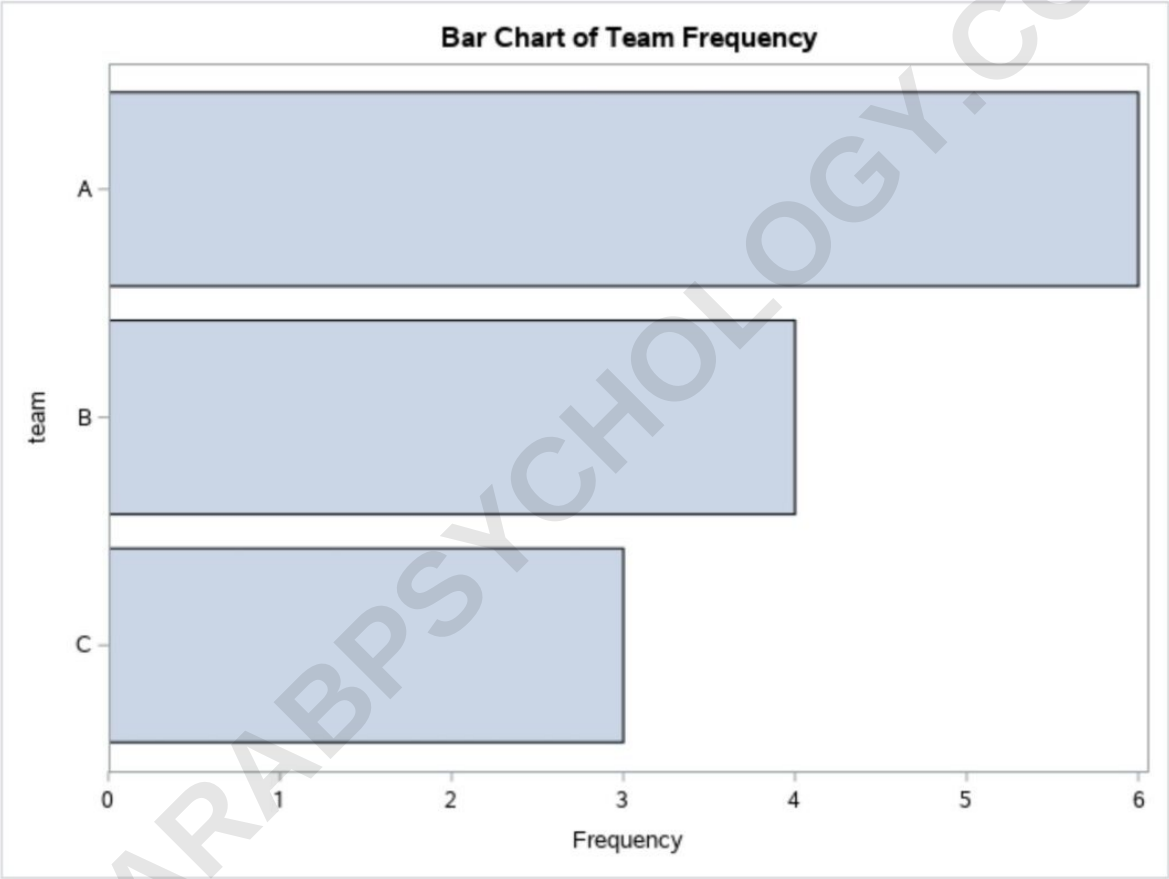
Functionally, the HBAR Statement is identical to VBAR; it calculates and plots the frequency of the categorical variable. The key difference is the axis orientation: categories are listed on the vertical axis, and the count is displayed along the horizontal axis. This subtle change can dramatically improve the user experience for reporting or presentation purposes.

To generate the horizontal version of the team frequency chart, the following minor modification is made to the previous code:

/*create horizontal bar chart to visualize frequency of teams*/

```
title "Bar Chart of Team Frequency (Horizontal)";
proc sgplot data = my_data;
hbar team;
run;
```

The resulting horizontal chart presents the exact same frequency data, but now the bars extend horizontally. This visualization is particularly useful when comparing many categories, as the text labels for each category remain fully legible without requiring rotation or truncation, enhancing overall visual clarity.



Understanding Grouped Bar Charts in SAS

While basic bar charts are excellent for univariate frequency analysis, complex datasets often require visualizing the interaction between two categorical variables. This is achieved in SAS using grouped bar charts, which are categorized into two primary types: stacked and clustered. Both types utilize the `GROUP` option within the `VBAR` or `HBAR` statement, specifying a second categorical variable to define sub-groups.

The choice between a stacked or clustered format depends entirely on the analytical goal. A **stacked bar chart** is ideal for showing the "part-to-whole" relationship, illustrating how the sub-groups contribute to the total frequency of the main category. Conversely, a **clustered bar chart** is superior for direct comparison of the absolute values of the sub-groups across different main categories, as the bars for each sub-group share the same baseline.

In the following examples, we will use the `team` variable as our primary category and the `position` variable (Guard or Forward) as our grouping variable. This allows us to visualize not just the total observations per team, but also the breakdown of positions within each team, providing a richer, two-dimensional view of the dataset's composition.

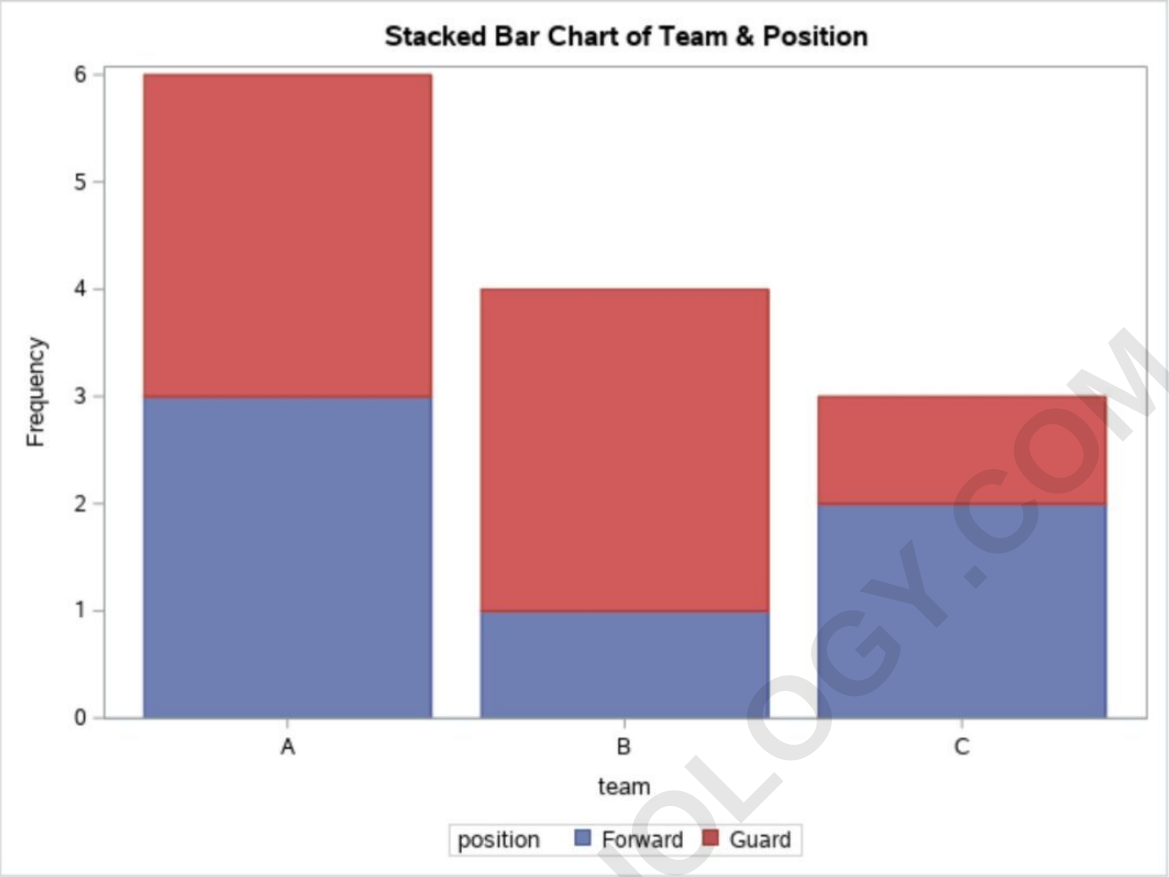
Example 2: Implementing a Stacked Bar Chart

A stacked bar chart provides an immediate visual summary of compound frequencies. To create this type of visualization in SAS, we simply add the `GROUP = variable2` option to the VBAR Statement. In our case, `variable1` is `team` and `variable2` is `position`.

When the code executes, PROC SGPLOT first tallies the total frequency for each team. It then segments the resulting bar according to the frequencies of the defined positions (Guard or Forward), assigning a distinct color to each position. This immediately shows the contribution of Guards versus Forwards to the total count for Team A, Team B, and Team C, thereby visualizing the internal structure of the teams.

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

The resulting stacked chart clearly depicts the frequency distribution. For instance, Team A has slightly more Forwards than Guards, while Team B has a balance, and Team C has more Guards. The stacked format is particularly intuitive when the goal is to assess the proportional composition of the main categories, as the total height of the bar still represents the overall frequency.



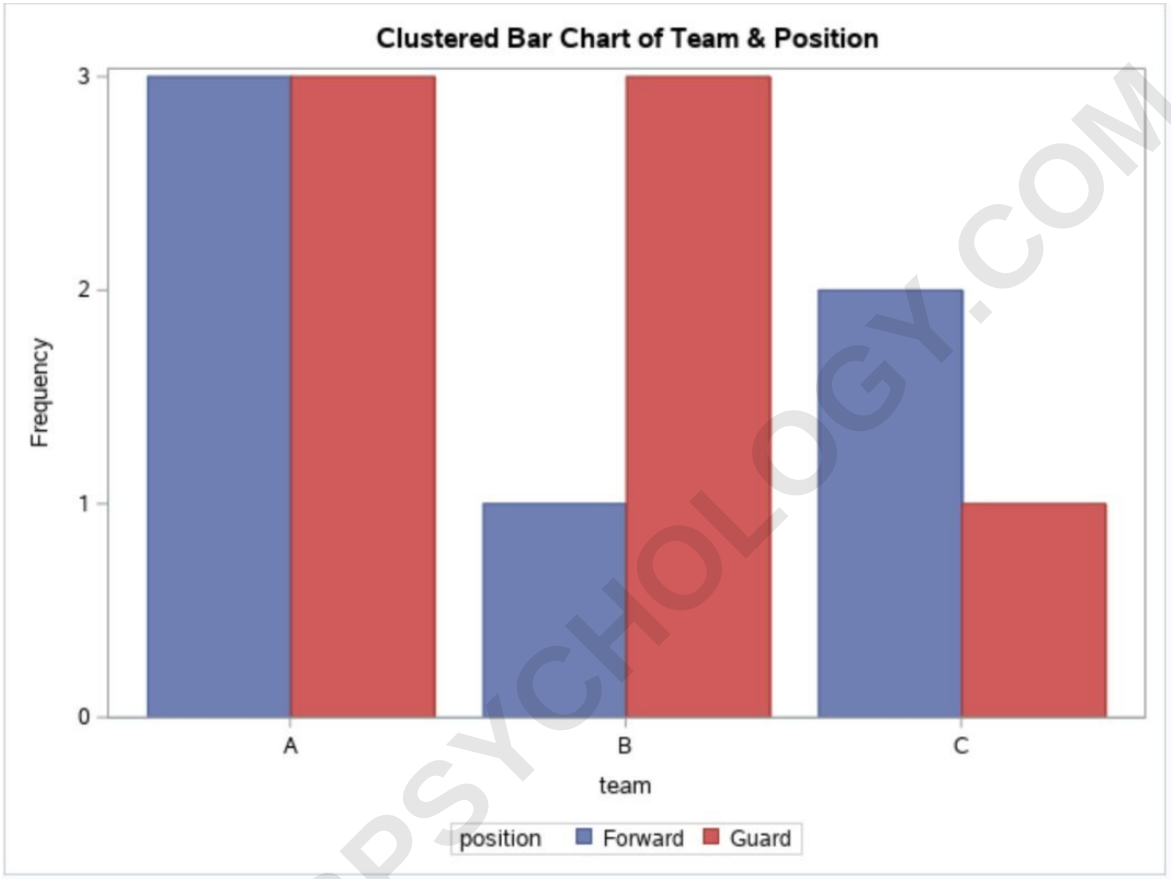
Example 3: Constructing a Clustered Bar Chart

If the primary analytical need is to compare the counts of Guards directly against the counts of Forwards across all teams, a clustered bar chart is the preferred visualization method. The clustered approach separates the grouped variables and places them adjacently, allowing for direct comparison against a common baseline (the zero axis).

To achieve this clustered display, we build upon the stacked chart code by adding a single, crucial option to the `VBAR` statement: `GROUPDISPLAY = CLUSTER`. This setting overrides the default stacking behavior and instructs PROC SGPLOT to cluster the bars based on the `position` variable.

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

The output provides the same frequency information as the stacked chart, but the visual interpretation is different. For each team, we see two separate bars (one for Guard, one for Forward). This configuration is significantly better for comparing the absolute counts--for instance, quickly noting that Team A has the highest count of Forwards compared to Teams B and C--without needing to mentally adjust for the stacked segments.



Advanced Customization and Bar Chart Best Practices

While the examples above focus on frequency counts, PROC SGPLOT is capable of much more sophisticated aggregation. Instead of plotting frequency (the default), you can plot the sum, mean, or median of a numeric variable (like `points`) by using the `RESPONSE` option in the `VBAR` or HBAR Statement, combined with the `STAT` option.

For example, to visualize the total points scored by each team, you would modify the basic code: `vbar team / response=points stat=sum;`. This level of customization ensures that the generated bar chart accurately reflects the specific metric required for the analysis. Furthermore, visual aesthetics are paramount in data reporting. SAS allows control over colors (using the `FILLATTRS` option), data labels (using the `DATALABEL` option), and axis formatting (using `YAXIS` or `XAXIS` statements).

When designing bar charts, always ensure that the axes are clearly labeled and that the visualization style--stacked versus clustered--is appropriate for the narrative you wish to convey. Clustered charts are generally preferred when the goal is comparing categories, whereas stacked charts are best when the goal is understanding the internal proportional breakdown of a whole. Mastering these subtle differences allows for the production of highly informative and authoritative statistical graphics in SAS.

Conclusion and Summary of Bar Chart Types

Generating effective bar charts in SAS is a straightforward process facilitated by the robust PROC SGPLOT procedure. By correctly employing the VBAR Statement or HBAR Statement, along with judicious use of the `GROUP` and `GROUPDISPLAY` options, analysts can produce visualizations that range from simple frequency counts to complex multivariate comparisons.

The three methods demonstrated--basic frequency, stacked, and clustered--provide the foundation for nearly all bar chart needs. Analysts should select the chart type that best highlights the critical patterns in the data: the basic chart for single-variable distribution, the stacked chart for part-to-whole relationships, and the clustered chart for direct group-to-group comparisons. Adhering to these principles ensures clean, accurate, and insightful graphical results.

The following resources offer additional guidance on creating and customizing visualizations within the SAS environment: