Learning SAS: Performing Frequency Analysis by Group Using PROC FREQ

Authored by

Mohammed looti

October 27, 2025

Introduction to Segmented Frequency Analysis in SAS

Effective data analysis requires a foundational understanding of how variables are distributed, particularly when dealing with **categorical data**. A **frequency table** serves as the cornerstone of initial data exploration, offering a concise summary of how often each unique value of a **variable** occurs within a dataset. This fundamental statistical summary is indispensable for gaining preliminary insights, verifying data integrity, and preparing the raw information for more sophisticated statistical modeling.

In the context of SAS software, the primary tool for generating these essential summaries is the **PROC FREQ** procedure. This powerful procedure is expertly designed to handle large volumes of data, providing not only simple raw counts but also percentages, cumulative frequencies, and statistics for one-way, two-way, and multi-way classifications. Its flexibility makes it one of the most frequently used procedures for **descriptive statistics**.

While overall frequency counts are helpful, modern data analysis often demands a more granular view. It is frequently necessary to analyze the distribution of one variable contingent upon the values of another--a process known as segmented or grouped analysis. This is precisely where the true power of **PROC FREQ** is unlocked through the application of the **BY statement**. Utilizing this statement allows analysts to move beyond generalized statistics, providing separate, detailed frequency tables for every distinct subgroup defined by the grouping variable, thereby isolating and illuminating differences across categories.

The Role of the BY Statement in PROC FREQ Syntax

The core objective of **PROC FREQ** is to tabulate **categorical data**. However, when the requirement shifts to performing this analysis independently for each level of an existing grouping variable, the integration of the **BY statement** is mandatory. This statement instructs the SAS System to treat each unique value of the specified grouping variable as a separate entity, generating a complete set of frequency tables for the analysis variable within the context of that specific group.

The fundamental syntax for executing a segmented frequency analysis in SAS is clean and efficient. It involves the procedure call, the dataset specification, the grouping directive, and the analysis variable definition.

```
proc freq data=my_data;
by var1;
tables var2;
run;
```

In this specific structure, the `PROC FREQ` statement initiates the tabulation procedure, clearly identifying the target dataset, `my_data`. The `BY var1;` statement is the critical directive, signaling to SAS that subsequent operations should be performed separately for every unique instance of the **variable** `var1`. The `TABLES var2;` statement then specifies which variable, `var2`, should be subjected to frequency tabulation within each of the groups defined by `var1`. Finally, the **RUN statement** executes the entire procedure. This method transforms a simple frequency count into a powerful, comparative statistical tool, enabling the analyst to compare distributions across populations or conditions directly.

Practical Example: Analyzing Team Composition

To clearly illustrate the application and utility of **PROC FREQ** with the **BY statement**, we will construct a realistic scenario. Imagine we are analyzing player data for two distinct sports teams. Our dataset, creatively named `my_data`, includes essential information such as the team identifier, the player's position (e.g., Guard or Forward), and their recent points scored. Our primary goal is to determine the distribution of player positions, but separated by their respective teams, allowing for a direct comparison of team structures.

The first crucial step is the creation and verification of this illustrative dataset in the SAS environment. This ensures that the data is correctly structured with the necessary grouping **variable** (`team`) and the analysis variable (`position`). We use a `DATA` step combined with `DATALINES` to input the sample data, followed by `PROC PRINT` for immediate visual inspection.

```
/*create dataset*/
data my_data;
input team $ position $ points;
datalines;
A Guard 22
A Guard 20
A Guard 30
A Forward 14
A Forward 11
B Guard 12
B Guard 22
B Forward 30
B Forward 9
B Forward 12
B Forward 25
;
run;
```

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

Upon executing the `PROC PRINT` command, the generated output confirms that the `my_data` dataset has been successfully loaded and structured. We can clearly observe the distinct categories within the `team` **variable** (A and B) and the distribution of player positions within the dataset. This visual verification is essential before launching any statistical procedure to avoid errors arising from incorrectly loaded or structured data.

Obs	team	position	points
1	A	Guard	22
2	A	Guard	20
3	A	Guard	30
4	A	Forward	14
5	A	Forward	11
6	B	Guard	12
7	B	Guard	22
8	B	Forward	30
9	B	Forward	9
10	B	Forward	12
11	B	Forward	25

Executing the Grouped Frequency Analysis

With the data successfully verified, the next logical step is to perform the grouped frequency analysis. Our objective is specifically to calculate the **frequency table** for the `position` variable, but critically, we need this calculation to be isolated and performed sequentially for each unique value of the `team` variable. This segmentation is achieved solely through the use of the **BY statement** within the **PROC FREQ** procedure.

```
/*calculate frequency of position, grouped by team*/
proc freq data = my_data;
by team;
tables position;
run;
```

Dissecting this code block reveals the mechanics of the grouped analysis. The initial statement,

`proc freq data = my_data;`, specifies the procedure and the input dataset. The core instruction, `by team;`, is what segments the data processing; it ensures that SAS reads the data and executes the subsequent frequency calculation independently for every contiguous block of observations sharing the same team value. Finally, `tables position;` directs the procedure to generate the requested **frequency table** for the `position` **variable** within each predefined team group.

This methodical approach yields two distinct frequency tables in the output--one for Team A and one for Team B. This is significantly more informative than a single, combined table, as it immediately highlights the structural differences in player roles between the two groups. This ability to segment and compare distributions is foundational to effective comparative **descriptive statistics**.

Interpreting Segmented Frequency Tables

The output generated after executing **PROC FREQ** with the **BY statement** is structured to provide clear, separate analysis blocks for each group. The visual segmentation makes comparative analysis straightforward and intuitive, allowing analysts to quickly identify how the distribution of the analysis variable changes across the defined subgroups.

The FREQ Procedure

team=A

position	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Forward	2	40.00	2	40.00
Guard	3	60.00	5	100.00

The FREQ Procedure

team=B

position	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Forward	4	66.67	4	66.67
Guard	2	33.33	6	100.00

As clearly illustrated by the resulting frequency tables, the output is partitioned by the `team` **variable**. Each section provides a comprehensive one-way **frequency table** for the `position` variable relevant only to that specific team. This structure enables precise, group-specific

observations regarding the composition of each team.

The key findings derived directly from the segmented output are as follows:

For Team A, the position "Forward" occurred with a frequency of **2**.
For Team A, the position "Guard" occurred with a frequency of **3**.
For Team B, the position "Forward" occurred with a frequency of **4**.
For Team B, the position "Guard" occurred with a frequency of **2**.

Beyond simple counts, the output also includes the **Percent**, **Cumulative Frequency**, and **Cumulative Percent** columns. The **Percent** column is vital, as it standardizes the raw counts, allowing for a fair comparison of the proportional representation of positions, regardless of the difference in total team size. The **Cumulative Frequency** and **Cumulative Percent** columns provide context on the running total of observations, which is particularly useful when analyzing variables with many ordered categories, though less critical for simple two-category variables like position. These statistics collectively deepen the interpretation of the grouped data, offering insights into relative distribution patterns.

Advanced Considerations and Best Practices

While the preceding example focused on a simple one-way frequency grouped by a single **BY variable**, the capabilities of **PROC FREQ** extend far beyond this basic application. The **TABLES statement** is exceptionally versatile; it allows the analyst to specify multiple analysis variables simultaneously. For instance, executing `tables position points;` would instruct SAS to generate two separate frequency tables--one for `position` and one for `points`--with both sets of tables segmented and processed according to the same **BY variable** (`team`).

Furthermore, **PROC FREQ** offers numerous options for fine-tuning the output and behavior of the procedure. Analysts can control the display order of categories using the `ORDER=` option (e.g., `ORDER=FREQ` to sort by frequency), manage how observations with missing values are handled using the `MISSING` option, or suppress the printed output entirely using `NOPRINT` when the only goal is to create an output dataset using the `OUT=` option for further processing. Leveraging these options allows for highly customized and effective analytical workflows tailored to specific reporting requirements.

A critically important technical requirement when employing the **BY statement** is ensuring that the input dataset is properly sorted by the **BY variable(s)**. If the data is not sorted, SAS will issue a warning, and the procedure will execute, but the results will be incorrect or misleading because observations belonging to the same group will not be processed contiguously. To guarantee accurate results, always precede **PROC FREQ** with a `PROC SORT` step: `proc sort data=my_data; by team; run;`. This preparatory step ensures that all records for a given team

are grouped together, fulfilling the prerequisite for the **BY statement** to function correctly and generate reliable segmented frequency tables.

Conclusion: Mastering Segmented Data Analysis

The combined use of **PROC FREQ** and the **BY statement** represents a fundamental and powerful technique for data exploration and analysis within the SAS environment. This methodology enables researchers and analysts to achieve segmented, granular insights, moving analysis beyond generalized totals to accurately understand patterns, differences, and unique distributions existing within specific subgroups. This capability is absolutely essential for rigorous comparative studies, quality control, and uncovering subtle nuances that are often obscured in complex datasets.

By adhering to the prescribed syntax, understanding the prerequisite of data sorting, and utilizing the flexibility of the **TABLES statement**, analysts can consistently apply this procedure to their own data. This enhances both the clarity and the analytical depth of statistical reports, transforming raw data into actionable knowledge regarding specific populations or conditions.

For analysts seeking to utilize the full range of options available for frequency analysis, consulting the official SAS documentation is highly recommended. The documentation provides exhaustive details on all available statements, complex options, and advanced examples, ensuring that users can leverage the full potential of **PROC FREQ** for even the most demanding descriptive statistical tasks.