

How to Easily Add Titles and Footnotes to Your SAS Output

Authored by
stats writer

December 1, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Add Titles and Footnotes to Your SAS Output*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=103020>

In the world of statistical programming, generating clear and professional output is just as important as the analysis itself. In SAS, you can dramatically enhance the readability and professional quality of your reports and graphs by using the **TITLE** and **FOOTNOTE** statements. These statements are fundamental tools for ensuring that every piece of output is properly labeled and contextualized for the end user. The TITLE statement is specifically designed to add descriptive text, labels, or captions to the top margin of your output, creating immediate clarity regarding the content being presented. Conversely, the FOOTNOTE statement functions similarly but places supplementary information, disclaimers, or data source citations at the bottom of the output page.

Both the **TITLE** and **FOOTNOTE** statements offer impressive flexibility, allowing users to include simple text strings, display system variables (like the current date or time), or even incorporate complex formatting options such as color, font size, and justification. For instance, a common application involves using the **TITLE** statement to clearly label the results, perhaps as "Analysis Results for Q3 Performance" or "Summary Statistics for Patient Cohort B." Meanwhile, the FOOTNOTE statement is invaluable for providing critical metadata, such as adding a note at the bottom of the output specifying the data extraction date, the SAS version used, or a citation for the original dataset source. Mastering these simple yet powerful statements is essential for any professional SAS programmer aiming for high-quality, easily interpretable reports.

Understanding the Basic Syntax of the TITLE Statement

The core function of the **TITLE** statement in SAS is to quickly associate a descriptive title with a specific procedure output, such as a table generated by PROC PRINT or a chart created by PROC SGLOT. When used without a number, the **TITLE** statement defines the primary, topmost title for the output. This statement is typically placed before the procedure step (PROC) that generates the output, ensuring that the title is rendered correctly alongside the results.

The structure of the basic **TITLE** statement is straightforward. It begins with the keyword `TITLE`, followed by the desired title text enclosed in quotation marks. If you define a title using only the `TITLE` keyword, it acts as `TITLE1`, setting the main heading. If you want to clear all previously defined titles or footnotes, you simply use the keyword without any text or options (e.g., `TITLE;`). This simple syntax is the foundation upon which more complex, multi-layered titling is built, providing immediate identification for any statistical report or graphical representation produced by the SAS system.

The following example demonstrates the most fundamental application of the TITLE statement, illustrating how to define and apply a single title to a procedure output. This method is ideal when only one line of descriptive text is required above the resulting table or graph, offering a clean and concise label for the data visualization or report section.

Method 1: Applying a Single Output Title

You can use the **TITLE** statement in SAS to quickly add a single, centered title to a table or chart. This approach is best suited for simple, direct labeling where no complex hierarchy or specialized formatting is necessary. The statement remains active for all subsequent procedure outputs until it is explicitly changed, replaced, or cleared. Therefore, careful placement within your SAS program is crucial to ensure that the correct title is associated with the appropriate output results.

This statement uses the following basic syntax structure when defining a singular title:

Syntax for Single Title Application

```
/* Define the primary title using the TITLE statement */
```

```
title "This is a title";
```

```
/* View the dataset using PROC PRINT, which inherits the defined title */
```

```
proc print data=my_data;
```

When this code executes, the string "This is a title" will appear centered at the top of the output generated by the PROC PRINT step. It is important to remember that the quoted text can contain any character string, and its default appearance (font, size, color) will be determined by the current ODS (Output Delivery System) style in use. While simple, this method ensures that every resultant table or graph is clearly labeled, which is a fundamental requirement for reproducible and professional reporting in SAS.

Method 2: Defining Multiple Titles and Hierarchy

For reports requiring greater detail or structured organization, SAS allows the definition of multiple title lines using numbered **TITLE** statements (TITLE1, TITLE2, TITLE3, and so on). This mechanism establishes a clear hierarchy, with TITLE1 always appearing highest on the page, followed sequentially by higher numbers. This is particularly useful for complex reports where the top line might be a general company or project name, the second line might detail the specific analysis type, and subsequent lines might provide dates or specific filters used.

The primary advantage of using numbered TITLE statements is the ability to apply distinct formatting options to each line independently. Unlike the single TITLE keyword, which assumes default formatting or applies globally defined styles, the numbered statements allow for granular control over aesthetics, including justification, font family, color, and size. This capability ensures that the most important title lines can be visually emphasized while supporting information remains legible yet less dominant.

The following example demonstrates how to define three separate title lines, each customized with unique attributes such as color, height (size), and text justification, providing a layered and visually appealing output heading.

Syntax for Multiple Styled Titles

```
/* Define titles with specific formatting options */  
title1 color="purple" height=25pt bold italic underline=1 "This is a Title";  
title2 font="Helvetica" justify=left height=18pt "Second Title";  
title3 color="green" justify=right height=14pt "Third Title";  
  
/* View dataset with the hierarchical titles applied */  
proc print data=my_data;
```

Advanced Styling Options for TITLE Statements

The real power of the numbered `TITLE` statements lies in the extensive range of options available for visual customization. These options allow programmers to control every aesthetic detail, making the output titles consistent with corporate branding or specific reporting standards. Understanding and utilizing these options ensures that titles not only convey information but also contribute to the overall professionalism of the statistical report.

When applying formatting to titles, the options must be placed between the `TITLEn` keyword and the title string itself. These options function as keywords that modify the subsequent title string until another title statement overrides them. This fine-grained control is critical for data presentation, especially when contrasting different sections or emphasizing key findings within the report. For example, using a distinct color for an alert or warning title line can immediately draw the user's attention to critical caveats.

Note the following key options that you can use to modify the appearance and placement of titles and footnotes:

COLOR: Specifies the font color. This accepts standard HTML color names (e.g., "red", "blue") or hexadecimal color codes for precise corporate branding.

HEIGHT: Controls the font size. This is often specified in typographical points (pt) or centimeters (cm) and dictates the prominence of the title line.

FONT: Determines the font family used for the title text (e.g., "Arial", "Helvetica"). The availability of fonts depends on the ODS destination (e.g., HTML, PDF, RTF).

JUSTIFY: Sets the horizontal alignment of the title. Options include `LEFT`, `RIGHT`, or `CENTER` (which is the default if omitted).

STYLE: Used for applying specific font styles. You can apply `BOLD`, `ITALIC`, or `UNDERLINE` to

modify the text presentation, ensuring certain words or phrases stand out within the title line.

These options can be combined freely on a single title line, allowing for highly complex and customized visual designs. It is generally recommended to define these styles at the beginning of the `TITLE` statement before the quoted text to ensure proper parsing by the `SAS` compiler.

Practical Application: Detailed Example 1 (Single Title)

To illustrate the simple application of the `TITLE` statement, we will walk through a complete `SAS` program. This example involves creating a small dummy `dataset` and then using the basic `TITLE` keyword in conjunction with `PROC PRINT` to produce a labeled output table. This method is the standard starting point for anyone generating routine data checks or quick reports in `SAS`.

The following code shows how to create and label a simple `dataset` called `original_data` and attach one descriptive title to its printed output. Notice the placement of the `TITLE` statement immediately before the `PROC PRINT` step; this ensures the title scope correctly covers the procedure that follows. The title will automatically be centered by default, providing a clean, non-distracting header for the resulting table.

```
/* Create dataset named original_data */
```

```
data original_data;
```

```
input team $ points rebounds;
```

```
datalines;
```

```
A 25 10
```

```
A 18 4
```

```
B 27 9
```

```
B 33 13
```

```
;
```

```
run;
```

```
/* Define the title for the next procedure */
```

```
title "This is a title";
```

```
/* View dataset using PROC PRINT */
```

```
proc print data=original_data;
```

```
run;
```

The result of this code execution is a standard printed table containing the team, points, and rebounds variables, with the phrase "This is a title" positioned clearly above the data columns. This method provides the quickest way to label output and is often sufficient for internal consistency checks or preliminary reporting. It establishes the context without requiring complex formatting

overhead.

This is a title			
Obs	team	points	rebounds
1	A	25	10
2	A	18	4
3	B	27	9
4	B	33	13

Practical Application: Detailed Example 2 (Multiple Styled Titles)

When generating official reports or presentations, standard centered titles may not suffice. This next example demonstrates the full potential of the numbered `TITLE` statements by applying three distinct title lines, each formatted to convey specific visual weight and hierarchy. We maintain the same underlying `dataset` structure but significantly enhance the header presentation.

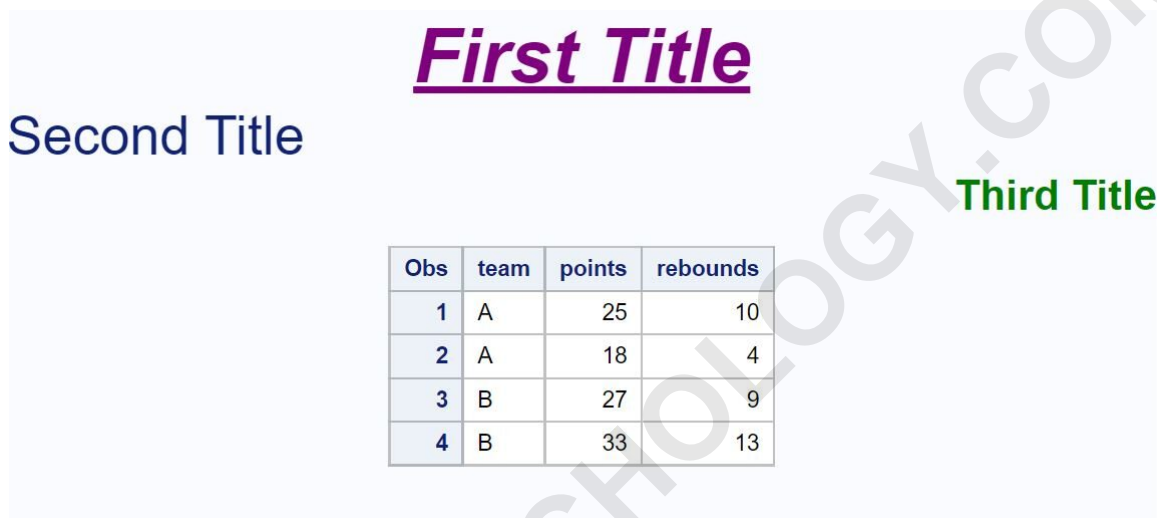
The following code shows how to implement multiple titles with custom designs, including specific colors, heights, and justifications, before calling `PROC PRINT`. Note how `TITLE1` uses large purple bold italic text, `TITLE2` is left-justified in a specific font, and `TITLE3` is smaller, green, and right-justified. This combination provides a rich, multi-dimensional header that instantly communicates different levels of information to the reader.

```
/* Create dataset named original_data */
data original_data;
input team $ points rebounds;
datalines;
A 25 10
A 18 4
B 27 9
B 33 13
;
run;

/* Define hierarchical and styled titles */
title1 color="purple" height=25pt bold italic underline=1 "First Title";
title2 font="Helvetica" justify=left height=18pt "Second Title";
title3 color="green" justify=right height=14pt "Third Title";
```

```
/* View dataset */  
proc print data=original_data;  
run;
```

The resulting output clearly displays the three distinct title lines, demonstrating how formatting options can be used effectively to differentiate between primary report titles, secondary descriptive information, and specific run parameters or dates, significantly improving the overall aesthetic and informational content of the report header.



The screenshot displays the output of a SAS PROC PRINT statement. It features three distinct title lines: a primary title 'First Title' in purple, a secondary title 'Second Title' in blue, and a tertiary title 'Third Title' in green. Below the titles is a data table with four rows and four columns: 'Obs', 'team', 'points', and 'rebounds'. The table contains the following data:

Obs	team	points	rebounds
1	A	25	10
2	A	18	4
3	B	27	9
4	B	33	13

Using the FOOTNOTE Statement for Supplementary Information

While the **TITLE** statement provides crucial context at the top of the output, the **FOOTNOTE** statement serves the equally vital role of adding supplementary details at the bottom of the page. Footnotes are essential for providing information that may distract from the main output visualization but is still necessary for data integrity, reproducibility, or legal compliance. Common uses include citing data sources, listing critical assumptions made during the analysis, or providing contact information for the report creator.

Similar to the **TITLE** statement, **FOOTNOTE** statements can be numbered (**FOOTNOTE1**, **FOOTNOTE2**, etc.) to allow for multiple lines of information, with **FOOTNOTE1** appearing closest to the output body. If the simple **FOOTNOTE** keyword is used without a number, it defaults to **FOOTNOTE1**. All formatting options available for titles (such as **COLOR**, **HEIGHT**, **FONT**, and **JUSTIFY**) are also fully functional within the **FOOTNOTE** statement, allowing for consistent branding or subtle presentation of secondary information.

A highly recommended practice is to use system variables within footnotes. For example, using the

`&SYSDATE` automatic macro variable allows the user to display the date the code was executed, enhancing the audit trail of the report. The syntax follows the same pattern as the title syntax, ensuring ease of use and consistency across the programming environment. For example: `FOOTNOTE "Report Generated on &SYSDATE";` provides dynamic, accurate documentation for every output generated.

Conclusion and Best Practices

The proper use of the **TITLE** and FOOTNOTE statements is a hallmark of professional SAS reporting. These simple statements move the output from being a raw data dump to a polished, contextually rich document. Always remember to prioritize clarity and relevance when crafting your titles and footnotes. Use **TITLE1** for the main topic and subsequent titles for descriptive sub-lines. Reserve footnotes for metadata and disclaimers.

Furthermore, it is a best practice to explicitly clear titles and footnotes when moving between different analytical blocks, especially if they are generating unrelated outputs. To clear all active titles, simply execute `TITLE;` (without quotes or options). To clear all active footnotes, execute `FOOTNOTE;`. This prevents titles defined for one procedure from inadvertently carrying over to a subsequent, unrelated procedure, ensuring that every piece of output remains accurately labeled and professionally formatted.

Mastering these techniques ensures that your statistical output is not only accurate but also highly presentable and easy to interpret.

Related SAS Tutorials

The following tutorials explain how to perform other common tasks in SAS, building upon the foundational knowledge of report formatting: