Learning How to Remove Variable Labels in SAS: A Step-by-Step Guide

Authored by

Mohammed looti

September 10, 2026

RECOMMENDED CITATION

Mohammed looti (2026). *Learning How to Remove Variable Labels in SAS: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from https://statistics.arabpsychology.com/?p=4223

Introduction to Variable Labels in SAS

Variable labels are a powerful feature within the SAS system, designed to provide descriptive context for variable names, particularly in generated reports or outputs. While variable names are limited to specific character counts and syntax rules, the associated label can be much longer and more informative, ensuring clarity for end-users or stakeholders reviewing statistical results. For instance, a variable named **'temp_c'** might have the label **'Average Daily Temperature in Celsius'**, making the output significantly easier to interpret.

However, there are specific scenarios in data processing and preparation pipelines where these descriptive labels must be temporarily or permanently removed. This often occurs when preparing data for export to external systems that do not recognize or correctly handle SAS metadata, or when simplifying data structures for advanced analytical procedures that require minimal overhead. Maintaining clean metadata is crucial for efficient data management, and the ability to selectively manage attributes like labels is fundamental to professional SAS programming.

Fortunately, the SAS environment provides highly efficient and specialized tools for metadata manipulation. The most robust and standard approach involves leveraging the **PROC DATASETS** procedure in conjunction with the **ATTRIB statement**. This combination allows programmers to precisely target and modify variable characteristics, including removing labels, either for a single variable or across an entire dataset, providing granular control over the data structure.

Understanding the Importance of PROC DATASETS

The **PROC DATASETS** procedure is the essential utility for managing and manipulating SAS libraries and datasets without requiring the data itself to be read or processed entirely. Unlike standard data manipulation procedures like **PROC DATA**, which create new copies of the dataset after modification, **PROC DATASETS** uses the **MODIFY** statement. This key feature allows in-place modification of metadata, which is highly efficient, especially when dealing with very large datasets where copying the entire file is resource-intensive and time-consuming.

When removing variable labels, we utilize the **MODIFY** statement followed by the dataset name, and then the **ATTRIB statement**. The ATTRIB statement is the mechanism through which various attributes--such as format, informat, length, and label--can be set or reset. To effectively remove a label, the programmer simply specifies the variable name (or variables) and sets the **LABEL=** option to a null string, indicated by two single quotes (**''**).

The syntax is straightforward, yet powerful. Below are the two primary methods for label removal utilizing this procedure. These methods are preferred because they directly interact with the dataset descriptor information, offering a clean and permanent removal of the label attribute from the variable definition within the SAS file structure.

Method 1: Remove Label from One Variable

```
proc datasets lib=work;
modify original_data;
attrib my_variable label='';
```

Method 2: Remove Label from All Variables

```
proc datasets lib=work;
modify original_data;
attrib _all_ label='';
```

Prerequisites: Setting Up the Sample Dataset

To demonstrate these label removal techniques effectively, we must first establish a sample dataset that includes assigned variable labels. This sample data, which we will call **original_data**, contains three numerical variables (x, y, and z) representing hypothetical sports statistics. Each variable is given a descriptive label during the data creation step using the **LABEL** statement within the **DATA** step.

This initial setup step is critical because it visually confirms the existence of the labels before we attempt to remove them. When we run **PROC CONTENTS** on this initial dataset, the output will clearly show the metadata, including the variable names and their corresponding descriptive labels. This allows us to verify that our removal procedures are successful later on. The variables are defined as follows: **x** (Rebounds), **y** (Points), and **z** (Assists).

The following code block executes the data creation process and then calls **PROC CONTENTS** to display the current state of the variable attributes. Note the use of the **LIB=WORK** option in the PROC DATASETS examples; this ensures we are modifying the temporary dataset stored in the default SAS work library.

```
/*create dataset*/
data original_data;
label x='REBOUNDS'
y='POINTS'
z='ASSISTS';
input x y z;
datalines;
6 22 5
8 14 9
```

```
9 31 10
9 40 7
3 12 3
2 20 5
;

/*view contents of dataset*/
proc contents data=original_data;
```

The execution of **PROC CONTENTS** confirms that all three variables currently possess their assigned labels, as shown in the output summary below. This initial state validates the data structure before we proceed with the modification step.

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Label
1	x	Num	8	REBOUNDS
2	y	Num	8	POINTS
3	z	Num	8	ASSISTS

Method 1: Removing a Label from a Single Variable

In many analytical projects, the requirement is not to strip all metadata, but rather to clean up or reset the attributes for a specific subset of variables. This precision minimizes unintended consequences on other variables whose labels may be essential for subsequent reporting tasks. Method 1 addresses this exact need by targeting only one variable for label removal.

To achieve this, we initiate **PROC DATASETS** and specify the target library (**LIB=WORK**) and the dataset (**original_data**) using the **MODIFY** statement. The critical component is the **ATTRIB statement**, where we name the variable (in this case, **x**) and explicitly set its **LABEL** attribute to a null string (**''**). This action overwrites the existing label metadata for variable **x**, effectively removing it.

We will specifically target the variable **x**, which currently holds the label **'REBOUNDS'**. The syntax is concise and immediately executes the metadata change upon submission. Following the modification, we can rerun **PROC CONTENTS** to confirm that the label for **x** is now empty, while the labels for **y** and **z** remain intact, demonstrating the targeted precision of this method.

The following code snippet demonstrates the process of using **PROC DATASETS** to remove the

label specifically from the variable 'x' in our sample dataset:

```
proc datasets lib=work;
modify original_data;
attrib x label='';
```

Upon reviewing the updated contents of the dataset, it is evident that the label previously associated with variable **x** has been successfully cleared. Variables **y** and **z** retain their labels, confirming that the modification was isolated and precise, which is often essential for maintaining the integrity of complex data models.

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Label
1	x	Num	8	
2	y	Num	8	POINTS
3	z	Num	8	ASSISTS

Method 2: Global Removal of All Variable Labels

There are scenarios, particularly when preparing a final dataset for distribution or integration into non-SAS environments, where all variable metadata, including labels, must be uniformly stripped. This is often necessary to avoid conflicts or unnecessary complexity in receiving systems. Method 2 provides a highly efficient solution for global label removal using a special keyword.

This method again utilizes the **PROC DATASETS** and **MODIFY** statements, but differs in the application of the ATTRIB statement. Instead of specifying individual variable names, we employ the special variable list keyword _**ALL**_. When used in the ATTRIB statement, _**ALL**_ instructs SAS to apply the specified attribute change--in this case, setting **LABEL=''**--to every variable present in the dataset.

The efficiency of using _**ALL**_ is paramount, especially when dealing with datasets containing hundreds or thousands of variables. Manually listing every variable would be impractical and prone to error. By using this keyword, the programmer ensures comprehensive removal of all descriptive labels with minimal code, maintaining clean, streamlined programming practices.

The following code demonstrates the syntax required to use _**ALL**_ for complete label removal across the dataset:

```
proc datasets lib=work;
modify original_data;
attrib _all_ label='';
```

After executing this global modification, a final check using **PROC CONTENTS** will reveal that none of the variables--x, y, or z--retain any descriptive labels. This confirms the successful application of the global attribute change, resetting the metadata for the entire dataset simultaneously.

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
1	x	Num	8
2	y	Num	8
3	z	Num	8

Advanced Considerations and Best Practices

While the **PROC DATASETS** approach is the standard for permanent, in-place metadata modification, SAS offers alternative techniques that might be suitable for temporary label suppression or specific reporting needs. For example, if the goal is simply to prevent labels from appearing in a single report, many reporting procedures (like **PROC PRINT** or **PROC REPORT**) offer options such as **LABEL NO** or **NO LABEL**, which suppress the display of labels without altering the dataset's metadata structure.

It is important to remember that when using **PROC DATASETS**, the change is permanent unless the dataset is immediately overwritten or restored from a backup. Therefore, careful consideration of the target library is essential. When working in a production environment, programmers should always test these destructive metadata changes in a temporary library (like **WORK**) or ensure they have write permission to the specific library being modified, particularly if it is a shared or master data repository.

Furthermore, the **ATTRIB statement** is highly versatile. If, instead of removing the label entirely, the requirement was to change the label, the same syntax is used, simply replacing the null string (**''**) with the new descriptive text. Understanding the power of the **ATTRIB** statement allows for comprehensive management of variable characteristics beyond just label removal, including adjusting variable lengths or applying custom formats and informats simultaneously.

Summary of Techniques and Further Resources

The most reliable and efficient way to manage variable labels in SAS is through the **PROC DATASETS** procedure combined with the **ATTRIB statement**. This method provides the flexibility to target single variables or apply changes globally using the _**ALL**_ keyword, all while executing modifications in place to conserve computing resources.

The techniques covered provide complete control over the variable metadata structure, ensuring data preparation meets rigorous quality standards, whether the final destination is a SAS procedure, an external database, or a simple report.

Note: You can find the complete documentation for **proc datasets** through the official SAS support website.

Additional Resources

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