

# How do I export data from SAS to a CSV file? Can you provide examples?

Authored by  
**stats writer**

July 1, 2024

## RECOMMENDED CITATION

stats writer (2024). *How do I export data from SAS to a CSV file? Can you provide examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164930>

To export data from SAS to a CSV file, follow these steps:

1. Open the SAS program and load the desired dataset.
2. Use the PROC EXPORT statement to specify the dataset and the desired file format (in this case, CSV).
3. Specify the file path and name for the CSV file.
4. Use the OUTFILE statement to specify the delimiter and other formatting options for the CSV file.
5. Run the program to export the data to the CSV file.

Here is an example of the PROC EXPORT statement:

```
PROC EXPORT DATA = mydata  
OUTFILE = "/path/to/mydata.csv"  
DBMS = CSV REPLACE;  
DELIMITER = ','  
RUN;
```

This will export the dataset "mydata" to a CSV file named "mydata.csv" located at the specified file path. The data will be separated by commas as specified by the DELIMITER option.

Other formatting options, such as specifying column names or labels, can also be included in the PROC EXPORT statement.

Overall, exporting data from SAS to a CSV file is a simple process that can be easily customized to fit specific needs.

## Export Data from SAS to CSV File (With Examples)

**You can use proc export to quickly export data from SAS to a CSV file.**

**This procedure uses the following basic syntax:**

```
/*export data to file called data.csv*/  
proc exportdata=my_data
```

```
outfile="/home/u13181/data.csv"  
dbms=csv  
replace;  
run;
```

Here's what each line does:

**data:** Name of dataset to export  
**outfile:** Location to export CSV file  
**dbms:** File format to use for export  
**replace:** Replace the file if it already exists

The following examples show how to use this function in practice.

**Related:**

Example 1: Export Dataset to CSV with Default Settings

Suppose we have the following dataset in SAS:

```
/*create dataset*/  
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 dataset*/
```

```
proc printdata=my_data;
```

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

**We can use the following code to export this dataset to a CSV file called data.csv:**

```
/*export dataset*/
```

```
proc exportdata=my_data
```

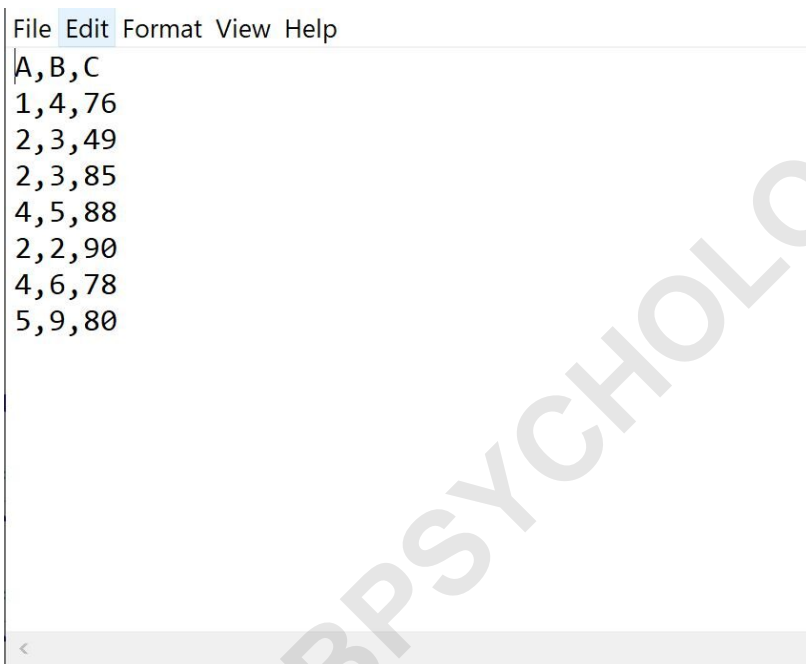
```
outfile="/home/u13181/data.csv"
```

```
dbms=csv
```

```
replace;
```

```
run;
```

I can then navigate to the location on my computer where I exported the file and view it:

A screenshot of a text editor window with a menu bar containing 'File', 'Edit', 'Format', 'View', and 'Help'. The text area contains the following CSV data:

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

A horizontal scrollbar is visible at the bottom of the text area.

The data in the CSV file matches the dataset from SAS.

**Example 2: Export Dataset to CSV with Custom Settings**

For example, the following code shows how to export a SAS dataset to a CSV file using a semi-colon as the delimiter and no header row:

```
/*export dataset*/  
proc exportdata=my_data  
outfile="/home/u13181/data.csv"  
dbms=csv  
replace;  
delimiter=";";  
putnames=NO;  
run;
```

I can then navigate to the location on my computer where I exported the file and view it:

```
1;4;76  
2;3;49  
2;3;85  
4;5;88  
2;2;90  
4;6;78  
5;9;80
```

Notice that the header row has been removed and the values are separated by semi-colons instead of

**commas.**

### **Additional Resources**

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

ARABPSYCHOLOGY.COM