How to Perform Grouped Univariate Analysis in SAS Using Proc Univariate

Authored by

stats writer

November 22, 2025

The Proc Univariate procedure in SAS is one of the most fundamental and powerful tools available for performing detailed, single-variable statistical analysis. When paired with the BY statement, this procedure allows analysts to execute this robust analysis separately for distinct subgroups within a larger dataset. This capability is absolutely essential for researchers and data scientists who need to understand how the statistical profile of a key variable changes across different categories, such as comparing performance metrics across different product lines, age groups, or, as we will demonstrate, athletic teams.

Using Proc Univariate with grouping functionality provides an efficient method for analyzing the distributional characteristics of a variable for each set of clustered observations. By automating the repetitive task of subsetting data and running independent analyses, SAS ensures consistency and significantly reduces the time required for data exploration. This procedure goes far beyond simple means and standard deviations, generating comprehensive descriptive statistics, moments, quantiles, and graphical outputs like box plots and histograms for every specified group simultaneously.

This comprehensive guide will walk through the exact syntax and practical application of leveraging the BY statement within Proc Univariate. We will focus on ensuring the input data meets the necessary prerequisites and demonstrate how to correctly interpret the segmented output generated by SAS. Mastering this technique is crucial for anyone engaging in comparative data analysis within the SAS environment.

2. Understanding the Purpose of Grouped Univariate Analysis

Grouped univariate analysis serves as a cornerstone of data quality assessment and preliminary statistical modeling. Its primary objective is to facilitate the comparison of a variable's central tendency, dispersion, skewness, and kurtosis across different populations defined by a categorical variable. Without this grouping capability, an analyst would only see the aggregated statistics for the entire dataset, which often masks significant variations or anomalies present only within specific subgroups. For instance, an overall mean might appear normal, but a grouped analysis could reveal that one subgroup is heavily skewed while another is tightly clustered.

The core utility lies in its ability to simultaneously compare distributions. Analysts often use this procedure to check for violations of statistical assumptions, such as homogeneity of variance, before proceeding to more complex inferential tests like ANOVA. If the spread (variance) of a variable, say income, differs wildly between groups (e.g., different cities), the grouped output from Proc Univariate provides immediate, quantifiable evidence of these disparities, allowing for necessary data transformations or selection of alternative statistical models. This makes the procedure invaluable for diagnostics.

Furthermore, running Proc Univariate with the BY statement is the most efficient way to generate

specialized graphical representations for each subgroup. When visualization options (like plotting or creating box plots) are specified, SAS automatically generates separate plots for each level of the grouping variable. This visual inspection complements the tabular output, providing immediate insight into potential outliers or differences in distribution shape that might influence subsequent analyses.

3. Basic Syntax and Implementation of PROC UNIVARIATE with BY

To successfully execute a grouped univariate analysis in SAS, you must utilize the BY statement in conjunction with the procedure call. It is crucial to remember a fundamental prerequisite of the BY statement: the dataset must be sorted by the grouping variable prior to execution. If the data is not sorted, SAS will issue a warning and may produce incorrect or incomplete results, as it processes observations sequentially based on the order in which they appear in the dataset.

The basic structure for running this grouped procedure is exceptionally simple, requiring only the procedure name, the data specification, the BY statement identifying the categorical variable, and the **RUN** command. This minimal syntax instructs SAS to calculate the full array of univariate statistics--including N, Mean, Standard Deviation, Skewness, Kurtosis, and all quartiles--for every numeric variable present in the input dataset, repeating the entire analysis once for each unique value found in the specified grouping variable.

The syntax below illustrates this foundational approach. Note the use of the **NORMAL** option, which requests tests for normality (such as the Shapiro-Wilk test), which is frequently desired when analyzing the shape of the data's distributional characteristics across groups.

```
proc univariate data=my_data normal;
by group_variable;
run;
```

It is important to understand that the output generated by this simple syntax will be extensive, potentially containing dozens of tables if your dataset holds many numeric variables. In the following sections, we will move beyond this abstract syntax and apply this powerful procedure using a concrete example based on sports data.

4. Detailed Example: Setting Up the SAS Dataset

To demonstrate the functionality of grouped univariate analysis, we will utilize a sample dataset containing performance metrics for basketball players. This dataset, named **my_data**, includes three variables: **team** (a categorical variable used for grouping), and two numeric performance variables, **points** and **rebounds**. Our goal is to calculate separate descriptive statistics for player

performance based on which team they belong to, providing insights into team-specific differences in scoring and rebounding capabilities.

The following SAS data step code is used to create and populate this dataset. We define the variables using the **INPUT** statement, specifying the dollar sign (**$**) after **team** to denote it as a character variable, while **points** and **rebounds** are treated as numeric variables. The data lines provide raw performance observations for players across two distinct teams, Team A and Team B.

```
/*create dataset*/
data my_data;
input team $ points rebounds;
datalines;
A 12 8
A 12 8
A 12 8
A 23 9
A 20 12
A 14 7
A 14 7
B 20 2
B 20 5
B 29 4
B 14 7
B 20 2
B 20 2
B 20 5
;
run;

/*view dataset*/
proc print data=my_data;
```


After the data is successfully created and loaded into the SAS session, we confirm its structure using **PROC PRINT**. This step ensures that the observations are correctly input and assigned to their respective teams before proceeding to the analytical procedures. The visual representation below confirms the dataset structure, showing players associated with Team A and Team B, along with their associated points and rebounds values.

Obs	team	points	rebounds
1	A	12	8
2	A	12	8
3	A	12	8
4	A	23	9
5	A	20	12
6	A	14	7
7	A	14	7
8	B	20	2
9	B	20	5
10	B	29	4
11	B	14	7
12	B	20	2
13	B	20	2
14	B	20	5

5. Applying PROC UNIVARIATE by Group

With the data established, we can now proceed to apply the grouped univariate analysis. A crucial preparatory step, often implicitly necessary, is sorting the data by the grouping variable (**team**). Although in some simple datasets the observations might already be in the correct order, relying on this is risky. Best practice dictates using **PROC SORT** immediately before the analysis, especially when using the BY statement. For this example, assuming the dataset is already sorted by the **team** variable (A then B), we execute the **PROC UNIVARIATE** call directly.

The following block of code instructs SAS to calculate descriptive statistics for every numeric variable in **my_data** (**points** and **rebounds**), performing these calculations separately for each level of the **team** variable. Because we did not specify a **VAR** statement yet, the procedure defaults to analyzing all suitable numeric columns available in the dataset.

```
proc univariate data=my_data;
by team;
run;
```

Upon execution, SAS will iterate through the dataset, recognizing the distinct groups defined by the **team** variable. For each unique team, it generates a complete set of univariate output tables. This results in a comprehensive report structure, organized sequentially by the grouping variable.

Specifically, the output generated by the procedure will include the following segments, repeated for every numeric variable:

Descriptive statistics for **points** for team **A**

Descriptive statistics for **rebounds** for team **A**

Descriptive statistics for **points** for team **B**

Descriptive statistics for **rebounds** for team **B**

6. Interpreting the Grouped Output

The output tables generated by the grouped univariate analysis are highly detailed, providing deep insight into the characteristics of each variable within its group context. We must carefully examine the output for both Team A and Team B regarding the **points** variable to identify how the scoring distributions differ between the two groups.

For Team A, the descriptive statistics section provides the mean, median, mode, standard deviation, and variance. These metrics help establish the central tendency and spread of scoring performance. Furthermore, the moments section details the skewness and kurtosis, which are critical for understanding the shape of the data's distributional characteristics--whether the data is symmetrically distributed or heavily tailed.

Examining the output for Team A's points:

The UNIVARIATE Procedure
Variable: points

team=A

Moments			
N	7	Sum Weights	7
Mean	15.2857143	Sum Observations	107
Std Deviation	4.42396073	Variance	19.5714286
Skewness	1.22128564	Kurtosis	-0.0509457
Uncorrected SS	1753	Corrected SS	117.428571
Coeff Variation	28.9417992	Std Error Mean	1.67209999

Basic Statistical Measures			
Location		Variability	
Mean	15.28571	Std Deviation	4.42396
Median	14.00000	Variance	19.57143
Mode	12.00000	Range	11.00000
		Interquartile Range	8.00000

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	9.141627	Pr > |t|	<.0001
Sign	M	3.5	Pr >= |M|	0.0156
Signed Rank	S	14	Pr >= |S|	0.0156

By comparing the corresponding output for Team B (not fully displayed here, but represented by the second image focusing on graphical output), we can draw comparative conclusions. For example, if Team A shows a lower mean but a higher standard deviation than Team B, it implies that Team A's scoring is less consistent, relying on a few high-scoring outliers, while Team B scores more consistently close to its average. The output also includes extreme values and quantile estimates, useful for identifying performance benchmarks within each squad.

A key component of the grouped univariate output is the graphical representation. SAS automatically provides visualizations, such as the box plot and histogram, for each group. These visual aids are powerful tools for quickly assessing differences in median (the center line of the box plot), interquartile range (the length of the box), and the presence of outliers (indicated by stars or circles outside the whiskers).

Quantiles (Definition 5)	
Level	Quantile
100% Max	23
99%	23
95%	23
90%	23
75% Q3	20
50% Median	14
25% Q1	12
10%	12
5%	12
1%	12
0% Min	12

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
12	3	12	3
12	2	14	6
12	1	14	7
14	7	20	5
14	6	23	4

7. Focusing Analysis: Using the VAR Statement

While running **PROC UNIVARIATE** without specifying variables provides a comprehensive overview of all numeric fields, this often results in large, unwieldy output when dealing with wide datasets. In scenarios where the analysis must be concentrated on only one or a small subset of variables--such as focusing solely on scoring performance and ignoring rebounding metrics--the VAR statement becomes indispensable.

The VAR statement must be placed between the **PROC UNIVARIATE** and **RUN** statements, and it takes precedence, restricting the analysis only to the variables listed within it. This significantly streamlines the output, making interpretation faster and reducing computational overhead, especially in production environments dealing with massive datasets.

For example, we can use the following syntax to calculate descriptive statistics only for the **points** variable, grouped by the **team** variable:

```
proc univariate data=my_data;
var points;
by team;
run;
```

This streamlined code generates only two sets of output tables: one for Team A's points and one for Team B's points, excluding the analysis of the **rebounds** variable entirely. Analysts should utilize the VAR statement whenever they have a specific analytical target, ensuring that the resulting report is focused and easier to navigate.

8. Key Considerations and Prerequisites

While Proc Univariate with grouping is powerful, its successful implementation relies on addressing several key technical considerations, primarily concerning data structure and variable types. First and foremost, as mentioned earlier, the integrity of the grouping hinges on the data being correctly sorted by the variable listed in the **BY** statement. If **PROC SORT** is omitted or incorrectly executed, SAS treats sequential non-matching group values as new groups, leading to fragmented and incorrect statistical summaries.

Secondly, the variable specified in the **BY** statement must be categorical, whether character or numeric, with a manageable number of unique levels. While SAS will technically run the procedure on a continuous variable, the resulting output--an analysis for virtually every single unique value--would be statistically meaningless and practically unreadable. The strength of this procedure lies in comparing the distributional characteristics across distinct, predefined groups.

Finally, analysts should be mindful of missing data. If the grouping variable contains missing values, SAS typically treats them as a separate, distinct group for which it calculates statistics. If the variables specified in the VAR statement have missing values, those observations are excluded from the analysis for that specific variable within that group, but the group itself is still analyzed based on the non-missing observations. Understanding how SAS handles these null values is essential for accurate interpretation.

9. Expanding the Analysis Scope

The core functionality demonstrated here--using **Proc Univariate** with the **BY** statement--can be significantly expanded using various options available within the procedure. For example, analysts can use the **OUTPUT** statement to save the calculated statistics (like mean, median, standard deviation) for each group back into a new SAS dataset. This is highly useful for integrating summary statistics directly into reports or using them as input for subsequent modeling stages.

Additionally, advanced users can specify options within **PROC UNIVARIATE** to customize the output, such as requesting specific percentile calculations, suppressing certain tables (like moments or extreme values) to keep the report concise, or adjusting the statistical tests performed (e.g., adding location tests like the T-test).

Feel free to specify as many variables as you'd like in both the VAR statement and the **BY** statement (if multiple grouping levels are needed, nested within a sorted data step) to calculate summary statistics for whichever variables are required for your comparative analysis. Mastering this nested analysis and output management allows for highly efficient and targeted statistical reporting in SAS.

10. Conclusion and Further Resources

The combination of Proc Univariate and the **BY** statement is fundamental to performing effective comparative data exploration in SAS. It provides a robust, standardized way to assess the individual distributional characteristics and summary statistics of numeric variables segmented across crucial categorical dimensions. This methodology is essential for diagnostic checks, baseline comparisons, and producing structured statistical reports across different subgroups.

By ensuring the input data is sorted and by judiciously using the VAR statement, analysts can harness the full power of this procedure, transforming raw data into actionable, group-specific insights.

The following tutorials explain how to perform other common tasks in SAS: