Learning Quartiles with SAS: A Step-by-Step Guide

Authored by

Mohammed looti

May 17, 2026

Introduction to Quartiles and Their Importance

In the comprehensive field of descriptive statistics, quartiles serve as essential tools for segmenting a numerical dataset into four equally sized parts. These measures, which are specific types of quantiles, offer crucial insight into the internal structure and spread of observations. By dividing the data based on position, quartiles help analysts move beyond simple averages to truly understand the central tendency, variability, and asymmetry within the data distribution. Specifically, they pinpoint the boundary points that define the lowest 25%, the middle 50%, and the upper 25% of the data.

Mastering the calculation and interpretation of quartiles is fundamental across numerous analytical disciplines. Whether you are assessing student academic performance, tracking fluctuations in market trends, or monitoring industrial process efficiency, quartiles provide a statistically robust framework. By identifying the 25th, 50th (the median), and 75th percentiles, they offer a more nuanced view than measures like the mean, particularly when dealing with data that is heavily skewed. Furthermore, quartiles are indispensable for constructing effective visualization tools, such as box plots, and for calculating the interquartile range (IQR), a highly resistant measure used for identifying potential outliers.

This definitive guide focuses on the practical application of calculating quartiles utilizing SAS, one of the industry's leading statistical software packages. We will detail the necessary syntax, walk through hands-on examples, and explain how to accurately interpret the results, covering both aggregated and grouped data analysis. The primary and most versatile procedure for obtaining these statistics in SAS is `PROC UNIVARIATE`, designed for comprehensive univariate analysis.

```
/*calculate quartile values for variable called var1*/
proc univariate data=original_data;
var var1;
output out=quartile_data
pctlpts = 25 50 75
pctlpre = Q_;
run;
```

Understanding the `PROC UNIVARIATE` Statement for Quartile Calculation

The `PROC UNIVARIATE` procedure is known for its extensive capability in generating a vast array of descriptive statistics, encompassing everything from moments and extreme values to the specific quantiles we seek. To successfully calculate quartiles using this procedure, several core statements must be included, each serving a distinct and critical function in the analysis pipeline.

The initial statement, `PROC UNIVARIATE` followed by the `DATA` option, explicitly specifies the input dataset containing the raw observations. Subsequently, the `VAR` statement is mandatory, as it designates the specific numeric variable or list of variables that the procedure should analyze. Without the correct variable designation, SAS cannot perform the necessary calculations.

A crucial step for extracting results for further programmatic use is employing the `OUTPUT OUT` statement. This command directs SAS to store the computed quartile values into a newly created dataset, which we named `quartile_data` in our generic example. This output dataset structure is essential for integration into subsequent SAS data steps or procedures.

The precise percentile points are defined using the `PCTLPTs` statement. For standard quartiles, the values 25, 50, and 75 are specified, corresponding precisely to the First Quartile (Q1), the Median (Q2), and the Third Quartile (Q3). To ensure clarity and organization in the output dataset, the `PCTLPRE` statement is used to assign a common prefix (e.g., Q_) to the newly generated variables (e.g., Q_25, Q_50, Q_75), making them instantly recognizable as calculated quartile boundaries.

Setting Up Your Data in SAS

Before any statistical computation can commence in SAS, a structured dataset must be prepared and loaded into the session. For instructional purposes, we will utilize the DATA step to manually input a small sample dataset. This method is highly efficient for demonstration or when dealing with limited amounts of data that need to be defined directly within the SAS program editor.

Our sample dataset, logically named `original_data`, is designed to analyze team performance. It includes two essential variables: `team`, a character variable distinguishing between Team A and Team B, and `points`, a numeric variable recording the score achieved. The structure begins with the INPUT statement, which defines these variables and their types (the dollar sign `$` denotes a character variable). The data itself is provided immediately following the DATALINES statement.

To ensure the data has been correctly read and structured, we execute a validation step. Following the data entry, the `PROC PRINT` command is utilized to display the contents of the newly created `original_data` table. This confirmation step is crucial for maintaining data integrity before proceeding to any statistical calculations, providing visual assurance that the input data is ready for analysis.

```
/*create dataset*/
data original_data;
input team $ points;
datalines;
A 12
A 15
```

```
A 16
A 21
A 22
A 25
A 29
A 31
B 16
B 22
B 25
B 29
B 30
B 31
B 33
B 38
;
run;

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

Obs	team	points
1	A	12
2	A	15
3	A	16
4	A	21
5	A	22
6	A	25
7	A	29
8	A	31
9	B	16
10	B	22
11	B	25
12	B	29
13	B	30
14	B	31
15	B	33
16	B	38

Calculating Quartiles for a Single Variable

Once the dataset is properly loaded into SAS, we can execute the core analysis procedure to determine the quartiles for the aggregated scores. Our objective is to calculate these positional statistics for the `points` variable across all teams combined. This involves implementing the `PROC UNIVARIATE` statement, targeting only the variable of interest using the `VAR points;` line.

The following block of code is structured to calculate the 25th, 50th, and 75th percentiles--the essential components of the quartile analysis. By utilizing the `OUTPUT OUT=quartile_data` option, the results are captured and stored efficiently in a new SAS dataset, rather than simply being displayed in the output window. The `PCTLPRE = Q_` prefix ensures that the resulting variables (Q_25, Q_50, Q_75) are clearly identifiable within this new dataset as the calculated quartile boundaries.

Following the successful execution of `PROC UNIVARIATE`, a subsequent `PROC PRINT` command is used to display the contents of the `quartile_data` dataset. This immediate output allows the analyst to review the computed quartile values (Q1, Q2, and Q3) without needing to scroll through the full univariate analysis report, providing a clean and focused view of the required statistics.

```
/*calculate quartile values for points*/
proc univariate data=original_data;
var points;
output out=quartile_data
pctlpts = 25 50 75
pctlpre = Q_;
run;

/*view quartiles for points*/
proc print data=quartile_data;
```

Obs	Q_25	Q_50	Q_75
1	18.5	25	30.5

Interpreting the Quartile Output

The numerical output generated by the `PROC UNIVARIATE` procedure is highly precise and requires careful interpretation to extract meaningful conclusions about the data's central tendency and statistical spread. Understanding what each quartile represents is key to leveraging this component

of descriptive statistics.

First Quartile (Q1): Representing the 25th percentile, Q1 is the boundary value below which 25% of all observations fall. In the context of our example, a Q1 value of **18.5** signifies that a quarter of the recorded team scores were 18.5 points or less.

Second Quartile (Q2): This is equivalent to the median and the 50th percentile. Q2 centrally divides the entire dataset into two equal halves. A calculated Q2 of **25** means that 50% of the scores are 25 points or below, and equally, 50% are 25 points or above.

Third Quartile (Q3): Known as the 75th percentile, Q3 marks the threshold below which 75% of the data points reside. Our result shows a Q3 of **30.5**, meaning three-quarters of the team scores were 30.5 points or less.

These three quartiles, when combined with the minimum and maximum values of the dataset, constitute the fundamental five-number summary. This summary provides a rapid and comprehensive overview of the data distribution. Furthermore, the interquartile range (IQR), calculated simply as Q3 minus Q1, offers a highly robust measure of statistical dispersion, defining the range of the middle 50% of the data while remaining less susceptible to the influence of extreme values or outliers.

Calculating Grouped Quartiles

In real-world data analysis, it is frequently necessary to calculate descriptive statistics for defined subgroups within a larger dataset. This comparative analysis is essential for understanding heterogeneity. Using our example, we may want to calculate the specific quartiles for the `points` variable for Team A and Team B independently. SAS simplifies this process through the powerful `BY statement`.

By integrating `BY team` to the `PROC UNIVARIATE` block, SAS is instructed to partition the data based on the unique values of the `team` variable. The procedure then performs the entire quartile calculation routine separately for each distinct group. This segmentation automatically produces individual sets of quartiles, enabling a detailed comparative assessment of the performance or characteristics between the subgroups.

A crucial technical consideration when using the `BY statement` is the requirement that the input dataset must be sorted by the grouping variable (`team`). While the sample data used here appears pre-sorted, in production environments, analysts must typically precede the `PROC UNIVARIATE` step with a `PROC SORT` command (e.g., PROC SORT DATA=original_data; BY team; RUN;) to ensure that the grouping mechanism functions correctly and produces valid comparative statistics.

```
/*calculate quartile values for points*/
proc univariate data=original_data;
var points;
by team;
output out=quartile_data_grouped
pctlpts = 25 50 75
pctlpre = Q_;
run;

/*view quartiles for points*/
proc print data=quartile_data_grouped;
```

Obs	team	Q_25	Q_50	Q_75
1	A	15.5	21.5	27
2	B	23.5	29.5	32

Analyzing Grouped Quartile Output

When you execute the code with the `BY statement`, the resulting output dataset provides segmented results, presenting the quartile statistics for each category independently. This structure is immensely beneficial because it facilitates a direct and clear comparison of the data distribution parameters across different groups, eliminating the need for manual data filtering or splitting.

Examining the output table above, we can directly compare the Q1, Median (Q2), and Q3 values for Team A versus Team B. For example, if Team A exhibits a lower median score but a significantly larger interquartile range compared to Team B, it suggests that Team A's performance is more dispersed or variable, whereas Team B's scores are tightly clustered around their central measure. Such fine-grained insights are fundamental in fields like business intelligence, medical research, and quality control, where group-specific analysis is paramount for informed decision-making. Utilizing SAS for this level of descriptive statistics ensures accuracy and efficiency.

Conclusion and Further Exploration

Calculating quartiles in SAS is a core competency in statistical analysis, easily achieved using the robust `PROC UNIVARIATE` procedure. By mastering the output generation using `PCTLPTs` and the segmentation capabilities provided by the `BY statement`, analysts can rapidly summarize the

spread and central tendency of their data, thereby gaining critical insights into the underlying data distribution and identifying potential structural issues or outliers.

Whether your task requires assessing the aggregated characteristics of a single variable or conducting sophisticated comparative analysis between subgroups, SAS provides a streamlined and reliable statistical environment. Developing proficiency in these fundamental techniques--especially the accurate interpretation of Q1, Median (Q2), and Q3--is essential for any data professional aiming to produce thorough and actionable data summaries.

For those seeking to extend their analysis beyond basic quartile calculation, `PROC UNIVARIATE` offers a wealth of additional options. We highly recommend investigating features such as graphical output generation (including histograms and box plots), calculation of custom percentiles, or the application of formal normality tests. These advanced explorations can provide even richer, multi-faceted insights into the characteristics of your dataset and enhance the overall depth of your statistical reporting.