

How to display values in percent format in SAS?

Authored by
stats writer

November 19, 2025

RECOMMENDED CITATION

stats writer (2025). *How to display values in percent format in SAS?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=96873>

In the realm of statistical computing, specifically when utilizing the SAS System, displaying numerical values in a professional and interpretable format is critical for effective data reporting. While data is typically stored internally as ratios (e.g., 0.50), human readability often requires these ratios to be presented as percentages (e.g., 50%).

To achieve this transformation within SAS, users rely on specialized formats. The primary tool for this task is the Percentw.d Format, which allows precise control over the visual representation of the percentage. Here, *w* denotes the total width allocated to the output field, and *d* specifies the number of decimal places that will be displayed after the decimal point.

For instance, applying the format statement `Percent7.2` informs SAS to reserve a total width of 7 characters for the final output, including the percentage sign, the decimal point, and any required signs, while ensuring that the percentage value is shown with exactly two digits following the decimal separator. This structure enables a raw value like 0.125 to be accurately and clearly presented as 12.50% in the resulting output table.

Understanding Percentage Formatting in SAS

Numerical data in SAS is inherently stored without predefined formatting instructions. When you load data that represents proportions or fractions--such as exam scores, market share, or response rates--these values exist as floating-point numbers between 0 and 1. To convey the meaning of these numbers effectively to non-technical stakeholders, it is essential to apply a display format that multiplies the stored value by 100 and appends the '%' symbol.

The PERCENT format option in SAS is specifically designed to handle this conversion automatically. It is a powerful tool because it controls only the appearance of the data during viewing or printing; it does not alter the underlying numerical value stored in the Dataset. This distinction is crucial: calculations can be performed on the raw proportional values, while reports display the formatted percentages.

The flexibility of the PERCENTw.d format allows users to tailor the output to specific reporting requirements. Whether the need is for high precision (many decimal places) for scientific reports or rounded integers for executive summaries, the format parameters offer complete control. Understanding how the width (*w*) and decimal places (*d*) interact is the key to generating clean and professional output in any SAS application, particularly within reporting procedures like PROC PRINT.

The Anatomy of the Percentw.d Format

The Percentw.d Format is defined by two crucial components: *w* and *d*. The *w* value, or the width parameter, dictates the total number of columns reserved for displaying the formatted percentage.

This total width must accommodate the numerical digits, the decimal separator, the percentage sign (%), and, if applicable, the negative sign. If the specified width is too small to display the full value, SAS will often display asterisks (*) to indicate truncation, which is why prudent width selection is important.

Conversely, the *d* value, or the decimal parameter, precisely controls the number of digits that appear to the right of the decimal point. This value determines the level of precision shown to the reader. When SAS applies the format, it automatically multiplies the internal fractional value by 100. It then ensures that the resulting percentage is rounded to the *d* number of decimal places specified, following standard rounding rules (rounding up if the next digit is 5 or greater).

For instance, consider the value 0.945. If formatted using `PERCENT7.1`, the output would be 94.5%. The width (7) accommodates the three digits, the decimal, the one decimal place, and the percent sign (e.g., 94.5%). If formatted using `PERCENT7.0`, the output would be 95%. In this case, *d*=0 triggers rounding the percentage (94.5) to the nearest integer (95), and the total width is still sufficient (e.g., 95%).

Choosing the correct combination of *w* and *d* is essential for both accuracy and aesthetics. If *w* is significantly larger than necessary, the resulting column may contain excessive white space, impacting report compactness. If *d* is set too high, it might introduce unnecessary precision that distracts the reader. Therefore, selecting a format like `PERCENT10.2` often represents a good balance, providing ample space (10 characters) and standard two-decimal precision.

Setting Up the Sample Dataset in SAS

To demonstrate the practical application of percentage formatting, we will first create a simple Dataset containing hypothetical exam scores. These scores are stored as fractional values, ranging from 0 to 1, representing the proportion of correct answers for several students in a class. This setup mirrors many real-world scenarios where raw scores must be converted for presentation.

The following code block utilizes the `DATA` step to generate a dataset named `my_data`. We use the `INPUT` statement to define two variables: `student` (character type, denoted by \$) and `exam_score` (numeric type). The subsequent `DATALINES` statement holds the raw, unformatted data which will be processed by the SAS system.

We then conclude the data creation process with the `RUN` statement and immediately follow it with a simple `PROC PRINT` step. This initial print procedure serves to visualize the raw data as it is stored internally, confirming that the scores are indeed fractions before any formatting is applied. This baseline view is essential for verifying the impact of the subsequent formatting steps.

```
/*create dataset*/  
data my_data;  
input student $ exam_score;  
datalines;  
Andy 0.945  
Bob 0.78  
Chad 0.865  
Derrick 0.77  
Eric 0.75  
Frank 0.64  
George 0.895  
Henry 0.98  
Isaac 0.68  
John 0.84  
;  
run;  
  
/*view dataset*/  
proc print data=my_data;
```

Obs	student	exam_score
1	Andy	0.945
2	Bob	0.780
3	Chad	0.865
4	Derrick	0.770
5	Eric	0.750
6	Frank	0.640
7	George	0.895
8	Henry	0.980
9	Isaac	0.680
10	John	0.840

Applying Percent Formatting with Decimal Precision

Once the baseline data structure is established, the next logical step is to apply the desired

percentage Format to the `exam_score` column. The goal here is to convert the raw proportional values (e.g., 0.945) into visually clear percentages (e.g., 94.5%). For initial reporting, maintaining at least one decimal place is often preferred to ensure a higher degree of accuracy is conveyed without overcomplicating the display.

We accomplish this formatting within the `PROC PRINT` step using the `FORMAT` statement. The syntax is straightforward: specify the variable name followed by the desired format structure. In this example, we use `percent10.1`. This statement applies the percentage conversion and ensures that the output will reserve 10 characters for the field width and display exactly one digit after the decimal point.

The chosen format, `percent10.1`, is highly illustrative of the format's power. The **10** provides generous space for scores ranging from 0% to 100% (e.g., `100.0%` uses 6 characters), ensuring no truncation occurs, even with small fluctuations. The **1** digit of precision strikes a balance between detailed reporting and clear presentation. Observe how the original fractional values are transformed in the output generated by the code below.

```
/*view dataset and display exam scores in percent format*/  
proc print data=my_data;  
format exam_score percent10.1;  
run;
```



Controlling Output Width and Decimals

When reviewing the results of the previous step, every value in the `exam_score` column is now correctly displayed in a percent format. This transformation highlights the two roles played by the parameters in `percent10.1`. The overall width, defined by **10**, guarantees that the entire formatted value—including the implicit multiplication by 100 and the addition of the percent symbol—fits within the reserved column space in the `PROC PRINT` output.

More specifically, the parameter **1** dictates the precise number of decimal places shown. If the underlying data (e.g., 0.945) results in a percentage (94.5%) that naturally requires fewer digits than the specified precision, SAS will still ensure that the exact number of decimal places is displayed, padding with zeros if necessary (though not required in this example). Conversely, if the unformatted percentage has more decimal places (e.g., 94.53%), SAS rounds the value to meet the `d` specification (94.5%).

Choosing an appropriate width (w) is primarily about preventing truncation errors, especially when dealing with extreme values (e.g., percentages over 100% or very large negative percentages). While 10 characters might seem large for scores between 0% and 100%, it provides a safety buffer. Furthermore, using the same width consistently across multiple reports aids in visual alignment and professional presentation, ensuring that all columns maintain a standard visual footprint regardless of the exact numerical content.

Displaying Percentages Without Decimal Places

In many summarized reports or dashboards, excessive precision can clutter the visual presentation. When the requirement is to show percentages rounded to the nearest whole number, we must modify the d parameter within the Percentw.d Format. To suppress the display of any fractional components, the decimal parameter is simply omitted, or explicitly set to zero (d=0).

If we use percent10. (omitting the decimal parameter entirely), SAS defaults to displaying zero decimal places. This command triggers two actions: first, the proportional score is converted to a percentage; second, this percentage is rounded to the nearest integer. It is vital to note that this is rounding, not simple truncation. For example, 94.5% is rounded up to 95%, while 77.0% remains 77%.

This rounding behavior is critical for accurate reporting. While the visual output is simplified, the data integrity is maintained through standard arithmetic rounding rules applied during the formatting process. The width (10) remains adequate for the display, allowing the resulting integer percentage value (e.g., 95%) and the percent symbol to fit comfortably within the column boundaries. The following code demonstrates this application and the resulting output.

```
/*view dataset and display exam scores in percent format without decimal places*/  
proc print data=my_data;  
format exam_score percent10.;  
run;
```

Obs	student	exam_score
1	Andy	95%
2	Bob	78%
3	Chad	87%
4	Derrick	77%
5	Eric	75%
6	Frank	64%
7	George	90%
8	Henry	98%
9	Isaac	68%
10	John	84%

Upon reviewing the final output table, it is clear that each numerical value in the `exam_score` column has been rounded to the nearest integer. For instance, Andy's score, originally 94.5%, is now displayed as 95%. This process confirms that SAS applies standard rounding when the decimal parameter is suppressed or set to zero, ensuring that fractional parts are handled appropriately for high-level reporting.

Key Considerations for SAS Format Usage

While the `PERCENTw.d` format is ideal for immediate reporting using procedures like `PROC PRINT`, advanced users should understand where and how formats are permanently stored and applied. The use of the `FORMAT` statement within a reporting procedure (as shown in the examples above) applies the format only for the duration of that procedure run. The underlying Dataset remains unchanged, and the variable retains its raw numerical values.

For scenarios where a format must be permanently associated with a variable, the `FORMAT` statement should be included within the `DATA` step when the dataset is created or modified. Alternatively, if custom formatting rules or multiple conditional formats are required, users can leverage `PROC FORMAT` to define specialized formats that can then be applied universally across different procedures and datasets.

The `PERCENT` format is one of many standard numeric formats available in SAS. Others include `COMMAw.d` for currency and large numbers, `DATEw.` for date interpretation, and `DOLLARw.d`. Mastery of these formatting options is a fundamental skill in SAS programming, ensuring that data presentation is accurate, professional, and tailored to the intended audience.