

# Learning Comparative Panel Plots Using PROC SGPANEL in SAS

Authored by  
**Mohammed loot**

November 14, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Learning Comparative Panel Plots Using PROC SGPANEL in SAS*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=1537>

## Introduction to Comparative Panel Plotting with PROC SGPANEL in SAS

In the specialized realm of [SAS](#) programming, the capability to generate clear, insightful visualizations is **absolutely paramount** for effective data analysis and transparent communication of results. Analysts frequently encounter complex [datasets](#) characterized by multiple categorical or grouping variables. In these scenarios, the critical need arises to systematically compare statistical distributions or identify underlying trends across numerous distinct subgroups simultaneously.

This requirement is precisely addressed by the [PROC SGPANEL](#) procedure in SAS, an indispensable utility for comparative graphics. This procedure empowers users to automatically generate and arrange numerous individual plots into a sophisticated [panel layout](#). By facilitating seamless side-by-side comparisons, **PROC SGPANEL** enables analysts to achieve a deeper, more granular understanding of intricate data structures that might otherwise remain obscured in aggregated views. It transforms complex comparisons into highly manageable visual tasks.

The core advantage of employing **PROC SGPANEL** lies in its efficiency, consistency, and automation. It meticulously eliminates the tedious process of manually producing, styling, and arranging individual graphs for every subset of the data. Instead, it systematically generates a series of graphs, where each element represents a specific subset defined by the unique values of one or more grouping variables. Furthermore, this procedure enforces crucial consistency in visual elements, such as synchronized scaling, unified axis limits, and overall presentation style. This consistency is fundamentally crucial, moving beyond mere aesthetics to facilitate accurate and reliable visual comparison across different groups, which is the primary analytical objective of comparative graphics.

This comprehensive guide will walk through two fundamental and highly practical methodologies for leveraging the advanced capabilities of **PROC SGPANEL**. We will first explore the straightforward process of creating comparative plots grouped by a single classification [variable](#), offering a clear approach to subgroup analysis. Subsequently, we will extend this methodology to demonstrate how to effectively group plots by multiple classification variables concurrently, thereby enabling the visualization of complex conditional distributions and interaction effects. Both methods will be illuminated using clear, practical SAS code examples, demonstrating how to efficiently visualize data distributions and uncover meaningful, hidden patterns within your data categories.

### Preparing Your Data: A Foundation for Panel Visualization

Before any specialized visualization capabilities of [PROC SGPANEL](#) can be initiated, it is critical that your raw information is appropriately structured and successfully loaded into a **SAS dataset**. For the purpose of our detailed demonstrations, we will utilize a hypothetical dataset that we name `my_data`. This sample dataset is specifically designed to contain performance statistics for

basketball players, including essential [variables](#) such as `team`, `position`, `points` scored, and `assists`.

The following [DATA step](#) block meticulously constructs this sample dataset using inline data entry. Every line documented within the `datalines` section represents a single observation--a player's individual record. Character variables, which include `team` and `position`, are explicitly designated by appending a dollar sign (\$) in the [INPUT statement](#). Conversely, numerical variables, such as `points` and `assists`, are specified without this dollar sign prefix. The concluding [RUN statement](#) is essential, as it executes the preceding instructions, thereby finalizing the creation of the `my_data` dataset, which is now structurally ready for visualization and analysis.

```
/*create dataset*/  
data my_data;  
input team $ position $ points assists;  
datalines;  
A Guard 14 4  
A Guard 22 6  
A Guard 24 9  
A Forward 13 8  
A Forward 13 9  
A Guard 10 5  
A Guard 20 6  
A Guard 34 9  
A Forward 33 8  
A Forward 15 5  
B Guard 24 4  
B Guard 22 6  
B Forward 34 2  
B Forward 15 5  
B Forward 23 5  
B Guard 10 4  
B Guard 12 6  
B Forward 30 2  
B Forward 15 5  
B Forward 11 5  
;  
run;  
  
/*view dataset*/  
proc print data=my_data;
```

To ensure the complete integrity of the data and confirm its correct structure, we utilize [PROC PRINT](#). This essential procedure generates a tabular display of all observations and variables, allowing for a quick, visual verification that every data point has been loaded accurately and is structurally sound. Once this preliminary check is successfully completed, the data is formally verified and fully prepared for sophisticated comparative visualization techniques using [PROC SGPANEL](#).

Obs	team	position	points	assists
1	A	Guard	14	4
2	A	Guard	22	6
3	A	Guard	24	9
4	A	Forward	13	8
5	A	Forward	13	9
6	A	Guard	10	5
7	A	Guard	20	6
8	A	Guard	34	9
9	A	Forward	33	8
10	A	Forward	15	5
11	B	Guard	24	4
12	B	Guard	22	6
13	B	Forward	34	2
14	B	Forward	15	5
15	B	Forward	23	5
16	B	Guard	10	4
17	B	Guard	12	6
18	B	Forward	30	2
19	B	Forward	15	5
20	B	Forward	11	5

## Method 1: Creating Panel Plots Grouped by a Single Variable

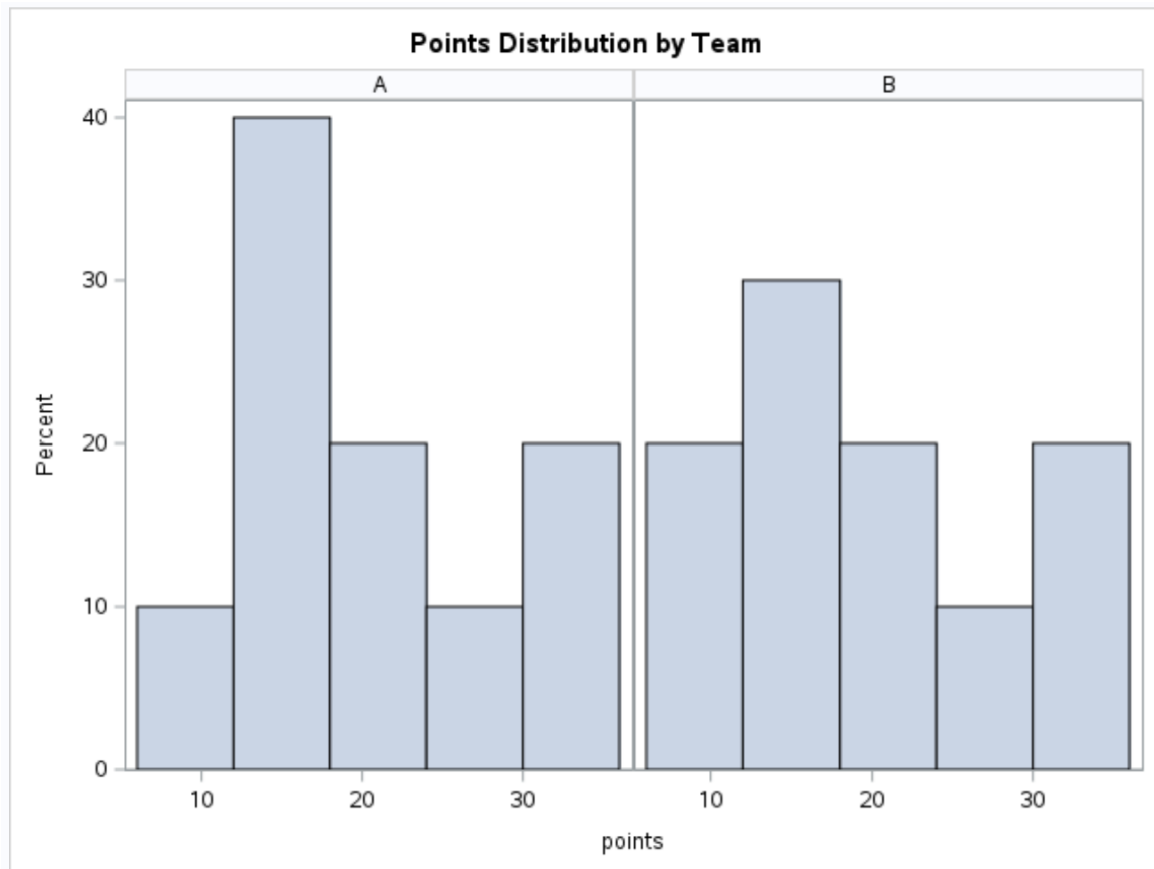
One of the most foundational and frequently utilized applications of [PROC SGPANEL](#) is the efficient generation of a series of plots, where each plot distinctly corresponds to a unique category defined within a single grouping [variable](#). This approach proves exceptionally valuable for analysts who need to quickly compare the distribution, measure of centrality, or key characteristics of a specific metric across different, well-defined groups. In our running example, we will concentrate

on visualizing the distribution of `points` scored, distinctly separated and displayed for each unique `team`.

The operational core of this specific visualization task is the [PANELBY statement](#). This statement is specifically used to designate the variable that dictates the grouping structure of the resulting panels. Here, `team` is designated as the variable that will form the individual panels. We include the `/ novarname` option within the statement to suppress the display of the variable name (`team`) directly above each individual panel, resulting in a cleaner, less cluttered, and more concise visual output. Following this, the [HISTOGRAM statement](#) is employed to specify that we intend to plot the distribution of the continuous `points` variable within the boundaries of each generated panel.

```
title "Points Distribution by Team";  
proc sgpanel data=my_data;  
panelby team / novarname;  
histogram points;  
run;
```

The resulting visual output from this code block will present two distinct [histograms](#), automatically arranged side-by-side in a horizontal configuration. Each histogram clearly represents the distribution of `points` for a specific team (Team A and Team B, respectively). This efficient side-by-side arrangement enables the user to conduct an immediate visual comparison of scoring patterns between the two teams, effectively highlighting differences in critical statistical measures such as central tendency, overall data spread, and distribution skewness. This default horizontal [panel layout](#) is standard for most analyses.



While the default [panel layout](#) in SAS typically organizes plots horizontally, situations often arise where a vertical arrangement is strongly preferred or necessitated. This might be due to practical constraints on horizontal space, or a focused analytical desire to emphasize the individual characteristics of each distribution for enhanced readability. Fortunately, **PROC SGPNEL** provides the necessary flexibility to precisely control the panel orientation through the use of the [ROWS option](#), which is specified within the **PANELBY statement**.

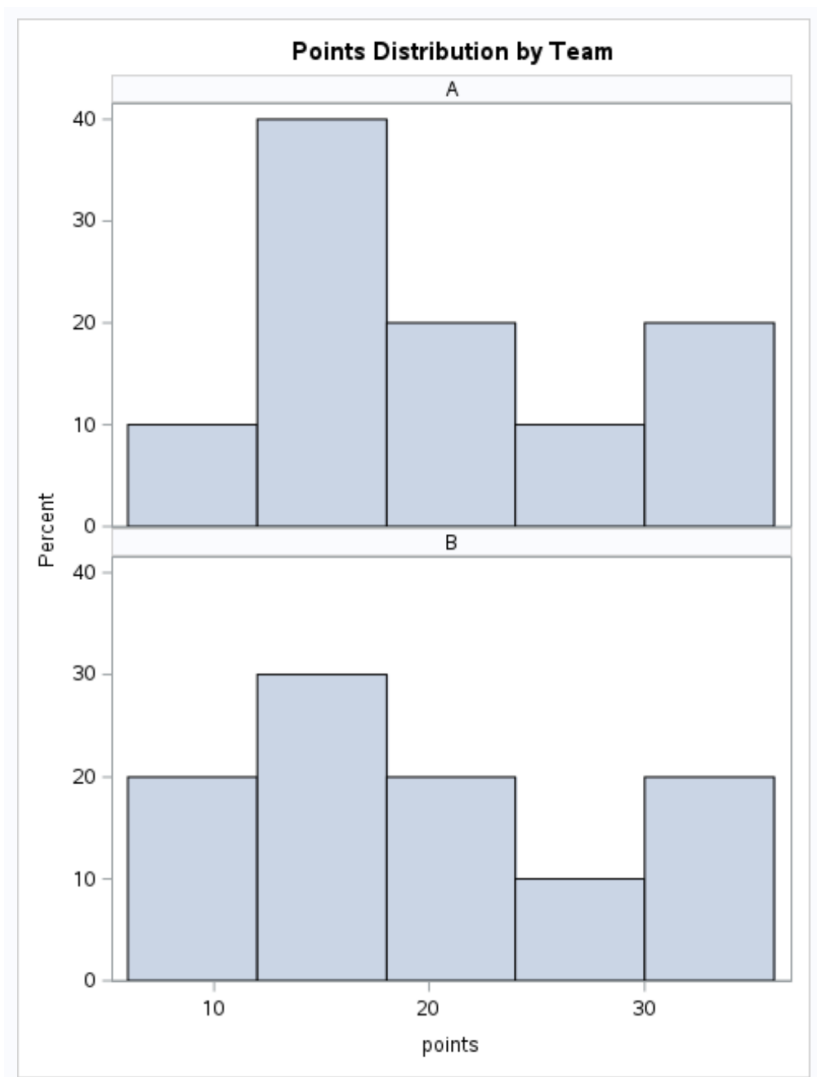
By simply adding the option `rows=2` to the [PANELBY statement](#), we instruct SAS to override the default arrangement and organize the two team-specific [histograms](#) into two separate rows instead of consolidating them into a single row. This modification is particularly beneficial when the total number of panels is small, allowing each plot to occupy more vertical space, thereby emphasizing and clearly distinguishing their individual distributions. The resulting code structure is minimally changed yet yields a significantly different and optimized visual arrangement.

```

title "Points Distribution by Team";
proc sgpanel data=my_data;
panelby team / rows=2 novarname;
histogram points;
run;

```

As clearly illustrated in the subsequent image, the histogram for Team A is prominently displayed at the top of the output, while the histogram for Team B is positioned directly beneath it. This vertical stacking offers a highly clear and organized view, making the comparison of their respective point distributions straightforward and immediate. It represents a powerful yet simple modification for optimizing visualizations based on specific aesthetic or presentation requirements, especially for reports that are optimized for vertical viewing.



## Method 2: Creating Panel Plots Grouped by Multiple Variables

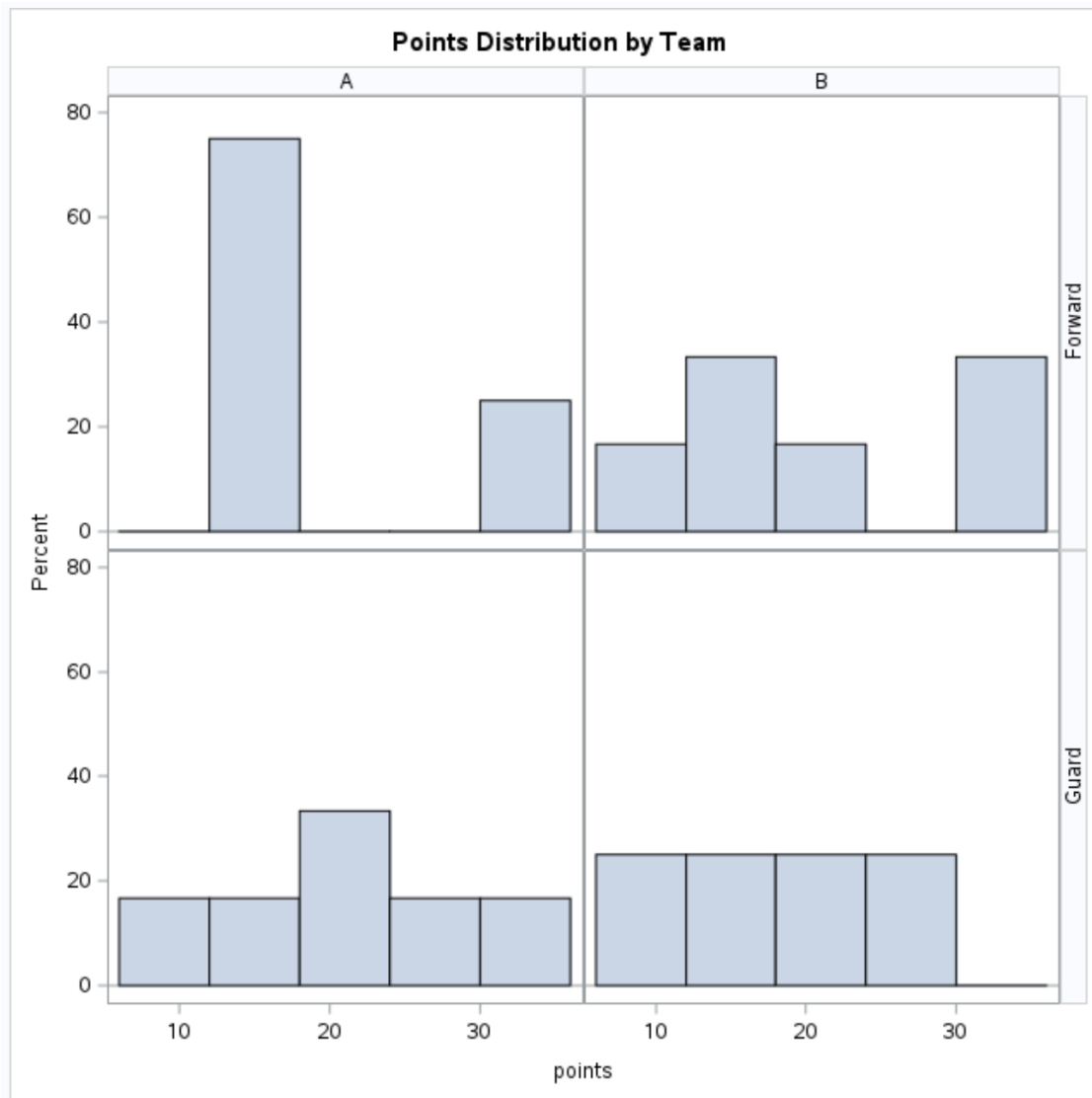
While grouping by a single classification [variable](#) is highly useful, [PROC SGPANEL](#) truly demonstrates its advanced analytical capability when you need to concurrently explore complex interactions or conditional distributions across two or more categorical variables. This advanced functionality facilitates a far more granular level of analysis, enabling the discovery of nuanced insights--such as interaction effects--that would likely remain obscured in simpler, aggregated

views of the [dataset](#).

To construct a [panel of plots](#) that are grouped by multiple variables, the process remains syntactically efficient: you simply list all the desired grouping variables within the **PANELBY statement**. For instance, to conduct a detailed examination of the `points` distribution stratified by both `team` and `position`, we specify `panelby team position`. This command instructs SAS to generate a unique panel for every distinct combination of `team` and `position` found within the input data, effectively creating a powerful cross-classification visualization.

For multi-variable paneling, a critical option is `layout=lattice`. This option is crucial because it ensures that the resulting panels are logically arranged in a well-defined grid-like structure, where each cell in the grid precisely represents a unique combination of the grouping variables. This highly structured [layout](#) is essential for preserving clarity, maintaining visual order, and guaranteeing ease of comparison, particularly when dealing with a substantial number of panels. As in the previous example, `novarname` is included for aesthetic cleanliness, and `histogram points` specifies the plot type and the continuous variable whose distribution is to be displayed within each cell of the lattice.

```
title "Points Distribution by Team and Position";  
proc sgpanel data=my_data;  
panelby team position / layout=lattice novarname;  
histogram points;  
run;
```



The final resulting output is a robust visual grid displaying four distinct [histograms](#). These include: Team A Guards, Team A Forwards, Team B Guards, and Team B Forwards. This comprehensive panel layout facilitates a highly detailed comparison of the `points` distributions across these specific subgroups. By visualizing the data in this cross-classified manner, an analyst can readily identify how scoring patterns might differ not only between the teams themselves but also based on the specific player role (position) within each respective team, providing a rich, multi-dimensional view of the performance data.

## Versatility with Other Plot Types and ODS Graphics

While our instructional examples have predominantly relied on [histograms](#) to effectively visualize data distributions, it is essential to understand that [PROC SGPANEL](#) is not restricted to this single plot type. Its true strength and power are derived from its remarkable versatility, permitting the

generation of a wide spectrum of graphical representations within the same clean, organized [panel layout](#). This inherent flexibility makes the procedure suitable for virtually any analytical need, ranging from the exploration of univariate distributions to the detailed examination of relationships between continuous variables.

The analyst can effortlessly substitute the `histogram` statement with various other specialized graphical statements to create different types of plots tailored to specific data insights. For example, popular and frequently used alternatives include:

**VBOX:** Used to generate [vertical box plots](#). Box plots are statistically robust tools, excellent for comparing summary distributions across multiple groups by clearly displaying the median, quartiles, and potential outliers.

**HBOX:** Similar in function to `VBOX`, but creates [horizontal box plots](#), which may offer enhanced readability or be preferred depending on the complexity or the number of categories involved in the analysis.

**SCATTER:** This statement produces [scatter plots](#), making it the ideal tool for visualizing the relationship between two continuous variables within the specific context of each panel. A key example would be plotting `points` against `assists` for every unique team-position combination to analyze correlations.

**DENSITY:** Creates kernel density estimates, which provide a smooth, continuous representation of the data distribution, highly valuable when analyzing continuous variables and comparing their underlying shapes.

**BAR:** Generates bar charts, which are perfectly suited for displaying frequencies, counts, or calculated sums associated with a categorical variable, offering a standard way to summarize discrete data.

The decision regarding the appropriate plot type should always be driven by the inherent nature of your data and the specific analytical insights you are aiming to extract and communicate to your audience. Regardless of the plotting statement used, **PROC SGPANEL** ensures a critical level of graphical consistency: all plots are generated with synchronized axes and uniform styling across the entire [panel layout](#). This consistency is the hallmark of high-quality comparative visualizations produced by the [SAS ODS Graphics](#) system, making direct comparisons straightforward and statistically reliable.

## Conclusion and Additional Resources

The [PROC SGPANEL](#) statement in [SAS](#) stands as a powerful, flexible, and essential tool for generating truly compelling comparative visualizations. By providing the capability to effortlessly group plots by one or multiple classification variables, it dramatically enhances the analyst's capacity to explore intricate [datasets](#) and efficiently uncover latent patterns or critical differences

between various subgroups. Whether the analytical task requires simple side-by-side comparisons or complex grid-based lattice layouts, **PROC SGPANEL** offers the precise syntax and extensive options necessary to achieve sophisticated visualization objectives with high efficiency and statistical precision.

We have successfully navigated the entire process of setting up the data, demonstrated how to construct panels based on both single and multiple grouping variables, and explored the broad spectrum of alternative plot types available within this robust procedure. Mastering **PROC SGPANEL** is an essential step that will undoubtedly elevate your data analysis and reporting capabilities within the SAS environment, enabling you to communicate complex data insights far more clearly and effectively to any audience.

For additional technical specifications, advanced customization techniques, and a complete listing of all statement options, analysts should always refer to the complete official [documentation for the PROC SGPANEL statement in SAS](#). It is highly recommended to experiment with different plot types and panel options--such as the `COLUMNS` and `ROWGUTTER` options--to fully grasp the extensive and robust capabilities of this procedure.

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