How to Easily Sort PROC FREQ Output in SAS

Authored by

stats writer

November 21, 2025

The output of PROC FREQ in SAS can be efficiently customized and sorted using the **ORDER=** option. This critical procedure option allows analysts to dictate the exact order in which categories appear within the resultant frequency table. You can choose to sort based on metrics like raw observation **count**, calculated **percent**, or even by a user-defined sort order. Furthermore, while the default frequency sort is descending, advanced techniques allow for ascending sort customization. This capability ensures quick analysis and easy visualization of the most relevant patterns within your statistical data.

1. Introduction: Mastering Output Sorting for PROC FREQ in SAS

The PROC FREQ procedure is one of the most fundamental and widely used tools within the SAS statistical software environment. It is primarily utilized for generating frequency tables, cross-tabulations, and various statistical measures related to categorical variables within your dataset. While the default output often sorts categories alphabetically or by internal data representation, data analysis frequently requires arranging results based on magnitude, such as count or percentage. To achieve this necessary level of customization, SAS provides the powerful **ORDER=** option, which is integral to controlling how the resulting frequency table is presented.

Understanding the nuances of the **ORDER=** option is essential for any SAS programmer looking to efficiently summarize and visualize descriptive statistics. This option grants granular control over the output arrangement, allowing users to specify sorting criteria based on key metrics like observation **count** or calculated **percent**. Furthermore, while the standard functionality offers descending sort based on frequency (highest to lowest), we will also explore an advanced workaround utilizing the PROC SORT procedure to achieve an ascending sort (lowest to highest), providing a comprehensive approach to output management. This detailed guide will ensure you can tailor PROC FREQ output precisely to meet your specific analytical needs, enhancing the clarity and impact of your statistical reports.

To sort categories by their counts in descending order, you must use **PROC FREQ** with the **ORDER=FREQ** option in SAS to create a frequency table in which the categories in the table are sorted based on frequency.

2. Utilizing the ORDER= Option for Sorting Categories

As a foundational element of customizing PROC FREQ, the **ORDER=** option is specified directly within the procedure statement. The key purpose of this parameter is to dictate the display order of the variable levels in the resulting tables. By default, if the **ORDER=** option is omitted, SAS typically sorts the categories based on the formatted values of the analysis variable--usually an alphabetical or character sort. However, when performing descriptive analysis, it is far more common to require a sort based on how frequently each category appears.

The most powerful and frequently used setting for this option is **ORDER=FREQ**. When this option is applied, SAS instructs the procedure to arrange the categories in the output based on their respective frequencies (counts). By default, this arrangement is always done in **descending order**, meaning the category with the highest count appears first in the frequency table. This is particularly useful when quickly identifying the most prevalent groups within the analyzed data.

There are several valid keywords for the **ORDER=** option, allowing for versatility in output presentation. Common values include:

ORDER=FREQ: Sorts categories by frequency count, descending (highest count first).

ORDER=DATA: Sorts categories by the order of their appearance in the input dataset.

ORDER=FORMATTED: Sorts categories alphabetically or numerically based on their formatted values (this is the default behavior if **ORDER=** is omitted).

ORDER=INTERNAL: Sorts categories based on their internal, unformatted values.

3. Implementing the Basic Syntax for Frequency Sorting

You can use the following syntax to apply frequency-based sorting:

```
proc freq data=my_data order=freq;
tables my_variable;
run;
```

To implement the **ORDER=FREQ** option, the syntax must be included immediately after the `proc freq` statement and before the standard `tables` statement that defines the variable of interest. This straightforward integration ensures that the sorting logic is applied before the table is generated and printed to the output window. Using this syntax provides an immediate and clear view of the distribution sorted by the importance of each category's presence within the data.

The general structure shown above involves specifying the procedure name, the dataset to be analyzed, the **ORDER=FREQ** option, and finally, the variable list within the `tables` statement. It is critical to remember that this option specifically manages the sequence of rows in the output table, which represent the levels of the categorical variable. It does not alter the underlying data structure itself, but only affects its presentation by the procedure.

The following example shows how to use this syntax in practice.

4. Demonstrative Example: Preparing the Input Data

For demonstration purposes, we will utilize a readily accessible and standardized dataset provided within the SASHELP library. Specifically, we will use the **SASHELP.BirthWgt** dataset. This dataset is a popular resource for statistical examples as it contains characteristics for a large cohort of 100,000 mothers who recently gave birth, offering rich categorical variables suitable for frequency table analysis.

Before executing the frequency analysis, it is good practice to inspect the input data structure and view the first few observations to understand the variables and their formats. We can achieve this quick inspection using the PROC PRINT procedure, combined with the **OBS=** option to limit the displayed rows.

The code below requests that SAS print the first 10 observations from the `sashelp.BirthWgt` dataset. This ensures we confirm the presence and structure of the variables we intend to analyze, such as the `Race` variable, which we will use for our frequency demonstration.

```
/*view first 10 observations from BirthWgt dataset*/
proc print data=sashelp.BirthWgt (obs=10);

run;
```

Obs	LowBirthWgt	Married	AgeGroup	Race	Drinking	Death	Smoking	SomeCollege
1	No	No	3	Asian	No	No	No	Yes
2	No	No	2	White	No	No	No	No
3	Yes	Yes	2	Native	No	Yes	No	No
4	No	No	2	White	No	No	No	No
5	No	No	2	White	No	No	No	Yes
6	No	No	2	White	No	No	No	
7	No	No	2	Asian	No	No	No	Yes
8	No	No	3	White	No	No	No	Yes
9	No	Yes	1	Black	No	No	No	No
10	No	No	2	Native	No	No	No	Yes

5. Generating the Default Frequency Output

To establish a baseline for comparison, we first execute PROC FREQ without specifying the **ORDER=** option. When analyzing the `Race` variable, we anticipate the output will default to sorting categories based on their formatted values, which in this case means alphabetical order.

The following code block generates a standard frequency distribution for the `Race` variable:

```
/*create frequency table for Race variable*/
proc freq data=sashelp.BirthWgt;
tables Race;
run;
```

The FREQ Procedure

Race	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Asian	5224	5.22	5224	5.22
Black	14133	14.13	19357	19.36
Hispanic	22139	22.14	41496	41.50
Native	942	0.94	42438	42.44
White	57562	57.56	100000	100.00

Upon reviewing the resulting frequency table, it becomes evident that the levels of the `Race` variable (e.g., 'Asian', 'Black', 'Hispanic', 'White') are arranged strictly according to their alphabetical position. This default sort, while consistent, may obscure immediate insights into which demographic groups are most represented in the dataset. Notice that the categories are currently sorted in alphabetical order.

6. Applying ORDER=FREQ for Descending Sort

To re-sort the output so that the categories are arranged based on their frequency count, we must modify the initial PROC FREQ statement to include the **ORDER=FREQ** option. This modification immediately instructs SAS to prioritize the display based on magnitude rather than lexicographical order.

To instead sort the categories by frequency, we can use the following syntax:

```
/*create frequency table for Race variable, sorted by frequency*/
proc freq data=sashelp.BirthWgt order=freq;
tables Race;
run;
```

The resulting output dramatically changes the presentation. The categories are now reorganized

starting with the highest count and proceeding sequentially to the lowest. This descending sort is the standard behavior when using **ORDER=FREQ** and is optimal for identifying the dominant characteristics within the dataset at a glance.

The FREQ Procedure

Race	Frequency	Percent	Cumulative Frequency	Cumulative Percent
White	57562	57.56	57562	57.56
Hispanic	22139	22.14	79701	79.70
Black	14133	14.13	93834	93.83
Asian	5224	5.22	99058	99.06
Native	942	0.94	100000	100.00

Notice that the categories are now sorted based on frequency from highest to lowest.

7. Advanced Sorting: Achieving Ascending Order (Lowest to Highest)

While **ORDER=FREQ** provides a descending sort, a common analytical requirement is to view the results in **ascending order**--from the lowest frequency count to the highest. Unfortunately, PROC FREQ does not inherently support an ascending sort directly through its standard options. When this specific presentation is needed, SAS programmers must employ a multi-step workaround involving saving the frequency results to a new dataset and then processing that intermediate dataset using PROC SORT.

This workaround leverages the ability of PROC FREQ to output its results into a SAS dataset via the **OUT=** option specified in the `TABLES` statement. Once the raw frequency counts and percentages are stored in a temporary dataset, the powerful sorting capabilities of PROC SORT can be utilized to arrange the observations (which represent the categories) according to the `Count` variable in the desired ascending order.

The primary steps for this advanced technique involve:

Executing PROC FREQ with the **NOPRINT** option and saving the output using **OUT=**.
Executing PROC SORT on the output dataset using the `BY Count` statement.
Optionally, using a DATA step to recalculate cumulative statistics based on the new sort order.
Displaying the final, sorted dataset using PROC PRINT.

8. Step-by-Step Implementation of the Ascending Sort Workaround

You can use the following workaround with the **PROC SORT** statement to sort based on frequency from lowest to highest:

```
/*create frequency table and store results in Racefreq*/
proc freq data=sashelp.BirthWgt noprint;
tables Race / out=Racefreq;
run;

/*sort Racefreq based on frequency from lowest to highest*/
proc sort data=Racefreq;
by count;
run;

/*create new dataset with cumulative freq and cumulative percent*/
data freq_low_to_high;
set Racefreq;
cumcount + count;
cumpercent + percent;
run;

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

In this sequence, the initial PROC FREQ uses the **NOPRINT** option to suppress the default output and the **OUT=Racefreq** option to save the results (including the `Count` variable) into a temporary dataset named `Racefreq`. Subsequently, PROC SORT is applied to this dataset, using the `BY count` statement to sort it in ascending order by default.

The subsequent DATA step is vital if cumulative statistics are required, as sorting the data invalidates the automatically calculated cumulative values from the initial PROC FREQ run. The DATA step iteratively calculates the new **cumcount** and **cumpercent** based on the newly sorted order. Finally, PROC PRINT displays the custom-sorted frequency table.

Obs	Race	COUNT	PERCENT	cumcount	cumpercent
1	Native	942	0.942	942	0.942
2	Asian	5224	5.224	6166	6.166
3	Black	14133	14.133	20299	20.299
4	Hispanic	22139	22.139	42438	42.438
5	White	57562	57.562	100000	100.000

Notice that the categories are now sorted based on frequency from lowest to highest.

9. Summary of Sorting Techniques in Frequency Analysis

Effective manipulation of output presentation is crucial for high-quality statistical reporting. By mastering the **ORDER=** option in PROC FREQ, SAS users can quickly toggle between alphabetical, data-entry, or descending frequency sorts, catering to immediate analytical needs. The standard approach using **ORDER=FREQ** remains the simplest method for highlighting the most frequent observations.

However, for specialized requirements such as ascending sort, the integration of multiple SAS procedures--specifically combining PROC FREQ to generate counts and PROC SORT to reorganize the intermediate dataset--is necessary. This technique demonstrates the flexibility of the SAS language, allowing complex data manipulation through procedural chaining.

By implementing these sorting strategies, analysts ensure that their frequency tables are not only accurate but also visually optimized for clear interpretation, regardless of whether the goal is to emphasize the most common or the rarest occurrences within the statistical data. These techniques are fundamental for robust descriptive statistics in the SAS environment.

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