Learning How to Concatenate Datasets in SAS: A Step-by-Step Guide

Authored by

Mohammed looti

October 31, 2025

Understanding Dataset Concatenation in SAS

In the realm of statistical computing and data management, the ability to combine disparate data sources efficiently is paramount. Dataset concatenation in SAS, or Statistical Analysis System, refers to the process of vertically stacking two or more datasets on top of each other. This operation is distinct from merging, where datasets are joined horizontally based on common key variables. Concatenation is typically used when data observations have been collected in multiple batches or files, but they all share an identical underlying structure.

The primary mechanism for performing this crucial data transformation relies on the foundational DATA step. The DATA step is the engine within SAS that reads, modifies, and creates new datasets. When combining data, the resulting output is a new, unified dataset containing all observations from the input files, maintaining the original variable definitions. This technique ensures data continuity and simplifies subsequent analysis, as analysts can work with a single, comprehensive file rather than managing multiple smaller segments.

To successfully execute concatenation, it is a critical prerequisite that all contributing datasets possess a consistent schema. While SAS is flexible enough to handle slight inconsistencies--such as a variable existing in one dataset but not another (in which case missing values are inserted)--the most robust and predictable results occur when the variable names, data types (character vs. numeric), and lengths are identical across all inputs. If the structures are highly divergent, concatenation may lead to unexpected results or require complex data cleaning beforehand.

The Essential SAS Syntax for Combining Data

The core of the concatenation process utilizes the powerful SET statement within the DATA step. Unlike the MERGE statement, which links records based on keys, the SET statement simply reads observations sequentially from the specified input datasets. The order in which the datasets are listed in the SET statement dictates the final order of observations in the resultant dataset.

The basic syntax is straightforward and highly efficient. We initiate a new DATA step, name the target dataset (the concatenated file), and then use the SET statement followed by a list of all source datasets we wish to combine. It is important to note that the input datasets remain unchanged; the DATA step is creating an entirely new dataset based on the combined information. This fundamental structure is illustrated below:

```
/*concatenate two datasets into one*/
data data3;
set data1 data2;
run;
```

In this example, `data3` is the resulting concatenated file. The SET statement first processes every observation from `data1`, and once `data1` is exhausted, it immediately begins reading observations from `data2`, continuing until all source files have been processed. This sequential processing is the definition of concatenation in the SAS environment.

Detailed Example: Preparing the Source Datasets

To demonstrate the practical application of this syntax, we will establish two separate datasets representing performance metrics collected at different times or locations. We will name these initial files `data1` and `data2`. Both datasets must contain the same variables: `firstName` (character), `lastName` (character), and `points` (numeric).

The following code block utilizes the DATA step along with the `INPUT` and `DATALINES` statements to define and populate these two distinct source files. This initial setup is crucial for ensuring that the subsequent concatenation step operates correctly, as it confirms that the variable types and order are consistent across the inputs.

```
/*create first dataset*/
data data1;
input firstName $ lastName $ points;
datalines;
Austin Smith 15
Brad Stevens 31
Chad Miller 22
;
run;

/*create second dataset*/
data data2;
input firstName $ lastName $ points;
datalines;
Dave Michaelson 19
Eric Schmidt 29
Frank Wright 20
Greg Gunner 40
Harold Anderson 35
;
run;

/*view datasets*/
proc print data=data1;
```

proc print data=data2;

Before proceeding with the combination, it is best practice to verify the structure and content of the source files. The PROC PRINT procedure is used here to display the contents of both `data1` (which has 3 observations) and `data2` (which has 5 observations). This verification step confirms that our source data is ready for the concatenation operation.

Obs	firstName	lastName	points
1	Austin	Smith	15
2	Brad	Stevens	31
3	Chad	Miller	22

Obs	firstName	lastName	points
1	Dave	Michaels	19
2	Eric	Schmidt	29
3	Frank	Wright	20
4	Greg	Gunner	40
5	Harold	Anderson	35

Executing the Concatenation and Verifying Results

With the source data validated, we can now execute the core concatenation logic. The goal is to create a new dataset, `data3`, that vertically stacks all eight observations from `data1` and `data2`. The sequential nature of the SET statement ensures that the rows from `data1` appear first in the output, immediately followed by the rows from `data2`.

We use the following concise code to perform this critical combination. This single DATA step is incredibly powerful and demonstrates the efficiency of SAS for handling large volumes of data manipulation.

/*concatenate two datasets into one*/
data data3;
set data1 data2;
run;

/*view new dataset*/

```
proc print data=data3;
```

Upon execution, the resulting dataset, `data3`, is created. We then use PROC PRINT once more to confirm the structure and content of the newly formed file. As expected, the resulting dataset now contains eight total observations, which is the sum of the three observations from `data1` and the five observations from `data2`.

Obs	firstName	lastName	points
1	Austin	Smith	15
2	Brad	Stevens	31
3	Chad	Miller	22
4	Dave	Michaels	19
5	Eric	Schmidt	29
6	Frank	Wright	20
7	Greg	Gunner	40
8	Harold	Anderson	35

The resulting dataset successfully contains all observations from the source files, demonstrating a successful vertical stacking operation. The key takeaway from this example is the simplicity of the syntax--a single `DATA` step and SET statement are sufficient to merge vast quantities of data records.

Critical Considerations for Successful Dataset Merging

While the syntax for concatenation is remarkably simple, managing the underlying structure of the data requires careful attention. The most fundamental requirement, as demonstrated above, is that all source files must share the same variable names. If, for instance, `data1` contained a variable named `Score` and `data2` contained a variable named `Points`, SAS would treat them as two entirely separate variables in the concatenated output. Records from `data1` would have a missing value for `Points`, and records from `data2` would have a missing value for `Score`. Therefore, standardizing variable names prior to concatenation is essential for maintaining data integrity.

Furthermore, variable attributes--specifically type (character or numeric) and length--are determined by the first dataset listed in the SET statement. If `data1` defines a variable as character with a length of $10, and `data2` defines the same variable as character with a length of $50, the concatenated dataset (`data3`) will inherit the length of $10 from `data1`. Data from `data2` that exceeds this length will be truncated, potentially leading to data loss or corruption without any

explicit warning in the log. Analysts must proactively harmonize variable attributes before combining data.

Finally, it is important to understand the scalability of this method. While our example concatenated only two datasets, the DATA step is fully capable of listing dozens or even hundreds of files in the SET statement. For very large-scale operations involving dynamically generated files, SAS programmers often employ advanced techniques such as macro loops or the `INFILE` statement combined with the `_ALL_` option to automate the listing of all relevant datasets in a library. Regardless of scale, the core principle remains the same: the SET statement dictates the sequential reading process.

Summary and Further Resources

Concatenation in SAS is a fundamental and efficient method for stacking datasets, relying solely on the power of the SET statement within the DATA step. While the execution syntax is simple, successful implementation requires strict attention to the uniformity of input data structures, ensuring consistent variable names and attributes across all files. This approach allows analysts to quickly consolidate segmented data into a single master file for comprehensive analysis.

To expand your knowledge of data combination techniques in SAS, consider exploring related operations. While concatenation stacks data vertically, the `MERGE` statement is used for horizontal joining (linking records side-by-side based on common IDs). Understanding the distinction between `SET` (concatenation) and `MERGE` (joining) is essential for mastering data manipulation in the DATA step environment.

Additional Resources

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

This section is reserved for links to related tutorials (e.g., How to Merge Datasets, How to Use PROC SORT, etc.) to guide the reader toward further specialization in SAS programming.