How to use the NMISS Function in SAS (With Example)

Authored by

stats writer

November 19, 2025

The NMISS function in SAS is an indispensable component for quantitative assessment of data quality, specifically designed to count the number of missing values present within specified variables in a dataset. This function is essential during the initial stages of any statistical project, providing analysts with a reliable and rapid metric for gauging data completeness before proceeding to advanced modeling or descriptive data analysis. While NMISS accepts one or more variable names as arguments, its core strength lies in its optimization for handling numerical data, where a missing value is universally represented by the system missing symbol (a single period).

The primary utility of the NMISS function is its ability to return a single integer representing the cumulative count of omissions for the variables under scrutiny. For instance, invoking `NMISS(VAR_A)` would yield the total count of missing observations in VAR_A. This capability is foundational for effective data cleaning workflows, as identifying the scope and distribution of missingness is the crucial first step toward deciding on appropriate intervention strategies, such as imputation or exclusion. Proper application of NMISS ensures that subsequent statistical computations are grounded in a clear and accurate understanding of the data's integrity and limitations.

It is vital for SAS programmers to acknowledge that the default behavior of NMISS is tailored for numeric variables. When used within general procedures like PROC MEANS, it automatically reports missing counts only for fields designated as numeric. Addressing missingness in non-numeric fields, such as character strings, requires specific programming techniques, often leveraging the function within the DATA step or the flexible query environment provided by PROC SQL. Acknowledging this distinction is key to performing a comprehensive assessment that covers all variable types within a complex dataset.

The Critical Importance of Missing Data Assessment

Quantifying missing values is far more than a simple clerical exercise; it directly impacts the validity and reliability of statistical outcomes. The volume, pattern, and underlying mechanisms of data omission can drastically skew analytical results, potentially introducing biases, diminishing statistical power, and leading to inaccurate inferences. Therefore, a prerequisite to any reliable modeling or hypothesis testing exercise is a precise quantification of the missing data extent. The NMISS function offers the most direct and efficient mechanism within the SAS system for achieving this initial measurement for all numeric fields.

The count derived from NMISS provides immediate insight into the severity of data quality challenges. A small number of missing records might suggest random data entry errors, potentially manageable through simple imputation methods. Conversely, if NMISS reveals a high concentration of missingness in a particular variable, it points toward a significant systemic issue--such as a data collection failure or variable redundancy--which necessitates strategic re-evaluation

of that variable's utility in the analysis. Understanding these quantitative patterns requires not only the count itself but also the contextual knowledge provided by this descriptive data analysis tool.

Furthermore, integrating NMISS counts into data summaries supports robust documentation and quality assurance practices, which are essential in regulated industries like pharmaceuticals or finance. By incorporating the NMISS statistic into summary reports generated by procedures like PROC MEANS, analysts ensure transparency regarding data inputs and limitations. This fundamental step ensures that all subsequent decisions regarding data handling, including definitive exclusion or advanced imputation, are fully auditable and justifiable, enhancing the overall credibility of the research or report.

Utilizing PROC MEANS with the NMISS Option for Summary Statistics

The standard and most streamlined approach for obtaining a dataset-wide count of missing values across all numeric variables is by implementing the `NMISS` option directly within the PROC MEANS procedure. While the NMISS function can be used in the DATA step for row-wise calculation, leveraging the procedural option allows SAS to efficiently generate aggregate statistical summaries that include both the count of non-missing observations (N) and the count of missing observations (NMISS).

Appending the `NMISS` keyword to the PROC MEANS statement instructs the procedure to calculate and display the missing count for every eligible variable defined in the input dataset. This method removes the necessity for manual iteration or complex programming, positioning it as the preferred technique for exploratory data analysis (EDA). The resulting summary table furnishes key descriptive statistics--such as the mean, standard deviation, and range--while simultaneously providing the essential metrics of data completeness (N and NMISS).

The following syntax represents the core implementation of this functionality. This command directs SAS to calculate comprehensive descriptive statistics for the dataset named **my_data**, explicitly requesting the inclusion of missing value counts:

You can use the **NMISS** functionality in SAS to count the number of missing values for each numeric variable in a dataset efficiently.

Here is one common way to execute this function in practice:

```
proc means data=my_data nmiss;
run;
```

This particular example will count the number of missing values for each **numeric variable** in the dataset called **my_data**, producing a summary table critical for initiating data cleaning procedures.

Practical Demonstration: Constructing a Dataset with Intentional Missingness

To fully illustrate the mechanism of the NMISS function, we will establish a sample dataset detailing statistics for various basketball players. This dataset, designated **my_data**, includes a mix of numeric (`points`, `assists`, `rebounds`) and character variables (`team`) and is intentionally populated with several missing values, represented by a period in the input stream, to mimic typical real-world data imperfections.

The process initiates with the DATA step to define the dataset structure and variable types, followed by the DATALINES statement to input the raw data records. This setup is paramount for ensuring that SAS correctly interprets the variable attributes. Special attention is given to the input of missing numerical data using the period (`.`) symbol, which SAS recognizes as a system missing value. Note that one row also features a missing value for the character variable, further complicating the data structure.

Following data creation, the PROC PRINT statement is executed to display the raw contents of the dataset. This step visually confirms the successful compilation of the data table and permits a manual, preliminary verification of the pattern and location of the missing entries before the NMISS analysis is run.

Example: Using NMISS in SAS to Count Number of Missing Values for Numeric Variables

Suppose we have the following dataset in SAS called **my_data** that contains information about various basketball players:

```
/*create dataset*/
data my_data;
input team $ points assists rebounds;
datalines;
A 10 2 .
A 17 5 .
A 17 . .
A 18 3 4
A 15 0 5
B . 4 5
B 29 0 8
B . 2 9
C 12 1 9
. 30 1 .
```

```
;
run;

/*view dataset*/
proc print data=my_data;
```

Obs	team	points	assists	rebounds
1	A	10	2	.
2	A	17	5	.
3	A	17	.	.
4	A	18	3	4
5	A	15	0	5
6	B	.	4	5
7	B	29	0	8
8	B	.	2	9
9	C	12	1	9
10		30	1	.

It is readily apparent that missing values (indicated by the period) are present across the `points`, `assists`, and `rebounds` numeric variable columns, alongside a missing observation in the categorical `team` column.

Executing the NMISS Count via PROC MEANS

Following the successful creation of the sample dataset, the subsequent phase in the data cleaning methodology involves precisely quantifying the extent of numeric missingness. As previously detailed, this quantification is optimally performed using the PROC MEANS procedure augmented by the `NMISS` option. This combined command efficiently executes the internal NMISS counting logic across all eligible numeric variables (`points`, `assists`, and `rebounds`), consolidating the results into a concise and easily interpretable summary table.

Crucially, the character variable `team` is automatically excluded from this PROC MEANS analysis, as the procedure is fundamentally geared towards statistical measures pertinent to continuous data. The structure of the output table is designed to be highly informative, providing the necessary statistical metrics that allow analysts to assess not only the count of missing records but also the reliability and robustness of the calculated averages and distributions for each metric.

We execute the following code to apply the NMISS function through PROC MEANS, thereby calculating the total number of missing observations per numeric variable:

```
/*count number of missing values in each variable*/
proc means data=my_data nmiss;
run;
```

The MEANS Procedure

Variable	N Miss
points	2
assists	1
rebounds	4

Analyzing the Procedural Output and Missing Data Patterns

The output table generated by PROC MEANS serves as quantified evidence of the data deficiencies within the numeric fields. Each row in the table corresponds to one variable, and the essential column labeled **NMISS** displays the precise count of missing values determined by the NMISS logic. The column labeled **N** is equally important, representing the count of non-missing, or available, observations that were used in the calculation of summary statistics like the mean.

Reviewing this data allows analysts to draw definitive conclusions about the completeness of their numerical measures, informing subsequent strategic decisions. If the NMISS count for a variable is disproportionately large compared to N, analysts may consider the variable too sparse for meaningful inclusion in predictive models, potentially justifying variable exclusion or the use of more sophisticated, resource-intensive imputation techniques to salvage data utility.

Based on the quantitative summary derived from our example table, we can accurately detail the missingness across the dataset:

The **points** variable exhibits **2** missing values. This signifies that 8 out of 10 observations were available for calculating central tendency and dispersion measures.

The **assists** variable shows only **1** missing value. This variable is highly complete, which suggests high confidence in its calculated summary statistics.

The **rebounds** variable records **4** missing values. This represents the highest rate of missingness in the dataset, necessitating careful consideration during the data cleaning phase.

Counting Missing Character Variables Using PROC SQL

A significant limitation of the default NMISS function, particularly when integrated with PROC MEANS, is its inherent inability to process character variables. In SAS, a missing character value is defined as a field containing only blank spaces or an empty string, which is conceptually distinct from the numeric system missing value (`.`). To accurately extend the missingness assessment to character fields, expert analysts utilize PROC SQL, which facilitates the direct application of functions within a highly flexible query structure.

Within the PROC SQL environment, the NMISS function can be applied to character variables. In this specific context, NMISS evaluates whether the variable's value conforms to the definition of a missing character string. This method is indispensable for ensuring a holistic data quality assessment, encompassing organizational codes, textual identifiers, and categorical data fields, which are just as susceptible to missingness errors as numerical measurements.

The following example code illustrates the targeted use of the NMISS function on the `team` variable--a character field--to determine its missing observation count:

```
/*count number of missing values for team variable*/
proc sql;
select nmiss(team) as missing_team_values
from my_data;
quit;
```

missing_team_values
1

The output generated by the PROC SQL query confirms that there is exactly **1** missing value recorded in the **team** column. This concludes the comprehensive assessment of missing data across both numeric and character data types within the sample dataset.

Advanced Considerations and Alternatives

While the **NMISS** function is highly effective for basic counting, the SAS system provides additional specialized functions and procedures for sophisticated data validation and quality control, particularly when analysts must address complex missing data patterns or implement advanced imputation models. A notable alternative is the `CMISS` function, often used within the DATA step, which is designed to count missing values across both numeric variables and character variables

simultaneously when they are explicitly listed as arguments.

For research that requires detailed pattern analysis--such as identifying whether data is Missing At Random (MAR) or Missing Completely At Random (MCAR)--procedures like `PROC MI` (Multiple Imputation) or `PROC FREQ` offer deeper diagnostics. These advanced procedures move beyond simple counting to analyze the type and underlying mechanism of missing data, which is essential for determining the most appropriate statistical modeling approach. Nevertheless, the rapid, foundational count provided by NMISS remains the quickest method for preliminary assessment of numeric data integrity.

In conclusion, mastering the use of **NMISS**, whether integrated into PROC MEANS for efficiency or utilized within PROC SQL for character-specific assessment, is a non-negotiable step in any professional statistical workflow. This simple yet powerful tool ensures efficient data cleaning and establishes a defensible foundation for subsequent statistical analysis.

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