What is the complete guide for using PROC IMPORT in SAS?

Authored by

stats writer

June 23, 2024

The complete guide for using PROC IMPORT in SAS is a comprehensive resource that outlines the steps and techniques for efficiently importing data into SAS using PROC IMPORT. This guide provides detailed instructions on how to use PROC IMPORT to read data from various external file formats, such as Excel, CSV, and text files, and convert them into SAS datasets. It also covers advanced features, such as specifying data types, handling missing values, and creating custom import templates. With this guide, users can easily and effectively import data into SAS for further analysis and manipulation.

The Complete Guide: Use PROC IMPORT in SAS

You can use the PROC IMPORT statement to import external data files into SAS.

This statement uses the following basic syntax:

```
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 SASdatafile: Location of file to importdbms: Format of file being importedreplace: Replace the file if it already existsgetnames: Use first row as variable names (Set to

NO if first row does not contain variable names)

You can use this general syntax to import virtually any type of file into SAS. You will only need to change the value for the dbms argument depending on the type of file you're importing.

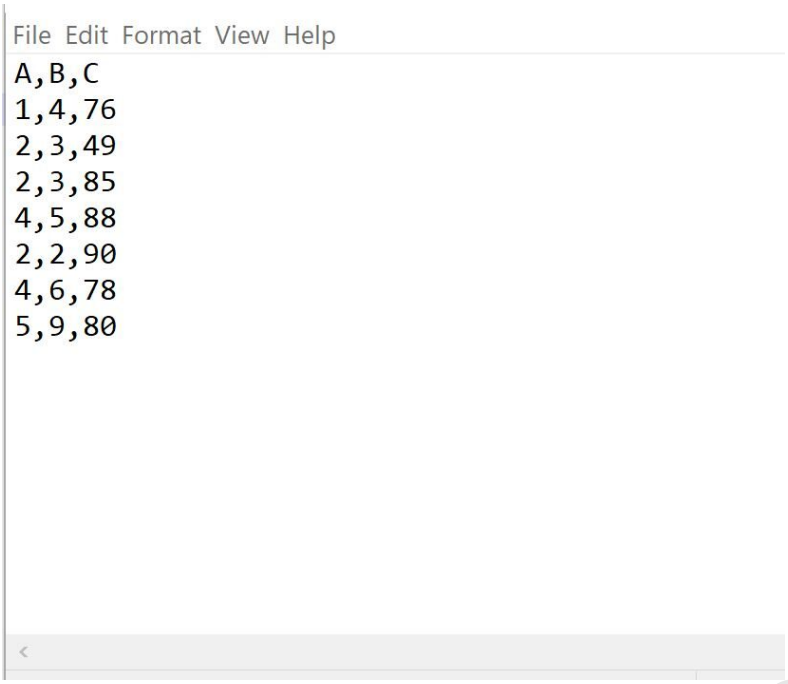
For example:

To import a CSV file, specify dbms=csvTo import an Excel file, specify dbms=xlsxTo import a Text file, specify dbms=dlm

The following examples show how to use PROC IMPORT to import each of these types of files.

Example 1: Use PROC IMPORT to Import CSV File

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



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.

Example 2: Use PROC IMPORT to Import Excel File

	A	B	C	D	E	F
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						
18						
19						
20						

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

```
/*import data from Excel file called my_data.xlsx*/
proc import out=new_data
datafile="/home/u13181/my_data.xlsx"
dbms=xlsx
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 Excel file.

Example 3: Use PROC IMPORT to Import Text File

Suppose we have the following text file called data.txt:

```
column1 column2
1 4
3 4
2 5
7 9
9 1
6 3
4 4
5 2
4 8
6 8
```


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

```
/*import data from text file called data.txt*/
proc import out=new_data
datafile="/home/u13181/data.txt"
dbms=dlm
replace;
getnames=YES;
```

```
run;

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

Obs	column1	column2
1	1	4
2	3	4
3	2	5
4	7	9
5	9	1
6	6	3
7	4	4
8	5	2
9	4	8
10	6	8

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

Note: Refer to the SAS for a complete list of optional arguments you can use when importing files.

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