

# Learning to Create Scatter Plots in SAS: A Step-by-Step Guide

Authored by  
**Mohammed loot**

October 31, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Create Scatter Plots in SAS: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=7430>

[Scatter plots](#) are fundamental tools in data visualization, offering an immediate graphical representation of the relationship between two continuous variables. In the [SAS](#) statistical software environment, generating these visualizations is straightforward and highly customizable, primarily utilizing the powerful ODS Graphics procedures. The most efficient and modern method for creating high-quality statistical graphics, including detailed scatter plots, is through the use of the [PROC SGPLOT](#) procedure. This guide outlines the essential syntax and provides detailed examples for generating both simple bivariate scatter plots and complex plots segmented by group variables, ensuring that analysts can effectively communicate correlation and distribution patterns within their datasets.

The core functionality required for visualization in SAS is encapsulated within the **ODS Graphics system**. This system, introduced in later versions of SAS, offers a significant improvement in the quality and ease of generating output compared to older procedures. When approaching scatter plot creation, analysts typically need to consider two main scenarios: plotting the relationship between two variables across the entire dataset, or plotting this relationship while distinguishing observations based on a categorical grouping variable. Mastering the basic syntax of `PROC SGPLOT` is the first step toward advanced statistical graphing capabilities.

## Understanding the Core SAS/GRAPH Procedure: PROC SGPLOT

The `PROC SGPLOT` procedure is the cornerstone of modern SAS visualization. It is part of the Statistical Graphics (SG) procedures designed to produce publication-quality graphs quickly and efficiently. Unlike earlier procedures that required complex setup for aesthetics, `PROC SGPLOT` uses a declarative syntax, meaning you simply declare the type of plot you want (e.g., `SCATTER`, `HISTOGRAM`, `BAR`) and specify the necessary variables. This simplicity makes it the preferred choice for analysts seeking to perform quick data exploration or generate standardized reports.

The fundamental structure of any `PROC SGPLOT` request involves identifying the dataset and then specifying one or more plot statements. For scatter plots, the crucial statement is the `SCATTER` statement itself. This statement requires the assignment of variables to the horizontal (X) and vertical (Y) axes. By default, `PROC SGPLOT` handles axis scaling, labeling, and marker selection, providing a clean, ready-to-use output that visually represents the covariance between the two chosen variables.

### Method 1: Create One Scatter Plot

To generate a basic [scatter plot](#) that shows the relationship between two continuous variables across the entire dataset, you use the following concise syntax. This approach is ideal for initial exploratory data analysis (EDA) where the goal is to quickly assess the presence, direction, and strength of a correlation before introducing complexity such as grouping or stratification.

```
proc sgplot data=my_data;  
scatter x=var1 y=var2;  
run;
```

## Method 2: Create Scatter Plots by Group

When the analytical objective shifts to comparing relationships across different subsets of data--for instance, comparing performance metrics between different operational teams or demographic groups--the `GROUP=` option becomes indispensable. By adding the `GROUP=` option to the `SCATTER` statement, `PROC SGPLOT` automatically assigns unique colors and, optionally, unique marker symbols to the data points belonging to each level of the specified grouping variable (`var3` in this case). This stratification greatly enhances the ability to conduct comparative visual analysis.

```
proc sgplot data=my_data;  
scatter x=var1 y=var2 / group=var3;  
run;
```

## Setting Up the Data Environment (The Sample Dataset)

To illustrate these methods effectively, we will utilize a small, simulated dataset focusing on sports performance metrics. This dataset, named `my_data`, contains three critical variables: `team` (a categorical variable indicating the team--A or B), `points` (a continuous variable measuring points scored), and `rebounds` (a continuous variable measuring rebounds collected). This structure allows us to demonstrate both a simple bivariate relationship (points vs. rebounds) and a grouped comparison (points vs. rebounds segmented by team).

The following SAS code block demonstrates the creation of this dataset using the `DATA` step and the `DATALINES` statement. It is essential for reproducibility that the data creation step is executed prior to any visualization procedures. The subsequent `PROC PRINT` command confirms that the data has been loaded correctly into the SAS environment, showing all observations and variables available for plotting.

```
/*create dataset*/  
data my_data;  
input team $ points rebounds;  
datalines;  
A 29 8  
A 23 6  
A 20 6
```

```
A 21 9
A 33 14
A 35 11
A 31 10
B 21 9
B 14 5
B 15 7
B 11 10
B 12 6
B 10 8
B 15 10
;
run;

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

Reviewing the printed output ensures that the variable types are correctly assigned--specifically, that `points` and `rebounds` are numeric and `team` is character (indicated by the dollar sign \$ in the `INPUT` statement). This verification is a standard practice in data analysis to prevent procedural errors during subsequent visualization steps.

Obs	team	points	rebounds
1	A	29	8
2	A	23	6
3	A	20	6
4	A	21	9
5	A	33	14
6	A	35	11
7	A	31	10
8	B	21	9
9	B	14	5
10	B	15	7
11	B	11	10
12	B	12	6
13	B	10	8
14	B	15	10

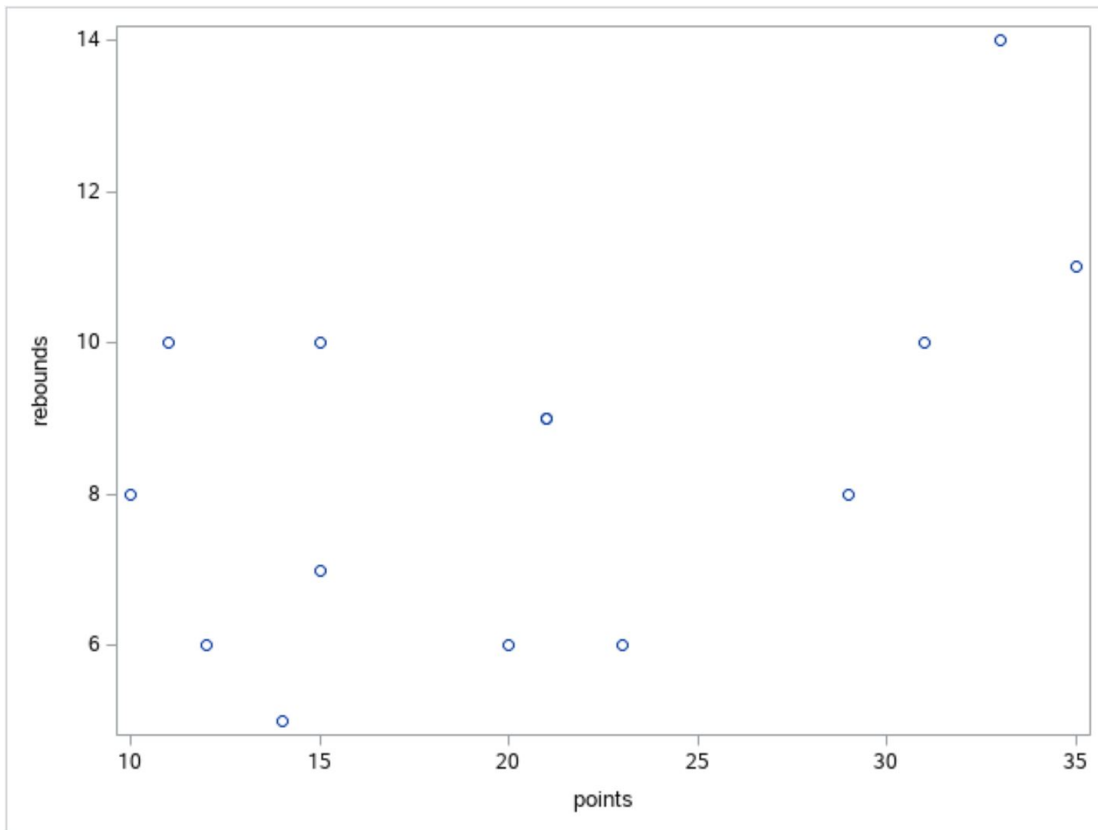
## Example 1: Creating a Simple Bivariate Scatter Plot

The most common application of the scatter plot is to explore the relationship between two variables without imposing any constraints or groupings. In our example, we want to understand if there is a relationship between the number of **points** scored and the number of **rebounds** collected, treating all observations (regardless of team) as a single population. This is a foundational step in [bivariate analysis](#).

The following code demonstrates the direct implementation of Method 1, using `points` as the X-variable and `rebounds` as the Y-variable. The simplicity of this code block highlights the efficiency of the `PROC SGPLOT` procedure in SAS. The output instantly provides a visual test of correlation: if the points cluster in an upward trend, a positive correlation is suggested; if they cluster in a downward trend, a negative correlation is suggested. If the points are scattered randomly, little to no linear relationship exists.

```
proc sgplot data=my_data;  
scatter x=points y=rebounds;  
run;
```

Upon execution, the resulting graph clearly maps the values: the X-axis displays the recorded values for the **points** variable, while the Y-axis displays the corresponding values for the **rebounds** variable. Each observation in the dataset corresponds to a single point plotted on this Cartesian plane. Analyzing the generated plot is crucial for statistical inference, as it immediately reveals potential outliers, the functional form of the relationship, and the density of the data at various points.



## Enhancing Visualization: Customizing Plot Appearance

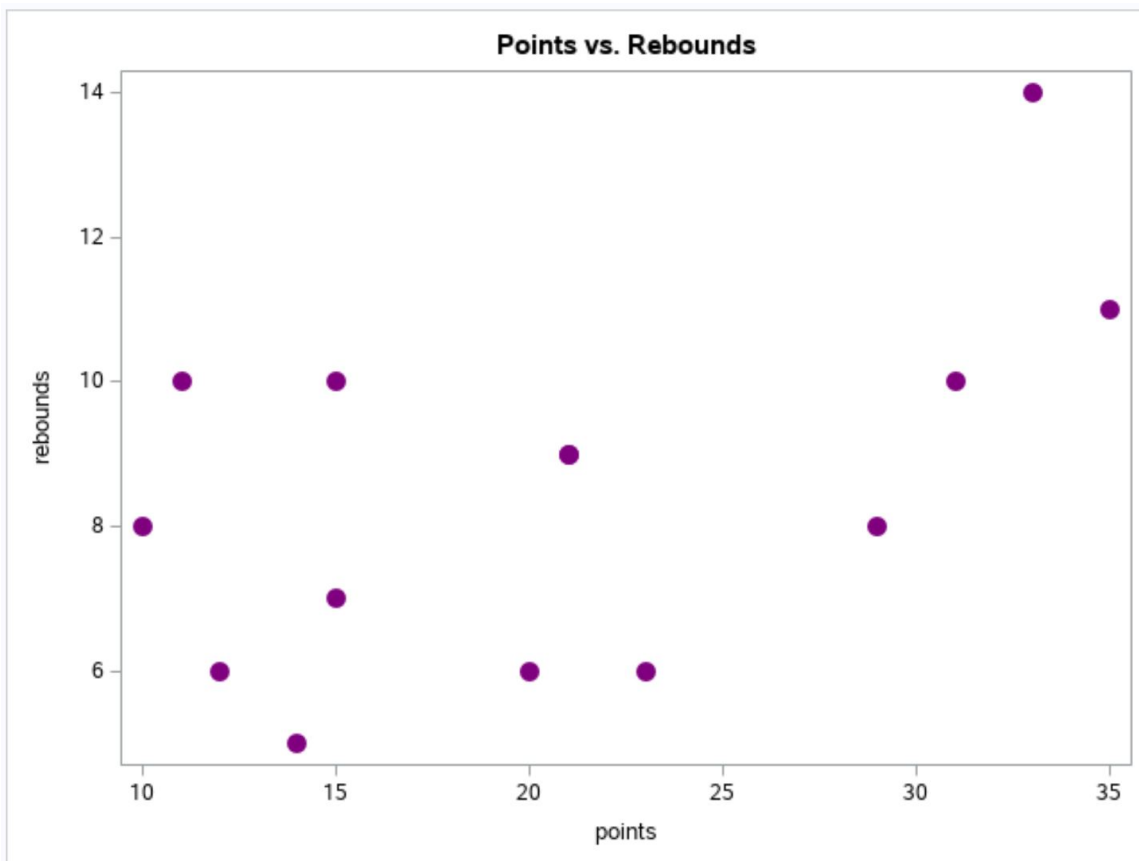
While the default [PROC SGPLOT](#) output is functional, it often benefits from aesthetic enhancements to improve clarity and professional presentation. Customization options allow the analyst to control elements such as the plot title, marker appearance (shape, size, color), and axis labels. Adding a descriptive title is essential, as it immediately informs the reader of the graph's purpose. The use of the `TITLE` statement, placed before the procedure step, provides a quick method for defining a clear, centered heading for the output graph.

Beyond simple titling, modifying the data markers themselves can significantly impact the visual appeal and readability of the plot, particularly when the data points overlap or when aiming for a specific corporate or publication aesthetic. The `SCATTER` statement supports the use of plot options, specified after a forward slash (/). The most important of these options for aesthetic control is the `MARKERATTRS` option. This option allows the user to specify characteristics like the marker `symbol` (e.g., `CircleFilled`, `Square`), the `size`, and the `color` of the markers.

The code below demonstrates how to apply a title and customize the markers to be filled circles, sized at 12 units, and colored purple. These modifications transform the basic output into a more visually compelling graphic, emphasizing the distinctiveness of each data point and potentially aiding in visual interpretation by increasing marker visibility. This level of detail in customization is

crucial for preparing graphics for formal reports or publications where visual standards are high.

```
title "Points vs. Rebounds";  
proc sgplot data=my_data;  
scatter x=points y=rebounds /  
markerattrs=(symbol=CircleFilled size=12 color=purple);  
run;
```



## Example 2: Generating Grouped Scatter Plots for Comparative Analysis

A significant strength of the [PROC SGPLOT](#) procedure is its ability to easily incorporate categorical variables to segment the visualization. This is achieved using the `GROUP=` option within the `SCATTER` statement, corresponding to Method 2. When the `GROUP=` option is specified, SAS automatically uses different colors (and optionally, different marker types) to distinguish the data points belonging to each unique value of the grouping variable. In this case, we use the `team` variable to separate the performance metrics of Team A from Team B.

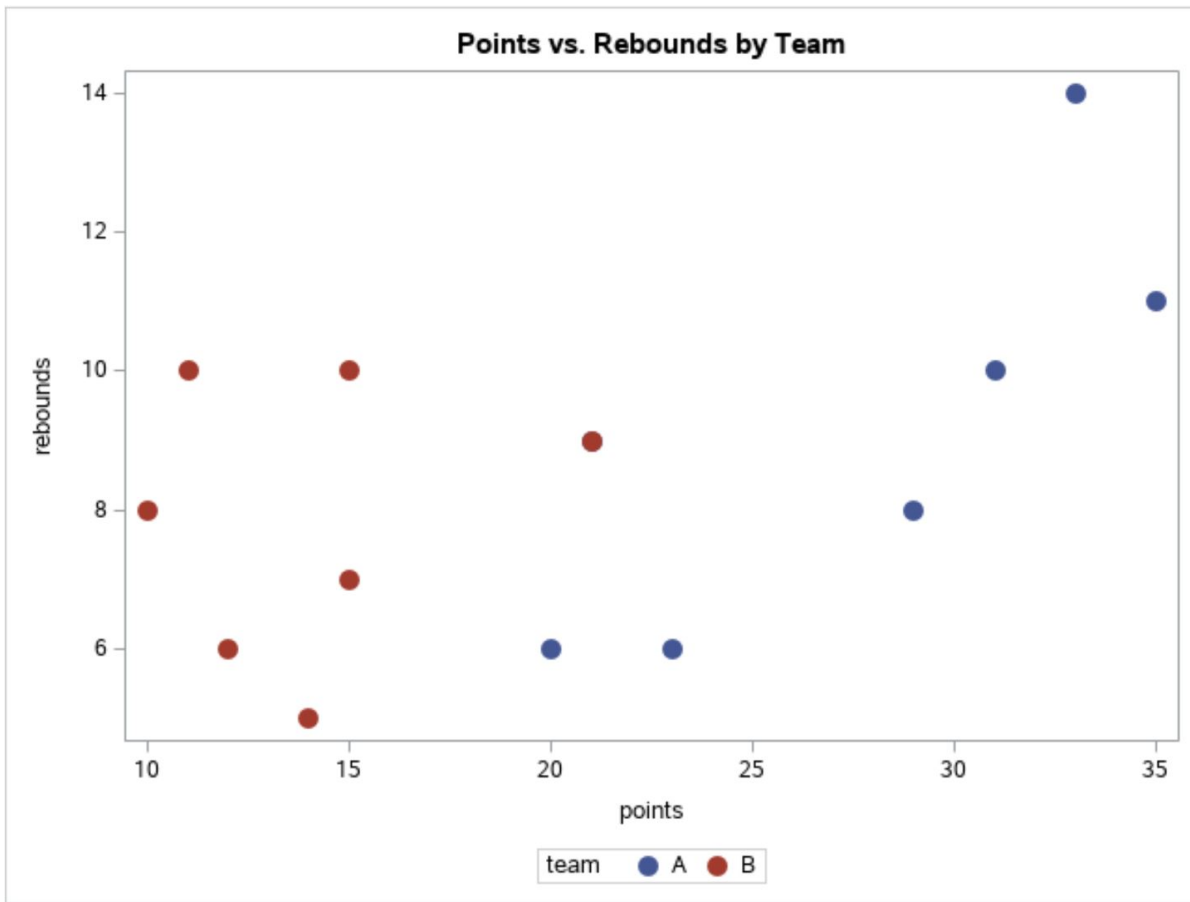
Grouping allows for immediate comparative analysis. For example, an analyst can visually assess whether the slope or clustering pattern (i.e., the correlation) between points and rebounds differs

substantially between Team A and Team B. This is crucial for identifying structural differences in the data that might be masked when viewing the data as a single aggregate population. The `GROUP=` option also automatically generates a legend, which is necessary for interpreting which color corresponds to which group, thereby completing the visual narrative.

In the following code block, we apply the `GROUP=team` option. We also maintain the aesthetic enhancements from Example 1, ensuring the markers are clearly visible, while allowing SAS to assign the default colors based on the group values. Note that a new title is used to reflect the comparative nature of the analysis.

```
title "Points vs. Rebounds by Team";  
proc sgplot data=my_data;  
scatter x=points y=rebounds /  
markerattrs=(symbol=CircleFilled size=12)  
group=team;  
run;
```

The resulting plot successfully segments the data, allowing us to quickly visualize the specific relationship between points and rebounds for both Team A and Team B individually. For instance, we might observe that Team A tends to achieve higher scores and rebounds overall, and the correlation within Team A's data appears stronger than the correlation observed within Team B's data points. This enhanced visual clarity is invaluable for presenting findings related to subgroup performance or characteristics.



## Conclusion and Further Visualization Resources

Creating effective scatter plots in [SAS](#) is a fundamental skill for any data analyst. By leveraging the modern `PROC SGPlot` procedure, analysts can quickly transition from raw data to sophisticated visualizations, whether the goal is simple bivariate exploration or complex comparative analysis using grouping variables. The flexibility offered by options like `MARKERATTRS` and the inherent efficiency of the ODS Graphics system ensure that the visualizations produced are not only statistically accurate but also aesthetically polished and ready for publication or presentation.

Understanding the relationship between continuous variables is often just one component of a larger statistical investigation. Scatter plots are frequently paired with other graphical tools, such as histograms or box plots, to provide a comprehensive view of data distributions and relationships. By continuously exploring the options available within `PROC SGPlot` and other `SAS/GRAPH` procedures, practitioners can unlock the full potential of their data visualization capabilities.

For those interested in expanding their knowledge of statistical visualization techniques within the SAS environment, the following resources provide guidance on creating other common chart types:

## **Additional Resources**

The following tutorials explain how to create other charts in SAS:

[How to Create Histograms in SAS](#)