

SAS is a commonly used statistical program utilized for a variety of analyses. SAS supports many types of files as input and the following tutorial will cover some of the most popular.

SAS Libraries:

Using a SAS library is the best way to save your SAS dataset. By default, data is placed in a temporary library called the `work` library. A library may be created by specifying a folder in which to store the SAS dataset.

```
libname adata "S:\BERD";
```

Creates a SAS library named `adata`. Any data stored in the `adata` library will be saved into the folder `S:\BERD`.

To save a data set called `example` in the `adata` library, we use the following line of code:

```
data adata.example; set example; run;
```

Inputting data:

Data may be directly input into SAS using a data step. This can take two forms.

The first initializes and names the variables using an input statement and then uses the datalines statement to input one observation in each line.

```
data ex1;
input id y x;
datalines;
1 2.1 3.2
2 3.4 4.1
3 2.0 2.9
4 4.2 6.0
5 1.5 2.7
;
run;
```

The second initializes and names the variables using an input statement and then uses the datalines statement to input multiple observations in each line. Text variables should be followed by '\$' in the input statement. To input multiple observations in each line, follow the variable names with '@@' in the input statement. This indicates to SAS that a new observation begins with every other entry.

```
data ex2;
input id $ area @@;
datalines;
y1 1571 y1 814 y1 2150 y1 1761 y2 838 y2 613
y2 1240 y2 1297 y3 1596 y3 1000 y3 1216 y4 2072
y4 1155 y4 773 y4 1666
;
```

Importing from text (.txt) file:

Next, we address importing the text file. If your data file is a simple text file, this is the statement you use:

```
proc import datafile="S:\BERD\Tutorials\Draft\Importing
Data\ex3.txt"
out=ex3
dbms=dlm
replace;
getnames=no;
run;
```

You will now have a SAS data set that you can analyze, create graphs with, or otherwise manipulate as needed.

This is the most basic .txt import, but there are other options that are included. Say, for example, the first line of your data set contained column names. The above function call would have those column names be the first entry in your data set (and names the columns VAR1 to VARn). To fix this and have the column names properly displayed you would change getnames=no to getnames=yes:

```
proc import datafile="S:\BERD\Tutorials\Draft\Importing
Data\ex3.txt"
  out=ex3
  dbms=dlm
  replace;
getnames=yes;
run;
```

This will take the first row of your data file and correctly interpret it as the names of the columns in your SAS data set.

If you have a data file where each row is separated by a character, such as “,”, for instance, you can use the `delimiter=` option.

```
proc import datafile="S:\BERD\Tutorials\Draft\Importing
Data\ex4.txt"
  out=ex4
  dbms=dlm
  replace;
getnames=yes;
delimiter=",";
run;
```

If you have a tab delimited data file, which are somewhat common, you should use:

```
proc import datafile="S:\BERD\Tutorials\Draft\Importing
Data\ex5.txt"
  out=ex5
  dbms=dlm
  replace;
getnames=yes;
delimiter='09'x;
run;
```

There are many more options you can utilize with `proc import` if you have a data set that won't import properly or just want to format it in a different way. If you have an interest or need of these additional options, go to:

<http://support.sas.com/documentation/cdl/en/proc/65145/HTML/default/viewer.htm#n18jyszn33umngn14czw2qfw7thc.htm>

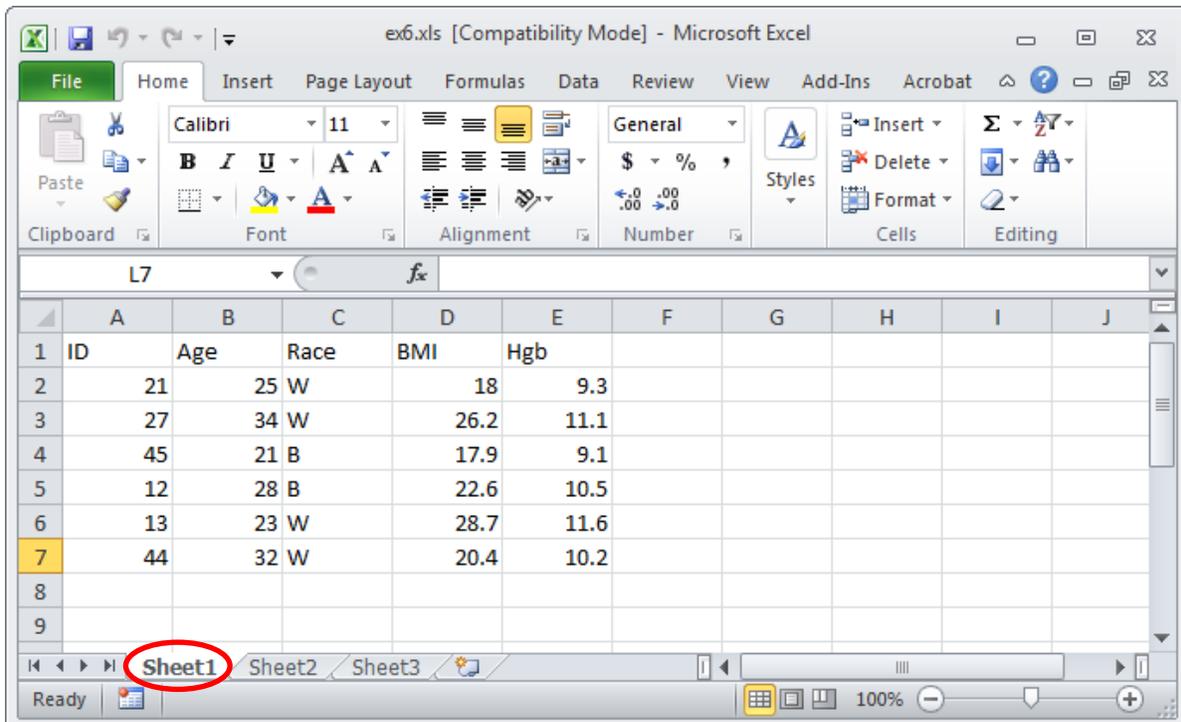
This will direct you to the SAS support site and give you a detailed view of all the options available to you.

Importing from excel (.xls, .xlsx, .csv) file:

To import from excel as an excel spreadsheet file:

```
proc import datafile="S:\BERD\Tutorials\Drafts\Importing
Data\ex6.xls"
  out=ex6
  dbms=xls
  replace;
sheet="Sheet1";
getnames=yes;
mixed=yes;
run;
```

Note that both .xls and .xlsx files may be imported. To import a .xlsx file, simply change xls to xlsx in both the `datafile` line and the `dbms` line. The name in the `sheet` line should be the specific sheet in your excel workbook you wish to import. For this example, the data is found on Sheet1.



To import from excel as a CSV (comma delimited) file:

```
proc import datafile="S:\BERD\Tutorials\Drafts\Importing
Data\ex7.csv"
  out=ex7
  dbms=csv
  replace;
getnames=yes;
run;
```

Importing from other Statistical Software:

SAS also has the ability to import data files from other statistical software packages, such as SPSS, STATA, and JMP. In this tutorial we will look at SPSS.

```
proc import datafile="S:\BERD\Tutorials\Drafts\Importing
Data\ex8.sav"
  out=ex8
  dbms=sav
  replace;
run;
```

A particularly nice feature about importing SPSS data files is that variable labels and lengths are retained in the import. Another nice feature of importing SPSS data files is that any variable formats (value labels) are automatically imported to SAS from SPSS.