

How to Easily Create Frequency Tables in SAS

Authored by
stats writer

December 1, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Create Frequency Tables in SAS*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=103468>

Understanding data distribution is fundamental to any statistical analysis. Frequency tables provide a concise, organized summary of categorical or discrete data, showing how often each value occurs. In the SAS system, generating these essential summaries is straightforward, relying primarily on the powerful PROC FREQ procedure.

This procedure is instrumental for analysts, allowing them to quickly calculate raw frequencies, percentages, and cumulative statistics for a given set of data. PROC FREQ is versatile enough to construct both one-way distributions for individual variables and two-way cross-tabulations (contingency tables) to examine relationships between pairs of variables. This guide provides practical examples demonstrating how to leverage this crucial tool for robust data exploration.

The **proc freq** statement is the core command for generating distribution summaries in SAS. It is designed to efficiently count the occurrences of unique values within one or more variables in a specified dataset. By mastering the arguments associated with **proc freq**, users can tailor output to meet specific reporting and analytical requirements, such as sorting results or handling missing observations.

The subsequent examples will utilize a common, built-in SAS resource: the **sashelp.BirthWgt dataset**. This comprehensive data source contains approximately 100,000 observations detailing various demographic and medical characteristics for mothers who recently gave birth, offering an excellent foundation for demonstration purposes.

To establish context, we first use **proc print** to display the initial ten observations from the **BirthWgt dataset**, ensuring we understand the structure of the underlying variables before proceeding with frequency calculations:

/*Reviewing the initial 10 observations from the sashelp.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

Example 1: Generating a Standard One-Way Frequency Table

The most straightforward application of `PROC FREQ` involves calculating the distribution for a single categorical variable. This process, often referred to as a one-way frequency analysis, provides a foundational summary of how observations are distributed across the categories of interest. In this scenario, we focus on the **Race** variable within the **BirthWgt** dataset.

To initiate the analysis, the **TABLES** statement is employed immediately following the **proc freq** statement. The **TABLES** statement specifies which variables SAS should analyze. Since we are only interested in **Race**, the syntax remains concise and highly readable:

```
/*Creating a standard one-way frequency table for the 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

The resulting output table is rich in detail and automatically includes four standard statistical measures, allowing for immediate interpretation of the data distribution. Understanding these components is essential for drawing accurate conclusions from the frequency tables:

Frequency: This is the absolute count, representing the total number of observations that belong to a specific category or level of the variable. For example, it shows the exact number of mothers categorized as "Hispanic."

Percent: This column displays the relative frequency, calculated as the frequency of the category divided by the total number of non-missing observations, expressed as a percentage. This measure is crucial for comparing category sizes relative to the whole sample.

Cumulative Frequency: This value is the running total of frequencies. It represents the sum of the frequencies of the current category and all categories preceding it in the table order. This is particularly useful when dealing with ordered data or when determining the count up to a certain point in the distribution.

Cumulative Percent: Similar to the cumulative frequency, this is the running total of the percentages. It indicates the percentage of the total observations that fall up to and including the current row's category.

Analyzing the output provides clear insights into the demographic makeup of the sample. For instance, the table reveals specific counts and proportions across the different racial categories:

We observe that the total count of mothers identifying as Hispanic is precisely **22,139**.

This count translates to a relative proportion, where **22.14%** of the total mothers in the analysis fall under the Hispanic category.

By examining the cumulative statistics, we find that the combined count of mothers classified as Asian, Black, or Hispanic totals **41,496** observations.

This combined group represents a significant portion of the sample, accounting for a **41.50%** cumulative percentage of the total population studied.

Example 2: Sorting Categories by Frequency Using the ORDER Option

By default, SAS sorts the categories displayed in a frequency table alphabetically or numerically based on the variable's category name. While this default ordering is suitable for many standard reports, data analysts often prefer to see categories ordered by their magnitude--that is, sorted by their actual frequency count--to easily identify the most common or rarest occurrences.

To override the default sorting behavior and rank the categories based on the raw frequency count, we introduce the **ORDER=FREQ** option directly into the **proc freq** statement. This parameter tells SAS to calculate the frequencies first and then arrange the output rows in descending order of these counts, prioritizing clarity in distribution analysis:

```
/*Creating a frequency table for Race, explicitly sorted by descending frequency count*/  
proc freq data=sashelp.BirthWgt order=freq;  
tables Race;  
run;
```

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

As visible in the revised output, the categories are now sequenced starting with the highest frequency count (White, 57,678 observations) and ending with the lowest (Other, 2,752 observations). Using **ORDER=FREQ** is particularly effective when working with variables that have numerous levels, as it quickly highlights the predominant groups within the output. This capability greatly enhances the interpretability of large datasets.

Example 3: Including Missing Values in Frequency Analysis

A key distinction of the PROC FREQ procedure is its handling of missing data. By default, PROC FREQ excludes observations containing missing values for the variables specified in the analysis. When generating frequency tables, this means that any observation where the variable (e.g., Race) is null will not be included in the total count or percentage base shown in the standard

output.

However, it is often critical for data quality checks and complete reporting to explicitly document the presence and count of missing observations. To force SAS to include a dedicated row for missing values in the frequency table, we utilize the **MISSING** option. This option is placed immediately after the variable name in the **TABLES** statement, separated by a forward slash (/).

We will combine the **MISSING** option with the **ORDER=FREQ** option used previously to demonstrate how these customizations work together. Notice that the **ORDER=FREQ** option remains on the **proc freq** statement, while **MISSING** is applied specifically to the table being generated for **Race**:

```
/*Frequency table for Race, sorted by frequency, explicitly showing missing values*/
proc freq data=sashelp.BirthWgt order=freq;
tables Race / missing;
run;
```

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

Upon reviewing the output generated with the **MISSING** option, we observe that no additional row was appended to the frequency table for Race. This outcome is highly informative: it confirms empirically that the **Race** variable in the **sashelp.BirthWgt** data contains zero missing values. If missing values had been present, a row labeled 'Missing' would appear, detailing its frequency and percentage relative to the total number of observations in the sample.

Example 4: Generating Frequency Tables for Multiple Variables Simultaneously

One of the efficiency features of PROC FREQ is its capacity to generate multiple one-way frequency tables within a single execution block. This capability significantly streamlines the

analytical workflow when an analyst needs to summarize the distribution of several independent variables from the same data source.

To instruct SAS to perform this batch analysis, simply list all required variable names separated by a space on the **TABLES** statement. For instance, if we wish to examine both the distribution of **Race** and the distribution of the mother's **AgeGroup**, we include both names. We will also retain the **ORDER=FREQ** option from the previous examples to ensure both resulting tables are sorted by frequency count for maximum clarity:

/*Generating sorted frequency tables for both Race and AgeGroup variables*/

proc freq data=sashelp.BirthWgt order=freq;

tables Race AgeGroup;

run;

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

AgeGroup	Frequency	Percent	Cumulative Frequency	Cumulative Percent
2	75633	75.63	75633	75.63
3	14122	14.12	89755	89.76
1	10245	10.25	100000	100.00

The output confirms that SAS successfully processed the request, producing two distinct frequency tables. The first table summarizes the **Race** variable (as seen previously), and the second table provides the comprehensive frequency distribution for **AgeGroup**. Both tables adhere to the **ORDER=FREQ** sorting criteria specified in the main procedure statement, demonstrating the efficiency of applying global options to multiple analyses simultaneously.

Further Applications: Creating Two-Way Frequency (Crosstabulation) Tables

While one-way tables summarize single variables, frequency tables gain substantial analytical power when used for two or more variables simultaneously, creating what is known as a crosstabulation or contingency table. This allows analysts to examine the joint distribution of two categorical variables, revealing relationships and interactions between groups.

To generate a two-way table in SAS, the syntax in the **TABLES** statement simply replaces the space separation with an asterisk (*). For instance, to cross-tabulate **Race** by **AgeGroup**, the command would be `TABLES Race*AgeGroup;`. The resulting output matrix would show the frequency count for every combination of race and age group.

Crosstabulation tables generated by the PROC FREQ procedure also inherently offer options for displaying various percentages--such as cell percentage, row percentage, and column percentage--which are critical for interpreting conditional probabilities and independence between the variables. These options are activated using additional keywords following the slash (/) in the **TABLES** statement, such as **CHISQ** for association tests or **ALL** for comprehensive descriptive statistics.