

How do I Export Data from SAS to Text File?

Authored by
stats writer

November 19, 2025

RECOMMENDED CITATION

stats writer (2025). *How do I Export Data from SAS to Text File?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=96991>

Exporting data from SAS to a text file is a fundamental and often necessary operation for data analysts and programmers. This process allows data generated or manipulated within the SAS environment to be easily shared with other applications, systems, or platforms that require flat file formats like CSV or tab-separated values.

The most efficient and reliable method for achieving this export is by utilizing the built-in **PROC EXPORT** procedure. This powerful procedure is designed specifically for translating SAS datasets into various external file types, offering extensive control over the output format, including the choice of delimiter, file naming conventions, and handling of variable names. Once configured, data can be seamlessly transferred, making it readily available for external processing.

Utilizing the PROC EXPORT Procedure

The core functionality for external data output in SAS relies on the **PROC EXPORT** statement. This procedure minimizes the complexity of writing custom DATA step code for file output, providing a quick and standardized way to generate various external file outputs, including standard text files.

Understanding the basic structure of the **PROC EXPORT** command is essential for successful data translation. The following syntax template illustrates the fundamental components required to export a dataset to an external text file:

```
/*export data to file called my_data.txt*/  
proc export data=my_data  
outfile="/home/u13181/my_data.txt"  
dbms=tab  
replace;  
run;
```

Key Parameters of the Export Statement

Each keyword within the **PROC EXPORT** statement serves a specific function in defining the source data and the characteristics of the target file. It is crucial to define these options accurately to ensure data integrity during the export process.

data: Specifies the name of the source dataset residing within the SAS environment that you intend to export. This parameter is mandatory.

outfile: Defines the complete file path and filename where the resulting text file will be saved on your operating system or server.

dbms: Determines the type of external file format. For generating delimited text files, `tab` is typically used, signaling that the output should conform to a text format structure.

replace: An optional but highly useful statement. When included, it instructs SAS to overwrite the output file if a file with the same name already exists in the specified `outfile` location, preventing execution errors.

The following practical examples show how to leverage this procedure for various export requirements.

Example 1: Export Dataset to Text File with Default Settings

Suppose we begin with a sample dataset in SAS containing information about various basketball players. This dataset, named `my_data`, holds metrics such as player rating, points, assists, and rebounds. We first define and view the data using the DATA step and `PROC PRINT`:

```
/*create dataset*/  
data my_data;  
input rating points assists rebounds;  
datalines;  
90 25 5 11  
85 20 7 8  
82 14 7 10  
88 16 8 6  
94 27 5 6  
90 20 7 9  
76 12 6 6  
75 15 9 10  
87 14 9 10  
86 19 5 7  
;  
run;  
  
/*view dataset*/  
proc print data=my_data;
```

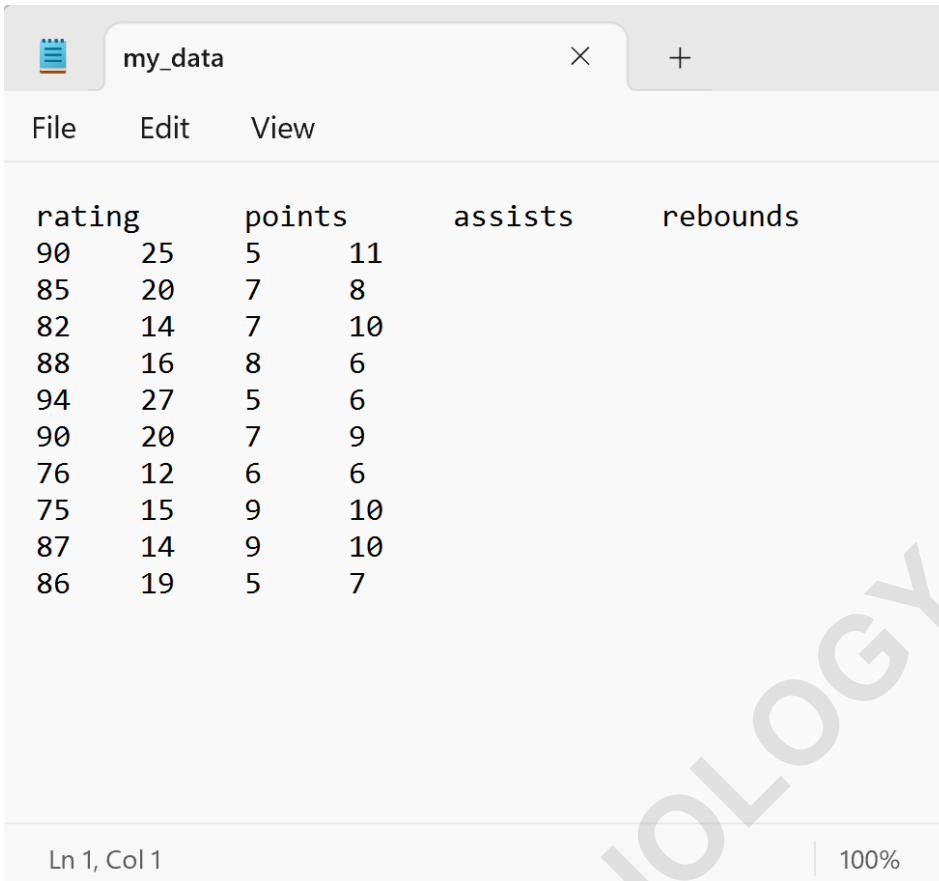
Obs	rating	points	assists	rebounds
1	90	25	5	11
2	85	20	7	8
3	82	14	7	10
4	88	16	8	6
5	94	27	5	6
6	90	20	7	9
7	76	12	6	6
8	75	15	9	10
9	87	14	9	10
10	86	19	5	7

Executing the Default Export

To perform the export using the default settings, which often result in a tab-separated file when `dbms=tab` is used, we execute the following `PROC EXPORT` code block. This command creates a file named `my_data.txt` at the specified output path:

```
/*export dataset*/  
proc export data=my_data  
outfile="/home/u13181/my_data.txt"  
dbms=tab  
replace;  
run;
```

Once the procedure successfully runs, we can navigate to the location on the computer where the file was exported to verify its contents:



rating	points	assists	rebounds
90	25	5	11
85	20	7	8
82	14	7	10
88	16	8	6
94	27	5	6
90	20	7	9
76	12	6	6
75	15	9	10
87	14	9	10
86	19	5	7

The resulting text file accurately mirrors the data contained within the source dataset from SAS, including the header row.

Example 2: Customizing Delimiters and Headers

While the default settings are often sufficient, data sharing frequently requires adherence to specific file formats. For instance, the receiving system might require a semicolon as a field delimiter, or demand that the header row (variable names) be excluded from the resulting text file.

The **PROC EXPORT** procedure accommodates these needs through the `DELIMITER` and `PUTNAMES` arguments, allowing for precise control over the output structure.

Applying Custom Settings for Export

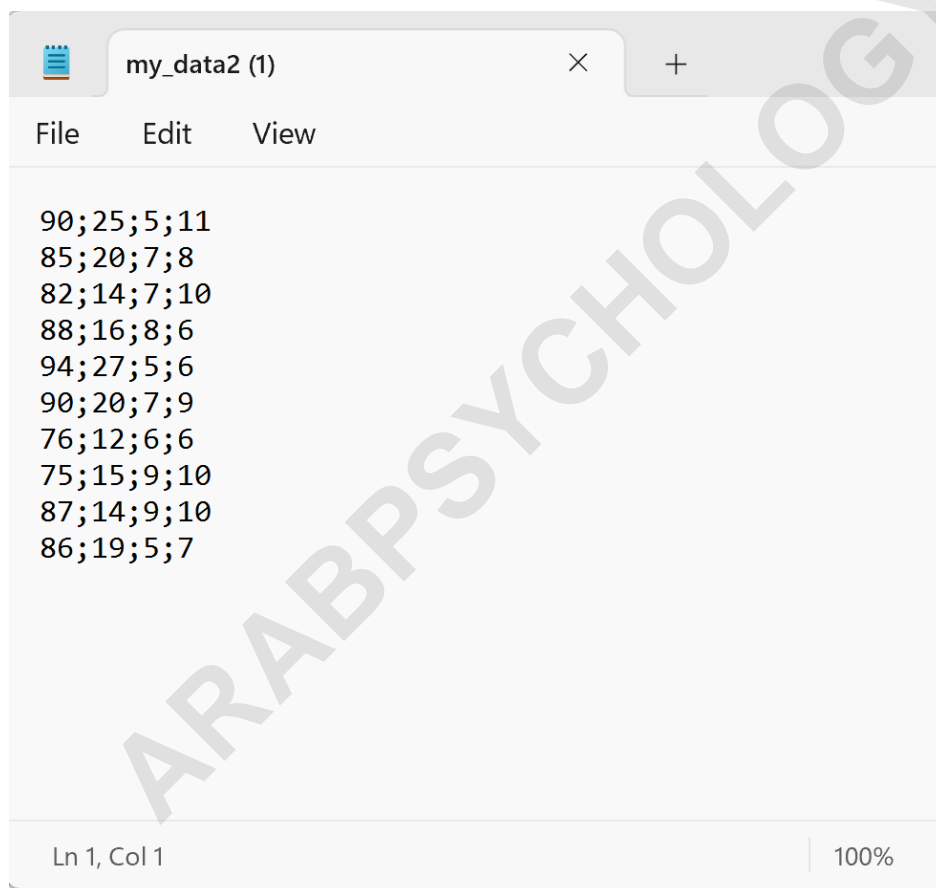
```
/*export dataset*/  
proc export data=my_data  
outfile="/home/u13181/my_data2.txt"  
dbms=tab  
replace;
```

```
delimiter=";";  
putnames=NO;  
run;
```

In this customized execution, we are generating a new file named `my_data2.txt`. The `DELIMITER=" ; "` option forces the use of a semi-colon as the field separator, while `PUTNAMES=NO` ensures that the variable names are suppressed, resulting in a cleaner data-only output file.

Verification of Custom Output

Navigating to the export location and opening the new file confirms that the custom instructions were correctly applied by SAS.



Notice that the header row has been removed, and the data values are clearly separated by semi-colons instead of the default tab or space delimiter. This demonstrates the flexibility of **PROC EXPORT** in handling specific external file requirements.

Further SAS Resources

For advanced options regarding file encoding, handling of complex data structures, or detailed information about other optional parameters, always refer to the complete official documentation for the **PROC EXPORT** statement.

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

[How to Import Text Files into SAS](#)

ARABPSYCHOLOGY.COM