How to Easily Calculate Descriptive Statistics in SAS

Authored by

stats writer

November 21, 2025

Calculating descriptive statistics is a fundamental step in any data analysis workflow, providing essential insights into the characteristics of your dataset. Within the SAS software environment, analysts rely primarily on two powerful procedures to generate these summaries: **PROC MEANS** and **PROC UNIVARIATE**. While **PROC MEANS** offers a fast and flexible way to calculate core metrics like the mean, median, standard deviation, and range, **PROC UNIVARIATE** provides a far more comprehensive suite of statistics, including measures of skewness, kurtosis, and detailed quantile information, suitable for deep exploration of a single variable's distribution. Mastering both procedures is crucial for any data professional working in SAS, allowing for the efficient creation of frequency tables, summary reports, and preliminary data assessments.

Understanding Descriptive Statistics in Data Analysis

Descriptive statistics are numerical values designed to summarize and describe the main features of a collection of information. Rather than making inferences about a larger population, they focus solely on characterizing the sample data currently in hand. These statistics help us address two key questions about our dataset: Where is the center of the data located (central tendency), and how spread out are the values (variability or dispersion)? A strong grasp of these measures is the cornerstone of effective data preparation and modeling. By utilizing procedures like **PROC MEANS** and **PROC UNIVARIATE** in SAS, users can quickly generate these summaries, enabling swift pattern detection and quality checks before proceeding to more complex statistical modeling.

The core components of descriptive statistics generally fall into two categories. Measures of central tendency include the mean (the average value), the median (the middle value when data is ordered), and the mode (the most frequent value). These metrics provide a single value that attempts to describe the typical case within the dataset. Conversely, measures of variability--such as the range, variance, and standard deviation--quantify the extent to which data points diverge from the central tendency. A low standard deviation indicates that data points tend to be close to the mean, while a high standard deviation suggests the data is widely spread out, which can often signal heterogeneity or the presence of outliers that warrant further investigation.

In the context of the SAS system, calculating these descriptive summaries is streamlined through powerful, specialized procedures. While other statistical software packages may require complex programming or coding to achieve comprehensive summaries, SAS simplifies the process using concise commands. For quick, high-level summaries across multiple variables or groups, analysts typically turn to **PROC MEANS**. When the analysis demands a detailed statistical profile of a single numeric variable, including moments, quantiles, and extreme values, **PROC UNIVARIATE** is the preferred choice. Understanding when and how to apply each of these procedures ensures efficient and accurate statistical reporting, a vital skill for any data analyst.

Preparing the Sample Data in SAS

To demonstrate the functionality of both primary procedures--**PROC MEANS** and **PROC UNIVARIATE**--we will utilize a small sample dataset containing fictional performance metrics for three different sports teams. This dataset includes variables representing the team identifier (a categorical variable), the points scored in a recent game (a quantitative variable), and the number of assists recorded (another quantitative variable). This data structure allows us to showcase not only simple summary calculations but also the powerful ability of SAS to generate statistics grouped by a classification variable, such as the team name.

The data step below creates and populates the dataset named **my_data**. This step defines the variables `team` (character, denoted by $), `points` (numeric), and `assists` (numeric). Following the data creation, a simple **PROC PRINT** statement is used to display the contents of the newly created dataset in the output viewer, confirming that the data input was successful and that all observations are correctly recorded. This verification step is a crucial best practice in any SAS programming endeavor, ensuring data integrity before statistical analysis begins.

The following code block shows the exact structure used to create this sample data in the SAS scripting environment. Note the use of the `datalines` statement for inline data entry, a common method for small, illustrative datasets.

```
/*create dataset*/
data my_data;
input team $ points assists;
datalines;
A 10 2
A 17 5
A 17 6
A 18 3
A 15 0
B 10 2
B 14 5
B 13 4
B 29 0
B 25 2
C 12 1
C 30 1
C 34 3
C 12 4
C 11 7
```

```
;
run;

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

Obs	team	points	assists
1	A	10	2
2	A	17	5
3	A	17	6
4	A	18	3
5	A	15	0
6	B	10	2
7	B	14	5
8	B	13	4
9	B	29	0
10	B	25	2
11	C	12	1
12	C	30	1
13	C	34	3
14	C	12	4
15	C	11	7

Method 1: Using PROC MEANS for Efficient Summary Statistics

The PROC MEANS procedure is arguably the most frequently used statistical procedure in SAS. Its primary function is to calculate summary statistics for quantitative variables efficiently. It is designed for speed and flexibility, making it ideal for exploratory data analysis (EDA) where analysts need quick snapshots of data center and spread. By default, **PROC MEANS** generates five fundamental statistics: N (count of non-missing observations), the Mean, the Standard deviation (Std Dev), Minimum, and Maximum values. These statistics provide a comprehensive overview of the variable's distribution across the entire dataset or specified subgroups.

To use PROC MEANS, the basic syntax requires specifying the procedure name and the dataset via the `DATA=` option. Furthermore, the mandatory `VAR` statement is used to list the numeric variables for which the statistics should be calculated. If the `VAR` statement is omitted, **PROC MEANS** will attempt to calculate summary statistics for every numeric variable in the specified

dataset, which can sometimes lead to cluttered or irrelevant output if the dataset is large. Analysts often customize the output using options like `N MEAN MEDIAN STDDEV MAXDEC=2` added to the procedure statement to request specific statistics or formatting enhancements.

The simplicity and efficiency of PROC MEANS make it suitable for initial data quality checks. For instance, quickly reviewing the Minimum and Maximum values can reveal data entry errors, such as negative scores where only positive values are expected. Furthermore, comparing the Mean and Median can offer immediate insights into the symmetry of the data distribution; if the mean is significantly different from the median, it suggests potential skewness or the influence of extreme outliers, prompting the analyst to investigate further using more detailed procedures like **PROC UNIVARIATE**.

Example 1: Basic PROC MEANS Implementation

In our first practical example, we apply PROC MEANS to calculate the summary statistics for the `points` variable across all fifteen observations in the `my_data` dataset. Since we are interested in the overall performance metrics, we do not include a grouping mechanism (like the `CLASS` statement) in this initial run. The concise nature of the syntax highlights the efficiency of SAS for routine statistical tasks.

```
/*calculate summary statistics for points variable*/
proc means data=my_data;
var points;
run;
```

Upon execution, PROC MEANS produces a clear, tabular output summarizing the requested variable. This table is easy to interpret and provides instantaneous descriptive insights. The output confirms that there are 15 total observations (N=15) contributing data to the analysis. The average score, or Mean, is calculated to be 18.2666667. The Minimum score recorded is 10, while the Maximum score is 34. The spread of the data, measured by the Standard deviation (Std Dev), is approximately 7.82, indicating a moderately high variability in points scored across the teams in this sample.

The MEANS Procedure

Analysis Variable : points				
N	Mean	Std Dev	Minimum	Maximum
15	17.8000000	7.8848861	10.0000000	34.0000000

The default output generated by the **PROC MEANS** procedure, as seen above, includes the following five core descriptive statistics, which are automatically calculated when no specific statistical keywords are provided in the procedure statement:

N: The total number of valid observations (non-missing values) used in the calculation.
Mean: The arithmetic average value of the variable.
Std Dev: The Standard deviation, quantifying the amount of variation or dispersion of a set of values.
Minimum: The smallest value recorded in the variable.
Maximum: The largest value recorded in the variable.

Grouping Analysis with PROC MEANS using the CLASS Statement

One of the most valuable features of PROC MEANS is its ability to segment the data and calculate descriptive statistics for defined subgroups, which is achieved using the `CLASS` statement. The `CLASS` statement specifies one or more variables (usually categorical or nominal) by which the input dataset should be divided. When a class variable is included, **PROC MEANS** calculates the requested statistics separately for each unique level of that grouping variable, providing comparative summaries across different categories.

This capability is essential for comparative analysis. For example, if we want to compare the scoring performance of Team A, Team B, and Team C, we need to calculate the Mean, Standard deviation, and range of `points` for each team individually, rather than for the aggregate dataset. The `CLASS team;` statement tells SAS to perform this segmentation before calculating the summary statistics specified in the `VAR` statement. This approach allows analysts to identify which groups contribute most to overall variability and which groups exhibit the highest or lowest central tendency.

The following code snippet demonstrates how to leverage the `CLASS` statement to calculate separate summary statistics for the `points` variable, grouping the output by the `team` variable. This provides a detailed comparison of scoring metrics across the three teams in our dataset, moving beyond the simple overall averages calculated in the previous step.

```
/*calculate summary statistics for points, grouped by team*/
proc means data=my_data;
class team;
var points;
run;
```

The MEANS Procedure

Analysis Variable : points						
team	N Obs	N	Mean	Std Dev	Minimum	Maximum
A	5	5	15.4000000	3.2093613	10.0000000	18.0000000
B	5	5	18.2000000	8.2885463	10.0000000	29.0000000
C	5	5	19.8000000	11.2338773	11.0000000	34.0000000

As evidenced by the output, the table now displays the summary statistics for the `points` variable segmented by each unique level of the `team` variable. We can immediately observe differences: Team A has an average of 15.4 points, Team B averages 18.2 points, and Team C averages 19.8 points. Furthermore, the variability, indicated by the Standard deviation, is highest for Team C (10.97), suggesting their performance is the most inconsistent compared to the other two teams. This grouped analysis is a powerful feature for preliminary comparative evaluation in SAS.

Method 2: Using PROC UNIVARIATE for Detailed Distribution Analysis

While PROC MEANS provides essential summary statistics, the PROC UNIVARIATE procedure is designed for a much more comprehensive, in-depth analysis of a single numeric variable's distribution. As its name implies, it focuses on univariate analysis, generating a vast array of descriptive metrics, graphical output (by default, in some SAS configurations), and tests for normality, which are generally not available through **PROC MEANS** without additional options. This procedure is crucial when an analyst needs to understand the fine details of a variable's shape, spread, and the location of its quantiles.

When executed, PROC UNIVARIATE generates several distinct tables, providing measures of central tendency (including the Mean, Median, and mode), moments (such as variance, skewness, and kurtosis), basic measures of location (min/max), and detailed quantile statistics (including quartiles and percentiles). Unlike **PROC MEANS**, which summarizes the data into a single, compact table, **PROC UNIVARIATE** spreads the output across multiple sections, offering a holistic view of the variable's probabilistic behavior.

The primary benefit of using PROC UNIVARIATE stems from the inclusion of higher-order statistics and detailed quantiles. Measures of skewness indicate the degree of asymmetry in the distribution, while kurtosis measures the "tailedness" of the distribution. These values are fundamental for determining if parametric assumptions for subsequent statistical tests (like t-tests or ANOVA) are met. Furthermore, the Interquartile Range (IQR), calculated from the 25th and 75th percentiles, provides a robust measure of spread that is less sensitive to extreme outliers than the

Standard deviation.

Example 2: Detailed Analysis using PROC UNIVARIATE

To illustrate the depth of analysis provided by PROC UNIVARIATE, we will again focus on the `points` variable in the `my_data` dataset. The basic syntax mirrors that of **PROC MEANS**, requiring only the procedure name, the dataset reference, and the `VAR` statement specifying the target numeric variable.

```
/*calculate detailed descriptive statistics for points variable*/
proc univariate data=my_data;
var points;
run;
```

The resulting output from PROC UNIVARIATE is significantly more detailed than the output from **PROC MEANS**. It is typically divided into sections. The first section provides basic statistics and moments: the Mean (18.2667), Median (15), and the Standard deviation (7.8247). It also includes the variance, skewness (0.916), and kurtosis (0.177). The positive skewness suggests that the data distribution has a longer tail extending towards higher point values, indicating a few high scores are pulling the mean away from the median.

The UNIVARIATE Procedure
Variable: points

Moments			
N	15	Sum Weights	15
Mean	17.8	Sum Observations	267
Std Deviation	7.88488608	Variance	62.1714286
Skewness	1.00931793	Kurtosis	-0.2991564
Uncorrected SS	5623	Corrected SS	870.4
Coeff Variation	44.2971128	Std Error Mean	2.03586883

Basic Statistical Measures			
Location		Variability	
Mean	17.80000	Std Deviation	7.88489
Median	15.00000	Variance	62.17143
Mode	10.00000	Range	24.00000
		Interquartile Range	13.00000

Note: The mode displayed is the smallest of 3 modes with a count of 2.

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	8.743196	Pr > |t|	<.0001
Sign	M	7.5	Pr >= |M|	<.0001
Signed Rank	S	60	Pr >= |S|	<.0001

The second main section focuses on extreme observations and detailed quantiles. The quantiles section is particularly important, as it lists specific percentiles, including the quartiles (25th, 50th/Median, 75th percentile). For instance, knowing the 75th percentile (Q3) of 25 means that 75% of the points scored are less than or equal to 25. The interquartile range (IQR), which is the difference between Q3 and Q1, is calculated to be 10.5 (25 - 14.5). These comprehensive statistics allow for a much deeper understanding of the score distribution than simple minimum and maximum values alone.

Quantiles (Definition 5)	
Level	Quantile
100% Max	34
99%	34
95%	34
90%	30
75% Q3	25
50% Median	15
25% Q1	12
10%	10
5%	10
1%	10
0% Min	10

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
10	6	18	4
10	1	25	10
11	15	29	9
12	14	30	12
12	11	34	13

In summary, the **PROC UNIVARIATE** procedure calculates an exhaustive set of descriptive statistics for the `points` variable, offering far more than central tendency and dispersion. This output includes:

Measures of Location: Mean, Median, Mode.
Measures of Variability: Standard deviation, Variance, Range, and Interquartile Range.
Measures of Shape: Skewness and Kurtosis.
Detailed Quantiles: Percentiles, Quartiles, and Extreme Values.

Applying Grouped Analysis with PROC UNIVARIATE

Just like **PROC MEANS**, PROC UNIVARIATE can utilize the `CLASS` statement to perform segmented analysis, calculating the full suite of detailed descriptive statistics for each unique group defined by a categorical variable. This is extremely useful when the analyst needs to compare the distributional shape (e.g., skewness and kurtosis) or the quantile structure of a variable across different groups, such as comparing the consistency and shape of scoring distributions across the

three different teams.

When the `CLASS` statement is included in the **PROC UNIVARIATE** code, the procedure executes the full univariate analysis routine for every level of the classification variable. For our example, running this code will produce three separate, extensive output sections--one for Team A, one for Team B, and one for Team C--each containing the moments, basic stats, and quantiles specific to that team's data subset.

This approach allows for a granular comparison. For instance, we might find that while Team A's scores are normally distributed (low skewness/kurtosis), Team C's scores are highly skewed due to one or two exceptionally high scores. Such insights are crucial for understanding performance heterogeneity and informing subsequent statistical modeling decisions.

The following code demonstrates the use of the `CLASS` statement within **PROC UNIVARIATE**:

```
/*calculate detailed descriptive statistics for points, grouped by team*/
proc univariate data=my_data;
class team;
var points;
run;
```

Executing this command produces three distinct groups of output tables. Each output block displays the detailed descriptive statistics--including moments, quantiles, and extreme observations--for the `points` variable, separated by each of the unique `team` values. Reviewing these three separate reports allows the analyst to conduct a thorough comparative assessment of the underlying score distributions for each subgroup within the `my_data` dataset.

Choosing Between PROC MEANS and PROC UNIVARIATE

The choice between PROC MEANS and PROC UNIVARIATE depends entirely on the analytical objective and the required level of detail. If the goal is rapid generation of core summaries for multiple variables, ensuring data validity, or creating compact summary reports, **PROC MEANS** is generally the more efficient tool. It focuses on the fundamental measures of central tendency (like the Mean) and dispersion (like the Standard deviation) across potentially many variables simultaneously.

Conversely, when the analysis requires an exhaustive examination of the distribution of a single variable--especially when checking for normality assumptions, identifying specific outliers, or calculating detailed percentile ranks--PROC UNIVARIATE is indispensable. It provides the crucial measures of skewness and kurtosis that describe the shape of the data, information vital for advanced statistical modeling. Although more verbose in its output, the depth of analysis provided

by **PROC UNIVARIATE** justifies its use in exploratory and diagnostic phases of data analysis.

In practice, many SAS professionals use the two procedures synergistically. **PROC MEANS** serves as the initial, high-level filter to quickly check data across all numeric variables. If a variable shows unusual characteristics (e.g., a large difference between the mean and median, or an unexpectedly high standard deviation), the analyst then zooms in on that specific variable using **PROC UNIVARIATE** to diagnose the distributional problem, determine the extent of skewness, and identify the exact percentile locations of potential outliers. Both are essential components of the SAS statistical toolbox for descriptive analysis.

Further SAS Analysis Tutorials

Once you have mastered the calculation of basic descriptive statistics using **PROC MEANS** and **PROC UNIVARIATE**, you can proceed to more complex data manipulation and statistical modeling tasks in the SAS environment. Understanding the distribution of your variables is a prerequisite for virtually all subsequent analyses, including correlation, regression, and hypothesis testing.

The following tutorials explain how to perform other common tasks in SAS, building upon the foundational knowledge of data summarization demonstrated here: