Getting Data in SAS

Neelesh Singh

Module Content

- Introduction to data inputs
- Inline data,
- external raw file
- SAS Data
- Temporary and Permanent Datasets
- Format and In format
- Import wizard
- Proc import
- Proc Contents
- Common issues with imports
- Using data _null_, put and _n_
- Getting Data from Server

SAS needs to know 3 things while data is created

- Where is the current Data
    - Instream – embedded in current program
    - Raw data file
    - SAS data file present in a folder
- What is the Format of Data
    - Set defined column size
    - Separated by a delimiter
    - Does it contain Special Characters eg Emails
    - Format of the variables
- Location of output data
    - Temporary data
    - Permanent data

Creating Inline Data

<u>Creating a Data Set with a single observation and multiple variables</u>

```
data test;
x=2; y=3;
run;
```

Creates a data set named test with one observation and 2 variables

<u>Note</u> :
Every sentence in SAS should end with a semicolon
Every data step should end with the run statement

<u>Creating a Data Set with multiple observations and multiple variables</u>

```
data test;
input a b;
cards;
12 34
37 45
32 88
run;
```

Creates a data set named test with multiple observations and 2 variables

<u>Note</u> :
The input statement is used to declare the variables in the event where there are multiple observations
The cards statement is a mandatory part of syntax before inputting values for the variables

Creating Inline Data

Creating a Data Set with a character variable

```
data test;
input a $ b $6.;
cards;
12 Jack
37 James
run;
```

Creates a data set named test with 2 character Variables. A $ after the variable name indicates a character variable

Alternative way of inputting values to variables

```
data test;
input a b;
datalines;
12 34
37 45
32 88
run;
```

The datalines statement acts as an alternative to the Cards statement

Creating Inline Data

Creating a Data Set with length specifications

```
data t;
input city $ 1-10 Street $ 11-21 Revenue 22-25 Cost 26-28;
cards;
BANGALORE Church st  234 23
CHENNAI  T NAGAR   657 67
DELHI   CP   564 34
KANPUR  MG ROAD  455 36;
run;
```

Taking input from an existing data set

Making an exact copy of a data set

```
data test1;
set test;
run;
```

Creates a temporary data set called test which is an exact replica of the data set test in the work directory

Note :
The data statement indicates the name of the data set that is being created
The set statement indicates the name of the data set from which the input is being taken

Creating a new variable from an existing data set

```
data test1;
set test;
y=2+b-a;
run;
```

The variable y is being created in the data set test1 which takes inputs from the data set test where the Variables a and b are present

Permanent and Temporary Datasets

Temporary datasets

```
PROC PRINT data = work.test1;
RUN;
Or
PROC PRINT data = test1;
RUN;
```

Permanent datasets

```
Data SAS1.Test;
SET WORK.test1;
RUN;

Data Test;
SET SAS1.test1;
RUN;
```

Reading permanent Datasets

Reading permanent Datasets

Importing dataset from a Raw File

- Reading a raw data file

```
data t;
infile 'C:\data\temp3.dat';
input city $ 1-10 Street $ 11-21 Revenue 22-25 Cost 26-28;
cards;
run;
```

- Reading a file with delimiter

```
infile 'C:\mydata\test.dat' dsd dlm='|' lrecl=1024;
infile 'C:\mydata\test.txt' dsd dlm='09'x truncover;  for TAB
```

- Skip the header using FIRSTOBS= option

```
infile 'C:\mydata\test.dat' dsd dlm='~' firstobs=2;
```

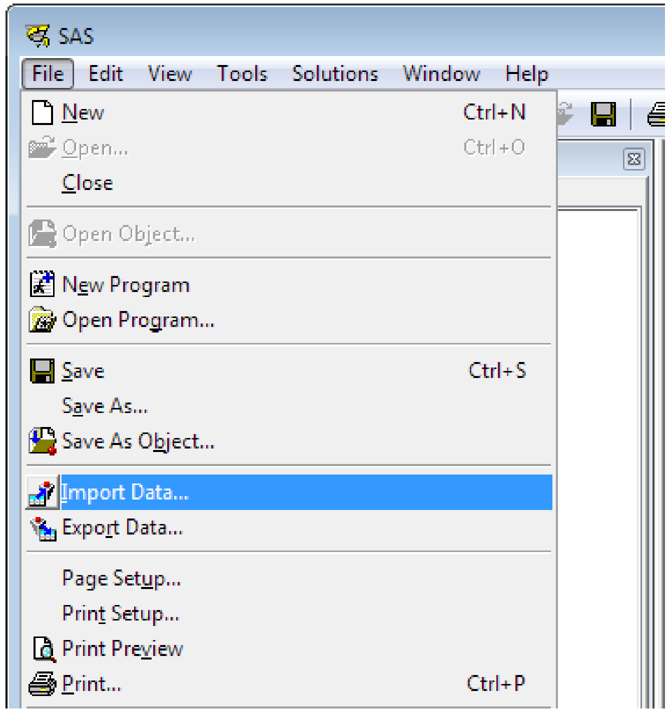
PROC Import

- Proc Import Procedure –
- Reads 600 records using tilde ~ delimiter

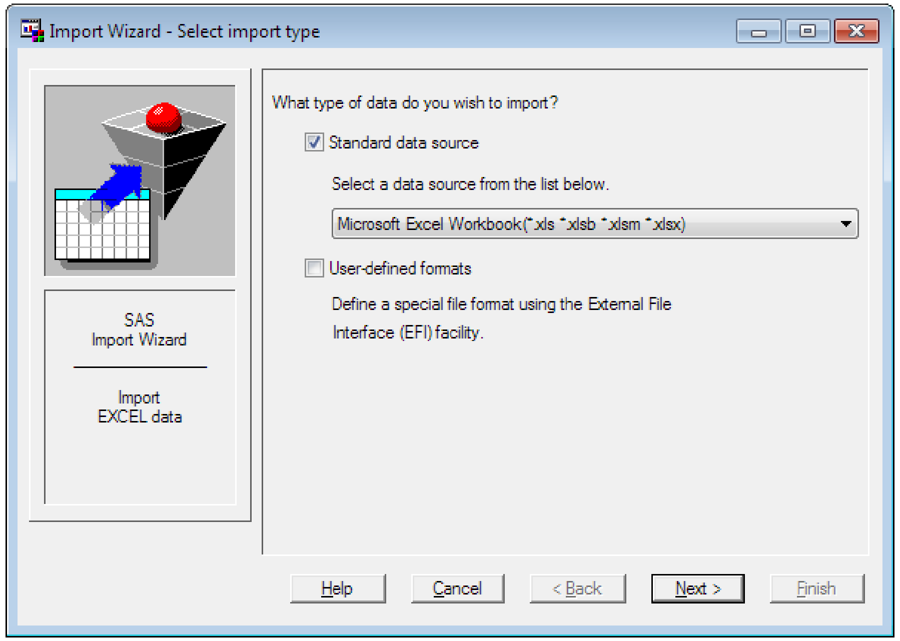
```
proc import datafile='c:\mydata\test.fil' dbms=dlm out=work.test replace;    delimiter="~";
getnames=yes;
 guessingrows=500;
run;
```

SAS import Wizard

1



2



What type of data do you wish to import?

[✓] Standard data source

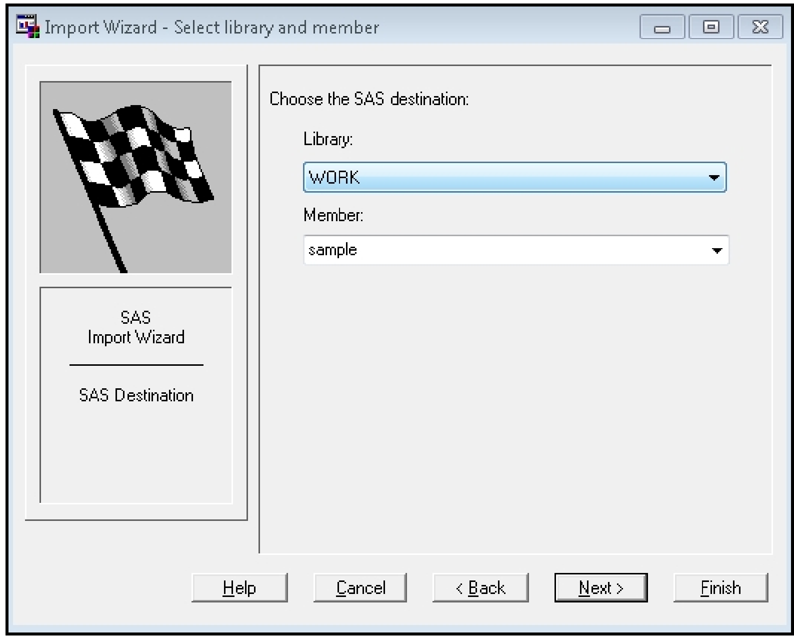
Select a data source from the list below.

Microsoft Excel Workbook(*.xls *.xlsb *.xlsm *.xlsx)

[ ] User-defined formats

Define a special file format using the External File Interface (EFI) facility.

3

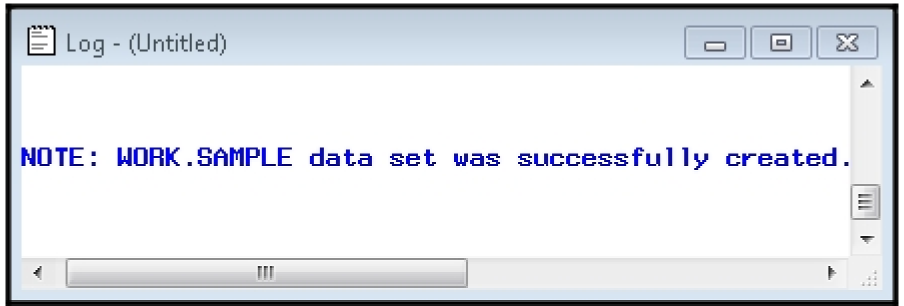


Choose the SAS destination:

Library: WORK

Member: sample

4



```
NOTE: WORK.SAMPLE data set was successfully created.
```

Proc Contents

- To view content of a dataset

```
PROC CONTENTS data = sashelp.class;
RUN;
```

- To view content of full library

```
PROC CONTENTS data = sashelp._ALL_ NODS;
RUN;
```

NODS allows data of each file to not get printed in the output

ALL is an automatic variable which references to all SAS files in the library

FORMAT and INFORMAT

- INFORMAT to set the input format in the data

```
DATA Class;
  INPUT name $ score 3. bday MMDDYY10.;
  DATALINES;
       John 56 09/14/1992
       Miranda 38 09/14/1987
       Amit 91 09/15/1985;
  RUN;
```

- FORMAT is to change output (view) format of the data

```
DATA students;
  SET Class;
  FORMAT bday MMDDYY10.;
RUN;
```

- FORMAT can be used in PROCs too

```
PROC PRINT DATA = students;
  VAR ids bday;
  FORMAT bday WORDDATE20.;
RUN;
```

PROC FORMAT can be used to change the format of data

To format the .(missing) As N/A

```
proc format;
     value Variable .='N/A' other=[12.1];
run;
```

Common issues with import

- Large Files
- Formatting
- Spaces
- Mixed formats
- Extra data in the end
- Data Quality issues

Using data _null_ , put and _n_

Using the _null_ and put Statements

```
data _null_;
x=5;
y=x**2;
put x y;
run;
```

The data _null_ creates a data set in SAS memory which can be used for future references
We cannot give the name _null_ to any dataset. This is reserved for SAS.This will not create a data set.
The put statement puts the output in the Log

Using the _n_ variable

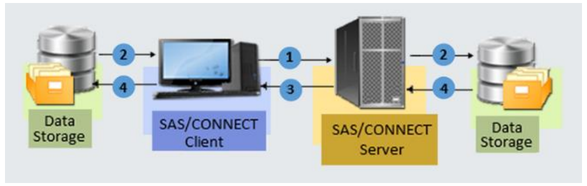
```
data test1;
set t1;
if _n_<=2;
run;
```

The variable _n_ identifies the number of iterations that the SAS process has undergone
This name cannot be assigned to any variable. It is reserved for SAS
This is an internal variable and does not get created in the data set but remains there in SAS memory from where it can be used for computations

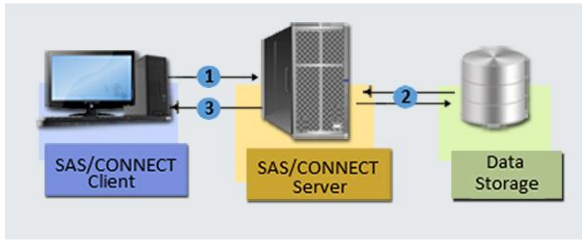
Data Access from Server using SAS/CONNECT

- SAS/CONNECT software is a SAS client/server toolset that provides the ability to manage, access, and process data in a distributed and parallel SAS environment.

- Data is transferred using the UPLOAD and DOWNLOAD procedures.



- Compute Services provides enable you to direct the execution of SAS programs to one or more server sessions. An RSUBMIT block, or a "remote submit," is a block of statements that are created in the client session using the RSUBMIT and ENDRSUBMIT statements.



Data Access from Database

Creating SAS dataset from DBMS table

```
proc sql ;
connect to oracle (user=neelesh password=nsingh path="ORCL");
create table work.banking as
select * from connection to oracle
(select A.*
from account A, transaction B
where A.cust_Id=B.cust_Id);
disconnect from oracle;
quit;
```

greatlearning
Learning for Life

THANK YOU