How to Generate Descriptive Statistics in SAS Using Proc Summary

Authored by

stats writer

December 1, 2025

The Proc Summary procedure in SAS is an indispensable tool for generating comprehensive descriptive statistics from a single dataset. This procedure is highly optimized for speed and efficiency, particularly when handling large datasets, making it a cornerstone of efficient data analysis within the SAS environment. When executed, Proc Summary processes data values from specified variables and consolidates them into a concise summary report, enabling analysts to quickly grasp the central tendencies and variability within their data.

The primary utility of Proc Summary lies in its versatility in calculating core statistical measures necessary for initial data exploration. These measures include central tendency metrics such as the mean and median, measures of dispersion like the standard deviation and variance, and positional statistics such as quartiles, minimum, and maximum values. Furthermore, it automatically computes the sum and the total count (N) of non-missing observations for the variables selected, providing a robust overview of the data distribution profile. Unlike PROC MEANS, Proc Summary does not print results to the output window by default, compelling the user to employ the **OUTPUT** statement to store results in a new SAS dataset for subsequent analysis or reporting.

Understanding these fundamental statistics is crucial as the first step in any analytical workflow. By rapidly generating these metrics, Proc Summary allows data analysts to quickly assess data quality, identify potential outliers or data anomalies, and evaluate the underlying distribution of variables before proceeding with more complex statistical modeling. The inherent efficiency of this procedure makes it an essential component for routine data management and validation tasks across various industries utilizing SAS for large-scale data processing.

Core Statistical Metrics Provided by Proc Summary

When utilizing the **proc summary** procedure in SAS without specifying any particular keywords in the OUTPUT statement, the procedure automatically calculates and outputs a standard set of statistics for the variables listed in the VAR statement. These metrics provide a comprehensive snapshot of the variable's characteristics, covering count, central tendency, and variability.

N: Represents the total number of non-missing observations used in the calculation for the specific variable. This is critical for assessing data completeness.
MIN: Reports the minimum observed value within the variable. This helps identify the lower boundary of the data range.
MAX: Reports the maximum observed value within the variable, establishing the upper boundary of the data range.
MEAN: Provides the arithmetic mean, which is the sum of all values divided by the count of non-missing values, indicating the central location of the distribution.
STD: Calculates the standard deviation, which measures the dispersion or variability of the

dataset. A higher STD indicates that the data points are spread out over a wider range of values.

While these five statistics are produced by default, analysts have the flexibility to request many other statistical outputs using specific keywords within the OUTPUT statement, such as SUM, VAR (variance), Q1, Q3 (quartiles), and P90 (percentiles). This modular approach allows the procedure to be tailored precisely to the statistical requirements of the analysis being conducted, ensuring that resources are used efficiently and only the necessary metrics are generated and stored.

Preparing the Data: Utilizing the SASHELP.FISH Dataset

To illustrate the practical application of the **proc summary** procedure, we will utilize the widely available **SAS built-in dataset** named **SASHELP.Fish**. This dataset is a standard resource used for demonstration and contains various measurements for 159 different fish caught in a lake in Finland. The variables include quantitative measures such as Weight, Lengths, and Height, alongside qualitative measures like Species.

Before running the summary procedure, it is good practice to inspect the structure and initial observations of the dataset to ensure data integrity and understand the available variables. We can use the **proc print** procedure coupled with the **OBS=10** option to display the first ten observations from this dataset, confirming the data format and variable types.

The following code executes this initial data viewing step:

```
/*view first 10 observations from 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

The output visualization above confirms the presence of variables such as Species, Weight, Length1, Length2, Length3, Height, and Width. This initial check is vital, especially when dealing with new or unfamiliar datasets, as it ensures that the variables we intend to summarize are correctly identified and available for processing in subsequent steps using **proc summary**.

Example 1: Summarizing a Single Variable

Our first practical example demonstrates the simplest application of **proc summary**: calculating the descriptive statistics for a single, quantitative variable. We will focus on the **Weight** variable, which represents the weight of each fish in grams. This exercise is foundational to understanding the basic syntax and the essential use of the **VAR** and **OUTPUT** statements.

The **VAR** statement specifies which variables are to be included in the summary calculation. Since **proc summary** suppresses output by default, the **OUTPUT OUT=** statement is mandatory; it directs the calculated statistics into a newly created SAS dataset, which we name **summaryWeight**. Once the summary procedure runs, we then use **proc print** to display the contents of this new output dataset, making the results visible to the user.

The following code snippet executes the calculation and displays the resulting statistics:

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

```
/*print 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 Proc Summary Output Dataset

The output dataset generated by **proc summary** follows a standard structure that may initially appear complex but is highly logical and essential for understanding grouped statistics later on. The dataset contains several automatically generated variables alongside the calculated statistics, which must be interpreted correctly.

Here is a detailed breakdown of the key columns in the output table:

**TYPE**: This column is critical when using grouping variables (CLASS statement). A value of **0**, as seen in this single-variable summary, signifies that the statistics were calculated across the entire dataset without any subgrouping. Higher values indicate specific combinations of class variables used in grouped analysis.
**FREQ**: This variable indicates the number of observations that contributed to the calculation of the descriptive statistics in that particular row. In the case of a full summary (where _TYPE_=0), this represents the total count of non-missing observations for the variable across the entire input dataset.
**STAT**: This categorical column specifies the name of the descriptive statistic presented in that row. The default statistics produced are N, MIN, MAX, MEAN, and STD.
Weight: This variable column contains the numerical value for the corresponding descriptive statistic listed in the _STAT_ column.

By examining the output, we derive specific, valuable insights into the fish weight data:

The total number of observations (N) used was **158** (meaning one observation was missing a weight value).
The minimum weight value observed was **0** grams.

The maximum weight value observed was **1,650** grams.
The mean weight value was approximately **398.70** grams.
The standard deviation of weight values was **359.09**, indicating a high degree of variability in the weights of the fish captured.

These five statistical measures collectively provide an excellent initial understanding of the distribution of values for the Weight variable, highlighting the range and the degree of spread around the average weight.

Example 2: Summarizing Multiple Variables

One of the greatest benefits of **proc summary** is its ability to handle multiple variables simultaneously with minimal change to the code structure. If an analyst needs to compare the central tendency and dispersion of several quantitative variables--for instance, Weight and Height--they simply list both variables within the **VAR** statement.

This approach significantly streamlines the analytical process, avoiding the need to run the procedure separately for each variable. All requested statistics (N, MIN, MAX, MEAN, STD) will be calculated for every variable specified and output into a single, comprehensive summary dataset. This makes comparisons between variables, such as variability in weight versus variability in height, straightforward and efficient.

We use the following code to calculate descriptive statistics for both the **Weight** and **Height** variables:

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

/*print 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

As shown in the output, the resulting dataset now includes separate columns for both **Weight** and **Height**, with the calculated statistical values aligned with the corresponding _STAT_ type. We can now easily observe the five default descriptive statistics for both variables side-by-side, facilitating direct comparison of their data distributions. For instance, comparing the standard deviation of Weight (359.09) against that of Height (42.82) immediately reveals that fish weights exhibit significantly higher absolute variability than fish heights within this sample.

Example 3: Grouped Summaries using the CLASS Statement

Often in data analysis, it is necessary to calculate statistics not just for the entire population, but for distinct subgroups within the data. To calculate descriptive statistics for a quantitative variable (like Weight) based on the categories defined by a nominal variable (like Species), we must use the powerful **CLASS** statement within **proc summary**.

The **CLASS** statement instructs **proc summary** to partition the input dataset according to the unique values (levels) of the specified classification variable(s). The procedure then calculates all requested summary statistics independently for each resulting subgroup. This capability is fundamental for comparative analysis, such as determining if the average weight varies significantly across different species of fish.

We use the following code to calculate descriptive statistics for **Weight**, but this time, the calculations are grouped by **Species**:

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

/*print 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

Analyzing Grouped Summary Results

The output table generated from the grouped analysis is considerably longer than the previous examples because it includes five summary rows for every unique level of the **Species** variable. Note that the output dataset now includes the **Species** column, indicating which group the statistics belong to. The **_TYPE_** variable, which was 0 previously, will now typically be 1, signifying that a classification variable was used to generate the summaries.

The analyst can now easily compare the characteristics of the Weight variable across the different fish species. For instance, focusing solely on the rows corresponding to the **Bream** fish, we can extract the following specific group statistics:

The total number of observations (N) for Bream was **34**.
The minimum weight value for Bream was **242** grams.
The maximum weight value for Bream was **1,000** grams.
The mean weight value for Bream was approximately **626** grams.
The standard deviation of weight values for Bream was **206.60**.

By repeating this observation process for every other species listed in the table, analysts gain a highly granular view of how weight distribution differs across the various types of fish. This level of detail is invaluable for hypothesis testing and targeted reporting. The variation in mean weights (e.g., Bream mean of 626 vs. other species) provides compelling evidence of significant biological differences, justifying further inferential statistical testing.

Advanced Output Options and Considerations

While the default output provides N, MIN, MAX, MEAN, and STD, **proc summary** offers extensive flexibility through the **OUTPUT** statement to request specialized statistics. Analysts can easily add metrics such as the Variance (VAR), Skewness (SKEW), Kurtosis (KURT), Sum of Weights (WTSUM), or specific percentiles (P1, P5, P99). To request these, one simply includes the corresponding keyword followed by the desired variable names in the OUTPUT statement. For instance, **OUTPUT OUT=MyData VAR=Weight P90=Weight_P90;** would add both variance and the 90th percentile of Weight to the output dataset.

Another important consideration involves handling missing values. By default, **proc summary** excludes missing values from the calculation of statistics like the mean or standard deviation, only counting non-missing observations towards N. However, if the analysis requires that missing values be treated as a valid category when grouping data, the **CLASS** statement can be augmented with the **MISSING** option. Furthermore, analysts should be aware that **proc summary** is often preferred over PROC MEANS when the primary goal is not screen visualization but rather the creation of a new, clean dataset containing aggregate statistics for use in subsequent data steps or other procedures within the analytical pipeline.

The ability to generate customized output datasets efficiently and to perform complex grouped analyses using the **CLASS** statement solidifies **proc summary**'s position as a powerful and highly versatile procedure for data aggregation and initial statistical assessment in the SAS programming language. Mastery of this procedure is fundamental for any professional involved in large-scale data processing and reporting.

Further Exploration in SAS

The following related tutorials provide guidance on performing other common and essential data manipulation and statistical tasks in SAS: