How to Easily Export Data from SAS to Excel

Authored by

stats writer

December 1, 2025

The ability to seamlessly transfer analytical results between software platforms is a cornerstone of modern data management. Exporting data from SAS, a powerful statistical analysis system, to Excel, the industry-standard spreadsheet application, is a common and essential task for data analysts and researchers. This process allows users to leverage the robust computational capabilities of SAS for complex modeling while utilizing the familiar interface of Excel for reporting, visualization, and sharing results with stakeholders who may not have access to the SAS environment.

While several methods exist for this data migration, the most straightforward and widely implemented technique involves the use of the PROC EXPORT procedure. This powerful SAS utility is specifically designed to write Dataset content directly into external file formats, including delimited files, databases, and, critically, Microsoft Excel workbooks. Understanding the correct syntax and parameters for PROC EXPORT is paramount for ensuring data integrity and successful transfer.

This comprehensive guide will detail the steps necessary to move your statistical results and data structures from the SAS environment into a usable Excel file format. We will focus primarily on the use of PROC EXPORT using practical, step-by-step examples that cover both single and multiple Dataset exports. By the end of this tutorial, you will possess the expertise to reliably automate your data export tasks, saving significant time in your analytical workflow.

The Primary Method: Utilizing PROC EXPORT for Data Transfer

The most efficient and standardized method for transferring a Dataset from the SAS system to a Microsoft Excel workbook is through the utilization of the PROC EXPORT procedure. This procedure is designed specifically for creating external files from existing SAS data structures. It simplifies the often-complex requirements of file formatting by handling the internal data conversion necessary to generate a clean, readable Excel file, typically in the modern **.xlsx** format.

To initiate the export process, the analyst must specify several key parameters within the **PROC EXPORT** statement. These parameters dictate which Dataset is being moved, the exact location and filename of the output file, and the specific file type, designated by the Database Management System (DBMS) option. By clearly defining these elements, SAS can execute the data translation without manual intervention, ensuring accuracy and consistency across different runs.

One crucial benefit of using **PROC EXPORT** is the control it grants over the final presentation within Excel, allowing the user to name the specific worksheet where the data will reside. This capability is particularly useful when consolidating results from multiple analyses into a single workbook, a topic we will explore in detail in a subsequent example. Before diving into the examples, let's examine the foundational syntax required for a successful export operation.

Deconstructing the PROC EXPORT Syntax

The core syntax of the PROC EXPORT procedure is both concise and highly effective, requiring just a few positional and keyword arguments to define the entire operation. The statement begins with the procedure name itself, followed by the necessary inputs and output specifications, concluding with the mandatory **RUN** statement to execute the code block.

The structure below illustrates a typical implementation of **PROC EXPORT** designed to create a new Excel workbook and place a specified Dataset onto a named sheet. Pay close attention to the parameter usage, especially the **DBMS** option, which explicitly tells SAS that the target file is a modern Excel spreadsheet format (.xlsx).

```
/*Standard syntax illustrating export of SAS data to an Excel .xlsx file*/
proc export data=my_data
outfile="/home/u13181/my_data.xlsx"
dbms=xlsx
replace;
sheet="First Data";
run;
```

Each keyword option in the PROC EXPORT statement serves a specific purpose, collectively ensuring that the data transfer meets all necessary requirements for both the SAS source and the Excel destination. Below is a detailed breakdown of these essential parameters:

DATA: This is a required parameter specifying the exact name of the Dataset (or temporary work table) within the SAS environment that you intend to export.
OUTFILE: This parameter specifies the full operating system path and filename for the resulting external file. For Excel exports, this must typically end in the **.xlsx** extension.
DBMS: Standing for Database Management System, this crucial option defines the file format of the output file. For modern Excel files, the value should be set to **XLSX**. (For older Excel versions, **XLS** might be used, but is generally discouraged.)
REPLACE: This optional keyword tells SAS to overwrite the **OUTFILE** if it already exists. If **REPLACE** is omitted and the file exists, the procedure will typically terminate with an error.
SHEET: This sub-option allows the user to specify the descriptive name that will appear on the tab of the Excel workbook. If omitted, SAS defaults to a generic name based on the **DATA** name.

Case Study 1: Exporting a Single Dataset

The most frequent use case for PROC EXPORT involves transferring a single, finalized Dataset into its own, dedicated Excel workbook or sheet. To demonstrate this foundational process, we will

first create a small, hypothetical Dataset named **MY_DATA** within the SAS session. This data simulates typical input containing three numeric variables (A, B, and C) and seven observations.

The following SAS code block uses the **DATA** step and **DATALINES** statement to input the raw data directly. Following creation, the **PROC PRINT** statement is included to verify the contents of the newly created **MY_DATA** before we proceed with the export operation, ensuring that the source data is exactly as intended.

```
/*Create the sample dataset MY_DATA in the SAS environment*/
data my_data;
input A B C;
datalines;
1 4 76
2 3 49
2 3 85
4 5 88
2 2 90
4 6 78
5 9 80
;
run;

/*View the dataset content to confirm structure and values*/
proc print data=my_data;
```

As displayed in the resulting output (or log entry, depending on the SAS environment), the **MY_DATA** table is successfully generated, containing 7 rows and 3 columns, plus an observation number column typically added by **PROC PRINT**. This visual confirmation is vital before proceeding to the file export step.

Obs	A	B	C
1	1	4	76
2	2	3	49
3	2	3	85
4	4	5	88
5	2	2	90
6	4	6	78
7	5	9	80

Executing the Single Dataset Export

With the **MY_DATA** Dataset successfully loaded into the SAS session, we can now execute the export command. We instruct PROC EXPORT to take the data, format it using the **DBMS=XLSX** option, and save it to the specified **OUTFILE** path. The inclusion of the **REPLACE** option ensures that if we run this code multiple times, the output file is automatically updated without generating a file access error.

For clarity and organization within the target spreadsheet, we utilize the **SHEET** statement to assign a descriptive name, "First Data," to the worksheet containing the exported content. This practice is strongly recommended, especially when the workbook is intended for external reporting or distribution, as it improves the usability of the final Excel file.

```
/*Executing the PROC EXPORT statement to create my_data.xlsx*/
proc export data=my_data
outfile="/home/u13181/my_data.xlsx"
dbms=xlsx
replace;
sheet="First Data";
run;
```

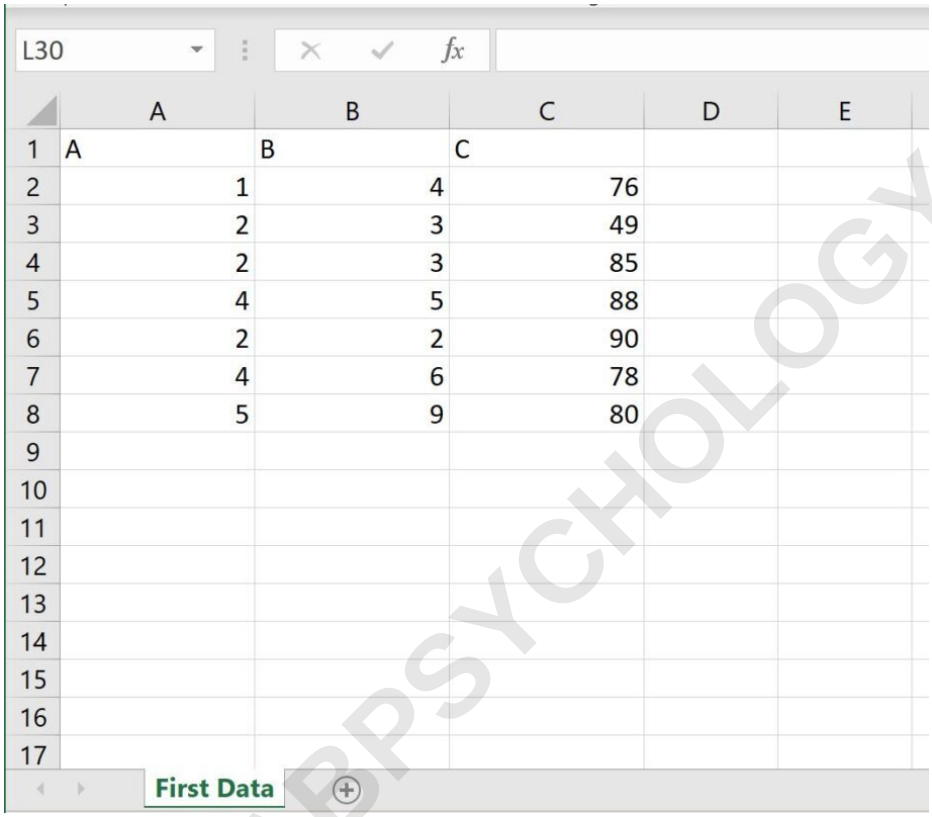
Upon successful execution, the SAS log will confirm the number of records written. The user can then navigate to the specified path (**/home/u13181/my_data.xlsx** in this example) to open and inspect the exported file in Microsoft Excel.

Validating the Exported Excel Workbook

Once the export procedure is complete, opening the generated file in Excel allows for immediate

validation. We confirm two primary aspects: first, that the data values and structure (columns A, B, C, and the 7 rows of data) perfectly match the original SAS Dataset; and second, that the metadata, specifically the sheet name, adheres to the instruction provided in the **PROC EXPORT** statement.

The image below confirms that the Dataset structure has been maintained, with variable names appearing correctly as column headers in the Excel sheet. Furthermore, the tab at the bottom of the workbook is correctly labeled "First Data," validating the use of the **SHEET** option.



Maintaining consistency between the SAS source and the Excel destination is critical for reliable data analysis pipelines. In the absence of specific formatting requests, SAS ensures that numerical data types are preserved and character data is correctly encoded, simplifying the downstream use of the data within the Excel environment for charting or formula application.

Case Study 2: Exporting Multiple Datasets to a Single Workbook

A more complex, yet common, requirement in analytical reporting is the need to consolidate results from several related Datasets into a single, cohesive Excel workbook, with each Dataset occupying its own sheet. This approach greatly streamlines the delivery of comprehensive reports. The key to achieving this successfully lies in careful management of the **OUTFILE** and **REPLACE** options within sequential **PROC EXPORT** calls.

To demonstrate this, we first establish two distinct SAS Datasets: **MY_DATA** (our original set) and **MY_DATA2**, which contains different variables (D, E, F) and a varying number of observations. Both datasets must be defined and loaded into the SAS work library before the export phase can begin.

```
/*Creating the first dataset (MY_DATA)*/
data my_data;
input A B C;
datalines;
1 4 76
2 3 49
2 3 85
4 5 88
2 2 90
4 6 78
5 9 80
;
run;
```

```
/*Creating the second, distinct dataset (MY_DATA2)*/
data my_data2;
input D E F;
datalines;
1 4 90
2 3 49
2 3 85
4 5 88
2 1 90
;
run;
```


The crucial difference when exporting multiple Datasets is that the first **PROC EXPORT** step must create the target Excel file, potentially using the **REPLACE** option. Subsequent **PROC EXPORT** calls targeting the exact same **OUTFILE** path will then append new sheets to the existing workbook, provided a unique **SHEET** name is supplied for each new export operation.

Sequential Export Commands

The process involves running the **PROC EXPORT** command twice, consecutively. The crucial detail here is that the **OUTFILE** path (**/home/u13181/my_data.xlsx**) must remain identical in both

procedure calls, indicating that both Datasets should reside in the same physical file. The **DBMS=XLSX** designation inherently supports multi-sheet writing to a single file.

For the first export (using **MY_DATA**), the **REPLACE** option ensures that the workbook is either created anew or overwritten if it already exists. The second export (using **MY_DATA2**) automatically appends a new sheet because the combination of **OUTFILE**, **DBMS=XLSX**, and a unique **SHEET** name instructs SAS not to overwrite the entire file but simply to add the new data as a new worksheet within the existing structure.

```
/*Exporting MY_DATA to the first sheet ("First Data")*/
proc export data=my_data
outfile="/home/u13181/my_data.xlsx"
dbms=xlsx
replace;
sheet="First Data";
run;
```

```
/*Exporting MY_DATA2 to the second sheet ("Second Data")*/
proc export data=my_data2
outfile="/home/u13181/my_data.xlsx"
dbms=xlsx
replace;
sheet="Second Data";
run;
```

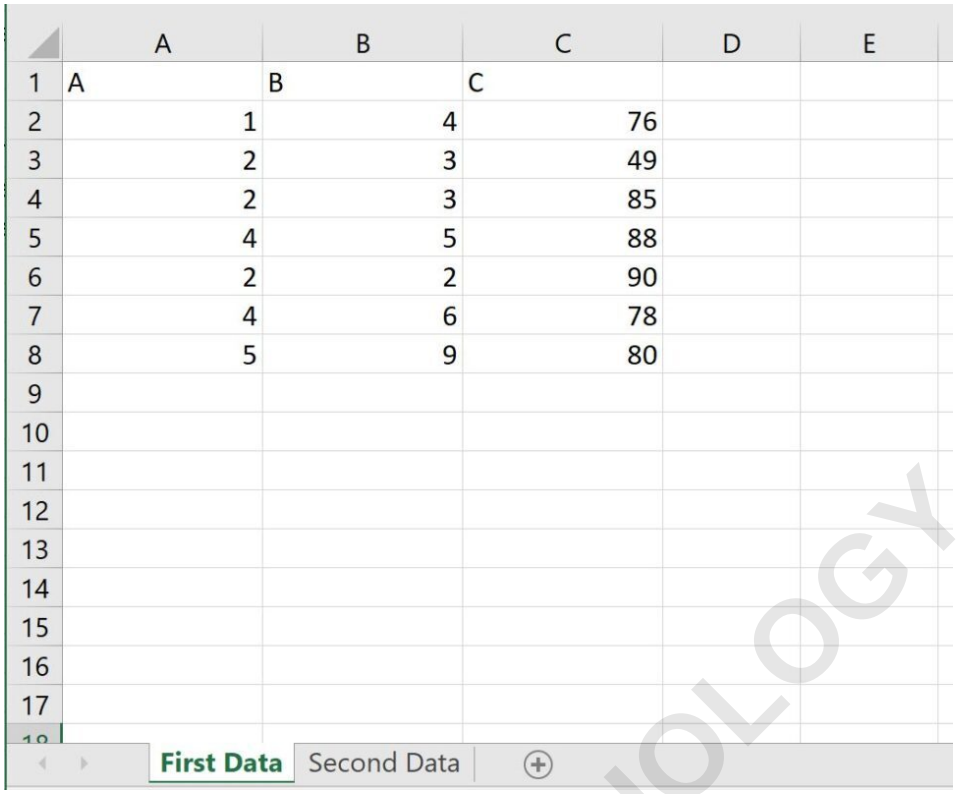

It is essential to note that while the **REPLACE** option is used in both statements, it applies specifically to the sheet name within the file if that sheet already exists. If **REPLACE** were omitted, and the sheet name "First Data" already existed in the target file, SAS would typically halt the procedure or append the data, depending on the server configuration. Using **REPLACE** ensures the sheet is freshly written each time.

Verification of Multi-Sheet Output

After running the sequential export procedures, the resulting **my_data.xlsx** file now contains all the data required, neatly organized into separate worksheets. Navigating to the file location and opening the workbook confirms that the structure and content are correct, demonstrating the utility of **PROC EXPORT** for complex reporting needs.

The first verification confirms that the worksheet labeled "First Data" holds the original **MY_DATA** content, consisting of variables A, B, and C. This sheet is displayed below, confirming the

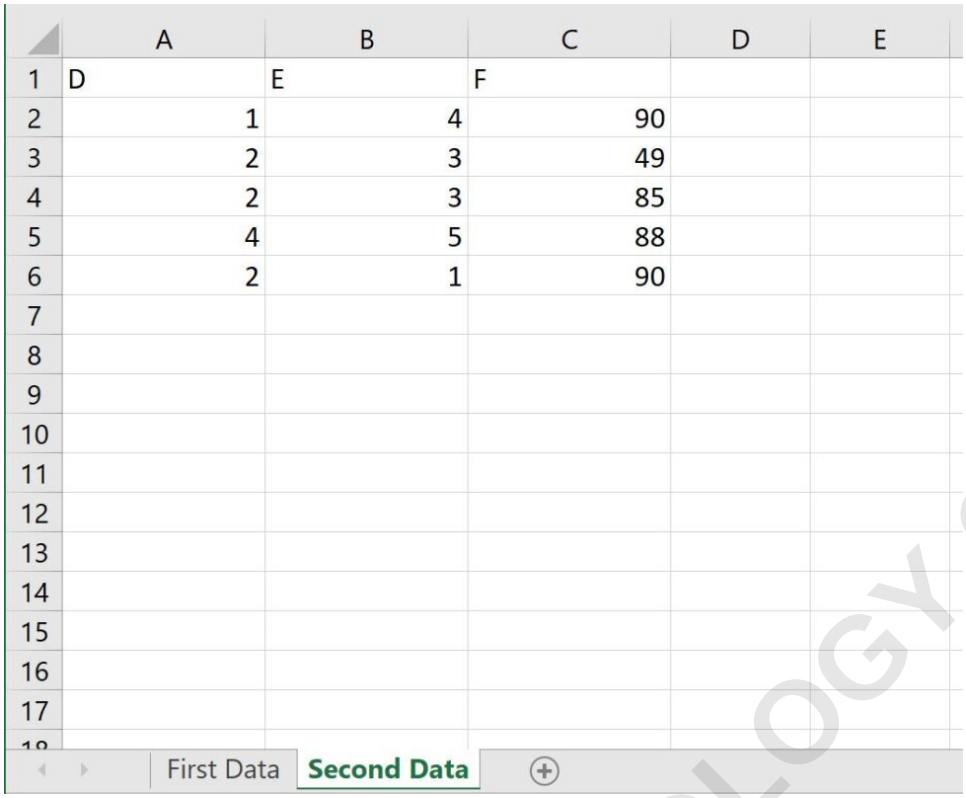
successful execution of the first **PROC EXPORT** call.



	A	B	C	D	E
1	A	B	C		
2	1	4	76		
3	2	3	49		
4	2	3	85		
5	4	5	88		
6	2	2	90		
7	4	6	78		
8	5	9	80		
9					
10					
11					
12					
13					
14					
15					
16					
17					

First Data Second Data

Subsequently, selecting the "Second Data" tab reveals the content of the **MY_DATA2** Dataset. This sheet contains the variables D, E, and F, confirming that the second **PROC EXPORT** step successfully appended the new data to the existing workbook without corrupting or replacing the previously written data on "First Data." This ability to manage multiple outputs is a powerful feature when collaborating between SAS and Excel.



	A	B	C	D	E
1	D	E	F		
2	1	4	90		
3	2	3	49		
4	2	3	85		
5	4	5	88		
6	2	1	90		
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					

First Data | Second Data

General Best Practices for PROC EXPORT

While PROC EXPORT is robust, implementing a few best practices can prevent common errors. Always ensure that the specified path in the **OUTFILE** parameter is accessible and writable by the SAS session, especially in server or enterprise environments. Path errors are the most frequent cause of failed exports. Furthermore, consistently using **DBMS=XLSX** is recommended for modern systems, as this format handles larger files and more complex data structures than its older **XLS** counterpart.

For analysts working with specialized formats, the **DBMS** option is highly flexible and supports exports to other formats like CSV, TXT, and various database systems. For example, using **DBMS=DLM** allows data to be exported as a text file where the delimiter can be explicitly defined, offering an alternative to Excel when simple, raw data transfer is required.

In summary, mastering the use of the **PROC EXPORT** procedure, particularly its ability to manage multiple sheets within a single file, is a fundamental skill for any SAS programmer. This methodology ensures data fidelity and organizational clarity when migrating data from SAS's analytical environment to the widely used Excel platform.

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