Use Proc Report in SAS (With Examples)

Authored by

Mohammed looti

March 28, 2026

RECOMMENDED CITATION

Mohammed looti (2026). *Use Proc Report in SAS (With Examples)*. PSYCHOLOGICAL STATISTICS. Retrieved from https://statistics.arabpsychology.com/?p=3347

The SAS environment provides powerful tools for statistical analysis and data manipulation, but few procedures match the flexibility and sophistication of the **PROC REPORT** procedure for generating high-quality, customized output. This procedure is specifically engineered to move beyond simple data listings, offering the ability to create complex, aggregated, and presentation-ready reports directly from your source data. Unlike basic data viewing tools, **PROC REPORT** grants the user precise control over virtually every aspect of the final product: from column layout and variable summarization to aesthetic formatting and group breaks. This makes it an indispensable asset for analysts who need to communicate findings clearly and professionally.

While simpler procedures like **PROC PRINT** are adequate for quick inspections or displaying raw rows, they lack the intrinsic capabilities for intricate formatting, complex aggregation, and conditional summarization that **PROC REPORT** excels at. When the goal shifts from merely displaying data to constructing a narrative or summarizing key metrics, leveraging the unique features of **PROC REPORT** becomes essential. It bridges the gap between raw data processing and polished, actionable business intelligence reporting, providing a significant step up in reporting capability within the SAS programming workflow.

The Foundational Syntax of PROC REPORT

At its core, **PROC REPORT** maintains a remarkably straightforward syntax, making it accessible even for users new to advanced SAS procedures. The most basic invocation requires only the procedure name and the specification of the input dataset. By default, when no additional statements are included, the procedure processes the entire dataset, displaying all observations (rows) and variables (columns) in their original order, very similar to a rudimentary data dump. This default behavior serves as the foundational baseline from which all customizations spring.

Understanding this foundational structure is the first step toward mastering the procedure. The basic command simply directs SAS to process the specified data object and apply default reporting rules, which include automatic column labels derived from variable names.

The following code illustrates the most basic invocation of **PROC REPORT**, demonstrating its default command structure:

```
/* Create a simple report */
proc report data=my_data;
run;
```

Executing this command produces a report that lists the observations from the specified dataset exactly as they are stored. While this simple output is functional, the true value of **PROC REPORT** becomes evident when analysts begin to incorporate customization statements. These statements

allow you to tailor the report's structure, filter content based on specific criteria, define informative column labels, and apply various display attributes to create a report that perfectly aligns with complex analytical and presentation needs. This extensibility is what establishes **PROC REPORT** as a go-to procedure for sophisticated reporting tasks.

For instance, consider the enhanced flexibility offered by including additional statements to refine the report's presentation, moving beyond the simple listing to a targeted summary:

```
/* Create a customized report for specific player statistics */
title 'Player Statistics for Dallas Mavericks';
proc report data=my_data;
where team='Mavs';
column conf team points;
define conf / display 'Conference' center;
run;
```

Essential Statements for Report Customization

The transition from a basic report to the customized example above is governed by several crucial statements that empower you to transform raw data into insightful summaries. Each statement plays a distinct, powerful role in shaping the final output, allowing the programmer to specify exactly what information is included and how it is visually represented.

Mastering these core statements is the key to unlocking the full potential of **PROC REPORT**. They provide the grammatical structure necessary to translate complex reporting requirements into executable SAS code.

The **TITLE** statement is used to assign a descriptive and prominent title to your report, significantly enhancing its readability and professional appearance. This is particularly useful for distinguishing between multiple reports or providing immediate context to your audience regarding the report's content.
The **WHERE** statement acts as a powerful conditional filter, allowing you to include only those observations (rows) that satisfy specific logical criteria. In our statistical example, using `WHERE team='Mavs'` ensures that only data relevant to that specific team is processed and displayed, creating a highly focused report.
The **COLUMN** statement is vital for explicitly selecting which variables (columns) to include in your report and specifying their precise display order from left to right. This ensures that only relevant information is presented and organized logically for optimal reader comprehension.
The **DEFINE** statement is arguably the most versatile command, offering granular control over how individual columns are presented. For example, `DEFINE conf / DISPLAY 'Conference' CENTER`

renames the internal variable 'conf' to the user-friendly label 'Conference' in the report output and horizontally centers its values, thereby improving visual clarity and aesthetics.

These statements collectively demonstrate the flexibility of **PROC REPORT** in tailoring output to specific requirements. The following section will walk through a practical, step-by-step example, illustrating how these concepts are translated into actionable code and resulting output. For a comprehensive understanding of all customization options, it is highly recommended to consult the official SAS documentation for PROC REPORT, which details hundreds of parameters for advanced reporting.

Practical Example: Setting Up the Sample Data

To demonstrate the robust capabilities of **PROC REPORT**, we will work with a sample dataset containing fictional basketball player statistics. This dataset, which includes details like team affiliation, points scored, rebounds, and conference, will serve as the foundation for generating both a basic default report and a highly customized, filtered report.

First, we must create the sample data within the SAS environment. We utilize a **DATA step** to input the statistics and then employ the simpler **PROC PRINT** procedure to display its raw contents, confirming the data structure before proceeding to report generation.

```
/* Create the sample dataset */
data my_data;
input team $ points rebounds conf $;
datalines;
Celtics 12 5 East
Celtics 14 7 East
Celtics 15 8 East
Celtics 18 13 East
Mavs 31 12 West
Mavs 32 6 West
Mavs 35 4 West
Mavs 36 10 West
Mavs 40 12 West
;
run;

/* View the dataset using PROC PRINT */
proc print data=my_data;
run;
```

Obs	team	points	rebounds	conf
1	Celtics	12	5	East
2	Celtics	14	7	East
3	Celtics	15	8	East
4	Celtics	18	13	East
5	Mavs	31	12	West
6	Mavs	32	6	West
7	Mavs	35	4	West
8	Mavs	36	10	West
9	Mavs	40	12	West

The image above visually confirms the complete structure of the 'my_data' dataset. We can observe four variables: 'team' (character), 'points' (numeric), 'rebounds' (numeric), and 'conf' (character). This raw data provides a clear view of the information we will transform using **PROC REPORT**.

Demonstrating Default and Customized Report Generation

Having successfully created and validated our source data, we can now proceed to generate reports that showcase the distinction between the default behavior of the procedure and its full customization potential. We begin by invoking the simplest form of **PROC REPORT**, which requires minimal code but establishes the baseline output structure.

This first step is crucial for understanding how the procedure interprets data when no specific instructions regarding filtering or formatting are supplied. It demonstrates the default assignment of column headers and the inclusion of all observations in their original sequence.

```
/* Create a report that displays the entire dataset */
proc report data=my_data;
run;
```

team	points	rebounds	conf
Celtics	12	5	East
Celtics	14	7	East
Celtics	15	8	East
Celtics	18	13	East
Mavs	31	12	West
Mavs	32	6	West
Mavs	35	4	West
Mavs	36	10	West
Mavs	40	12	West

As expected, this basic report simply reproduces the full dataset, similar to the output produced by the simpler **PROC PRINT**. It serves as a necessary baseline before we introduce custom elements designed to enhance clarity and focus.

To harness the true power of **PROC REPORT**, we will now generate a highly customized report. This report is specifically designed to focus exclusively on the 'Dallas Mavericks' players, display only a selected subset of relevant columns, and apply custom labels and alignment to improve the professional quality of the presentation.

```
/* Create a customized report */
title 'Player Statistics for Dallas Mavericks';
proc report data=my_data;
where team='Mavs';
column conf team points;
define conf / display 'Conference' center;
run;
```

Player Statistics for Dallas Mavericks

Conference	team	points
West	Mavs	31
West	Mavs	32
West	Mavs	35
West	Mavs	36
West	Mavs	40

Analyzing the Customized Report Output

Observe the significant improvements and targeted information presented in this customized report compared to the basic version. The final output is not just a data listing; it is a professional document designed for targeted analysis. Each visual enhancement directly corresponds to the statements we meticulously added to the **PROC REPORT** code, confirming the procedure's immense capability for output control.

A detailed analysis of the resulting report highlights the effect of each customization statement:

The report now proudly features a clear and informative title, "Player Statistics for Dallas Mavericks," which was applied using the **TITLE** statement. This immediately tells the viewer the report's purpose and context.
Crucially, the observations are focused exclusively on the Dallas Mavericks. This necessary data precision was achieved by employing the **WHERE** statement, which efficiently filtered out all data points related to other teams from the original dataset before the report was generated.
Only the 'conf', 'team', and 'points' variables are displayed, presented in that specific order. This streamlined, three-column view is the direct result of the **COLUMN** statement, ensuring that only the most relevant statistical data points are visible to the reader.
The 'conf' column is now correctly labeled 'Conference' and its values (West) are neatly centered for better presentation. These subtle but important visual improvements were implemented through the **DEFINE** statement using the **DISPLAY** and **CENTER** options, making the report significantly more professional and easier to consume.

This example merely scratches the surface of what is possible with **PROC REPORT** in SAS. The procedure offers extensive capabilities for complex aggregation, advanced summarization (using compute blocks), creating computed variables, and applying complex formatting rules, allowing you to generate virtually any report structure you can imagine, including cross-tabular layouts and

hierarchical grouping.

To unlock the full potential of this powerful procedure and discover how to further refine your outputs, we strongly encourage you to delve into the official SAS documentation for PROC REPORT. It provides a comprehensive explanation of all available statements and options, enabling you to produce reports that meet your most exacting specifications.

Additional Resources for SAS Programming

To further enhance your proficiency in the SAS environment, the following tutorials explain how to perform other common statistical and data management tasks: