Learn How to Count Distinct Values in SAS: A Step-by-Step Guide

Authored by

Mohammed looti

October 31, 2025

Understanding the composition and uniqueness of data elements is a fundamental step in data analysis using SAS (Statistical Analysis System). Specifically, counting distinct values within a variable allows analysts to determine the cardinality of a column, which is vital for quality checks, aggregation, and reporting. In SAS, the most efficient and versatile way to achieve this is by leveraging the robust capabilities of PROC SQL.

This guide explores two essential methods for calculating distinct counts within a dataset. The first method focuses on determining the total number of unique entries in a single column across the entire dataset. The second method demonstrates how to calculate distinct counts conditionally, grouping the results based on the values of another variable. Mastery of these techniques is crucial for anyone performing complex data manipulations in the SAS environment.

Method 1: Counting Distinct Values in a Single Column

When the goal is to quickly ascertain the overall uniqueness within a variable--for instance, determining how many unique product IDs or customer names exist--the single-column distinct count method is utilized. This approach uses the powerful combination of the `COUNT` aggregation function and the `DISTINCT` keyword within the `PROC SQL` step.

The `DISTINCT` keyword acts as a filter, ensuring that only unique occurrences of the specified variable are included in the count calculation. This method provides a single, summary statistic reflecting the total number of unique elements found in the variable across all observations in the source table.

The general structure for this operation is concise and highly readable, making it a standard practice in SAS programming for data auditing and exploration.

```
proc sql;
select count(distinct var1) as distinct_var1
from my_data;
quit;
```

Method 2: Counting Distinct Values by Group

In many analytical scenarios, understanding uniqueness relative to a specific category is necessary. For example, a user might need to know how many unique products were sold by each region, or how many unique scores were achieved by each team. This requires counting distinct values conditionally, using the GROUP BY clause.

When the `GROUP BY` clause is introduced, the `PROC SQL` procedure partitions the dataset based on the grouping variable (e.g., `var1`). The subsequent `COUNT(DISTINCT var2)` calculation is then

applied independently to each of these subsets. The result is a summary table displaying the grouping variable alongside the count of unique values for the target variable within that group.

This method is powerful for generating aggregated reports and performing sub-group analysis, offering deeper insight into data distribution than a simple overall distinct count. It is an indispensable tool for business intelligence and statistical reporting.

```
proc sql;
select var1, count(distinct var2) as distinct_var2
from my_data
group by var1;
quit;
```

Setting Up the Sample Dataset

To illustrate these two methods clearly, we will utilize a small, representative dataset named `my_data`. This dataset contains information about basketball teams and their corresponding scores (points). The variables included are `team` (a character variable, denoted by the `$` sign) and `points` (a numeric variable).

The creation of this sample data is handled using a standard SAS `DATA` step combined with the `DATALINES` statement, which allows us to input the raw data directly into the program. We then use `PROC PRINT` to confirm the structure and contents of the resulting table before proceeding with the distinct counting examples.

```
/*create dataset*/
data my_data;
input team $ points;
datalines;
Mavs 10
Mavs 13
Mavs 13
Mavs 15
Mavs 15
Rockets 9
Rockets 10
Rockets 10
Spurs 18
Spurs 19
;
```

```
run;

/*view dataset*/
proc print data=my_data;
```

The resulting dataset, which serves as the foundation for our analysis, clearly shows the repeated values we intend to analyze for uniqueness.

Obs	team	points
1	Mavs	10
2	Mavs	13
3	Mavs	13
4	Mavs	15
5	Mavs	15
6	Rockets	9
7	Rockets	10
8	Rockets	10
9	Spurs	18
10	Spurs	19

Example 1: Calculating Total Distinct Teams

Our first example applies Method 1 to determine the total number of unique teams present in the `my_data` dataset. We are interested in the cardinality of the `team` column. This is achieved by passing the `team` variable, modified by the `DISTINCT` keyword, into the `COUNT` function within our `PROC SQL` statement.

The code below executes this calculation, aliasing the output variable as `distinct_teams` for clarity in the resulting table. This result confirms how many different entities are represented across all observations.

```
/*count distinct values in team column*/
proc sql;
select count(distinct team) as distinct_teams
from my_data;
quit;
```

The output generated by SAS confirms the calculation.

distinct_teams
3

As demonstrated by the output, there are **3** distinct values in the `team` column. We can easily verify this outcome by manually reviewing the input data, where the only unique team entries are 'Mavs', 'Rockets', and 'Spurs'. This confirms the accuracy and utility of the `COUNT(DISTINCT...)` syntax for high-level uniqueness assessment.

Example 2: Calculating Distinct Points by Team Group

In this more advanced example, we apply Method 2 to count the unique scores (`points`) achieved by each individual team (`team`). This requires us to use the `GROUP BY` clause, ensuring that the distinct count is calculated separately for the 'Mavs' group, the 'Rockets' group, and the 'Spurs' group.

The `SELECT` statement specifies two outputs: the grouping variable (`team`) and the calculated distinct count (`COUNT(DISTINCT points)`), aliased as `distinct_points`. The `GROUP BY team` clause instructs PROC SQL to perform the aggregation independently for every unique value found in the `team` variable.

```
/*count distinct values in points column, grouped by team*/
proc sql;
select team, count(distinct points) as distinct_points
from my_data
group by team;
quit;
```

The resulting table is an aggregated report that clearly summarizes the distinct point totals for each category.

team	distinct_points
Mavs	3
Rockets	2
Spurs	2

By reviewing the output, we gain precise insights:

The **Mavs** group had **3** distinct point totals (10, 13, and 15, even though 13 and 15 appeared twice).

The **Rockets** group had **2** distinct point totals (9 and 10).

The **Spurs** group had **2** distinct point totals (18 and 19).

The resulting table successfully shows the number of distinct values in the `points` column, segmented and calculated independently for each of the teams, demonstrating the power and flexibility of using `COUNT(DISTINCT...)` in conjunction with the GROUP BY clause in SAS.

Additional Resources for SAS Programming

Counting distinct values is a foundational skill in data manipulation. However, the SAS programming language offers numerous procedures and techniques for handling complex data preparation and analytical tasks. We encourage continued learning to maximize efficiency when working with large datasets.

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