Use Proc Summary in SAS (With Examples)

Authored by

Mohammed looti

November 1, 2025

Understanding PROC SUMMARY in SAS: A Foundation for Data Analysis

The PROC SUMMARY procedure stands as one of the most essential and highly efficient utilities within the SAS system for generating fundamental descriptive statistics. While similar in function to `PROC MEANS`, **PROC SUMMARY** is typically the preferred choice when the primary objective is to create a structured output dataset containing the summary metrics, rather than simply displaying the results immediately in the output viewer. This focus on dataset creation makes it indispensable for automated reporting and subsequent analytical steps.

By design, when applied to one or more numeric variables within a given dataset, **proc summary** automatically calculates a standard, fixed set of metrics. These statistics are crucial because they provide a rapid and comprehensive overview of the data's central tendency, spread, and overall distribution. Understanding these default outputs is the first step toward leveraging the procedure's full power.

The core set of statistics automatically generated for each analyzed variable includes:

N: The total count of valid, non-missing observations used in the summary calculation.
MIN: The smallest numerical value recorded for the variable in the dataset.
MAX: The largest numerical value recorded for the variable in the dataset.
MEAN: The arithmetic average, representing the central value of the distribution.
STD: The standard deviation, a metric quantifying the amount of variation or dispersion of a set of values relative to the mean.

Setting the Stage: Utilizing the SASHELP.Fish Sample Dataset

To effectively illustrate the practical application and diverse capabilities of **proc summary**, we will rely on a readily accessible, built-in SAS dataset. The chosen resource is the `sashelp.Fish` dataset, which contains a rich collection of physical measurements--including weight, height, and various length metrics--for 159 distinct fish specimens captured from a lake in Finland. This dataset is perfectly suited for demonstrating single-variable summaries, multi-variable comparisons, and complex grouped calculations.

Before proceeding with any detailed statistical summaries, it is standard analytical practice to first inspect the structure and content of the source data. This preliminary inspection ensures data quality and confirms the variables and data types available for analysis. We utilize **proc print** for this purpose, allowing us to quickly visualize the initial records.

We can use **proc print** to view the first 10 observations from this dataset, confirming the structure we will be working with:

```
/*View first 10 observations from the Fish dataset*/
proc print data=sashelp.Fish (obs=10);

run;
```

Obs	Species	Weight	Length1	Length2	Length3	Height	Width
1	Bream	242	23.2	25.4	30.0	11.5200	4.0200
2	Bream	290	24.0	26.3	31.2	12.4800	4.3056
3	Bream	340	23.9	26.5	31.1	12.3778	4.6961
4	Bream	363	26.3	29.0	33.5	12.7300	4.4555
5	Bream	430	26.5	29.0	34.0	12.4440	5.1340
6	Bream	450	26.8	29.7	34.7	13.6024	4.9274
7	Bream	500	26.8	29.7	34.5	14.1795	5.2785
8	Bream	390	27.6	30.0	35.0	12.6700	4.6900
9	Bream	450	27.6	30.0	35.1	14.0049	4.8438
10	Bream	500	28.5	30.7	36.2	14.2266	4.9594

Example 1: Generating a Summary for a Single Continuous Variable

One of the most frequent uses of **proc summary** is to calculate a straightforward set of summary statistics for a singular variable of interest. In this foundational example, our focus is on the `Weight` variable, aiming to gain a rapid understanding of the distribution of fish weights across the entire dataset. Executing this basic summary requires a precise sequence of three core statements: the procedure call itself, the `VAR` statement to specify the variable(s) being analyzed, and the mandatory `OUTPUT` statement, which dictates the name and location of the resulting summary dataset.

The following code snippet demonstrates how to calculate the standard descriptive statistics specifically for the `Weight` variable and store these aggregated results in a newly created dataset named `summaryWeight`. Notice how clean and focused the syntax remains for this fundamental task.

```
/*Calculate descriptive statistics for the Weight variable*/
proc summary data=sashelp.Fish;
var Weight;
output out=summaryWeight;
run;
```

```
/*Print the generated output dataset*/
proc print data=summaryWeight;
```

Obs	_TYPE_	_FREQ_	_STAT_	Weight
1	0	159	N	158.00
2	0	159	MIN	0.00
3	0	159	MAX	1650.00
4	0	159	MEAN	398.70
5	0	159	STD	359.09

Interpreting the output table generated by **proc summary** requires familiarity with the special system variables that the procedure automatically generates and includes in the output dataset. These variables provide essential context regarding how the statistics were calculated:

**TYPE**: This critical column indicates the level of aggregation. A value of 0 signifies that the statistics were computed using all observations in the input dataset; this means no grouping (or `CLASS`) variables were specified or active.
**FREQ**: This represents the frequency count, or the total number of input records utilized to derive the descriptive statistics reported in that specific row.
**STAT**: This column explicitly labels the type of descriptive statistic being reported (N, MIN, MAX, MEAN, STD).
Weight: This column contains the actual numerical value corresponding to the statistic named in the `_STAT_` column for the variable `Weight`.

By carefully reviewing this summary output, we gain immediate and actionable insights into the weight distribution: the total number of valid observations analyzed was **158**; the minimum weight recorded was **0**, and the maximum weight reached **1,650**. Crucially, the mean weight for all fish stands at **398.70**, with a corresponding standard deviation of **359.09**, strongly suggesting a relatively wide and diverse spread in fish weights around the average.

Example 2: Analyzing Central Tendency Across Multiple Variables Concurrently

One of the major strengths of PROC SUMMARY is its inherent flexibility, allowing analysts to efficiently calculate summary statistics for multiple numeric variables using a single procedure call. This capability is exceptionally valuable when the goal is to compare the central tendencies and measure the variability across several different continuous metrics within the same body of data,

streamlining the analytical process significantly.

To execute this operation, the analyst simply needs to list all required variables--in this demonstration, `Weight` and `Height`--within the `VAR` statement, separated by spaces. SAS then automatically computes the full suite of default statistics (N, MIN, MAX, MEAN, STD) for each variable independently, storing all results consolidated into the specified output dataset. This avoids the need for repetitive procedure calls.

For instance, we can use the following code to calculate comprehensive descriptive statistics for both the `Weight` and `Height` variables, placing the results into `summaryWeightHeight`:

```
/*Calculate descriptive statistics for Weight and Height variables*/
proc summary data=sashelp.Fish;
var Weight Height;
output out=summaryWeightHeight;
run;

/*Print the output dataset*/
proc print data=summaryWeightHeight;
```

Obs	_TYPE_	_FREQ_	_STAT_	Weight	Height
1	0	159	N	158.00	159.000
2	0	159	MIN	0.00	1.728
3	0	159	MAX	1650.00	18.957
4	0	159	MEAN	398.70	8.971
5	0	159	STD	359.09	4.286

The resulting output dataset, `summaryWeightHeight`, clearly and concisely presents the five standard descriptive statistics for both `Weight` and `Height` variables. These results are systematically organized and easily identifiable through the `_STAT_` variable. This consolidated, side-by-side view facilitates immediate and straightforward comparison of the scale, range, and variability between these two critical physical measurements, allowing analysts to quickly spot differences in their distributions.

Example 3: Granular Analysis Using Categorical Grouping with the CLASS

Statement

In most real-world analytical scenarios, researchers require summary statistics not for the dataset as a whole, but calculated separately for distinct subgroups defined by a categorical variable. This process of partitioning the data is absolutely vital for comparative analysis--for instance, determining if the average weight, maximum height, or variability in measurements differs significantly across the various species of fish.

To execute this essential subgroup analysis within PROC SUMMARY, we introduce the incredibly powerful **CLASS statement**. The purpose of the `CLASS` statement is to instruct SAS to calculate all specified statistics for every unique value (or level) found within the classification variable, effectively creating separate summary blocks for each group.

For this demonstration, we calculate the standard descriptive statistics for `Weight`, but we mandate that these calculations are grouped based on the unique values found in the `Species` variable.

```
/*Calculate descriptive statistics for Weight, grouped by Species*/
proc summary data=sashelp.Fish;
var Weight;
class Species;
output out=summaryWeightSpecies;
run;

/*Print the output dataset*/
proc print data=summaryWeightSpecies;
```

Obs	Species	_TYPE_	_FREQ_	_STAT_	Weight
1		0	159	N	158.00
2		0	159	MIN	0.00
3		0	159	MAX	1650.00
4		0	159	MEAN	398.70
5		0	159	STD	359.09
6	Bream	1	35	N	34.00
7	Bream	1	35	MIN	242.00
8	Bream	1	35	MAX	1000.00
9	Bream	1	35	MEAN	626.00
10	Bream	1	35	STD	206.60
11	Parkki	1	11	N	11.00
12	Parkki	1	11	MIN	55.00
13	Parkki	1	11	MAX	300.00
14	Parkki	1	11	MEAN	154.82
15	Parkki	1	11	STD	78.76
16	Perch	1	56	N	56.00
17	Perch	1	56	MIN	5.90
18	Perch	1	56	MAX	1100.00
19	Perch	1	56	MEAN	382.24
20	Perch	1	56	STD	347.62
21	Pike	1	17	N	17.00
22	Pike	1	17	MIN	200.00
23	Pike	1	17	MAX	1650.00
24	Pike	1	17	MEAN	718.71
25	Pike	1	17	STD	494.14
26	Roach	1	20	N	20.00
27	Roach	1	20	MIN	0.00
28	Roach	1	20	MAX	390.00
29	Roach	1	20	MEAN	152.05
30	Roach	1	20	STD	88.83

The resulting output dataset, `summaryWeightSpecies`, is significantly more detailed compared to the non-grouped examples. It meticulously displays the full set of descriptive statistics for each unique `Species` of fish present in the dataset. It is important to observe how the `_TYPE_` variable changes; it now holds different values (1, 2, 3, etc.) corresponding to the level of grouping applied. In typical **proc summary** output, the highest type value usually represents the most granular summary--the statistics calculated specifically for the individual species groups.

Focusing specifically on the statistics for the group where `Species` equals "Bream," we can quickly extract key insights relevant only to that species:

The total number of observations counted for Bream was **34**.

The minimum weight recorded for Bream was **242**.
The maximum weight recorded for Bream was **1,000**.
The mean weight value for Bream was **626**.
The standard deviation of weight values for Bream was **206.60**, indicating its weight distribution relative to its mean.

This highly detailed, species-specific output powerfully illustrates the utility of the `CLASS` statement in partitioning data for granular statistical analysis. It enables researchers to perform sophisticated comparative analyses by comparing distributional characteristics accurately across all defined categorical groups.

Additional Resources for Advanced SAS Programming

Mastering PROC SUMMARY is a fundamental and crucial step toward achieving proficiency in SAS data preparation and analysis. For those looking to expand their statistical and programming skills further within the SAS environment, the following tutorials explain how to perform other common data manipulation and analytical tasks: