

How to Easily Create Histograms in SAS: A Step-by-Step Guide

Authored by
stats writer

December 1, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Create Histograms in SAS: A Step-by-Step Guide*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=103176>

Generating histograms within the SAS statistical software environment is a fundamental yet powerful technique for visualizing the distribution of continuous variables. This process is highly versatile, primarily utilizing the PROC UNIVARIATE procedure combined with the HISTOGRAM statement, though other methods like the VBAR statement (often used in PROC SGPLOT or PROC GCHART for bar charts that mimic distributional visualization) also exist for specific charting needs. The initial step requires precise identification of the numerical variable whose distribution you wish to analyze.

While the HISTOGRAM statement provides a straightforward approach for standard frequency distributions, more specialized statements, such as the VBAR statement, offer enhanced capabilities, especially when dealing with categorical data or when aiming for custom aesthetic effects. Beyond the basic plot, SAS grants significant control over visualization elements. Users can easily integrate additional options to customize appearance, such as defining titles, specifying the ordering of bins, or setting the precise width of the histogram bars to fine-tune the graphical representation of the data structure.

This tutorial will explore three distinct and highly practical methods for creating histograms: generating a single distribution plot, producing a comparative panel of distributions, and overlaying multiple distributions onto a single axis. These methods are essential tools for data analysts seeking to understand data skewness, kurtosis, and the overall shape of their variable distributions before proceeding to deeper statistical modeling.

Overview of Histogram Generation Techniques

The following techniques leverage PROC UNIVARIATE, the specialized procedure designed for detailed descriptive statistics and frequency distribution analysis, to generate one or more histograms. These approaches scale from simple single-variable plots to complex comparative visualizations, allowing users to deeply explore data structure based on grouping variables.

Method 1: Creating a Simple, Single Histogram

This is the most basic application, focusing solely on the distribution of one continuous variable. The core syntax involves calling PROC UNIVARIATE and specifying the target variable using the VAR statement, followed by the HISTOGRAM statement repeating the variable name.

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

Method 2: Generating a Panel of Comparative Histograms

When comparative analysis is required, you can partition the data based on a categorical variable (or grouping variable). By introducing the CLASS statement, SAS automatically generates side-by-side histograms, one for each level defined in the classifying variable. This is invaluable for visual comparisons of distributional shape across different groups.

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

Method 3: Overlaying Distributions within a Single Plot

For direct, high-contrast comparisons, overlaying distributions is often preferred over panels. This technique uses the same CLASS statement as Method 2, but crucially, it adds the `/ OVERLAY` option to the HISTOGRAM statement. This forces SAS to plot all resulting histograms on a unified axis system, enabling immediate observation of shifts in central tendency and spread.

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

The following examples show how to use each method with a specific sample dataset in SAS:

Preparing the Sample SAS Dataset for Visualization

To effectively demonstrate the three methods outlined above, we will first create a small, representative dataset named `my_data`. This dataset simulates basketball statistics, containing observational data for two variables of interest: `points` (a continuous variable representing scores) and `team` (a categorical variable used for grouping). The structure of the data is critical for illustrating how PROC UNIVARIATE handles different levels of data complexity, particularly when comparing distributions across groups.

The following SAS Data Step code defines the dataset, using the DATALINES statement for direct input of the raw data. Note that the `team` variable is defined as a character variable using the dollar sign (\$) input modifier, which is essential for its subsequent use as a grouping variable in the

CLASS statement. The `points` variable is the continuous numerical measure we will analyze using histograms.

```
/*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 generates the visual output below, allowing us to confirm the structure and content of the dataset before generating any plots. This verification step ensures that the variables `team`, `points`, and `rebounds` are correctly loaded into the SAS working environment.

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: Generating a Single Variable Distribution Plot

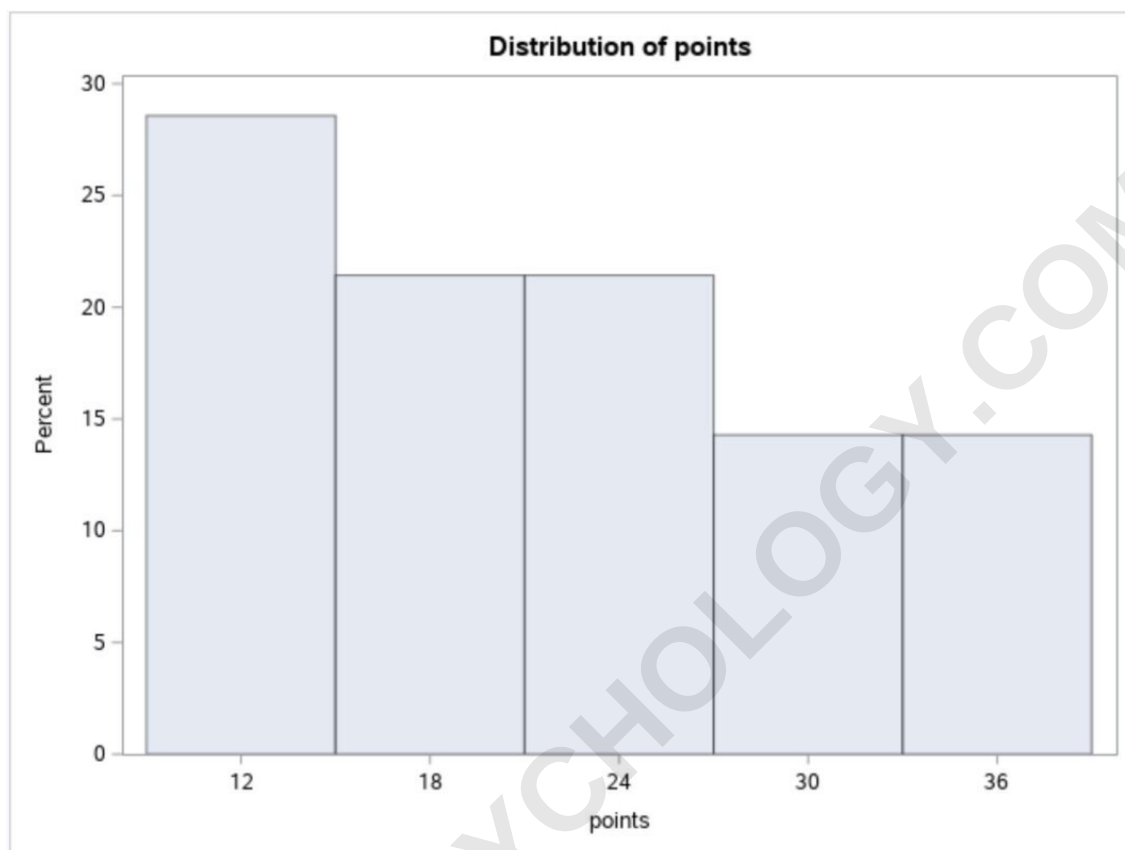
This first example illustrates the simplest application of the distribution plotting capabilities in SAS. Our goal is to visualize the overall frequency distribution of the **points** variable across all observations in the `my_data` dataset, without any division by team or group. This approach is fundamental for assessing the shape of the data--identifying potential outliers, verifying assumptions of normality (or lack thereof), and locating the central tendency of the scores.

The code utilizes the `PROC UNIVARIATE` statement followed by the `VAR` statement, which specifies `points` as the variable to be analyzed. Crucially, the `HISTOGRAM` statement directs SAS to generate the graphical output. Since no additional options or grouping variables are included, SAS uses its default binning algorithm to segment the range of `points` values into appropriate intervals.

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

The resulting plot below provides a clear visual summary. The **x-axis** represents the range of values observed for the **points** variable, while the **y-axis** typically displays the frequency, count,

or, in many default `PROC UNIVARIATE` outputs, the percentage of observations that fall within each defined bin or interval. Analyzing this output shows us where the majority of scores cluster--in this case, suggesting a concentration around the lower and upper twenties.



Example 2: Comparative Analysis Using a Panel of Histograms

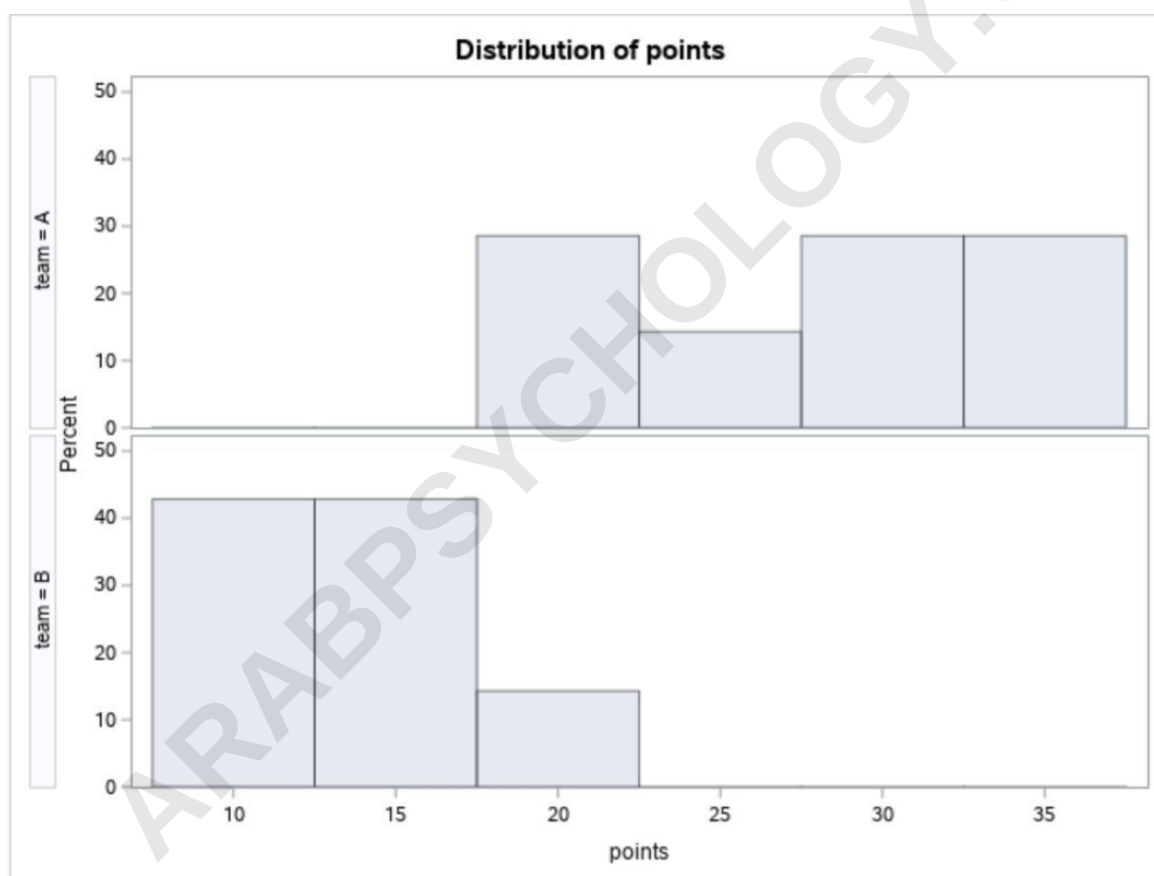
A frequent requirement in statistical analysis is comparing the distribution of a metric across different subsets of data defined by a categorical factor. This second example demonstrates how to create a panel of separate histograms, visualizing the distribution of **points** separately for each level of the **team** variable (A and B). This technique is crucial for determining if the underlying data generation process or central tendency differs significantly between groups.

To achieve this comparison, we introduce the CLASS statement (`class team;`) into the `PROC UNIVARIATE` block. The CLASS statement instructs SAS to perform the analysis--and generate the corresponding histogram--for each unique value found in the `team` variable. Because we omit the `/ OVERLAY` option, the output consists of distinct plots stacked vertically or arranged in a panel format.

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

```
class team;  
var points;  
histogram points;  
run;
```

Upon viewing the generated panel plot, the differences between the groups become immediately apparent. The distribution for team A is visibly shifted to the right compared to team B, confirming that players on team A generally achieve a higher number of **points**. A key advantage of this panel visualization is the shared horizontal axis (x-axis), which ensures that the scaling of the points variable is consistent across both teams. This standardization facilitates reliable, direct visual comparison between the two groups' performance distributions.



Example 3: Superimposing Distributions Using the OVERLAY Option

While panel plots (Example 2) are effective, visualizing multiple distributions simultaneously within a single frame often provides the clearest and most immediate comparison. This technique, known as overlaying, is achieved by adding a simple yet powerful option to the HISTOGRAM statement. The ability to overlay distributions is extremely useful when the analyst needs to compare minor

differences in shape or tail behavior between groups that might be masked in separate panel plots.

This method requires the same setup as creating the panel plot: we use `PROC UNIVARIATE` and specify the grouping variable (`team`) using the `CLASS` statement. The crucial difference lies in appending the `/ OVERLAY` option immediately after the `HISTOGRAM points` command. This modifier directs SAS to plot the frequency polygons or density curves derived from the two groups (Team A and Team B) on the same set of axes, typically differentiating them using color or line style.

`/*create histogram for points variable*/`

`proc univariate data=my_data;`

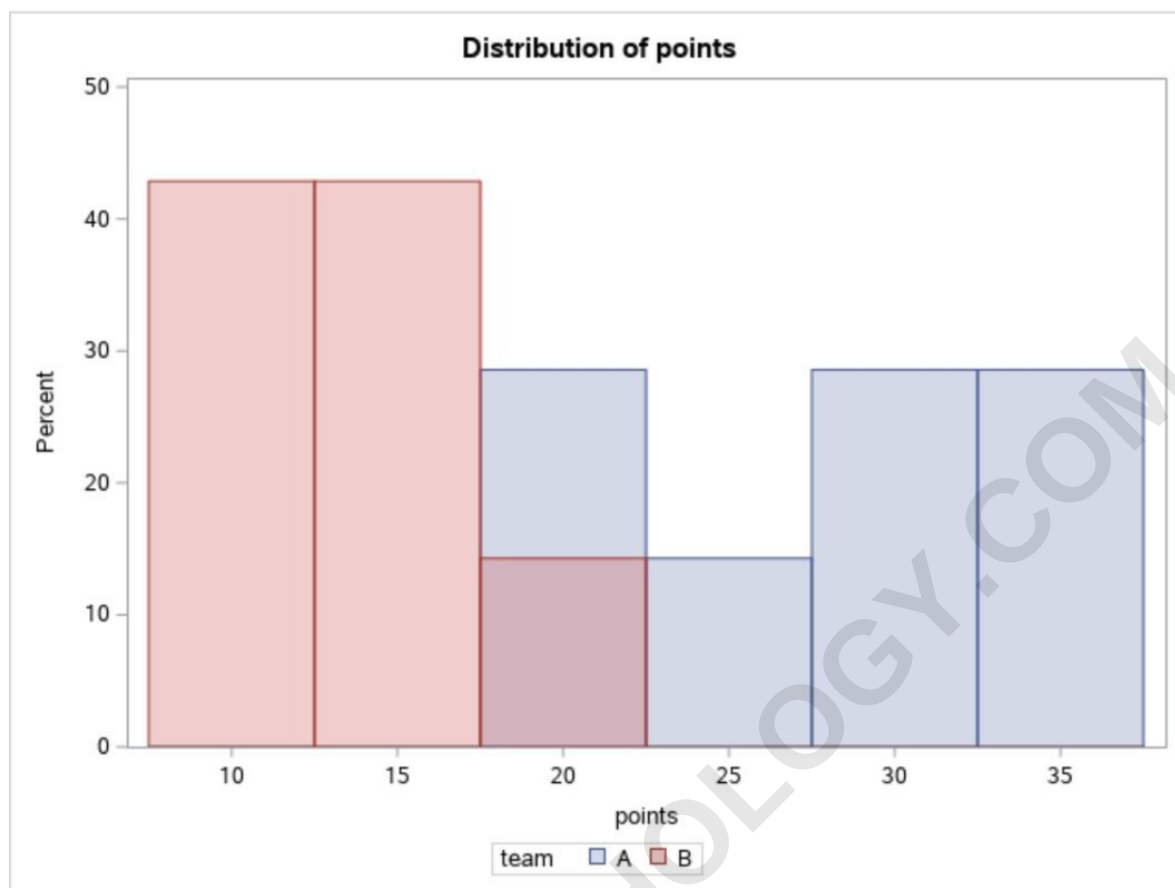
`class team;`

`var points;`

`histogram points / overlay;`

`run;`

The resulting visualization displays the two distributions overlaid, making the contrast between Team A's higher performance range and Team B's lower performance range extremely easy to perceive. This plotting method, which often defaults to density curves rather than bar heights in modern SAS graphical output, is particularly effective for high-impact communication of distributional differences, avoiding the need to visually switch focus between separate plots.



Further Customization and Related SAS Visualization Techniques

While the examples above focus on the core functionality of generating histograms using PROC UNIVARIATE, advanced users can customize these plots extensively. Options exist within the HISTOGRAM statement to specify the exact number of bins (NBINS=), define specific bin boundaries (MIDPOINTS=), or fit theoretical density curves (e.g., NORMAL, KERNEL) directly onto the histogram for formal comparison against expected distributions. Mastering these options allows for precise control over how data is represented graphically.

For example, if you suspect your data follows a specific distribution, adding HISTOGRAM points / NORMAL; can generate a superimposed normal curve, aiding in the visual assessment of normality assumptions crucial for many parametric statistical tests. Furthermore, if you are looking for highly customized graphic output beyond the capabilities of the traditional PROC UNIVARIATE output, procedures like PROC SGLOT offer greater flexibility in aesthetics and customization, including specialized density plots and bar charts.

Beyond histograms, SAS offers a wide array of visualization tools essential for comprehensive data exploration. We encourage readers to explore additional tutorials detailing how to create other

common chart types, such as box plots (which visualize quartiles and potential outliers), scatter plots (for examining bivariate relationships), and bar charts (for summarizing categorical data), thereby rounding out their data visualization skillset in SAS.

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

ARABPSYCHOLOGY.COM