

# How to Easily Extract the First N Rows from a SAS Dataset

Authored by  
**stats writer**

December 1, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Easily Extract the First N Rows from a SAS Dataset*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=103076>

Efficiently managing and analyzing large volumes of information often requires focusing only on a subset of the data. In SAS, a powerful statistical software suite, selecting the initial observations (the first N rows) of a dataset is a common requirement for quick validation, sampling, or debugging. While there are several techniques, the most straightforward methods involve leveraging the inherent capabilities of the DATA step, specifically using either the `FIRSTOBS=` and `OBS=` options within the SET statement, or by employing the powerful automatic variable `_N_`.

The `FIRSTOBS=` option determines which record number the SET statement should begin reading, while the `OBS=` option dictates the final record to be read. For instance, to retrieve the first 10 observations from an input file, one would specify `SET mydata(firstobs=1 obs=10)`. However, for conditional reading within the program logic--which is the focus of the examples below--using the automatic counter variable `_N_` often provides greater flexibility and clarity, especially when combined with conditional logic like the `IF-THEN OUTPUT` structure in the DATA step.

## The Automatic Variable `_N_`: Tracking Observation Count

In the world of DATA step programming, the automatic variable `_N_` serves as a fundamental counter, tracking the number of times the DATA step has iterated. Critically, `_N_` represents the sequential observation number currently being processed. Since it is automatically generated by SAS and does not appear in the final output dataset, it is perfectly suited for controlling execution flow and filtering records based solely on their physical position in the source file.

By implementing conditional logic that evaluates the value of `_N_`, we can precisely control when the `OUTPUT` command is executed. This allows us to create a new dataset containing only the desired starting rows. This technique provides superior programmatic control compared to relying solely on dataset options like `FIRSTOBS=` and `OBS=`, as the filtering happens inside the core logic of the program rather than during the file access stage.

## Two Primary Methods for Selecting Initial Rows Using Conditional Logic

We will now explore two specific, highly efficient implementations utilizing the `_N_` variable within the DATA step environment. These methods are preferred for their clarity and ability to handle large datasets effectively. The first implementation focuses on selecting a single record, while the second is a generalized pattern for selecting any positive integer N of initial records.

### Method 1: Selecting Only the First Row

To select exclusively the first observation in a dataset, we check if the automatic counter, `_N_`, is strictly equal to 1. When this condition is met, the program executes the `OUTPUT` statement, writing the current observation to the new dataset. Because we are not using the implicit output feature of the DATA step (by not including an implicit output statement at the end), all subsequent records

are automatically discarded, resulting in a single-row output file.

```
data first_row;  
set original_data;  
if _N_ = 1 then output;  
run;
```

## Method 2: Selecting the First N Rows

For selecting the first N rows--where N is any desired quantity--we utilize the less than or equal to relational operator ( $\leq$ ). This condition instructs SAS to continue writing records to the output file as long as the automatic row counter `_N_` does not exceed the specified limit (e.g., 5). This approach offers maximum flexibility, as N can be easily parameterized or changed without altering the underlying program structure.

The code below demonstrates how to select the first five observations. Once the value of `_N_` exceeds 5, the condition fails, and the `OUTPUT` statement is bypassed for all remaining iterations of the `DATA` step, ensuring only the top five records are retained.

```
data first_N_rows;  
set original_data;  
if _N_ <= 5 then output; /*select first 5 rows*/  
run;
```

## Setting Up the Demonstration Dataset for Execution

To confirm the functionality of these two methods, we must first establish a source dataset. The following code uses the `DATALINES` statement to create `original_data`, a small, readable dataset containing basketball statistics. This initial setup ensures that the subsequent subsetting examples have a clear, verifiable source file, making the demonstration reproducible for users.

The dataset `original_data` contains 10 observations, allowing us to clearly visualize the effect of selecting N=1 and N=5 rows. Following the data creation, `PROC PRINT` is used to display the entire source dataset, providing a baseline comparison before any filtering occurs.

```
/*create dataset*/  
data original_data;  
input team $ points rebounds;  
datalines;
```

```
Warriors 25 8
Wizards 18 12
Rockets 22 6
Celtics 24 11
Thunder 27 14
Spurs 33 19
Nets 31 20
Mavericks 34 10
Kings 22 11
Pelicans 39 23
;
run;
```

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

Obs	team	points	rebounds
1	Warriors	25	8
2	Wizards	18	12
3	Rockets	22	6
4	Celtics	24	11
5	Thunder	27	14
6	Spurs	33	19
7	Nets	31	20
8	Maverick	34	10
9	Kings	22	11
10	Pelicans	39	23

### Example 1: Practical Execution of Single Row Selection

Here we apply Method 1, executing the code to isolate only the first record. The `SET` statement reads the `original_data` record by record into the Program Data Vector (PDV). For the very first record (`_N_ = 1`), the condition is true, and the `OUTPUT` statement writes the observation (Warriors, 25, 8) to the new dataset, `first_row`. For all subsequent records, `_N_` is greater than 1, causing the conditional output to be skipped.

The subsequent `PROC PRINT` step confirms that the new dataset contains only one observation.

This confirms the technique's effectiveness for targeted selection of the starting record. This is especially useful in ETL processes where header information or initial configuration records need to be handled separately from the main data body.

```
/*create new dataset that contains only the first row*/
```

```
data first_row;
```

```
set original_data;
```

```
if _N_ = 1 then output;
```

```
run;
```

```
/*view new dataset*/
```

```
proc print data=first_row;
```

Obs	team	points	rebounds
1	Warriors	25	8

As clearly illustrated by the resulting table, the new dataset contains only the first row of the original dataset. This precise subsetting demonstrates the efficiency and reliability of using the `_N_` automatic variable for isolating specific observations based on their position.

## Example 2: Practical Execution of Selecting the First Five Observations

This example demonstrates the broader applicability of the `_N_` method by selecting the first five records. We use the condition `IF _N_ <= 5 THEN OUTPUT;` within the `DATA` step to control the output stream. For records 1 through 5, the condition is true, and data is written. Starting with record 6 (Spurs), the condition fails, and those records are discarded from the output dataset `first_N_rows`.

This method is highly scalable. If you needed to select the first 1,000 observations from a dataset containing millions of rows, this approach minimizes overhead because the `OUTPUT` operation is conditionally executed only for the required top portion of the file. The flexibility of simply changing the `N` value makes this a cornerstone technique for quick data inspection and prototyping in [SAS](#).

```
/*create new dataset that contains first 5 rows of original dataset*/
```

```
data first_N_rows;
```

```
set original_data;
```

```
if _N_ <= 5 then output;
```

```
run;
```

```
/*view new dataset*/  
proc print data=first_N_rows;
```

Obs	team	points	rebounds
1	Warriors	25	8
2	Wizards	18	12
3	Rockets	22	6
4	Celtics	24	11
5	Thunder	27	14

The resulting output table confirms that the new dataset contains the first five rows--from Warriors to Thunder. To select a different number of starting rows, simply change the value used in the comparison with `_N_` in the code above. This methodology is fundamental to creating representative samples or verifying the integrity of the initial records in any input file.

### Alternative Approach: Leveraging PROC SQL for Top N Selection

While the `DATA` step is the native and most flexible environment in SAS, database professionals often prefer using `PROC SQL`. SQL offers a declarative way to achieve the same result through the `OUTOBS=` option, which limits the number of rows written to the output table.

The equivalent SQL syntax for selecting the first five rows would be: `PROC SQL OUTOBS=5; CREATE TABLE first_5 AS SELECT * FROM original_data; QUIT;`. This method is concise and leverages SQL optimization engines. While `PROC SQL` is generally easier to read for simple subsetting tasks, the `_N_` method in the `DATA` step offers more potential for complex data manipulation and calculation before the output operation.

### Conclusion: Mastery of Positional Subsetting in SAS

Mastering how to select the first N rows is essential for efficient data handling in SAS. Although options like `FIRSTOBS=` and `OBS=` provide data access control at the `SET` statement level, the most robust and programmatically clear technique involves using the automatic variable `_N_` within an `IF-THEN OUTPUT` structure. This ensures that the subsetting logic is transparent and easily adjustable, regardless of the size or complexity of the input dataset.

By applying `IF _N_ <= N THEN OUTPUT;`, analysts can confidently extract the required initial portion of their data for analysis, testing, and debugging, thereby optimizing resource consumption and improving the overall efficiency of their SAS programs.

The following tutorials explain how to perform other common tasks in SAS, building on the foundational skills demonstrated here:

ARABPSYCHOLOGY.COM