Chapter 115

Importing Data

Introduction

NCSS can import from a wide variety of spreadsheets, databases, and statistical systems. When you import a file, the entire dataset is replaced with the imported data, so make sure that your data are saved before performing this operation. You will be prompted to save your data before performing the import if your data have not yet been saved. If you are importing from a database file like Access with more than one table non-empty table, you will be asked to select one table to import. If you are importing from a spreadsheet program like Excel with more than one non-empty sheet, you will be asked to select one or more sheets to import.

The table below presents a list of the types of files that can be imported by **NCSS**.

List of Imported Files

File Name	File Type	File Extension
Access	Database	ACCDB, MDB
dBase	Database	DBF
Epi Info	Database	REC
Excel	Spreadsheet	XLSX, XLS
Gauss	Stat System	DAT
JMP	Stat System	JMP
LimDep	Stat System	CPJ
Lotus 123	Spreadsheet	WK1, WK3
Matlab	Stat System	MAT
Minitab	Stat System	MTW
NCSS 97-2007 Spreadsheet	Stat System	S0
NCSS 97-2007 Database	Stat System	S0Z
Paradox	Database	DB
Quattro Pro	Spreadsheet	WKQ, WB*, WQ*
R	Stat System	RDATA
SAS	Stat System	SD2, SAS7BDAT, XPT, TPT
SPLUS	Stat System	DAT
SPSS	Stat System	SAV, POR
Stata	Stat System	DTA
Statistica	Stat System	STA
Symphony	Spreadsheet	WR1, WRK
Systat	Stat System	SYS, SYZ
Text (Delimited)	Text File	TXT, CSV
Text (Fixed-Width)	Text File	TXT, PRN

Import Limitations

You will likely run out of memory if the file you are importing results in more than about 25 million cells. You should limit the number of columns you are importing if you have a very large number of rows.

How to Import a File

The Import Wizard guides you through the steps of importing a file. Follow the on-screen instructions to complete the import.

Selecting a File to Import

Before the Import Wizard starts, you must select the type of file to import. To do this, select *File* > *Import* from the Data Window menu and then select the type of file to import. The most-commonly imported files are displayed in the call-out menu. If the file type you want to import is not on this list, select *Other File Format*. Either way, this will bring up a file selection dialog prompting you to select a file to import. You can change the file type by clicking on the dropdown box at the bottom of the dialog window near the *File Name* box. When you click the dropdown box, a complete list of importable file types will be displayed. Select the file type you want, navigate to and select the file on your computer, and then click *Open*. This will start the Import Wizard with the selected file. You cannot change the file after the wizard has started; to change the file you must cancel the wizard and start again.

The Import Wizard

Once you have selected a file to import, the Import Wizard will guide you through the steps necessary to import a file. The number of steps and the information required in the wizard depend on the type of file you are importing. There are three basic steps:

1. Data File Preview
2. Choosing How Columns are Imported
3. Running the Import Operation

Data File Preview

You will first be presented with a preview of the data file you have selected to import. This will show you the data as it exists in the data file before it is imported. For some file types, you will be prompted to provide additional information about the file you are importing. The information is described later in this chapter.

Choosing How Columns are Imported

The final step of the Import Wizard will ask you to select the columns to import and specify the data type of each imported column. There are three possible data types: *General (Numeric or Text)*, *Text*, and *DateTime*. By default, the data type of each imported column is set to *General*, which accepts numeric and text values.

We will now briefly describe how each data type affects the imported data.

General (Numeric or Text)

The *General* data type is the default data type for **NCSS** dataset columns. Data may either be text values or numeric. Dates are treated as text. Numeric values in columns with Data Type = General are converted to 16-digit floating-point numbers. Non-numeric text values as not converted.

Text

The *Text* data type is useful when you have large numeric identification values that you do not want converted to 16-digit floating-point numbers. Numeric text values will still be interpreted as numbers in transformations and calculations for data analysis.

For example, if the value 987654321234567898765432 1 is imported with *Data Type* = *General*, it will be converted to the maximum floating-point representation of 9.87654321234568E+24. Some of the digits are removed because a floating-point number can only store up to 16 digits. If the same value is imported with *Data Type* = *Text*, the imported value will maintain all of the digits 98765432123456789876543 21 because it is stored as a text value.

DateTime

The *DateTime* data type is useful when you want to format numeric values as dates. DateTime values are considered numeric in transformation and data analysis calculations. Data values in DateTime data columns are still stored with 16-digit floating point accuracy even though they are formatted as dates and times.

Running the Import Operation

Finally, press the *Finish* button to run the import. When you import a file, the entire dataset is replaced with the imported data, so make sure that your data are saved before performing this operation. You will be prompted to save your data before performing the import if your data have not yet been saved. The import operation will import only those columns you have selected.

Once you have imported a data file, we suggest you save it as an **NCSS** dataset with extension **.NCSS*. Remember, the file exists only in your computer's memory until it is saved. If you want to avoid importing the file over and over, save it as an **NCSS** dataset after you have imported it.

Special Instructions for Certain File Types

Some file types require additional information to complete the import procedure. This section explains the additional information that is required by these file types.

General Text File Options

NCSS can import data from either delimited or fixed-width text (or ASCII) files. These files may have the extensions **.txt*, **.prn*, or **.csv*. Files with the extension **.txt* may either be delimited or fixed-width. Files with the extension **.prn* must fixed-width. Files with the extension **.csv* are always comma-delimited.

When importing either delimited or fixed-width text files, you must specify the *Number of Lines per Record* and the *Record Containing Column Names* (if any).

Number of Lines per Record

This option specifies how many rows in a text file are to be considered as a single row in the resulting dataset. This is usually set to "1".

Record Containing Column Names

This option specifies which row in the text file (if any) contains the labels that will be used as column names. As you change this number, the text file preview will highlight the selected row in red. If there are not any column names in the file, set this option to "None".

Importing Delimited Text Files

A delimited text file is one whose columns are separated by some character that is consistent through the whole file. Commonly, delimited text files have columns separated by a comma or a tab character. When importing a delimited text file, you must specify what character is used to separate columns in the file. The Import Wizard will try to determine from the text file itself, which character is used. If the wizard does not automatically select the correct character, you can change it to one of the available options. If you choose "Other", then you must specify a delimiter in the box provided. The preview at the bottom of the screen will show you how the data will appear after import if the selected delimiter is used.

Importing Fixed Width Text Files

A fixed width (or fixed format) text file is one in which each data value occurs at the same position on each row, with each column represented by a fixed number of characters. The data may appear as a solid string of numbers. A format statement is needed to tell the program how to break the data apart into columns.

Specify the Fixed Format

The fixed width syntax is based on three single-letter commands and the slash character. These commands are combined to form the format statement. These commands will be discussed next, followed by examples of format statements.

C is for Variable

The character *C* is used to designate a column. The actual syntax is *rCn*. The *r* indicates the number of times the format segment is repeated. The *n* represents the number of positions (characters) that are used. If *r* is omitted, it is assumed to be one. Following are some examples of this type of format.

Format	Meaning
C1	The column is the next single character on the row.
C3	The column is the next three characters on the row.
2C4	Two columns, each four characters long.
3C1,2C2	Three columns that are each one character long followed by two columns that are each two characters long.

X is for Skip

The character *X* is used to designate the skipping of certain character positions on the row. The actual syntax is *Xn*. The *n* represents the number of positions (characters) that are skipped. Following are some examples of this type of format.

Format	Meaning
X1	Skip the next character position along the row.
X2	Skip the next two character positions along the row.
X25	Skip the next twenty-five character positions along the row.
X2,X8	Skip two and then eight character positions. Of course, you would usually write X10 instead.

T is for Transfer

The character *T* is used to transfer to a specific character position on the current row. This character position becomes the next position processed by the format decoder. If your format includes multiple rows, you cannot use the *T* command to move back to a previous row or ahead to the next row.

Format	Meaning
T1	Transfer to the first position.
T22	Transfer to position twenty-two.

/ is for Next Row

The character / is used to transfer to the beginning of the next row.

Examples of Fixed Format Statements

The above format commands, except for the slash, are put together using commas. The slash serves as its own separator and does not need to be combined with a comma. Note that the values are assigned to the dataset columns in sequence.

Below are some examples of how these format commands can be placed together to form the format statement.

File Segment	Format Statement	Interpreted Values
12345 7890	10C1	1, 2, 3, 4, 5, missing value, 7, 8, 9, 10
12 4567890	5C2	12, 4, 56, 78, 90
1234567890	C2,X4,C3	12, 789
1234567890	C2,T7,C3	12, 789
1234567890	C3,T1,3C1	123, 1, 2, 3
1234567890	C2,C3,C5	12, 345, 67890
1234567890	(combined with the next line)	
2345678901	2C5/C3	12345, 67890, 234

Importing SAS or Access Files

Each Microsoft Access database contains one or more tables. Each table may be thought of as an independent database. When you are importing data from an Access database, you will have to designate a single table to import. Each table has its own set of columns. Only one Access table may be imported at a time. Some SAS files also include multiple tables. You can only import one table at a time from these files as well.

File Encoding

For some file types (dBase, Epi Info, Gauss, JMP, LimDep, Lotus 123, Matlab, Minitab, Paradox, Quattro, R, SAS, SPlus, SPSS, Stata, Statistica, Symphony, and Systat) you must specify the file encoding. **NCSS** will attempt to determine what file encoding is appropriate for the source file, but sometimes errors related to the file encoding do occur. If you do not specify a file encoding and **NCSS** is unable to determine the appropriate encoding, then the default system encoding will be used.

If the file you are importing contains non-English or non-ASCII characters, we suggest that you review the data after the import is complete to make sure that these characters were imported correctly. If they were not, try importing the file again using a different source file encoding.