How to Easily Calculate Group Means in SAS

Authored by

stats writer

December 1, 2025

To perform effective data analysis, analysts frequently need to summarize numerical data based on specific categories or groups within their dataset. Calculating the arithmetic mean by group is perhaps the most fundamental statistical operation in this context, providing a clear central tendency measure for subsets of observations. In SAS, the leading statistical software suite, this calculation is straightforward but can be achieved through multiple powerful procedures. The most common tool for this task is the **MEANS procedure**, which is specifically designed for generating summary statistics, including the mean, standard deviation, and count, across various grouping levels.

However, for users familiar with database querying languages, the **PROC SQL procedure** offers a highly flexible and powerful alternative. By utilizing the `GROUP BY` clause within PROC SQL, users can aggregate data and calculate means (or any other aggregate function) based on one or more grouping variables, integrating seamlessly with SQL syntax conventions. This article explores both the conceptual framework and practical implementation of these methods, focusing particularly on the robust functionality provided by PROC SQL for achieving highly customized grouped summaries.

The Importance of Grouped Summary Statistics in Data Analysis

Summary statistics are the foundation of exploratory data analysis (EDA). When we calculate the overall mean of a variable, we get a single number that represents the entire population or dataset. While useful, this single value often masks critical underlying patterns or disparities that exist between different subpopulations. For example, knowing the average salary of all employees in a company is less informative than knowing the average salary categorized by department, location, or years of experience. Grouped summary statistics allow us to dissect the data, revealing nuanced insights and confirming or refuting initial hypotheses about the data distribution.

The ability to calculate the **mean by group** is crucial in fields ranging from market research and finance to epidemiology and quality control. In finance, one might calculate the average return grouped by stock sector; in healthcare, the average recovery time grouped by treatment protocol. SAS, designed specifically for rigorous statistical processing, provides efficient algorithms to handle these grouped calculations even on very large datasets. Procedures like PROC MEANS and PROC SQL manage the internal grouping, sorting, and calculation steps automatically, saving the analyst significant manual effort.

When preparing to calculate means by group, it is essential to first identify the categorical variables that will serve as the grouping factors (sometimes called classification variables) and the continuous variable for which the mean will be calculated (the analysis variable). Ensuring that the data is clean and that the grouping variables are correctly formatted is a prerequisite for generating accurate and reliable summary statistics. The following sections demonstrate how two primary

SAS procedures handle this grouping task.

The Essential Tool: Understanding the MEANS Procedure in SAS

The **MEANS procedure** is the traditional and perhaps most streamlined method in SAS for producing descriptive statistics for groups of observations. It is highly optimized for this purpose and offers a vast array of options for formatting output and selecting specific statistics. To instruct PROC MEANS to calculate the mean for specific subsets, you typically use the `CLASS` statement.

If you were to use PROC MEANS instead of PROC SQL, the basic syntax would look like this:

```
PROC MEANS DATA=my_data;
```

`CLASS var1;` (Specifies the grouping variable)

`VAR var2;` (Specifies the variable to calculate the mean on)

`OUTPUT OUT=summary_table MEAN=mean_var2;` (Saves the results to a new dataset)

```
RUN;
```

While PROC MEANS is powerful and handles internal sorting efficiently, many analysts prefer the highly readable and standardized syntax of SQL, especially when the goal is a simple aggregation (like calculating only the mean) or when integrating the aggregation step into a larger data manipulation query. This preference leads us to the increasingly popular use of PROC SQL for grouped statistical calculations.

Alternative Approach: Leveraging PROC SQL for Group Aggregation

The PROC SQL procedure provides a standardized, ANSI-compliant SQL interface within the SAS environment. When calculating the mean by group using SQL, the core mechanism is the `GROUP BY` clause, which partitions the data into subsets based on the values of one or more non-aggregated columns. Once partitioned, the aggregate function, `MEAN()`, is applied independently to the analysis variable within each subset. This approach is highly intuitive for anyone with a background in relational databases.

Method 1: Calculate Mean by One Group

When calculating the mean based on a single categorical variable, the structure is straightforward. We select the grouping variable, apply the `MEAN()` function to the variable of interest, and specify the grouping variable in the `GROUP BY` clause. The following code snippet illustrates how this is performed, calculating the mean of `var2` for every distinct value found in `var1`.

```
proc sql;
select var1, mean(var2) as mean_var2
from my_data
group by var1;
quit;
```

Method 2: Calculate Mean by Multiple Groups

If the requirement is to calculate the mean based on the interaction of two or more variables--for instance, grouping players by both 'Team' and 'Position'--the `GROUP BY` clause is simply extended to include all classification variables. This creates a more detailed level of aggregation, where the mean is calculated for every unique combination of the specified grouping variables. In the example below, the mean of `var3` is calculated for every intersection of `var1` and `var2`.

```
proc sql;
select var1, var2, mean(var3) as mean_var3
from my_data
group by var1, var2;
quit;
```

Setting Up the Data Environment (Dataset Creation)

To demonstrate the practical application of these methods, we will use a small sample dataset named `my_data`. This dataset simulates basketball statistics, containing observations classified by `team` (A or B) and `position` (Guard or Forward), along with the numerical variable `points` scored by each player. This setup provides two distinct categorical variables that can be used for single or multi-level grouping. We create this data using the standard SAS `DATA step` and the `DATALINES` statement.

The data generation code below initializes the dataset, and a subsequent `PROC PRINT` statement is used to display the contents, ensuring data integrity before proceeding with the statistical calculations. Understanding the input data is paramount, as the resulting means will directly reflect the structure and values contained within these observations.

```
/*create dataset*/
data my_data;
input team $ position $ points;
datalines;
A Guard 15
A Guard 12
```

A Guard 29
A Forward 13
A Forward 9
A Forward 16
B Guard 25
B Guard 20
B Guard 34
B Forward 19
B Forward 3
B Forward 8
;
run;

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

After executing the data step, the resulting dataset looks exactly as defined, providing the basis for our group mean calculations.

Obs	team	position	points
1	A	Guard	15
2	A	Guard	12
3	A	Guard	29
4	A	Forward	13
5	A	Forward	9
6	A	Forward	16
7	B	Guard	25
8	B	Guard	20
9	B	Guard	34
10	B	Forward	19
11	B	Forward	3
12	B	Forward	8

Technique 1: Calculating the Mean Based on a Single Grouping Variable

Our first example focuses on calculating the average points scored, grouped solely by the `team` variable. This single-group aggregation provides an overall comparison between Team A and

Team B regarding their scoring output. The query below uses PROC SQL to select the `team` variable and apply the `MEAN()` function to the `points` variable, aliasing the result as `mean_points` for clarity.

The output of this calculation is a concise summary table, showing one row for each unique team. This is highly efficient for comparisons where the internal structure (like position) is temporarily irrelevant, focusing only on the high-level performance metric.

```
/*calculate mean of points by team*/
proc sql;
select team, mean(points) as mean_points
from my_data
group by team;
quit;
```

Analyzing the Results of Single-Group Aggregation

Upon execution, SAS generates a result set that clearly distinguishes the average performance of Team A from Team B. This direct comparison is a powerful outcome of grouped statistical procedures.

team	mean_points
A	15.66667
B	18.16667

From the output table, we can immediately ascertain the following statistics:

Players on **Team A** scored an average of **15.66667** points.

Players on **Team B** scored an average of **18.16667** points.

This initial analysis suggests that, overall, Team B has a slightly higher average scoring output than Team A. However, this aggregated view might still be too coarse. To gain a deeper understanding, particularly regarding how different roles contribute to these team totals, we must introduce a second grouping factor. The next section demonstrates how to achieve this more granular level of analysis using multiple grouping variables.

Technique 2: Calculating the Mean Across Multiple Categorical Variables

Often, the most meaningful statistical insights come from analyzing the interaction between two or more factors. In our basketball example, understanding the average points scored requires distinguishing between the scoring roles (Guard vs. Forward) within each team. This requires a two-level aggregation, grouping first by `team` and then by `position`.

To perform this multi-group calculation, we modify our PROC SQL statement to include both `team` and `position` in the `SELECT` clause and correspondingly list both variables in the `GROUP BY` clause. The PROC SQL engine will then calculate the mean points for the four unique combinations: A/Guard, A/Forward, B/Guard, and B/Forward.

```
/*calculate mean of points, grouped by team and position*/
proc sql;
select team, position, mean(points) as mean_points
from my_data
group by team, position;
quit;
```

Interpreting Multi-Group Summary Statistics

The output generated by the dual grouping provides a much richer set of statistics, revealing the internal structure of the teams' performance.

team	position	mean_points
A	Forward	12.66667
A	Guard	18.66667
B	Forward	10
B	Guard	26.33333

By examining these results, we can draw more granular conclusions:

Team A, Guards: Average points = 18.66667

Team A, Forwards: Average points = 12.66667

Team B, Guards: Average points = 26.33333

Team B, Forwards: Average points = 10.00000

This detailed breakdown shows a significant difference in scoring contribution. Team B's overall higher mean (18.16667) is largely driven by its Guards, who score substantially higher than both Team A's Guards (26.33 vs 18.66) and Team B's own Forwards. Conversely, Team A's Forwards outperform Team B's Forwards (12.66 vs 10.00). This exemplifies why grouped analysis is critical: it isolates the source of variation, providing actionable insights that the overall team mean could never convey.

Choosing Between PROC MEANS and PROC SQL

While this tutorial used PROC SQL exclusively, analysts in SAS often debate which procedure is superior for grouped summary statistics. Both PROC MEANS and PROC SQL are highly effective, but they excel in different areas.

PROC MEANS is generally preferred when the primary goal is generating a high volume of statistical metrics (e.g., N, Max, Min, Standard Deviation, Quartiles) in addition to the mean, or when creating output datasets with specific statistical keywords. It is also often faster for very large, simple aggregation tasks because it is highly specialized.

PROC SQL, on the other hand, is the optimal choice when:

The analyst is already working within a SQL workflow.

Complex filtering (using `WHERE`) or joining multiple tables is required before or during aggregation.

Only one or two specific aggregate functions (like `MEAN()` or `COUNT()`) are needed, and the output format needs to be clean and minimal.

Conclusion and Further SAS Capabilities

Calculating the mean by group is a core skill for any data analysis performed in SAS. Whether you choose the dedicated power of the **MEANS procedure** or the flexible, query-based approach of **PROC SQL**, SAS provides robust methods for deriving these essential summary statistics. The ability to switch seamlessly between single-variable and multi-variable grouping allows analysts to move from broad, high-level comparisons to highly specific, granular performance metrics, thereby enhancing the quality and depth of their statistical reporting.

For tasks beyond simple means, SAS also offers procedures such as PROC SUMMARY (a non-outputting equivalent of PROC MEANS, often used when creating subsequent data steps) and PROC TABULATE (for generating highly customized, presentation-quality summary tables). Mastering grouped summary statistics using both procedural and SQL methods is a fundamental step toward maximizing your productivity in the SAS environment.

You can use the following methods to calculate the mean of values by group in SAS:

Method 1: Calculate Mean by One Group

```
proc sql;
select var1, mean(var2) as mean_var2
from my_data
group by var1;
quit;
```

Method 2: Calculate Mean by Multiple Groups

```
proc sql;
select var1, var2, mean(var3) as mean_var3
from my_data
group by var1, var2;
quit;
```

The following examples show how to use each method with the following dataset in SAS:

```
/*create dataset*/
data my_data;
input team $ position $ points;
datalines;
A Guard 15
A Guard 12
A Guard 29
A Forward 13
A Forward 9
A Forward 16
B Guard 25
B Guard 20
B Guard 34
B Forward 19
B Forward 3
B Forward 8
;
run;

/*view dataset*/
```

proc print data=my_data;

Obs	team	position	points
1	A	Guard	15
2	A	Guard	12
3	A	Guard	29
4	A	Forward	13
5	A	Forward	9
6	A	Forward	16
7	B	Guard	25
8	B	Guard	20
9	B	Guard	34
10	B	Forward	19
11	B	Forward	3
12	B	Forward	8

Example 1: Calculate Mean by One Group

The following code shows how to calculate the mean of points by team:

```
/*calculate mean of points by team*/
proc sql;
select team, mean(points) as mean_points
from my_data
group by team;
quit;
```

team	mean_points
A	15.66667
B	18.16667

From the output we can see that players on team A scored a mean of **15.66667** points and players on team B scored a mean of **18.16667** points.

Example 2: Calculate Mean by Multiple Groups

The following code shows how to calculate the mean of points, group by team and position:

```
/*calculate mean of points, grouped by team and position*/
proc sql;
select team, position, mean(points) as mean_points
from my_data
group by team, position;
quit;
```

team	position	mean_points
A	Forward	12.66667
A	Guard	18.66667
B	Forward	10
B	Guard	26.33333

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