How to Easily Create a Scatterplot with Regression Line in SAS

Authored by

stats writer

December 1, 2025

Generating a scatterplot overlayed with a regression line in SAS (Statistical Analysis System) is a fundamental task for visualizing bivariate relationships in statistics. This powerful combination allows analysts to immediately assess the correlation between two variables (the predictor and the response) and visualize the goodness of fit of a simple linear model. Unlike older graphical procedures, modern SAS relies heavily on the Statistical Graphics procedure, or PROC SGPLOT, which offers highly intuitive syntax and robust customization options for creating publication-quality graphics with minimal effort.

The core principle involves instructing PROC SGPLOT to first map the raw data points and then superimpose the calculated linear fit. This is achieved through specific statements within the procedure block. Historically, this required complex steps, but PROC SGPLOT simplifies the command structure by integrating the regression fitting directly into the visualization process via the `REG` statement. Mastering this technique is crucial for anyone performing exploratory data analysis or presenting statistical results using SAS.

This comprehensive guide will walk you through the essential steps, from the basic implementation to advanced customization techniques, ensuring you can generate clear, informative, and visually appealing statistical graphics. We will explore the critical parameters, such as defining axes, customizing line attributes, and controlling marker appearance, all within the framework of valid and efficient SAS code.

The PROC SGPLOT procedure is the preferred method for generating statistical graphics rapidly in SAS. We will use it extensively to create a scatterplot augmented by a **regression line**.

The following detailed examples demonstrate how to utilize this procedure effectively in a practical statistical programming environment.

Understanding the Role of PROC SGPLOT in Data Visualization

The **Statistical Graphics Procedure (PROC SGPLOT)** is the engine behind modern data visualization in SAS. Introduced to replace less flexible legacy procedures, `PROC SGPLOT` excels at creating sophisticated, high-quality graphs with minimal coding effort. It utilizes a declarative syntax where the programmer simply specifies the type of plot desired (e.g., histogram, boxplot, scatterplot) and the variables involved. This efficiency makes it the go-to tool for rapid prototyping and generating visualizations for immediate analysis or reporting.

Unlike procedures that require data manipulation or intermediate steps to calculate statistics before plotting, `PROC SGPLOT` handles many common statistical overlays, such as calculating and drawing the regression line, directly within its environment. When creating a combined scatterplot and regression fit, we leverage multiple statements within the single procedure block. This modular approach allows for easy layering of graphical elements, enabling complex visualizations where

raw data points, fitted curves, confidence intervals, and categorical distinctions coexist seamlessly.

The procedure is structured around three primary components: the procedure statement itself (`PROC SGPLOT`), plot statements that define the graphical elements (like `REG` or `SCATTER`), and global statements that control the overall appearance (like `TITLE` or `XAXIS`). Understanding this hierarchy is key to effectively commanding the graphical output. For fitting a simple linear model, the `REG` statement is paramount, as it simultaneously generates the scatter points and calculates the associated linear relationship between the specified Y (response) and X (predictor) variables.

The Mechanics of Creating a Basic Scatterplot with Regression Line

To produce the combined scatterplot and regression line, the programmer typically needs only two essential lines of code following the initial `PROC SGPLOT` call. First, we must declare the data set we intend to visualize, which is standard practice in SAS programming. We will use the readily available `sashelp.class` data set for demonstration purposes, which contains data on student heights and weights.

The core functionality rests within the REG statement. While one might initially assume separate statements for the scatter points (e.g., `SCATTER`) and the line (e.g., `SERIES`), the `REG` statement in `PROC SGPLOT` is designed to handle both components implicitly. When the REG statement is used, it automatically generates the scatter of the raw data points (the relationship between Y and X) and calculates and draws the best-fit linear relationship based on standard ordinary least squares methodology. This eliminates the need for separate plotting commands, streamlining the code dramatically.

For a basic visualization, the syntax requires mapping the dependent variable (Y) and the independent variable (X). For instance, if we wish to model the `Height` based on `Weight`, `Height` becomes the Y variable and `Weight` becomes the X variable. The resulting output provides an immediate visual summary: the dispersion of the data points illustrates variance and potential outliers, while the slope and intercept of the straight line provide the estimated parameters of the linear relationship.

Example 1: Generating the Standard Regression Plot

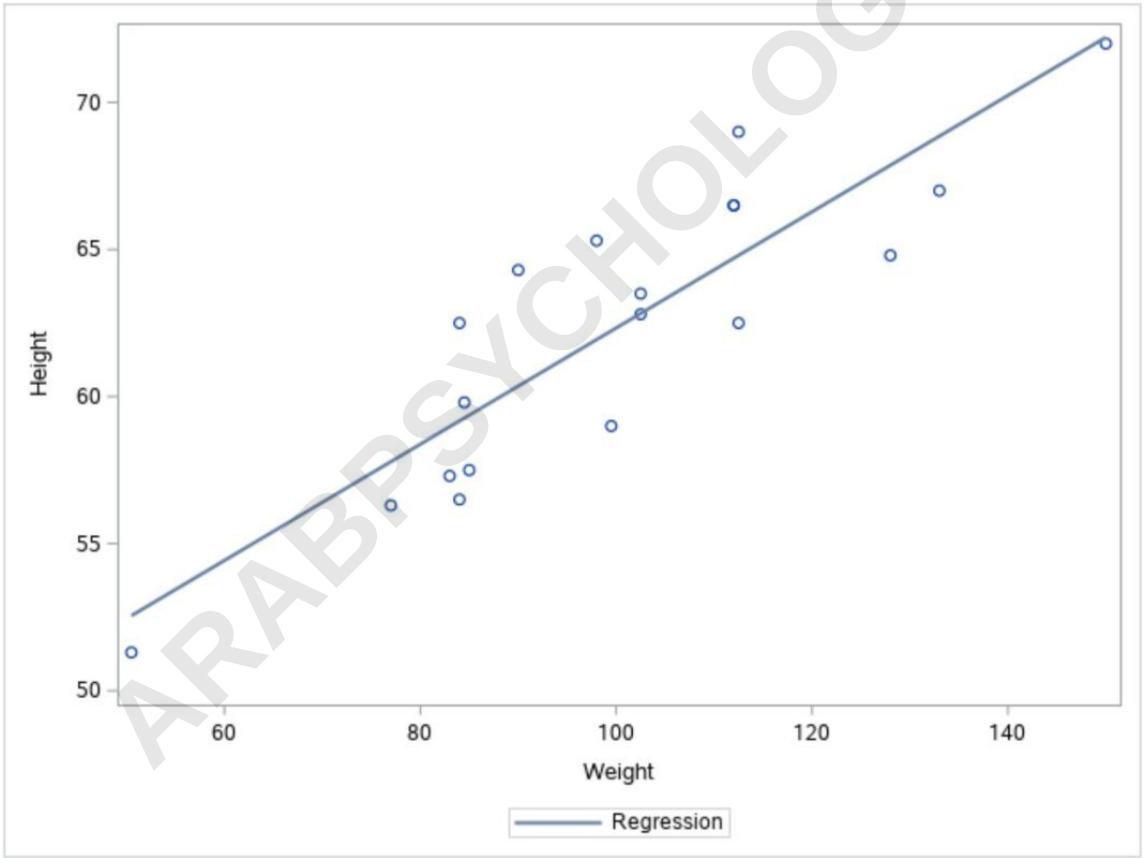
The following syntax demonstrates the simplest method for creating a scatterplot with an embedded regression fit using the widely used `sashelp.class` data set. This example serves as the foundation upon which all subsequent customizations are built, showcasing the efficiency of the PROC SGPLOT procedure in its most minimal form.

We initiate the process by calling `PROC SGPLOT` and specifying our data source. The subsequent `REG` statement immediately defines the bivariate relationship we are interested in visualizing. Note

how the use of the `REG` statement alone achieves the goal of plotting both the raw observations and the fitted linear model without requiring additional scatter statements or complex statistical calculations beforehand.

Observe the clean structure of the code block below. The comments clearly identify the purpose, and the execution is terminated by the `RUN` statement, a required element for executing any SAS procedure. This minimal code produces a standard output featuring default colors, markers, and automatically generated axes and legend, offering a quick diagnostic view of the data relationship.

```
/*create scatterplot with regression line*/
proc sgplot data=sashelp.class;
reg y=height x=weight;
run;
```



In the resulting graphic, the individual dots represent the raw data **observations** from the dataset, mapping each student's weight against their corresponding height. The solid blue line running through the center of the data cloud displays the calculated **fitted regression line**. This line summarizes the linear trend, indicating how much height is expected to increase for a one-unit increase in weight, providing a visual estimate of the model parameters derived through linear

regression techniques.

Advanced Customization Features in PROC SGPLOT

While the basic plot generated in Example 1 is useful for initial data exploration, professional reports often require extensive graphical customization to meet specific style guides or to enhance clarity. PROC SGPLOT is highly flexible, allowing users to override nearly every default graphical setting. This capability is essential for creating plots that are not only statistically accurate but also visually impactful and easily interpretable by a wider audience.

The customization options span across several categories, managed primarily through global statements and specific plot statement options. For instance, the clarity of the visualization can be dramatically improved by adding descriptive titles and explicitly labeling the X and Y axes, especially when variables have non-intuitive names or complex units. Furthermore, controlling the visual aesthetics--such as the colors, thickness, and marker shapes--is crucial for differentiating multiple series or highlighting specific features of the data.

When working with custom plots, it is important to manage the automatic features of PROC SGPLOT, such as the legend. Although helpful, the default legend might be unnecessary if only a single regression line is plotted, or if custom labels are provided. By utilizing specific procedure options, we gain granular control over these elements. The following list summarizes some of the key customization points available through **proc sgplot**:

Add a descriptive **title** to the chart using the `TITLE` statement.
Modify the axis labels using the `XAXIS` and `YAXIS` statements.
Suppress the automatic legend using the `NOAUTOLEGEND` option in the procedure call.
Customize the color, line style, and thickness of the regression line using the `LINEATTRS` option within the REG statement.
Customize the appearance (color, size, and symbol) of the data points in the scatterplot using the `MARKERATTRS` option.

Example 2: Achieving Advanced Plot Customization

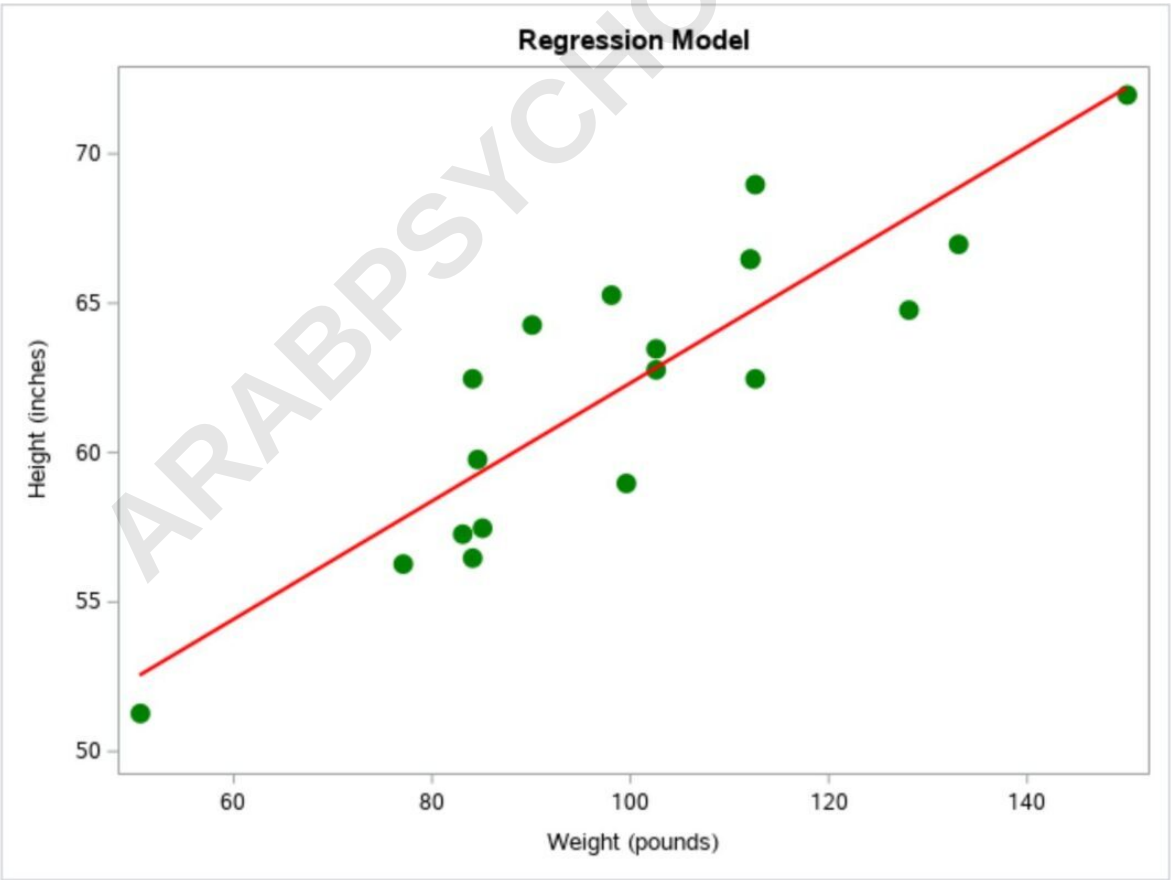
This example incorporates several customization options discussed previously to create a publication-ready graphic. We will modify the title, change the axis descriptions to include units, suppress the default legend, and drastically alter the appearance of both the fitted regression line and the scatter markers.

The `NOAUTOLEGEND` option is added to the `PROC SGPLOT` statement to prevent the automatic generation of a legend, providing a cleaner look when only one series is present. The `TITLE`, `XAXIS`, and `YAXIS` statements are utilized to provide clear, human-readable labels for the plot and

its dimensions, greatly improving interpretability. Pay close attention to the syntax required for these global statements.

The most significant modifications are applied within the REG statement using the attribute options, `LINEATTRS` and `MARKERATTRS`. These options utilize sub-options (like `COLOR`, `THICKNESS`, and `SYMBOL`) to define the specific aesthetics. This level of detail allows for precise control over the final visual output, making the resulting graph highly specific to reporting requirements.

```
/*create custom scatterplot with regression line*/
proc sgplot data=sashelp.class noautolegend;
title 'Regression Model: Height vs. Weight';
xaxis label='Weight (pounds)';
yaxis label='Height (inches)';
reg y=height x=weight /
lineattrs=(color=red thickness=2)
markerattrs=(color=green size=12px symbol=circlefilled);
run;
```



As evident from the resulting image, all specified modifications have been successfully implemented. The plot now features a descriptive title, clearly labeled axes including units, and distinct visual attributes: the regression line is now bold red, and the individual data points are large, filled green circles. This level of modification transforms a standard diagnostic plot into a highly tailored graphical output suitable for formal presentations or academic papers.

Fine-Tuning Aesthetics: LINEATTRS and MARKERATTRS Options

The `LINEATTRS` and `MARKERATTRS` options are the workhorses of aesthetic control within the `REG` statement. They allow the user to define exactly how the fitted curve and the raw data points are rendered, respectively. Understanding how to manipulate these attributes is essential for creating graphics that adhere to accessibility standards or institutional branding guidelines.

The `LINEATTRS` option takes sub-options that control the appearance of the fitted line. Key among these are `COLOR`, which dictates the hue (e.g., `red`, `blue`, `cxRRGGBB` hex codes); `THICKNESS`, which specifies the line weight (often measured in pixels); and `PATTERN`, which defines the line style (e.g., solid, dashed, dotted). By setting `THICKNESS=2` and `COLOR=red` in the previous example, we ensured the regression line stood out prominently against the background and the data markers, emphasizing the statistical model over the individual observations.

Similarly, the `MARKERATTRS` option controls the appearance of the individual data points in the scatterplot. Essential sub-options include `COLOR`, `SIZE` (which controls the diameter of the marker, often in pixels or points), and `SYMBOL`. SAS offers a wide array of built-in symbols (e.g., `CIRCLEFILLED`, `SQUARE`, `DIAMOND`) that can be selected based on design preference or to categorize different subsets of data if group variables were introduced. In Example 2, specifying `SYMBOL=circlefilled` and `SIZE=12px` ensured that the individual observations were large and clearly distinguishable.

Interpreting the Output: What the Regression Line Tells Us

The primary statistical purpose of adding a regression line to a scatterplot is to visually summarize the linear relationship between the predictor (X) and response (Y) variables. The line itself represents the predicted mean value of Y for any given value of X, calculated using the Ordinary Least Squares (OLS) method. The slope of this line indicates the direction and magnitude of the relationship: a positive slope suggests that as X increases, Y tends to increase, while a negative slope indicates an inverse relationship.

When interpreting the visual output from PROC SGPLOT, analysts should focus on two key aspects: the fit and the residuals. The fit is represented by how closely the line tracks the overall cloud of data points. If the data points cluster tightly around the line, it suggests a strong linear relationship and a good fit for the model. Conversely, if the points are widely scattered, the linear

model may not accurately represent the underlying data structure.

Furthermore, observing the pattern of the residuals--the vertical distance between each data point and the regression line--is vital. If the data points show a non-random pattern around the line (e.g., a curved or fan shape), it suggests that the assumption of linearity is violated or that heteroscedasticity is present. While the REG statement in `PROC SGPLOT` provides the basic line, more advanced statements like `SERIES` or specialized procedures might be necessary to visualize residuals or confidence intervals for a complete diagnostic assessment of the linear model.

Further Resources for SAS Visualization

To deepen your understanding of graphical capabilities in SAS, exploring related procedures and options is highly recommended. While `PROC SGPLOT` is excellent for single-panel plots, procedures like `PROC SGPANEL` and `PROC SGRENDER` offer ways to create multipanel displays and utilize custom GTL (Graph Template Language) code, respectively. These tools are invaluable when dealing with complex datasets requiring faceted visualization or highly specific design requirements.

Specific tutorials explaining how to perform other common tasks in SAS: