

## Importing Excel Data into SAS

PROC IMPORT is the SAS procedure used to read data from excel into SAS.

### PROC IMPORT Syntax:

```
PROC IMPORT  
DATAFILE="filename"  
OUT=SAS-data-set  
DBMS=identifier  
REPLACE;  
SHEET="Sheet-name";  
GETNAMES=YES;  
DATAROW=N;  
RANGE="range-name";  
RUN;
```

1. **DATAFILE=** option tells SAS where to find the Excel file that you want to import  
(*Complete filename path*).

For example : **DATAFILE = "C:\Desktop\age.xls"**

2. **OUT=** option tells SAS to create a dataset with any name of your choice. By default, the imported dataset is saved on WORK library  
(*temporary library*)

Examples :

i. **OUT = Age** .

In this statement, PROC IMPORT uses the WORK library. This implies  
OUT = Age is equivalent to OUT = Work.Age .

ii. **OUT = Input.Age**.

In this statement, PROC IMPORT uses the Input library (*Permanent library*).

3. **DBMS=** option tells SAS the type of file to read.

Examples :

- i. **DBMS = XLS** for Excel 97-2003 workbooks
- ii. **DBMS = XLSX** for Excel 2007 - 2013 workbooks

4. **REPLACE** is used to overwrite the existing SAS dataset (If any) mentioned in the **OUT=** option.

5. **SHEET=** option is used to specify which sheet SAS would import.

Examples :

- i. **SHEET = "Sheet1"** - To import data from worksheet named **sheet1**.
- ii. **SHEET = "Goal"** - To import data from worksheet named **Goal**.

6. **GETNAMES= YES** tells SAS to use the first row of data as variable names.

By default, PROC IMPORT uses **GETNAMES= YES**. If you type **GETNAMES= NO**, SAS would not read variable names from first row of the sheet.

7. **DATAROW=** option is used to specify starting row from where SAS would import the data.

For example : **DATAROW =5** tells SAS to start reading data from row number 5.

**Note :**

- i. When **GETNAMES=YES**, **DATAROW** must be greater than or equal to 2.
- ii. When **GETNAMES=NO**, **DATAROW** must be greater than or equal to 1

8. **RANGE=** option is used to specify which range SAS would import.

Examples :

- i. **RANGE="Sheet1\$B2:D10"**

This would tell SAS to import data from range B2:D10 from sheet1

## The CHART Procedure

- The procedure CHART is typically used to draw low-resolution vertical or horizontal bar charts, block charts, or pie charts.
- The general structure of a proc chart step is

```
PROC CHART < options >;  
  BLOCK variables < / options >;  
  BY variables;  
  HBAR variable < / options >;  
  PIE variable < / option >;  
  STAR variable < / option >;  
  VBAR variable < / option >;
```

- The PROC statement options in PROC chart is the “DATA=” option for naming the data set to be analyzed.
- The variables that appear in any of the statements HBAR, VBAR, BLOCK, PIE, or STAR statements above specifies the variables for which these charts are produced.
- The options that may be specified following the slash in each of the HBAR, VBAR, BLOCK, PIE, or STAR statements enable one to customize the appearance of the charts.
- Options for HBAR, VBAR, BLOCK, PIE, or STAR statements
  - “LEVELS=” specify the number of bars representing each chart variable when the variables given in the VBAR statement are continuous.
  - “SYMBOL= *character*” defines the symbol to be used in the body of standard HBAR and VBAR charts with no subgrouping.
  - “MIDPOINT= *values*” defines the range of value for the chart variable each bar or section represents by specifying the range midpoints.
  - “REF= *n*” (*n* is integer if frequency on y-axis) or “REF= *p*” ( $0 \leq p \leq 100$  if cumulative frequency on y-axis) request that a single reference line be drawn on the response axis.
  - “TYPE= *statistics*” specifies what the bars in the chart represent (by default: TYPE=FREQ).

Option	Results
TYPE=FREQ	Frequency count
TYPE=PCT	Percentages
TYPE=CFREQ	Cumulative frequencies
TYPE=CPCT	Cumulative percentages
TYPE=SUM	Totals
TYPE=MEAN	Means

- “GROUP= *variable*” produce side by side charts, with each chart representing the observations having a given value of the GROUP=variable.
- “SUBGROUP= *variable*” requests that each bar be subdivided into characters that show the SUBGROUP= variable’s contribution to the bar.

### Example 5.1(a): PROC PLOT

```
DATA HTWT;
INPUT SEX $ WEIGHT HEIGHT;
CARDS;
M      68      155
F      61      99
F      63      115
M      70      205
M      69      170
F      65      125
M      72      220
;
PROC PLOT;
PLOT WEIGHT*HEIGHT='*' / BOX
                                VAXIS=60 TO 74 BY 1
                                HAXIS=90 TO 230 BY 10
                                VREF=64;
RUN;
```

### Example 5.1(b): PROC PLOT

```
PROC SORT;
BY WEIGHT;

PROC PLOT;
BY SEX;
PLOT WEIGHT*HEIGHT='*' / BOX
                                VAXIS=60 TO 74 BY 1
                                HAXIS=90 TO 230 BY 10
                                VREF=64;
RUN;
```