

# how to Plot Means with Standard Error Bars in SAS?

Authored by  
**stats writer**

November 19, 2025

## RECOMMENDED CITATION

stats writer (2025). *how to Plot Means with Standard Error Bars in SAS?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=97032>

## 1. The Importance of Error Bars in Statistical Visualization

When presenting descriptive statistics, simply reporting the mean value for a group or category often provides an incomplete picture of the data. While the mean indicates the central tendency, it fails to quantify the variability or uncertainty associated with that estimate. This is where error bars become indispensable tools in statistical visualization, offering crucial context regarding the precision of the calculated averages.

Error bars typically represent various measures of variability, such as the Standard error of the mean (SEM), the standard deviation (SD), or confidence intervals (CI). For comparative plots, particularly those showing differences between groups, the Standard error is often preferred because it estimates how far the sample mean is likely to be from the true population mean. Visualizing these error bounds helps researchers and analysts assess potential overlaps or significant differences between groups directly on the graph.

In the context of SAS, generating high-quality statistical graphics that include these indicators is straightforward using specialized procedures. This detailed guide focuses specifically on how to effectively calculate the necessary components and utilize the powerful visualization capabilities of SAS to produce clear, informative plots of group means accompanied by Standard error bars. Achieving this requires a two-step approach: first, calculating the required statistics (mean and error bounds), and second, executing the plotting routine.

## 2. Understanding the SAS Approach: Using the SGPLOT Procedure

SAS provides multiple avenues for statistical plotting, but for generating custom, high-impact graphics efficiently, the SGPLOT procedure (Statistical Graphics Plotting) is the modern standard. The SGPLOT procedure is declarative, meaning the user specifies the visual components (scatter points, lines, error bars) and maps variables to specific axes and aesthetic roles.

Unlike some other SAS procedures that automatically calculate and plot error bars (like **PROC GLM** or **PROC MEANS** with ODS graphics), the SGPLOT procedure often requires that the error boundaries be pre-calculated and available as columns in the dataset. This approach offers maximum flexibility, allowing the user to plot any custom statistic (e.g., standard deviation, 90% CI, or Standard error) as error limits. The specific error type must be chosen carefully based on the analytical goal, with SEM being widely used for comparing group differences.

The core components within **PROC SGPLOT** used for error bar visualization are the SCATTER statement, which defines the central data point (the mean), and specific options within that statement: YERRORLOWER and YERRORUPPER. These options dictate the variable names in the input dataset that hold the lower and upper bounds of the error bar, respectively. Consequently, before invoking the SGPLOT procedure, we must ensure our data preparation step correctly generates

these three critical variables: the mean value, the lower error bound, and the upper error bound. This necessity directs us toward using PROC SQL for the aggregation step.

### 3. Step 1: Data Preparation and Calculation using PROC SQL

To calculate the group statistics required for plotting, we utilize the powerful capabilities of the PROC SQL procedure. PROC SQL allows for efficient data manipulation and aggregation, making it ideal for summarizing datasets and calculating derived statistics like the mean and Standard error for specific groups.

The goal of this step is to transform our raw dataset, which contains individual observations, into a summary dataset where each row represents a group (e.g., 'team') and contains the aggregated statistics necessary for plotting. We need to calculate four distinct statistics for each group: the group identifier (team), the group mean (meanPoints), the lower bound of the Standard error (lowStdPoints), and the upper bound of the Standard error (highStdPoints).

The standard error (SEM) is calculated by taking the standard deviation and dividing it by the square root of the sample size. However, SAS's PROC SQL provides a convenient built-in function, STDERR(), which computes the Standard error of the mean directly. This greatly simplifies the calculation process. We calculate the bounds by subtracting and adding the Standard error value from the group mean. Note the use of the GROUP BY clause, which partitions the data and applies the aggregation functions (MEAN, STDERR) separately to each group defined by the team variable.

Here is the required syntax using PROC SQL to derive these necessary variables. Note that we create a new table, groupPlot, which will serve as the input for the subsequent visualization step:

```
/*calculate mean and standard error of points for each team*/  
proc sql;  
create table groupPlot as  
select  
team,  
mean(points) as meanPoints,  
mean(points) - stderr(points) as lowStdPoints,  
mean(points) + stderr(points) as highStdPoints  
from my_data  
group by team;  
quit;
```

## 4. Step 2: Generating the Plot with PROC SGPLOT

Once the intermediate dataset (`groupPlot`) containing the calculated means and error bounds is ready, we transition to the visualization phase using the `SGPLOT` procedure. This procedure is remarkably flexible, allowing us to combine multiple plot types (like scatter plots and line plots) to build a complex graphic tailored to our specific analytical needs. For plotting means with error bars, we typically rely on the `SCATTER` statement supplemented by the error options.

The `SGPLOT` procedure call starts by specifying the summary dataset (`data=groupPlot`). We then use the `SCATTER` statement, mapping the categorical grouping variable (`team`) to the X-axis and the calculated mean (`meanPoints`) to the Y-axis. The critical step is integrating the pre-calculated error bounds using the `YERRORLOWER=lowStdPoints` and `YERRORUPPER=highStdPoints` options. These options instruct SAS to draw vertical lines originating from the mean point and extending down to the lower bound and up to the upper bound, thus forming the Standard error bars.

In many mean-plot visualizations, it is beneficial to connect the means across the groups using a line, particularly if the X-axis represents an ordinal variable or simply to guide the eye across the data trend. This is achieved using the `SERIES` statement, which plots a continuous line between the coordinates defined by `x=team` and `y=meanPoints`. Both the `SCATTER` and `SERIES` statements often include the `GROUP=team` option, although in this specific aggregated dataset structure, it primarily ensures consistent grouping and coloring if custom aesthetics were applied. The combination of `SCATTER` for the points and error bars, and `SERIES` for the connecting line, produces a comprehensive visualization.

The comprehensive visualization code is structured as follows, seamlessly combining the necessary plot elements to render the final graphic:

```
/*create plot with mean and standard error bars of points for each team*/  
proc sgplot data=groupPlot;  
scatter x=team y=meanPoints /  
yerrorlower=lowStdPoints yerrorupper=highStdPoints group=team;  
series x=team y=meanPoints / group=team;  
run;
```

## 5. Detailed Example: Plotting Basketball Team Performance

To illustrate this methodology clearly, let us apply the syntax to a practical scenario involving basketball data. Suppose we have a dataset, named `my_data`, which tracks the points scored by players across three different teams (A, B, and C). Our objective is to visualize the average points

scored by each team and assess the precision of these averages using Standard error bars.

First, we must define and load our sample dataset into the SAS environment. This step ensures that the procedures have the necessary raw data to perform the calculations. The dataset is simple, comprising two variables: the categorical variable `team` and the numeric variable `points`. Creating sample data in this manner (using the `DATALINES` statement) is a standard practice for demonstrating SAS code effectiveness.

The code block below sets up the initial dataset and uses SAS's **PROC PRINT** to display the contents, allowing us to verify the input data structure and ensure all observations were loaded correctly before proceeding to the aggregation phase:

```
/*create dataset*/  
data my_data;  
input team $ points;  
datalines;  
A 29  
A 23  
A 20  
A 21  
A 33  
B 14  
B 13  
B 17  
B 14  
B 15  
C 21  
C 22  
C 20  
C 25  
C 24  
;  
run;  
  
/*view dataset*/  
proc print data=my_data;
```

Upon running the **PROC PRINT** statement, the output confirms the successful creation of the dataset, showing the distribution of points across the three teams. This is the raw data that will be summarized and plotted:

Obs	team	points
1	A	29
2	A	23
3	A	20
4	A	21
5	A	33
6	B	14
7	B	13
8	B	17
9	B	14
10	B	15
11	C	21
12	C	22
13	C	20
14	C	25
15	C	24

## 6. Execution of Data Aggregation and Plotting

With the raw data established, the next logical step involves executing the `PROC SQL` aggregation step. This step is crucial for transforming the individual point values into the summarized metrics required for graphical representation. We specifically calculate the mean points, and the lower and upper bounds derived from the Standard error calculation for each team (A, B, and C).

The syntax below, identical to the structure previously introduced, executes the necessary calculations and saves the output in the `groupPlot` table, which acts as the data source for the visualization. It is essential that the column names created here (`meanPoints`, `lowStdPoints`, `highStdPoints`) exactly match the variables referenced in the subsequent `PROC SGLOT` call.

```
/*calculate mean and standard error of points for each team*/
```

```
proc sql;  
create table groupPlot as  
select  
team,  
mean(points) as meanPoints,  
mean(points) - stderr(points) as lowStdPoints,  
mean(points) + stderr(points) as highStdPoints  
from my_data
```

```
group by team;
```

```
quit;
```

```
/*create plot with mean and standard error bars of points for each team*/
```

```
proc sgplot data=groupPlot;
```

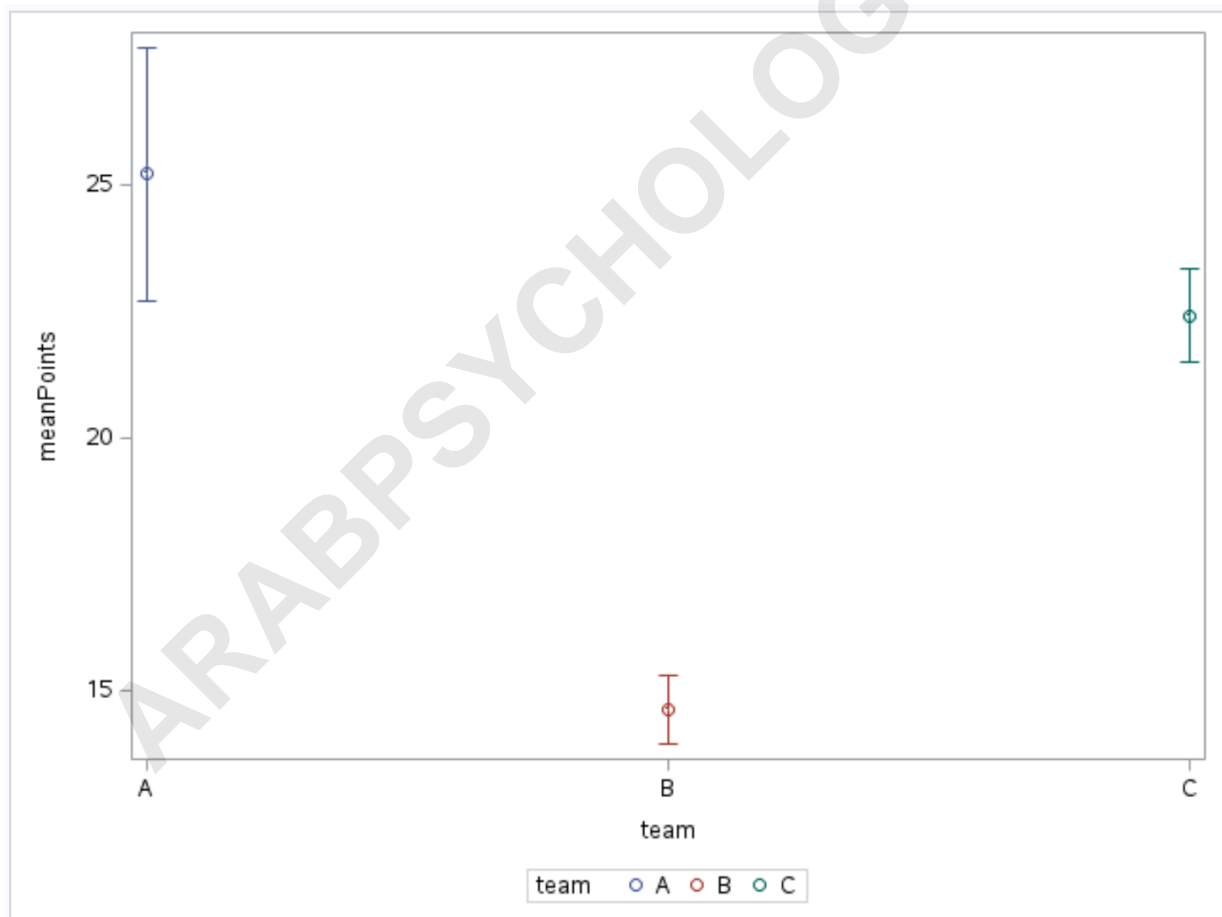
```
scatter x=team y=meanPoints /
```

```
yerrorlower=lowStdPoints yerrorupper=highStdPoints group=team;
```

```
series x=team y=meanPoints / group=team;
```

```
run;
```

Running the combined PROC SQL and SGPLOT procedure yields the desired visualization. The output is a clear graph where each team's average performance is marked by a data point, anchored by the vertical error bar representing the Standard error:



## 7. Interpreting the Generated Visualization and Data Table

The resulting graph effectively visualizes two key metrics simultaneously: the central tendency (the

mean, shown by the circle markers) and the precision of that estimate (the Standard error, shown by the vertical bars). The tiny circles represent the average point value for each respective team. The vertical lines extending above and below the circles indicate one Standard error unit in both directions, assuming a standard normal distribution for the sampling means.

A smaller error bar implies that the sample mean is a relatively precise estimate of the population mean, suggesting low variability within that team's performance or a larger sample size. Conversely, a longer error bar suggests greater variability or uncertainty regarding the true population average. Visual comparison between the teams is immediate: Team A has the highest mean score, while Team B exhibits the lowest mean score and the shortest error bar, indicating the most consistency in performance across its players given the sample size.

While the graph provides a powerful visual summary, it is often necessary to review the exact calculated values. We can use **PROC PRINT** on the summarized table, `groupPlot`, to confirm the numerical data used to construct the graph. This step is vital for transparency and numerical reporting, complementing the visual analysis with precise quantitative results:

```
/*print mean and standard error of points for each team*/  
proc print data=groupPlot;
```

This table output explicitly shows the mean points (`meanPoints`) alongside the precise lower and upper bounds (`lowStdPoints` and `highStdPoints`) derived from the Standard error calculation, providing a quantitative validation of the visual output. Analysts can use these bounds to perform rough comparisons; for instance, if the error bars of two means do not overlap, there is a strong suggestion of a statistically significant difference between the two group averages.

Obs	team	meanPoints	lowStdPoints	highStdPoints
1	A	25.2	22.7020	27.6980
2	B	14.6	13.9218	15.2782
3	C	22.4	21.4726	23.3274

## 8. Customization and Advanced Error Bar Options in SGPLOT

The flexibility of the SGPLOT procedure extends far beyond simple mean plotting. Users can extensively customize the appearance of the error bars, markers, and overall plot aesthetics. Common customizations include adjusting marker symbols, line thickness, colors based on group identity, and adding relevant titles and axis labels for clarity. For example, the `YERRORLOWER` and `YERRORUPPER` options can be paired with style options to change the appearance of the error bars

themselves, such as their cap size or color, enhancing the visual appeal and focus of the graphic.

For instance, if the analyst wished to display a 95% Confidence Interval (CI) instead of the standard error, the only necessary change would occur in the initial `PROC SQL` step. The calculation for the bounds would need to incorporate the appropriate t-statistic (or Z-score for large samples) multiplied by the Standard error, replacing the simple addition/subtraction of the Standard error value. The subsequent `SGPLOT procedure` syntax would remain functionally identical, only referencing the newly calculated CI bounds, demonstrating the modularity of the SAS coding environment.

Furthermore, **PROC SGPLOT** is designed to handle overlays. If you needed to compare the mean points against a predetermined benchmark, you could easily add a horizontal reference line using the `REFLINE` statement. This modular approach ensures that even complex statistical visualizations remain clean, reproducible, and easy to modify, leveraging the data preparation strength of `PROC SQL` and the graphic power of the `SGPLOT procedure` in SAS. This versatility makes **PROC SGPLOT** an invaluable tool for professional statistical reporting.

## 9. Conclusion and Further SAS Resources

Generating visualizations of group means coupled with standard error bars is a fundamental requirement in statistical reporting. SAS provides a highly effective and precise method for achieving this by combining the data aggregation capabilities of `PROC SQL` with the specialized plotting functions of the `SGPLOT procedure`. This two-step process--calculate bounds, then plot points and error limits--ensures that researchers maintain explicit control over the statistics being visualized.

By consistently applying the principles of pre-calculating mean and Standard error boundaries and correctly mapping these variables to the `YERRORLOWER` and `YERRORUPPER` options in the `SCATTER` statement, analysts can produce descriptive graphics that are not only visually engaging but also statistically robust. The ability to clearly see the magnitude of uncertainty around each group mean significantly enhances the interpretability of comparative analyses, facilitating sound data-driven conclusions.

For those interested in exploring additional advanced visualization techniques or different types of charts within the SAS environment, several procedures offer specialized plotting capabilities. We recommend exploring the official documentation for further tutorials on generating various chart types, including histograms, box plots, and heat maps:

The **GCHART** procedure for traditional business graphics.

The **SGMAP** procedure for geographical visualizations.

The `SGPLOT procedure` for a wide range of analytical plots and custom graphics creation.