

Create Frequency Tables in SAS (With Examples)

Authored by
Mohammed looti

November 1, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Create Frequency Tables in SAS (With Examples)*.
PSYCHOLOGICAL STATISTICS. Retrieved from
<https://statistics.arabpsychology.com/?p=7689>

In the realm of [statistical computing](#) and **data analysis**, the ability to generate concise summaries of categorical data is a fundamental requirement. The [proc freq](#) procedure in [SAS](#) represents the most efficient and robust tool available for this purpose. It allows analysts to rapidly create [frequency tables](#) for one or more variables within a specified [dataset](#). This procedure is indispensable for initial data exploration, data validation, and comprehensively reporting the distribution of qualitative variables.

A frequency table offers a simple yet comprehensive overview of how values are distributed across distinct categories, providing raw counts, crucial percentages, and cumulative statistics. Mastering the syntax and options associated with this procedure is essential for any professional working with quantitative data in the SAS environment, as it provides an immediate and clear understanding of data composition before undertaking complex statistical modeling.

Understanding the Sample Data: The BirthWgt Dataset

To effectively demonstrate the versatility and power of **proc freq**, we will utilize the widely accessible, built-in SAS sample [dataset](#) known as `sashelp.BirthWgt`. This robust dataset comprises records for 100,000 mothers who recently gave birth, tracking various characteristics pertinent to maternal and infant health research. Analyzing this specific data allows us to understand real-world distributions, such as race demographics or age groupings, establishing a strong foundation for our practical examples.

Prior to executing detailed frequency analyses, it is always standard practice to inspect the structure and initial content of the source data. This preliminary data exploration step ensures that variables are correctly loaded and interpreted, and it verifies the integrity of the [dataset](#) structure. We can use the [proc print](#) command to visualize the first few observations.

The following SAS code snippet illustrates the use of `OBS=10`, an important option used to limit the output display. This provides a quick, manageable snapshot of the variable types and the overall structure of the data:

```
/*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

Example 1: Generating a Basic Univariate Frequency Table

The most straightforward and common application of the [proc freq](#) procedure is generating a univariate table, which serves to summarize the distribution of a single categorical variable. For our initial demonstration, we will focus specifically on the `Race` variable housed within the `sashelp.BirthWgt` dataset. The core syntax is remarkably simple, requiring only the procedure name and the mandatory `TABLES` statement, followed immediately by the variable of interest.

This initial analysis provides vital statistical context by helping us understand the demographic composition of the observed mothers. By default, [SAS](#) sorts the categories alphabetically based on the variable values and automatically calculates four standard metrics essential for statistical interpretation and reporting.

We utilize the following precise code to execute this basic [frequency table](#) analysis 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

Interpreting the Standard PROC FREQ Output Metrics

Upon execution, the resulting output table is clearly structured into four distinct columns. Each column serves a specific statistical and descriptive purpose, providing a complete picture of the variable's distribution:

Frequency: This metric represents the **absolute count**, indicating the total number of observations that fall precisely into a given category. It is the raw numerical count of occurrences.

Percent: This calculates the **relative proportion** of the total non-missing observations accounted for by the specific category, expressed as a percentage. This powerful metric allows for unbiased comparison of distributions, irrespective of the overall sample size.

Cumulative Frequency: This is a **running total**, representing the sum of the frequencies up to and including the current row. This is particularly useful when analyzing ordinal data or quickly determining how many observations fall below a specific threshold.

Cumulative Percent: This represents the **running total of the percentages** up to and including the current category. It efficiently indicates the combined percentage of observations accounted for by the current category and all preceding categories.

Interpreting the statistics derived from this output provides immediate and actionable insights into the demographic structure of the dataset:

The raw count (Frequency) shows that the total number of Hispanic mothers was **22,139**.

The percentage of total mothers who were Hispanic was **22.14%**.

Using the Cumulative Frequency, we find that the total number of mothers who were Asian, Black, or Hispanic aggregates to **41,496**.

This combined group accounts for a Cumulative Percent of **41.50%** of the entire sample size.

Example 2: Controlling Output Order using the ORDER= Option

By default, **SAS** sorts the rows in a generated **frequency table** based on the alphabetical or numeric order of the category names. While this default ordering is systematic, it frequently fails to prioritize the statistically most relevant categories, especially when the primary analytical goal is the rapid identification of the largest population groups or modes.

To significantly enhance the statistical clarity and overall readability of the output, particularly when dealing with variables that exhibit high cardinality, analysts can leverage the `ORDER=` option within the `PROC FREQ` statement. This option grants the user control to specify alternative sorting criteria. The most common alternative is setting `ORDER=FREQ`, which ensures that the output categories are sorted in descending order based on their raw frequency count. Consequently, the most common categories immediately appear at the top of the table.

The following modified code illustrates the application of the `ORDER=FREQ` option. Sorting the `Race` variable output by frequency allows analysts to quickly and intuitively identify the dominant categories within the population, improving the speed of data interpretation:

```
/*create frequency table for Race variable, sorted by frequency*/
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 clearly demonstrated by the resulting table, the categories are now arranged precisely from the highest frequency (White, 58,504 observations) down to the lowest. This frequency-sorted presentation is widely preferred in formal statistical reports and presentations, as it immediately draws the reader's attention to the most significant proportions of the data, providing a more intuitive sense of the data's distribution and skewness.

Example 3: Handling Missing Values with the MISSING Option

In all stages of data quality assurance, understanding and accurately quantifying [missing values](#) is critically important for maintaining data integrity and preventing analytical bias. By default, when the `proc freq` procedure encounters an observation where the variable is missing (typically represented by a dot for numeric variables or blanks for character variables), it automatically excludes that observation entirely from both the frequency counts and all subsequent percentage calculations.

To ensure truly comprehensive data reporting and to validate the quality of the input data, analysts frequently need to explicitly include the count of missing observations within the frequency table itself. This crucial step is accomplished by appending the `/ MISSING` option directly to the `TABLES` statement. When this option is properly utilized, a dedicated row for [missing values](#) is included in the output, and all percentages displayed are calculated based on the grand total number of observations, including those that are missing.

We incorporate the `MISSING` option alongside the previously defined `ORDER=FREQ` option to generate a frequency distribution that rigorously accounts for all 100,000 observations in the source [dataset](#), irrespective of whether they possess a valid value for the `Race` variable:

```
/*create frequency table for Race variable, sorted by frequency*/  
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 careful examination of the resulting table, we observe that no additional row specifically dedicated to missing observations has appeared. This outcome is highly positive, as it confirms that the `Race` variable within the `sashelp.BirthWgt` dataset is effectively complete, containing

zero instances of missing data. If missing data had been present, an explicit row would display the raw count and corresponding percentage, immediately alerting the analyst to potential data cleaning requirements before proceeding with deeper statistical modeling or reporting.

Example 4: Creating Frequency Tables for Multiple Variables Concurrently

A significant practical advantage of the [PROC FREQ](#) procedure is its impressive capacity for the rapid, simultaneous processing of multiple variables. Instead of requiring analysts to execute redundant, repetitive code steps for each variable, this procedure allows the generation of several univariate [frequency tables](#) in one highly efficient operation. This capability dramatically streamlines the initial exploratory data analysis (EDA) phase and saves significant time.

To achieve this concurrent processing, the analyst simply lists all desired variable names within the `TABLES` statement, ensuring they are separated by a space. For this final example, we expand our analysis to include both the `Race` variable and the `AgeGroup` variable. This provides a parallel demographic summary, enabling us to examine two separate distributions (race and age) using a single, consolidated block of code. We will also retain the `ORDER=FREQ` option to ensure both resulting tables are organized based on descending frequency counts for optimal visual interpretation.

The following integrated code demonstrates how to command [proc freq](#) to output two distinct frequency summaries concurrently:

```
/*create frequency table for Race and AgeGroup variables, both sorted by frequency*/  
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 resulting output clearly confirms that two independent frequency tables were generated. The first table accurately summarizes the distribution of `Race`, and the second details the distribution of `AgeGroup`. Crucially, both tables adhere to the specified sorting preference (descending frequency), providing immediate and comprehensive insight into the demographic structure of the maternal population captured in the [dataset](#). This powerful, integrated technique is indispensable for efficient data auditing and reporting tasks within the [SAS](#) programming environment.