

How can I import CSV files into SAS, and what are some examples of how to do so?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I import CSV files into SAS, and what are some examples of how to do so?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164945>

SAS (Statistical Analysis System) is a powerful software used for data analysis and management. One of its key functionalities is the ability to import CSV (Comma Separated Values) files, which are a popular format for storing and exchanging data.

To import a CSV file into SAS, you can use the IMPORT procedure or the IMPORT wizard. The IMPORT procedure is suitable for advanced users who want more control over the import process, while the IMPORT wizard is a user-friendly tool for beginners. Both methods allow you to specify the location of the CSV file, the delimiter used, and any other formatting options.

Examples of how to import CSV files into SAS include importing a customer database for demographic analysis, importing a sales data file for forecasting, and importing a survey data file for statistical analysis. Additionally, SAS also allows you to import CSV files directly from a URL, making it convenient for accessing data from online sources.

In conclusion, importing CSV files into SAS is a straightforward process that can be done using the IMPORT procedure or the IMPORT wizard. This functionality allows users to easily access and analyze data in various industries and applications.

Import CSV Files into SAS (With Examples)

You can use proc import to quickly import data from a CSV file into SAS.

This procedure uses the following basic syntax:

```
/*import data from CSV file called my_data.csv*/  
proc import out=my_data  
datafile="/home/u13181/my_data.csv"  
dbms=csv  
replace;  
getnames=YES;  
run;
```

Here's what each line does:

out: Name to give dataset once imported into SAS
datafile: Location of CSV file to import
dmbms: Format of file being imported
replace: Replace the file if it already exists
getnames: Use first row as variable names (Set to NO if first row does not contain variable names)

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

Related:

Example 1: Import Data from CSV File into SAS

Suppose we have the following CSV file called `my_data.csv`:

File Edit Format View Help

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

We can use the following code to import this dataset into SAS and call it new_data:

```
/*import data from CSV file called my_data.csv*/  
proc import out=new_data  
datafile="/home/u13181/my_data.csv"  
dbms=csv  
replace;  
getnames=YES;  
run;
```

```
/*view dataset*/proc printdata=new_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

The data shown in the SAS output matches the data shown in the CSV file.

Note: We used `getnames=YES` when importing the file since the first row of the CSV file contained variable names.

Example 2: Import Data from CSV File into SAS with No Header and Custom Delimiter

Suppose we have the following CSV file called data.csv:

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

Notice that this file has no header row and the values are separated by semi-colons instead of commas.

We can use the following code to import this dataset into SAS and call it new_data:

```
/*import data from CSV file called data.csv*/  
proc import out=new_data  
datafile="/home/u13181/data.csv"  
dbms=csv  
replace;  
delimiter=";";  
getnames=NO;  
run;
```

```
/*view dataset*/proc printdata=new_data;
```

Obs	VAR1	VAR2	VAR3
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

The data shown in the SAS output matches the data shown in the CSV file.

By default, SAS provides the variable names as VAR1, VAR2, and VAR3.

Additional Resources

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