A Practical Guide to Handling Missing Data: Removing Rows with Missing Values in SAS

Authored by

Mohammed looti

November 16, 2025

Achieving high data quality is the fundamental prerequisite for any robust analytical endeavor. Yet, one of the most persistent and pervasive obstacles faced by data analysts and statisticians is the unavoidable presence of missing values within datasets. These data gaps can arise from numerous sources, including incomplete data entry, non-response bias in surveys, or corrupted system imports. Regardless of their origin, missing values have the potential to severely undermine the reliability, validity, and statistical power of all subsequent analyses. Fortunately, the SAS statistical software environment provides an efficient and powerful set of tools designed specifically to manage and resolve these data integrity challenges.

This comprehensive guide details the precise methodology for executing a complete-case analysis by systematically removing all observations (rows) that contain any missing values from a SAS dataset. Our primary focus will be on leveraging the highly efficient and straightforward syntax that utilizes the powerful CMISS function. Mastering this technique is an essential step in streamlining your overall data cleaning pipeline, enabling rapid preparation of pristine data suitable for sophisticated modeling and reporting tasks.

The core strategy for performing this 'listwise deletion'--ensuring that every single row is entirely free of missing entries across all variables--is executed within a specialized DATA step in SAS. This powerful methodological approach facilitates the creation of a refined, new dataset by sequentially processing the original data and systematically excluding any observation that fails the stringent data integrity check. The necessary syntax is remarkably concise and represents a foundational skill for any SAS programmer focused on high-quality data preparation:

```
data new_data;
set my_data;
if cmiss(of _all_) then delete;
run;
```

This block of code efficiently generates a new output dataset, designated as **new_data**, by drawing and processing records from the source dataset, **my_data**. The most critical operation is the application of the CMISS function across **all** variables (indicated by the special keyword _**all**_). If the CMISS function detects even a single missing entry within a row, that specific observation is immediately terminated and prevented from being written to the output file using the `then delete` statement. Consequently, **new_data** is guaranteed to contain only complete observations, which is vital for ensuring the statistical reliability of all subsequent calculations and modeling exercises. We will now explore the implications of missing data and the mechanics of the CMISS function in greater detail.

The Critical Significance of Missing Data and Its Impact on Analysis

Before deploying any technical solution, it is essential to fully grasp the inherent characteristics and potential negative consequences of missing values. Within the SAS programming environment, missing data are explicitly denoted using specific conventions: numeric missing values are represented by a single period (**.**), whereas character missing values are represented by a blank space. A failure to properly identify and handle these data gaps can severely distort statistical outcomes, leading to inefficient parameter estimates, a significant reduction in statistical power, and potentially guiding analysts toward fundamentally flawed conclusions regarding the underlying data structure and relationships.

The origins of missing data are highly diverse and often complicated to trace, ranging from participant refusal to answer sensitive questions in survey research to system errors during automated collection processes. While determining the specific mechanism of missingness--such as whether the data are Missing Completely at Random (MCAR) or Missing Not at Random (MNAR)--is crucial for highly advanced statistical treatment, the initial phase of data cleaning frequently involves a direct, albeit sometimes conservative, removal of incomplete records. This technique is formally referred to as 'listwise deletion' or the 'complete-case analysis' method.

Maintaining absolute analytical integrity necessitates rigorous data cleaning protocols. As demonstrated, removing rows that contain any missing data is one of the most straightforward and effective methods to ensure a dataset is fully populated and reliable. However, analysts must proceed with extreme caution: this technique relies on the assumption that the missingness mechanism is non-systematic (ideally MCAR). If the data are systematically missing (e.g., if specific demographic groups consistently fail to report certain variables), simply deleting these rows can introduce substantial selection bias into the resulting sample, potentially invalidating the entire scope of the analysis. Therefore, this powerful data preparation method must be applied judiciously, balancing the need for completeness against the risk of bias.

Leveraging the CMISS Function for Comprehensive Missing Value Detection

The cornerstone of our solution is the CMISS function, a specialized utility embedded within SAS that is engineered specifically to count the total number of missing arguments supplied to it within a given observation. This function offers remarkable versatility, capable of simultaneously evaluating both numeric variables (identifying the period marker '.') and character variables (identifying blank spaces). This dual capability is crucial, as it facilitates a truly comprehensive and single-step detection of missingness across heterogeneous data types present in a single row.

The standard syntax for invoking the CMISS function follows the structure: `CMISS(argument-1, argument-2, ..., argument-n)`. Each argument placeholder can be populated by a single variable name, a defined list of variables, or a variable array. The function's output is a simple

integer value that precisely quantifies how many of the specified inputs are missing for the current observation under processing. Critically, a return value of **0** serves as confirmation that no missing values were found among the specified arguments, while any integer greater than unambiguously signals the presence of one or more missing entries within that specific data record.

To effectively meet our objective--which is removing any row containing *any* missing value, irrespective of variable type or column location--the special argument **_all_** becomes indispensable. When this argument is incorporated into the function call, written as `cmiss(of _all_)`, it instructs SAS to automatically evaluate every single variable currently loaded in the input buffer. This powerful shorthand provides the most concise and resource-efficient method available in SAS for identifying observations that are incomplete in any aspect whatsoever, drastically reducing the programming effort required compared to manually listing hundreds of variables.

Implementing the Clean Sweep: Detailed DATA Step Syntax

The core mechanism for executing this critical row removal technique is the SAS DATA step, which is recognized as the foundational building block for all data creation, transformation, and modification processes within the SAS programming language. A clear understanding of the purpose and function of each statement within the syntax is paramount to achieving effective and reliable implementation:

```
data new_data;
set my_data;
if cmiss(of _all_) then delete;
run;
```

The DATA step is formally initiated with the statement **`data new_data;`**, which explicitly declares the programmer's intention to generate a new SAS dataset named **new_data**. Following this, the **`set my_data;`** statement commands SAS to begin reading and processing observations iteratively from the specified source dataset, **my_data**. This iterative, row-by-row processing loop is central to SAS's robust data manipulation capabilities.

The core conditional logic is housed within the statement **`if cmiss(of _all_) then delete;`**. As SAS processes each individual row, the CMISS function performs a rigorous check across every variable (thanks to the **_all_** argument) for any sign of missingness. If the function determines that one or more missing values are present (meaning CMISS returns a value greater than 0), the conditional statement is satisfied, and the **`then delete;`** command is immediately executed. This command crucially prevents the current observation from being written to the output

dataset, **new_data**. The process is finalized with the `run;` statement, which executes the entire DATA step and officially completes the creation of the cleaned dataset.

Practical Demonstration: Cleaning a Basketball Performance Dataset

To solidify the understanding of this essential technique, let us apply the method to a realistic data scenario. Imagine we are analyzing a dataset within SAS that tracks key performance metrics for professional basketball teams. This dataset, which we name **my_data**, includes columns detailing the team name, total points scored, and total assists. Due to common data collection issues--such as scoring errors or incomplete record submission--several observations within this dataset contain critical missing values.

Our initial procedural step involves generating and reviewing this sample dataset to visually confirm the rows that suffer from missing data. It is important to recall that for numeric variables like 'points' and 'assists', the missing entries are represented by the standard SAS period (**.**). Observing the initial dataset is crucial, as it provides the baseline against which we will measure the successful outcome of our data cleaning operation:

```
/* Create sample dataset named MY_DATA */
data my_data;
input team $ points assists;
datalines;
Mavs 113 22
Pacers 95 .
Cavs . .
Lakers 114 20
Heat 123 39
Kings . 22
Raptors 105 11
Hawks 95 25
Magic 103 26
Spurs 119 .
;
run;

/* View the initial dataset contents */
proc print data=my_data;
```

Obs	team	points	assists
1	Mavs	113	22
2	Pacers	95	.
3	Cavs	.	.
4	Lakers	114	20
5	Heat	123	39
6	Kings	.	22
7	Raptors	105	11
8	Hawks	95	25
9	Magic	103	26
10	Spurs	119	.

As the output table confirms, several rows contain problematic missing values. Specifically, teams such as the Pacers and Spurs are missing 'assists', the Kings are missing 'points', and the Cavs are entirely missing data for both performance metrics. Our objective is now clearly defined: we must utilize the powerful SAS DATA step combined with the CMISS function to generate a new dataset that exclusively retains the observations where all variables are complete and fully populated.

We now implement the core logic required for listwise deletion to remove all incomplete observations. This automated process will systematically generate the clean dataset, **new_data**, ensuring it is immediately suitable for rigorous analytical use:

```
/* Execute the DATA step to filter out rows with any missing values */
data new_data;
set my_data;
if cmiss(of _all_) then delete;
run;

/* View the resulting cleaned dataset */
proc print data=new_data;
```

Obs	team	points	assists
1	Mavs	113	22
2	Lakers	114	20
3	Heat	123	39
4	Raptors	105	11
5	Hawks	95	25
6	Magic	103	26

The final output clearly validates the successful execution of the code. Every single row that previously exhibited missing values has been accurately and completely excluded from the resulting dataset. The new dataset, **new_data**, now consists exclusively of complete observations, establishing a strong and reliable foundation for conducting rigorous statistical analysis without the complicating factor of data gaps or integrity concerns.

Advanced Considerations and Alternative Missing Data Strategies

The technique utilizing `cmiss(of _all_) then delete;` provides an exceptionally robust and highly efficient methodology for managing missing values by mandating a complete-case analysis. Its inherent power stems from its simplicity: it checks for missingness across every single variable in the dataset with minimal programming overhead. However, while listwise deletion is the most straightforward approach, analysts must recognize that it is not always the optimal choice for every dataset, especially those suffering from a significant amount of data loss.

Deleting entire rows based on the presence of even a single missing entry can sometimes be an overly aggressive strategy. If a source dataset has a high frequency of missing data, this approach can drastically reduce the effective sample size, consequently diminishing the statistical power of the analysis and potentially introducing severe bias if the mechanism of missingness is non-random. In scenarios where overall data loss is significant or where missingness is strictly limited to variables considered auxiliary or less critical to the primary research question, analysts should explore more nuanced and conservative approaches.

One common refinement involves restricting the missingness check only to a select, crucial group of variables (e.g., `if cmiss(points, assists) then delete;`), thereby retaining observations that are only missing non-essential or auxiliary data. Alternatively, a more sophisticated and frequently preferred strategy is statistical imputation, where analysts estimate and systematically fill in the missing values based on established statistical methods, such as mean substitution, regression modeling, or multiple imputation techniques. Analysts should always carefully weigh the characteristics of their data, the presumed mechanism of missingness, and the trade-offs between

sample size reduction and potential bias before finalizing their data preparation strategy. The method presented here is best utilized when the quantity of missing data is minimal or when the research objective strictly mandates a truly complete-case analysis.

Further Resources for Advanced SAS Data Management and Data Cleaning

Achieving expertise in SAS data management demands a comprehensive understanding of various specialized functions and procedural steps that go far beyond simple row deletion. To significantly enhance your capabilities in complex data manipulation and ensure adherence to the highest standards of data cleaning, we strongly recommend exploring related topics and advanced tools available within the SAS environment:

Exploring methods for handling specialized types of missing values, such as extended numeric missing values (A-Z, underscore), which require distinct handling procedures.
Utilizing the **NMISS** function, which is specifically optimized for counting only numeric missing values, offering a precise alternative to `CMISS` when character variables are irrelevant to the missingness check.
Implementing advanced techniques for statistical imputation, which includes complex methods like hot-deck imputation and multiple imputation, designed to preserve the original sample size while statistically estimating missing entries.
Developing proficiency in conditionally filtering observations using the **WHERE** statement and other advanced logical constructs.
Learning essential procedures for restructuring datasets, such as employing **PROC TRANSPOSE** to effectively reshape data between wide and long organizational formats.

By dedicating time to these supplementary tutorials and documentation, you will cultivate a deeper, more robust understanding of SAS's extensive data processing capabilities, enabling you to confidently address even the most challenging and intricate data preparation tasks encountered in professional analysis.