

Learning Weighted Averages in SAS: A Step-by-Step Guide

Authored by
Mohammed loot

October 31, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning Weighted Averages in SAS: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=7358>

Understanding Weighted Averages and Their Importance

In data analysis, the simple arithmetic mean often fails to capture the true central tendency when observations have varying levels of importance. This is where the concept of a [weighted average](#) becomes essential. Unlike the standard average, which treats all values equally, the weighted average assigns a specific multiplier, or "weight," to each data point, reflecting its relative significance. For instance, in finance, returns are often weighted by portfolio allocation, and in statistical surveys, samples might be weighted by their frequency or reliability. Using the correct method ensures that your analytical results accurately reflect the underlying data structure and its real-world implications.

Calculating a weighted average involves two primary steps: multiplying each value by its corresponding weight, summing these products, and then dividing that total by the sum of all weights. This process ensures that data points with larger weights exert a greater influence on the final result, providing a more robust and contextually relevant measure of central tendency. The power and flexibility of [SAS](#), particularly through its [PROC SQL](#) procedure, make this calculation straightforward and highly efficient, whether you are dealing with a simple dataset or complex grouped data.

The following two methods demonstrate how to effectively calculate weighted averages using the robust capabilities available within the [SAS](#) environment.

Method 1: Calculating a Simple Weighted Average

The most straightforward calculation involves finding a single weighted average across an entire [dataset](#). This approach is ideal when you need a summary statistic for a variable where a specific column dictates the importance or frequency of each observation. We utilize the [PROC SQL](#) procedure, which provides standard SQL capabilities within the SAS environment, making the syntax familiar to many data analysts.

The core of the calculation relies on the aggregate function `SUM()`. We calculate the numerator by summing the products of the `weight` variable and the `value` variable. The denominator is simply the sum of all weights. The resulting value is then assigned a clear alias, such as `weighted_average`, for easy retrieval and use in subsequent analysis. This method produces a single scalar output representing the overall weighted mean of the specified variable.

```
proc sql;  
create table new_data as  
select sum(weight * value) / sum(weight) as weighted_average  
from original_data;  
quit;
```

Method 2: Calculating Weighted Average by Group

Often, analysts need to calculate weighted metrics not just for the entire dataset, but separately for different categories or segments. This technique is known as calculating a weighted average by group. This is particularly useful in business intelligence or statistical reporting where metrics need to be segmented by attributes like region, department, or product type. By introducing the `GROUP BY` clause into the [PROC SQL](#) statement, we instruct [SAS](#) to perform the summation and division independently for each distinct value found in the specified [grouping variable](#).

In this advanced application, the structure remains largely the same as Method 1, but we include the [grouping variable](#) (e.g., `sales_rep`) in the `SELECT` statement. The crucial addition is the `GROUP BY` clause, which partitions the data before the aggregate calculations are executed. The resulting output table will contain one row for each unique group, along with its specific weighted average, providing segmented insights essential for detailed reporting and performance monitoring.

```
proc sql;
create table new_data as
select grouping_variable,
sum(weight * value) / sum(weight) as weighted_average
from original_data
group by grouping_variable;
quit;
```

Setting Up the Sample Dataset for Demonstration

To illustrate the practical application of these two methods, we will use a sample [dataset](#) containing sales information. This dataset includes a categorical variable (`sales_rep`) and two numerical variables (`price` and `amount`). In our examples, we will treat `price` as the variable whose average we seek, and `amount` as the weight, representing the volume or importance of that price observation.

The following code block demonstrates how to create this sample data in [SAS](#) using a simple `DATA` step and `DATALINES`. This step is necessary to ensure reproducibility and provide a solid foundation for the subsequent [PROC SQL](#) calculations. The data represents sales transactions recorded by two different sales representatives, A and B.

```
/*create dataset*/
data original_data;
input sales_rep $ price amount;
datalines;
```

```
A 8 1
A 5 3
A 6 2
B 7 2
B 12 5
B 14 4
;
run;

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

Upon running the print procedure, we confirm the structure of our source data, `original_data`, which will be the basis for both weighted average calculations.

Obs	sales_rep	price	amount
1	A	8	1
2	A	5	3
3	A	6	2
4	B	7	2
5	B	12	5
6	B	14	4

Example 1: Calculating the Overall Weighted Average

In this first demonstration, we apply Method 1 to determine the overall weighted average price across all sales representatives (A and B combined). Here, the variable we are averaging is price, and the corresponding weight is amount. This calculation yields a single, representative price that accounts for the total volume sold at each price point.

We execute the [PROC SQL](#) statement exactly as defined previously, replacing the generic variable names `value` and `weight` with our specific column names, `price` and `amount`. The resulting table, `new_data`, will hold only one value: the calculated [weighted average](#).

```
/*calculate weighted average of price*/
proc sql;
create table new_data as
select sum(amount * price) / sum(amount) as weighted_average
```

```
from original_data;  
quit;  
  
/*view weighted average of price*/  
proc print data=new_data;
```

The execution of this code confirms that the overall weighted average price, factoring in the volume of sales, is approximately 9.71. This result is significantly influenced by the higher prices associated with larger amounts in the dataset, demonstrating why weighting is crucial when analyzing volumetric data.

Obs	weighted_average
1	9.70588

The final calculated [weighted average](#) of price turns out to be **9.70588**.

Example 2: Calculating Weighted Average by Sales Representative

In our final example, we apply Method 2 to segment the data and calculate the weighted average price separately for each `sales_rep`. This provides granular insights into which representatives are achieving higher average prices based on their sales volume. We use `sales_rep` as the [grouping variable](#), effectively partitioning the `original_data` into two distinct subsets (A and B).

The code includes `sales_rep` in the `SELECT` statement and adds the necessary `GROUP BY sales_rep` clause. This modification instructs [PROC SQL](#) to perform the weighted summation calculation independently for all records belonging to Sales Rep A, and then repeat the process for Sales Rep B. The output table, viewed using `PROC PRINT`, clearly displays these segmented results, highlighting performance differences.

```
/*calculate weighted average of price, grouped by sales_rep*/  
proc sql;  
create table new_data as  
select sales_rep,  
sum(amount * price) / sum(amount) as weighted_average  
from original_data  
group by sales_rep;  
quit;
```

```
/*view results*/  
proc print data=new_data;
```

The results show a clear distinction between the two representatives. Sales Rep B has a much higher weighted average price, indicating that their larger sales volumes were consistently associated with higher price points compared to Sales Rep A, whose weighted average is lower.

Obs	sales_rep	weighted_average
1	A	5.8333
2	B	11.8182

The [weighted average](#) of price for sales rep A is **5.8333**.

The [weighted average](#) of price for sales rep B is **11.8182**.

Conclusion and Additional Resources

Mastering the calculation of [weighted averages](#) is a fundamental skill for accurate data manipulation and reporting. Utilizing [SAS](#), specifically the powerful features available in [PROC SQL](#), allows analysts to quickly generate accurate, context-sensitive metrics, both at the overall [dataset](#) level and segmented by relevant groups.

These techniques are highly transferable and can be adapted to various statistical challenges, ensuring that your analyses reflect the true importance of each observation. For those looking to further enhance their data processing skills in SAS, the following resources provide guidance on other essential tasks:

How to perform other common tasks in SAS.