

SAS: Use PROC FREQ with ORDER Option

Authored by
Mohammed looti

November 16, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *SAS: Use PROC FREQ with ORDER Option*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=2595>

The Importance of Ordering in Frequency Analysis

Effective [data analysis](#) hinges on the ability to swiftly extract meaningful patterns from raw information. A fundamental step in this process involves understanding the exact distribution of [categorical variables](#) within a dataset. The resulting frequency distribution, often presented as a table, serves as the primary quantitative summary, detailing how often each unique value appears. While calculating accurate counts is essential, the true power of this summary lies in its presentation. If the categories are ordered randomly or illogically, the resulting table can become dense and challenging to interpret, potentially burying critical insights.

When working with large, real-world datasets, analysts are typically interested in identifying the "vital few"--the categories that account for the majority of observations, or conversely, the rare categories that represent outliers or unusual events. If a frequency table defaults to sorting categories based on alphabetical labels, this crucial information may not be immediately apparent. For instance, in market research, knowing the top three most frequent purchase types is far more valuable than seeing those types listed alphabetically amongst dozens of others. Strategic sorting is therefore not merely a cosmetic choice but a functional requirement for achieving maximum clarity and analytical impact.

The [SAS](#) software suite, a cornerstone for statistical computing, offers robust tools for this task, primarily through the powerful **PROC FREQ** procedure. By default, **PROC FREQ** organizes output based on the formatted values of the categories, which almost always results in an alphabetical sort. This default behavior, while consistent, often fails to serve the primary analytical need: prioritization by prevalence. This article provides a comprehensive guide to utilizing the **ORDER** option within [PROC FREQ](#) to impose frequency-based sorting, ensuring your statistical reports immediately convey the most relevant trends and distributions.

Mastering the ORDER=FREQ Option

To move beyond the limitations of alphabetical ordering and introduce statistically meaningful prioritization, [PROC FREQ](#) utilizes the specific parameter: **ORDER=FREQ**. When this option is specified within the procedure statement, [SAS](#) is explicitly instructed to arrange all distinct categories in descending sequence, based on their absolute occurrence counts (their frequencies). This methodology ensures that the most common items--those contributing the largest number of observations to the dataset--are positioned at the very beginning of the generated output.

This descending frequency sort is invaluable across numerous analytical domains. For example, in quality control, it instantly identifies the most common types of failures. In demographic analysis, it pinpoints majority populations. This immediate visual hierarchy dramatically speeds up the process of interpretation, allowing analysts and stakeholders to focus their attention on the most impactful categories without having to manually scan the entire count column. The implementation of this

feature is remarkably simple, requiring only a minor, yet powerful, modification to the standard **PROC FREQ** syntax.

The required structure to enable this frequency-based sorting is highly intuitive and applies the ordering logic globally to the table creation process. The **ORDER=FREQ** option is placed directly on the main **PROC FREQ** statement, ensuring the sorting is applied before the output is generated. This is the syntax template for effective implementation:

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

In the example above, `my_data` represents the specific [SAS](#) dataset being analyzed, and `my_variable` is the target [categorical variable](#). The inclusion of `order=freq` is the sole mechanism required to transform a default alphabetical report into a statistically prioritized report. This small adjustment significantly enhances the utility of the output, optimizing the process of extracting actionable intelligence from vast volumes of information.

Practical Application: Exploring the BirthWgt Dataset

To fully illustrate the practical benefits of the **ORDER=FREQ** option, we will utilize a widely accessible, built-in [SAS](#) dataset: `sashelp.BirthWgt`. This comprehensive dataset contains 100,000 observations related to maternal and childbirth characteristics and serves as an excellent foundation for executing demographic and health-related [data analysis](#). By examining the distribution of a characteristic like maternal race within this large sample, we can clearly demonstrate the difference between default and frequency-prioritized reporting.

Before diving into the frequency analysis, adhering to best practices dictates a preliminary inspection of the source data. This vital step confirms data structure, verifies the presence of relevant [categorical variables](#), and ensures the integrity of the observations. We typically use the [PROC PRINT](#) procedure for this preliminary review, allowing us to quickly visualize the first few records of the dataset before applying complex statistical methods. Limiting the output to ten records is often sufficient to confirm data readiness:

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

The resulting output from [PROC PRINT](#) provides immediate visual confirmation of the dataset's

column structure, variable names, and data types. This successful verification confirms that the data is prepared for the subsequent statistical procedures, specifically the frequency analysis of the `Race` variable.

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

Comparing Default and Frequency-Based Sorting

To truly appreciate the analytical lift provided by frequency sorting, we must first establish a baseline using the default behavior of `PROC FREQ`. By executing the procedure on the `Race` variable without including the `ORDER` option, the resulting [frequency table](#) will sort categories alphabetically based on the text labels:

```
/*create frequency table for Race variable using default (alphabetical) sort*/
proc freq data=sashelp.BirthWgt;
tables Race;
run;
```

The output below clearly shows the categories arranged in strict alphabetical order. While this arrangement is systematic, it forces the reader to scan the entire 'Frequency' column to locate the most populous racial groups. In essence, the default setting prioritizes lexicographical order over statistical significance. This can significantly slow down the initial exploration phase of [data analysis](#), as the most critical insights regarding data distribution are not immediately presented.

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

The introduction of the **ORDER=FREQ** option provides an immediate, powerful solution. By instructing [SAS](#) to sort the categories based on their respective counts, from largest to smallest, we leverage the power of prioritization. This technique is particularly beneficial when applying principles like the Pareto distribution (the 80/20 rule) to identify the core components driving the data distribution. The syntax change required is minimal:

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

As the resulting table demonstrates, the analytical advantage is immediate and profound. The categories are now prioritized by prevalence, instantly revealing the demographic structure of the sample and clearly highlighting the dominant groups. This enhanced presentation significantly lowers the cognitive burden on the analyst, allowing for quicker, more reliable interpretation and facilitating robust, data-driven decision-making.

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

Implementing Ascending Frequency Sort: A Procedural Workaround

While **ORDER=FREQ** is the standard mechanism for descending sorting (most frequent categories first), there are specific statistical scenarios where the inverse order is required--sorting categories in ascending sequence, from the least frequent to the most frequent. This ascending sort is essential when the goal is to identify rare events, anomalies, or minority subgroups. Unfortunately, the current version of **PROC FREQ** does not offer a direct option like **ORDER=ASCENDING_FREQ**. To achieve this specific reporting requirement, experienced **SAS** programmers must implement a flexible, multi-step procedural workaround involving the sequential use of core components.

This robust technique leverages the interoperability between various **SAS** procedures. The process begins by executing **PROC FREQ**, but we utilize the **NOPRINT** option to suppress the visual output, simultaneously instructing the procedure to export the calculated frequencies into a temporary **DATA step** dataset using the **OUT=** statement. Once the raw counts are stored, the second step involves using **PROC SORT** on this new temporary dataset. The powerful **BY COUNT** statement is used here to explicitly sort the dataset based on the frequency variable (**COUNT**) in ascending order, thereby achieving the desired prioritization.

The final steps involve processing the sorted data. A subsequent **DATA step** is often employed to calculate new derived variables, such as running totals for **cumulative frequency** and cumulative percentages, transforming the raw counts into a complete analytical report. Finally, the **PROC PRINT** procedure is used to display the final, fully sorted table. This four-step sequence is essential for achieving the required ascending sort:

/*Step 1: Create frequency table and store results in Racefreq dataset, suppressing output*/

```
proc freq data=sashelp.BirthWgt noprint;  
tables Race / out=Racefreq;  
run;
```

/*Step 2: Sort Racefreq based on frequency (COUNT) from lowest to highest*/

```
proc sort data=Racefreq;  
by count;  
run;
```

/*Step 3: Create a new dataset with cumulative frequency and cumulative percent using DATA step*/

```
data freq_low_to_high;  
set Racefreq;  
cumcount + count;  
cumpercent + percent;  
run;
```

/*Step 4: View the final results using PROC PRINT*/

```
proc print data=freq_low_to_high;
```

The final table, presented below, successfully achieves the desired ascending sort order. This multi-step process underscores the immense flexibility inherent in [SAS](#) programming, allowing users to combine procedures to meet highly specific data manipulation and reporting demands that go beyond the standard options provided by a single procedure.

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

Key Considerations for Optimal Frequency Reporting

While the **ORDER=FREQ** option is highly effective, generating optimal statistical output requires strategic thought beyond simple syntax. When leveraging [PROC FREQ](#), analysts should adhere to several best practices to ensure that the results are both computationally efficient and highly communicative to the intended audience.

Align Order with Analytical Goal: The choice of ordering method must be dictated by the specific analytical question being addressed. If the goal is rapid identification of dominant factors, **ORDER=FREQ** is superior. However, for comparisons against historical reports or external standards, maintaining the default alphabetical order (or using **ORDER=FORMATTED**) might be necessary to ensure consistency and prevent confusion. Conversely, **ORDER=DATA** preserves the physical order of the first appearance in the dataset, which is rarely used but important for specific debugging tasks.

Consider Performance Implications: Although [SAS](#) procedures are designed for efficiency, sorting large tables--especially those created via procedural workarounds (like the ascending sort method requiring outputting and re-sorting temporary datasets)--can significantly increase processing time and memory consumption when dealing with exceptionally large datasets. Analysts must weigh the analytical benefit of the sort against the computational cost.

Ensure Reporting Transparency: When presenting [frequency tables](#) that deviate from the standard alphabetical sequence, it is professionally imperative to clearly annotate the report with the sorting logic used. Explicitly stating that the table is sorted by "Frequency (Descending)"

prevents misinterpretation by readers who may default to assuming alphabetical organization. This transparency is crucial for maintaining the credibility and accuracy of the statistical findings.

Explore Advanced PROC FREQ Features: The functionality of **PROC FREQ** extends far beyond simple count tables. Analysts should familiarize themselves with advanced options such as `OUT=` for creating output datasets, `NOPRINT` for batch processing, and various table options that enable the calculation of inferential statistics, including chi-square tests, exact tests, and measures of association, all within the same procedure execution.

Conclusion: Enhancing Statistical Interpretability

The effective summarization of categorical data forms the bedrock of insightful statistical reporting. The capability to easily sort [frequency tables](#) by their counts, moving beyond simple alphabetical arrangement, fundamentally improves the interpretability and analytical value of these summaries. By mastering the `ORDER=FREQ` option within [PROC FREQ](#), [SAS](#) users gain the immediate ability to prioritize data visualization, rapidly identifying prevalent patterns and informing robust decision-making processes.

Moreover, understanding the procedural workaround for achieving ascending frequency sorts--a technique that combines **PROC FREQ**, [PROC SORT](#), and the [DATA step](#)--demonstrates the versatility and power inherent in the [SAS](#) language. These combined techniques are essential for generating statistical reports that are not only accurate but also presented in the most analytically effective and insightful manner, catering to highly diverse reporting requirements.

Additional Resources for SAS Programming

To further refine your [SAS](#) programming expertise and explore more advanced methods for data manipulation and statistical modeling, the following official resources are highly recommended:

[SAS/STAT Procedures Guide](#)

[SAS Training and Certification](#)

[SAS Support Communities](#)