

SAS Tutorial: Creating Effective Pie Charts for Data Visualization

Authored by
Mohammed loot

November 14, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *SAS Tutorial: Creating Effective Pie Charts for Data Visualization*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=1279>

In the foundational realm of [data visualization](#), [pie charts](#) remain an essential and intuitive tool for illustrating proportions and compositional breakdowns within any categorical [dataset](#). This comprehensive, expert-level guide is specifically designed to demonstrate the effective methods for generating various styles of [pie charts](#) using [SAS](#), the powerful statistical software favored globally by data analysts and researchers. We will specifically focus on leveraging the versatile [statement](#) known as **PIE**, which is nested within the robust [PROC GCHART](#) procedure, to produce compelling and highly insightful visual representations of your compiled input data.

This tutorial systematically walks you through four distinct and practical examples of [pie chart](#) creation within the [SAS](#) environment. Each example is meticulously structured to highlight different functionalities and advanced visualization techniques available through [PROC GCHART](#). The techniques covered range from calculating simple category [frequency](#) counts to implementing sophisticated customization features, such as visually separating key slices (exploding) and precisely formatting data labels for maximum clarity. All subsequent examples will utilize a consistent, easily replicable sample [dataset](#) containing relevant observational information about various basketball players, including their respective team affiliations and the individual points they scored during a game period.

Establishing the Sample Dataset in SAS

Before commencing the visualization process, it is a critical prerequisite to properly establish the sample [dataset](#) that will be analyzed and manipulated. The foundation of any successful statistical or graphical analysis in [SAS](#) is ensuring that the data is correctly structured and validated. Our specific sample data, which we have named **my_data**, contains two crucial [variables](#): **team**, defined as a character variable representing the basketball team affiliation, and **points**, a numeric variable quantifying the individual points scored by a player. This dual structure enables us to perform both categorical analysis (by team) and quantitative analysis (by points scored).

The following [SAS](#) code snippet demonstrates the straightforward creation of this necessary sample [dataset](#). We employ the standard **DATA** and **DATALINES** statements, which are the fundamental methods for inputting observational data directly into the SAS system. Following the successful creation of the data structure, we utilize the ubiquitous **PROC PRINT** procedure to display the entire contents of the newly created dataset. This step serves as an essential data integrity check, verifying the accuracy of the data before we proceed with the graphical visualization procedures using [PROC GCHART](#).

```
/*create dataset*/  
data my_data;  
input team $ points;  
datalines;
```

```
Mavs 14
Mavs 22
Mavs 19
Mavs 31
Heat 14
Heat 25
Warriors 31
Warriors 35
Warriors 36
Jazz 29
;
run;

/*view dataset*/
proc print data=my_data;
```

Obs	team	points
1	Mavs	14
2	Mavs	22
3	Mavs	19
4	Mavs	31
5	Heat	14
6	Heat	25
7	Warriors	31
8	Warriors	35
9	Warriors	36
10	Jazz	29

As clearly illustrated by the output generated from the execution of **PROC PRINT**, our **my_data dataset** is now correctly formatted and prepared for the visualization stage. The printed table explicitly verifies the associations between the specific basketball teams and their corresponding individual player scores, providing a solid, verifiable foundation for all our subsequent [pie chart](#) analyses conducted using [PROC GCHART](#). This ensures that the visualizations we create accurately reflect the underlying data structure.

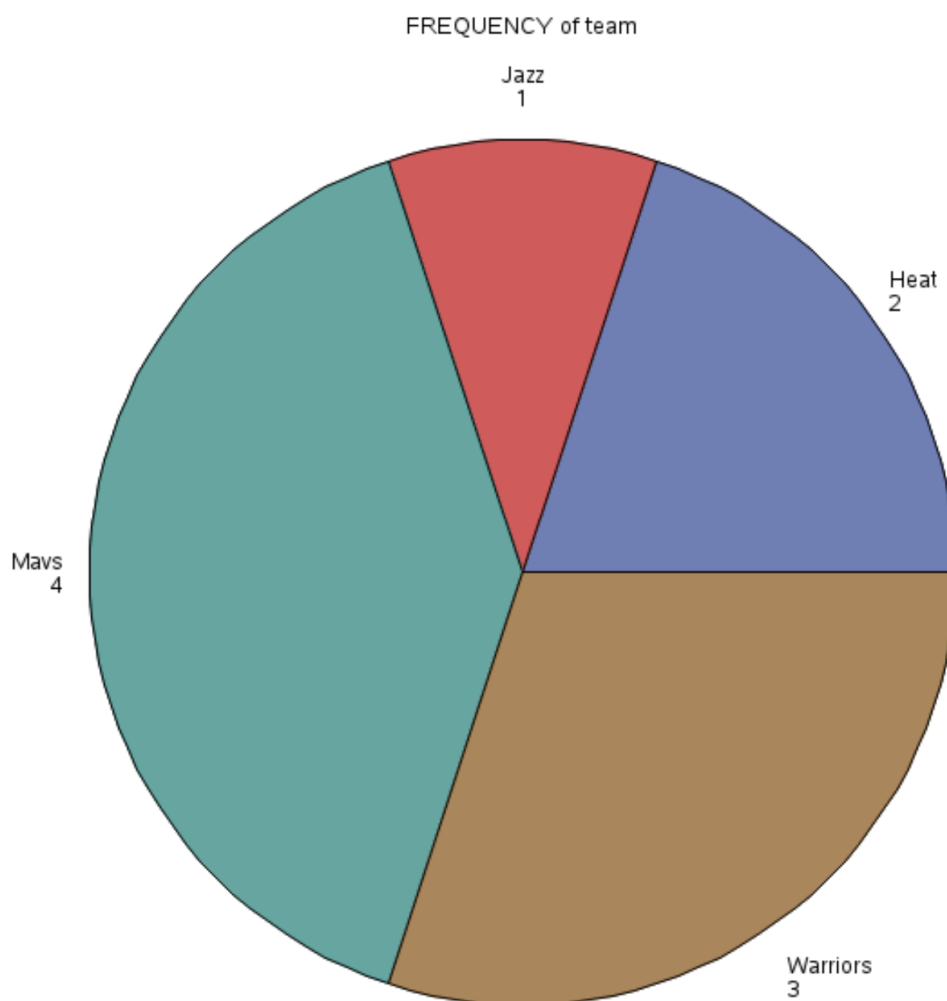
Example 1: Visualizing Categorical Frequencies

Our initial visualization example focuses on the most fundamental application of the [pie chart](#): visualizing the statistical [frequency](#), or the simple count, of each unique categorical value within a designated [variable](#). This direct approach offers immediate insight, proving highly useful for rapidly understanding the overall distribution, balance, and representation of categories present across your entire [dataset](#). For the purposes of this demonstration, we will analyze the **team** [variable](#) to count exactly how many times each team name appears in the data, thereby determining its relative proportional size within the resulting pie chart.

The following [SAS](#) code block clearly demonstrates the simple, efficient structure required to generate a basic [pie chart](#) based solely on observational counts. The resulting chart will effectively display the [frequency](#) of each unique team listed in the **team** column of our **my_data** [dataset](#). The core of this operation relies on the straightforward execution of the [PROC GCHART](#) procedure, immediately followed by the powerful **PIE** [statement](#), which specifies the grouping variable (in this case, **team**).

```
proc gchart data=my_data;  
pie team;  
run;  
quit;
```

Upon successful execution, this minimal code generates a [pie chart](#) where each distinct slice represents a unique basketball team. Crucially, the size (or angle) of each resulting slice is directly and mathematically proportional to the relative [frequency](#) of that specific team within the entire [dataset](#). This visualization method provides an immediate, highly effective summary of the team distribution, visually highlighting which categories hold the largest representation of observations in the data.



As clearly depicted in the resulting chart, the slices accurately illustrate the count of observations associated with each unique team in the **team** [variable](#). For example, if the 'Warriors' team appears three times and the 'Heat' team appears twice in the raw data, the 'Warriors' slice will be rendered proportionally larger to reflect its higher count. This visualization technique effectively highlights which teams have more data entries recorded in our sample, offering a quick and clear categorical comparison that is easily digestible.

Example 2: Summarizing Quantitative Data

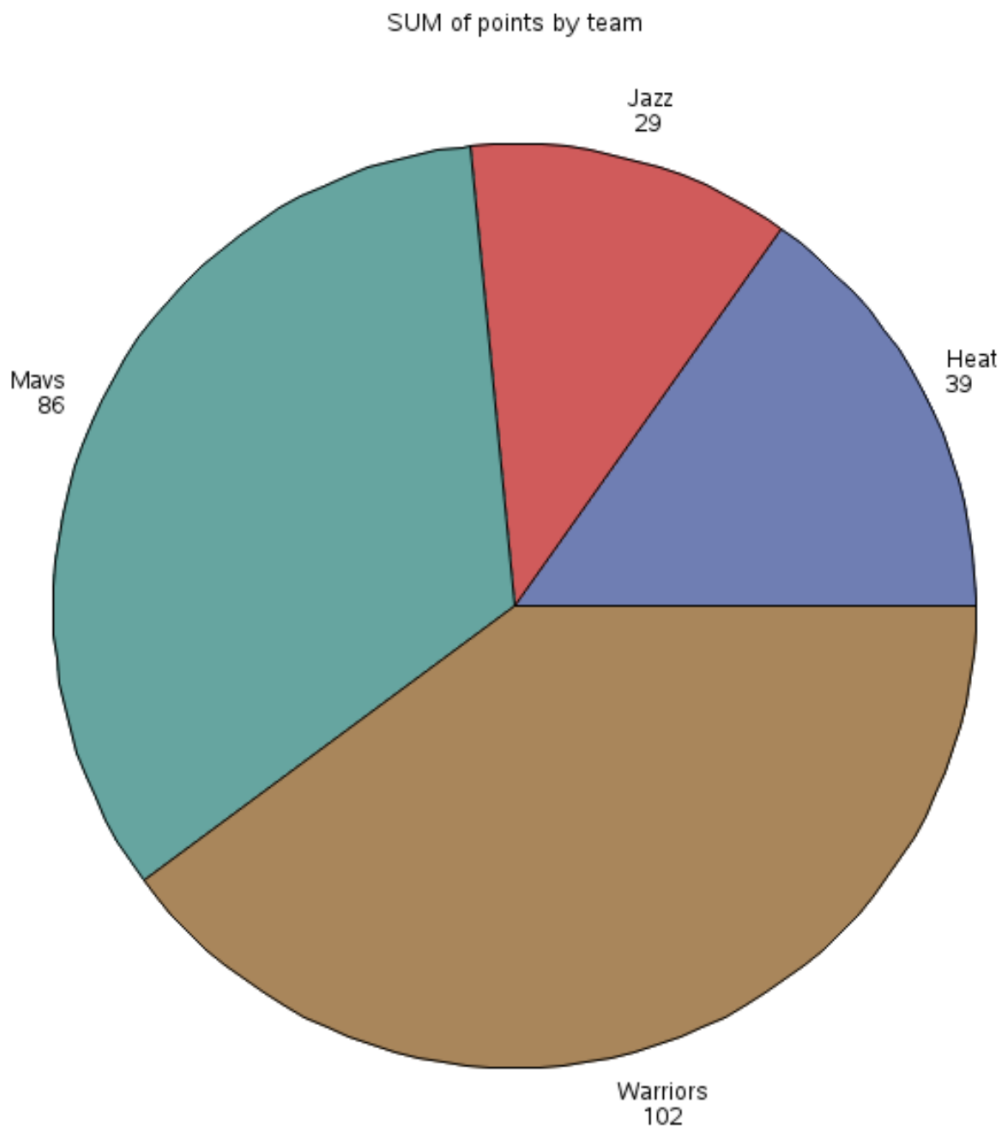
Moving beyond the simple counting of frequencies, [pie charts](#) offer the necessary flexibility to effectively display the [sum](#) of a designated numerical [variable](#) for each underlying category. This capability is exceptionally useful when the primary analytical objective is to visualize precisely how a total aggregate quantity--such as overall budget, gross sales figures, or, in the context of our example, total points scored--is distributed across various distinct groups. In this specific and more advanced example, our goal is to calculate the total points scored by players belonging to each

team and visualize that proportion.

To successfully achieve this quantitative summarization, we introduce the crucial **SUMVAR statement** within the context of **PROC GCHART**. The **SUMVAR statement** allows the user to explicitly specify a numeric [variable](#) whose individual values must be aggregated (summed up) for every instance of the chart's primary grouping [variable](#). For this critical analysis, we will designate the **points** column as our [sum](#) variable, while the **team** column remains our definitive primary grouping category.

The following [SAS](#) code illustrates the precise syntax required to create a sophisticated [pie chart](#) that displays the aggregated [sum](#) of values found in the **points** column, intelligently broken down by each unique value in the **team** column. This graphical output will provide a clear, visual breakdown of the total scoring contribution made by each individual team within our sample [dataset](#), effectively emphasizing their overall collective performance rather than just their player count.

```
proc gchart data=my_data;  
pie team / sumvar=points;  
run;  
quit;
```



In the resulting [pie chart](#), each generated slice now accurately represents the aggregated total of points scored by the corresponding team. In significant contrast to the earlier frequency chart, the proportional size of these slices is now dictated by the [sum](#) of all individual **points** entries associated with that particular team, rather than simply the count of observations. This powerful feature allows stakeholders and viewers to instantly compare the overall scoring contributions of each team at a single, easy-to-interpret glance, facilitating rapid performance evaluation.

For instance, if the 'Warriors' players collectively scored a substantially higher total number of points than the 'Mavs' players, the 'Warriors' slice will be rendered proportionally much larger, irrespective of the number of players involved in the scoring. This specific type of visualization is invaluable for understanding the effective and actual distribution of a quantitative [variable](#) across different qualitative, defined categories, providing a true measure of contribution.

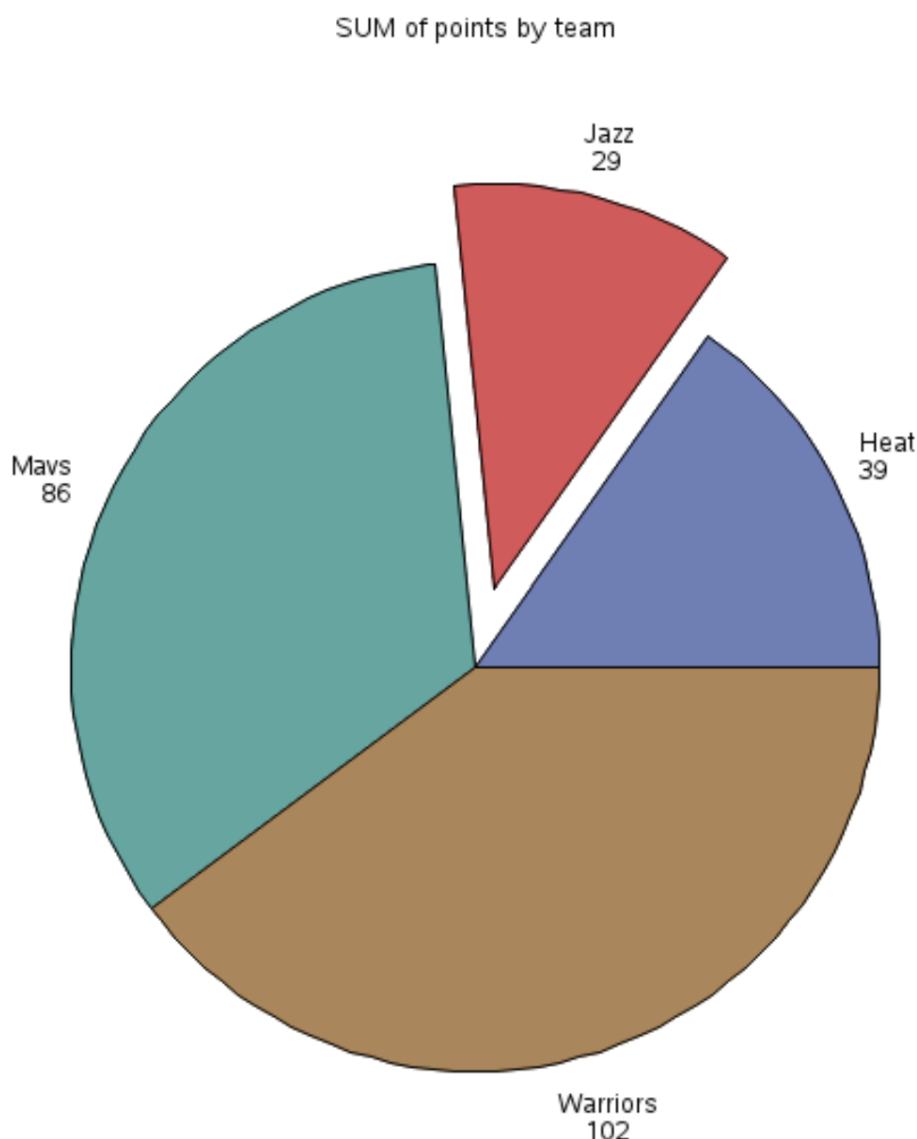
Example 3: Highlighting Key Segments with Exploding Slices

A frequent and important requirement in professional [data visualization](#) is the explicit need to draw immediate, specific attention to one or more particular categories within the generated chart. This necessary visual emphasis can be achieved highly effectively by employing the technique of "exploding" one or more slices, visually separating them slightly from the main, central body of the [pie chart](#). This technique serves as a potent visual cue, ideal for emphasizing a key data point, highlighting a statistically significant outlier, or underscoring a category that warrants special consideration, discussion, or managerial review.

To generate a [pie chart](#) featuring one or more exploded slices, SAS's [PROC GCHART](#) procedure thoughtfully provides the versatile **EXPLODE** [statement](#) option. This [statement](#) grants the user the ability to specify precisely which category's slice should be moved slightly away from the center of the chart, thereby immediately drawing the viewer's eye. In this practical demonstration, we will continue to display the [sum](#) of **points** for each team, but we will visually highlight the contribution specifically made by the 'Jazz' team.

The following [SAS](#) code demonstrates the simple yet effective usage of the **EXPLODE** [statement](#). By strategically adding the parameter `EXPLODE='Jazz'` directly to the **PIE** statement, we instruct SAS to create a [pie chart](#) where the slice corresponding to the 'Jazz' team is explicitly "exploded" away from all other segments. This intentional visual separation ensures that the 'Jazz' contribution immediately stands out to the viewer, serving as a powerful and undeniable visual cue that guides the interpretation.

```
proc gchart data=my_data;  
pie team / sumvar=points explode='Jazz';  
run;  
quit;
```

Observing the resulting chart, the visual effect is immediate and pronounced: the slice corresponding to the 'Jazz' team has been distinctly separated and slightly pulled away from the cluster of the rest of the [pie chart](#) segments. This technique proves particularly useful when your primary intention is to highlight a specific category or a limited set of categories that hold significant analytic importance or demand additional scrutiny within your overall presentation. It is an indispensable tool for guiding the audience's focus and effectively emphasizing key findings or anomalies.

Example 4: Enhancing Readability with Custom Labels

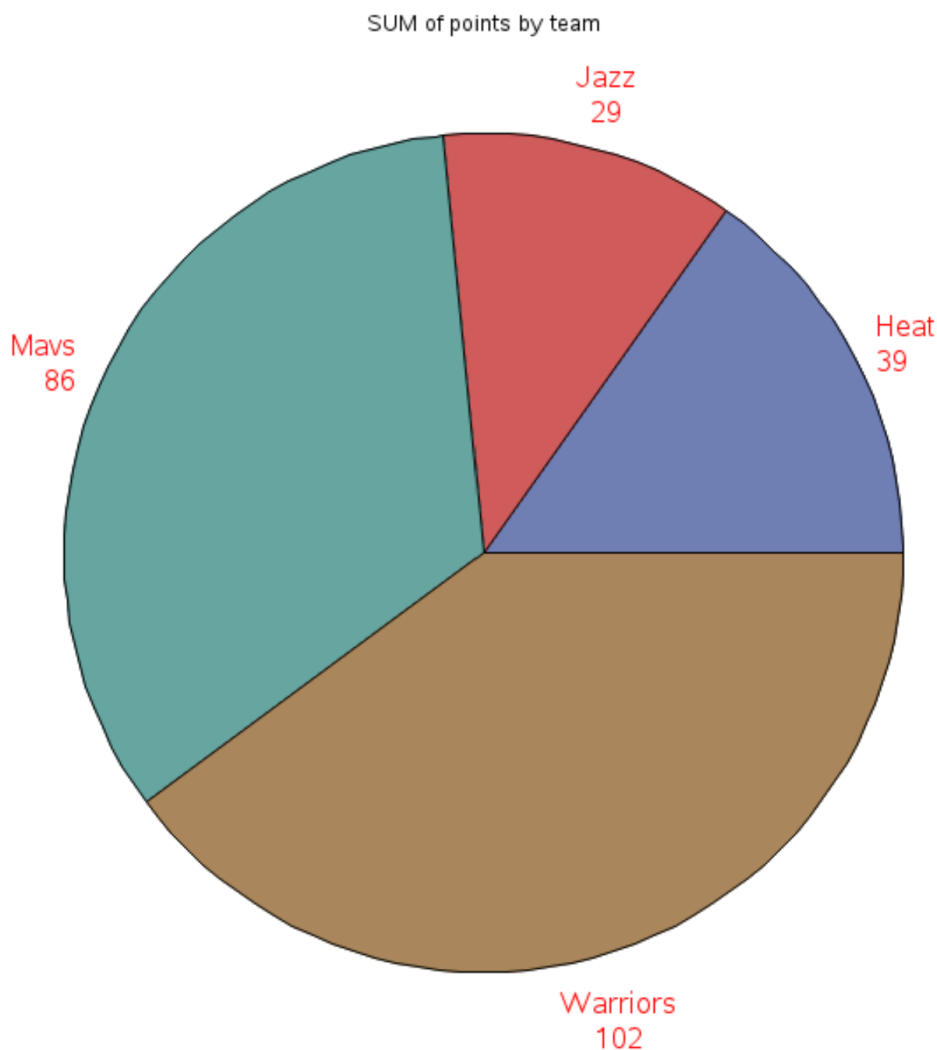
Customizing the visual appearance and placement of labels on your [pie chart](#) is a profoundly crucial step that can significantly enhance both its fundamental readability and its overall aesthetic appeal. By judiciously adjusting attributes such as the font height, color, style, and positioning, you

can ensure that your labels are consistently clear, visually prominent, and perfectly aligned with your broader [data visualization](#) objectives. This final, advanced example demonstrates the straightforward process of applying such detailed custom formatting within [SAS pie charts](#) using specialized parameters.

The **PLABEL** [statement](#), utilized within **PROC GCHART**, provides extensive, granular control over the textual labels displayed on each slice of the chart. Within this powerful statement, you are able to specify various [arguments](#) (or parameters) to modify characteristics like the font height, the color scheme, the text style, and many other visual attributes. For the specific goals of this example, we will focus on dramatically increasing the default font size and changing the font color of the labels to a bold red, ensuring they are both larger and far more striking against the background graphic.

The following [SAS](#) code illustrates the precise syntax required for using the **PLABEL** [statement](#) to create a [pie chart](#) with significantly enhanced labels. We specifically instruct SAS to set the font height (using the **h** [argument](#)) to 1.5 units and simultaneously set the font color to red, utilizing the **color** [argument](#). This strategic combination dramatically improves label visibility and overall visual impact, making the chart easier to read from a distance.

```
proc gchart data=my_data;  
pie team / sumvar=points plabel=(h=1.5 color=red);;  
run;  
quit;
```



Upon carefully reviewing the generated graphical output, it is immediately apparent that the labels positioned directly on the [pie chart](#) slices now appear with a noticeably larger font size and a distinct, bold red color, fully satisfying our detailed customization request. This immediate improvement in label visibility significantly enhances the overall clarity and professional impact of the chart, making the interpretation of the data faster, more accurate, and much more accessible for the intended audience.

Specifically, the **h** [argument](#) within the **PLABEL** [statement](#) precisely controls the font height scaling factor, while the **color** argument dictates the exact font color utilized for the slice text labels. These highly flexible [arguments](#) offer users extensive options for tailoring their [pie chart](#) labels to meticulously meet virtually any presentation requirement, ultimately ensuring that the critical data message is conveyed effectively, aesthetically, and with maximum impact.

Summary and Further Learning

This expert guide has successfully provided a comprehensive, step-by-step overview of the methodology for creating four distinct and functional types of [pie charts](#) in [SAS](#), all utilizing the essential [PROC GCHART](#) procedure. By mastering the practical application of the core **PIE**, **SUMVAR**, **EXPLODE**, and **PLABEL** statements, you are now equipped to effectively visualize both categorical data distributions and summarized quantitative distributions, enabling you to communicate complex data insights with enhanced clarity and superior visual impact.

For those interested in exploring more diverse [data visualization](#) options and extending their analytical capabilities further within the powerful [SAS](#) environment, the following related resources offer highly valuable, official guidance on generating other powerful types of charts and graphics:

[How to Create Bar Charts in SAS](#)

[Generating Scatter Plots and Regression Lines in SAS](#)

[Mastering PROC SGPLOT for Advanced Graphics](#)