

Learning to Create Histograms in SAS: A Step-by-Step Guide with Examples

Authored by
Mohammed looti

October 31, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learning to Create Histograms in SAS: A Step-by-Step Guide with Examples*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=7435>

Introduction to Histograms and SAS Utilization

Histograms are fundamental [statistical graphics](#) used extensively in [data analysis](#) to visually represent the distribution of numerical data. They provide a clear visual summary of the major features of the distribution of the sample, including its shape, central tendency, and variability. In the SAS environment, the most efficient and powerful procedure for generating these distributional plots is the [PROC UNIVARIATE](#) procedure. While other procedures exist, **PROC UNIVARIATE** is specialized in providing detailed descriptive statistics alongside high-quality graphics, making it the standard choice for exploratory data analysis of a single variable.

Understanding the underlying distribution of your data is a critical first step in almost any statistical modeling effort. A histogram segments the range of values into intervals (bins) and displays the frequency (or percentage) of observations falling into each bin. This visual representation helps analysts quickly identify potential outliers, determine if the data is normally distributed, or detect skewness and multimodality. Mastering the creation of histograms in SAS is therefore essential for any serious data professional using the platform.

This tutorial outlines three distinct methods for generating histograms using SAS. These methods progress from the simplest case--plotting a single variable--to more complex visualizations involving grouped data and overlaid distributions. Each technique leverages the robust capabilities of **PROC UNIVARIATE**, demonstrating how minor adjustments in syntax can yield vastly different and highly informative charts suitable for various analytical purposes.

Three Primary Methods for Histogram Generation in SAS

The following three standard approaches cover the majority of use cases when generating a [histogram](#) within the SAS environment. We will utilize the **PROC UNIVARIATE** command for all three methods, as it offers superior control over distributional analyses and visualization compared to other procedures like PROC GCHART or PROC SGPLOT when focusing purely on the variable distribution. These methods are listed below, followed by detailed examples utilizing a sample [dataset](#).

The primary distinction between these methods lies in the inclusion of the `CLASS` statement, which allows for grouping, and the `/ OVERLAY` option, which controls the visualization style when multiple groups are present.

Method 1: Create One Histogram: This is the simplest form, used to visualize the distribution of a single continuous variable across the entire dataset.

```
proc univariate data=my_data;  
var var1;
```

```
histogram var1;  
run;
```

Method 2: Create Panel of Histograms: This method introduces the `CLASS` statement, which instructs SAS to create a separate histogram for the analysis variable (`var1`) for each unique level found in the classification variable (`var2`). The resulting output is a panel of individual plots, perfect for comparing distributions across defined groups.

```
proc univariate data=my_data;  
class var2;  
var var1;  
histogram var1;  
run;
```

Method 3: Overlay Histograms: By adding the `/ OVERLAY` option to the `HISTOGRAM` statement (in conjunction with the `CLASS` statement), SAS plots all the group-specific distributions onto a single set of axes. This is extremely useful for direct, side-by-side visual comparison of the distribution shapes and locations relative to one another.

```
proc univariate data=my_data;  
class var2;  
var var1;  
histogram var1 / overlay;  
run;
```

Setting Up the Sample Dataset for Analysis

To effectively demonstrate these three methods, we must first establish the sample dataset that will be used throughout the examples. This synthetic dataset, named `my_data`, represents hypothetical performance metrics for basketball players across two different teams, A and B. It contains three variables: `team` (a categorical variable), `points` (a continuous variable representing points scored), and `rebounds` (another continuous variable). The structure of this data will allow us to demonstrate both univariate (Method 1) and grouped (Methods 2 and 3) analyses.

The following SAS code block utilizes the `DATA` step and the `DATALINES` statement to create and populate the `my_data` dataset directly within the SAS session. It is important to note the use of the dollar sign (\$) after the `team` variable in the `INPUT` statement, which designates it as a character variable, suitable for use in the `CLASS` statement later on.

```
/*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;
```

Executing the `PROC PRINT` command confirms the successful creation and population of the dataset, providing a tabular view of the observations. This verification step is crucial before proceeding with graphical analysis to ensure the data integrity is sound and matches expectations. The dataset contains 14 observations split unevenly between the two teams. This initial structure confirms we are ready to proceed with the visualization techniques outlined previously.

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 Single Histogram Using PROC UNIVARIATE

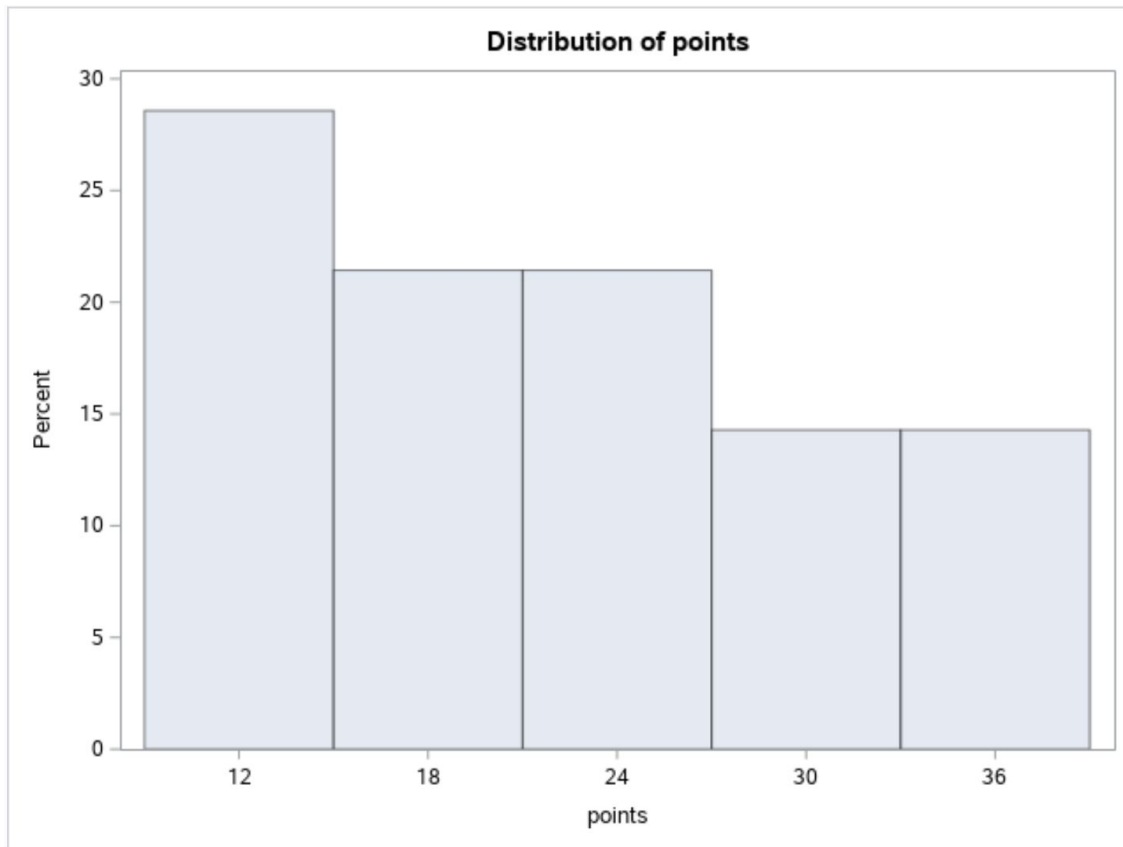
The first and most basic application of the histogram procedure involves visualizing the distribution of a single continuous variable across all observations in the dataset, without any grouping. For this example, we focus on the **points** variable, which measures player scoring performance. This visualization will help us understand the overall scoring pattern across all players regardless of their team affiliation.

To achieve this, we use [PROC UNIVARIATE](#), specifying `my_data` as the input dataset. The critical statements here are `VAR points;`, which selects the variable for analysis, and `HISTOGRAM points;`, which explicitly requests the creation of the graphical output. Since no `CLASS` statement is included, the procedure treats all observations as a single group.

```
/*create histogram for points variable*/  
proc univariate data=my_data;  
var points;  
histogram points;  
run;
```

Upon execution, the resulting [histogram](#) is displayed. The **x-axis** represents the range of values observed for the **points** variable, segmented into equal-width bins. The **y-axis**, by default in SAS's univariate output, displays the percentage of observations in the [dataset](#) that fall into each

respective bin. Analyzing this plot allows us to quickly assess the concentration of scoring figures, identify the mode (the bin with the highest frequency), and note the overall spread of the data, which appears to be somewhat bimodal or skewed based on the combined data distribution.



Example 2: Generating a Panel of Histograms for Group Comparison

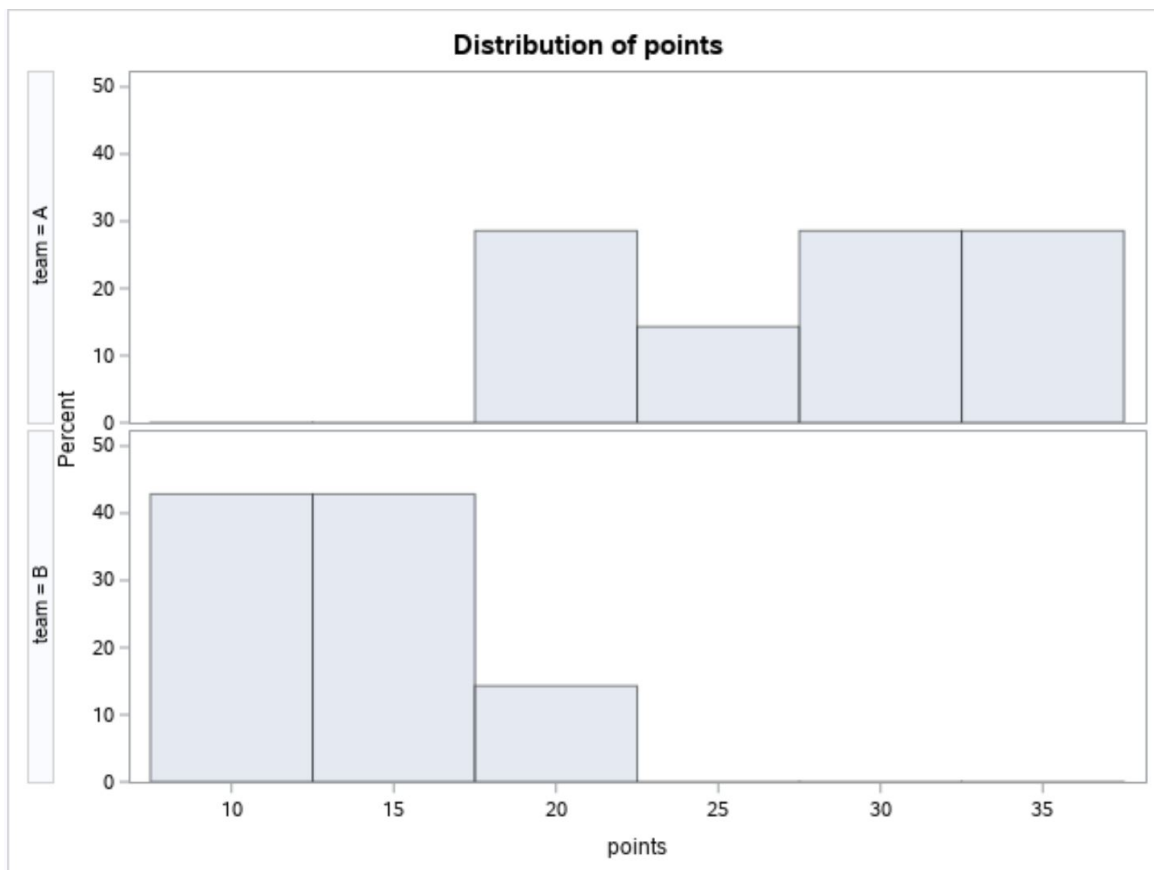
Often in [data analysis](#), the distribution of a variable must be compared across different subgroups defined by a categorical variable. This is where the concept of a panel of histograms becomes invaluable. A panel visualization keeps the distributions separate but aligns them spatially, facilitating direct comparison without overlap. In our scenario, we want to compare the distribution of **points** scored between **team A** and **team B**.

To achieve this grouping, we introduce the `CLASS` statement into our **PROC UNIVARIATE** code. The syntax is straightforward: `CLASS team;` specifies that the subsequent analyses should be partitioned by the unique values (A and B) found within the `team` variable. Importantly, we keep the `HISTOGRAM points;` statement without any additional options.

```
/*create histogram for points variable*/  
proc univariate data=my_data;  
class team;
```

```
var points;  
histogram points;  
run;
```

The output is two distinct histograms, one for each team, stacked vertically or arranged side-by-side depending on the SAS output environment settings. The crucial feature of this panel plot is that the two histograms share a common **x-axis** scale. This shared scaling is fundamental, as it allows analysts to compare the relative locations of the data distributions with ease. Visual inspection confirms that the players on team A generally score higher points (distribution shifted right) than the players on team B (distribution shifted left and more tightly clustered). This rapid visual insight into distributional differences is the primary benefit of generating a panel plot.



Example 3: Overlaying Histograms for Direct Visualization

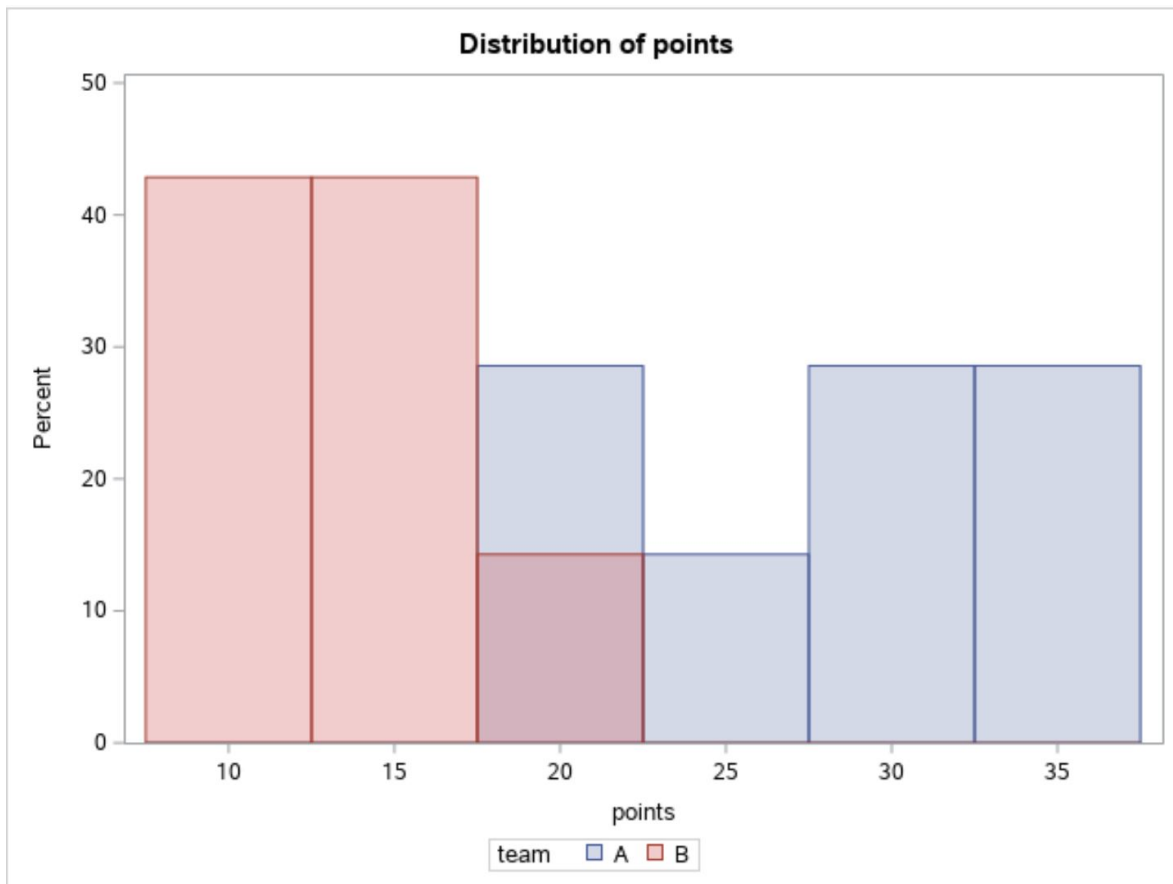
While a panel of histograms is excellent for comparison, sometimes the analyst requires a single, unified chart where the distributions are explicitly superimposed. This technique, known as overlaying histograms, is particularly useful when the goal is to show the degree of separation or overlap between two or more groups directly on the same set of coordinates.

To overlay the group histograms, we must still use the `CLASS` statement (as established in Example 2) to define the groups (Team A and Team B). The key modification is adding the `/ OVERLAY` option immediately following the `HISTOGRAM` statement. This option signals to [PROC UNIVARIATE](#) that instead of plotting separate panels, it should combine the distributions into one plot, using different colors or patterns to distinguish between the groups.

```
/*create histogram for points variable*/  
proc univariate data=my_data;  
class team;  
var points;  
histogram points / overlay;  
run;
```

The resulting graphical output places the histogram for Team A and the histogram for Team B onto the same axes. SAS automatically assigns distinct colors to the bars of each group, and typically includes a legend to identify which color corresponds to which team. This visualization is highly effective when attempting to illustrate the degree of difference in central tendency and spread. For instance, in our example, the overlaid plot clearly shows the minimal overlap between the higher scoring range of Team A and the lower scoring range of Team B.

This type of plot is invaluable for presentations and reports where space is limited or where the primary message is centered around the contrast between group distributions. Analysts should exercise caution when overlaying many distributions (more than three or four), as too much overlap can lead to visual clutter and hinder interpretability. However, for comparing two groups, the overlaid histogram provides a clean, impactful, and summarized view of the data distribution.



Further Exploration and Resources

Generating histograms is just one facet of the powerful analytical capabilities available within the SAS programming environment. While **PROC UNIVARIATE** is the designated tool for distributional analysis and the creation of these specific plots, SAS offers a wide array of procedures for creating various other types of [statistical graphics](#), including box plots, scatter plots, and bar charts.

For those seeking to further customize their graphical output--such as adjusting bin widths, specifying density curves (e.g., normal or kernel density estimates) on the histogram, or changing colors and titles--the **PROC UNIVARIATE** documentation provides extensive options that can be added to the `HISTOGRAM` statement. Additionally, for users who require highly customized or complex visualizations outside the scope of basic distributional plots, procedures like `PROC SGLOT` offer greater flexibility in combining multiple plot types into a single figure.

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