Title

stata.com

import excel — Import and export Excel files

Description

`import excel` loads an Excel file, also known as a workbook, into Stata. `import excel` *filename*`, describe` lists available sheets and ranges of an Excel file. `export excel` saves data in memory to an Excel file. Excel 1997/2003 (`.xls`) files and Excel 2007/2010 (`.xlsx`) files can be imported, exported, and described using `import excel`, `export excel`, and `import excel, describe`.

`import excel` and `export excel` are supported on Windows, Mac, and Linux.

`import excel` and `export excel` look at the file extension, `.xls` or `.xlsx`, to determine which Excel format to read or write.

For performance, `import excel` imposes a size limit of 40 MB for Excel 2007/2010 (`.xlsx`) files. Be warned that importing large `.xlsx` files can severely affect your machine's performance.

`import excel auto` first looks for `auto.xls` and then looks for `auto.xlsx` if `auto.xls` is not found in the current directory.

The default file extension for `export excel` is `.xlsx` if a file extension is not specified.

Quick start

Check the contents of Excel file `mydata.xls` before importing

```
import excel mydata, describe
```

Same as above, but for `mydata.xlsx`

```
import excel mydata.xlsx, describe
```

Load data from `mydata.xls`

```
import excel mydata
```

Same as above, but load data from cells A1:G10 of `mysheet`

```
import excel mydata, cellrange(A1:G10) sheet(mysheet)
```

Read first row as lowercase variable names

```
import excel mydata, firstrow case(lower)
```

Import only `v1` and `v2`

```
import excel v1 v2 using mydata
```

Save data in memory to `mydata.xlsx`

```
export excel mydata
```

Same as above, but export variables `v1`, `v2`, and `v3`

```
export excel v1 v2 v3 using mydata
```

Menu

import excel

File > Import > Excel spreadsheet (*.xls;*.xlsx)

export excel

File > Export > Data to Excel spreadsheet (*.xls;*.xlsx)

Syntax

Load an Excel file

```
import excel [using] filename [, import_excel_options]
```

Load subset of variables from an Excel file

```
import excel extvarlist using filename [, import_excel_options]
```

Describe contents of an Excel file

```
import excel [using] filename, describe
```

Save data in memory to an Excel file

```
export excel [using] filename [if] [in] [, export_excel_options]
```

Save subset of variables in memory to an Excel file

```
export excel [varlist] using filename [if] [in] [, export_excel_options]
```

import_excel_options	Description
`sheet("`*sheetname*`")`	Excel worksheet to load
`cellrange([`*start*`][:`*end*`])`	Excel cell range to load
`firstrow`	treat first row of Excel data as variable names
`case(preserve|lower|upper)`	preserve the case (the default) or read variable names as lowercase or uppercase when using `firstrow`
`allstring[("`*format*`")]`	import all Excel data as strings; optionally, specify the numeric display format
`clear`	replace data in memory
`locale("`*locale*`")`	specify the locale used by the workbook; has no effect on Microsoft Windows

`allstring("`*format*`")` and `locale()` do not appear in the dialog box.

export_excel_options	Description
Main	
`sheet("`*sheetname*`"[, modify|replace])`	save to Excel worksheet
`cell(`*start*`)`	start (upper-left) cell in Excel to begin saving to
`firstrow(variables|varlabels)`	save variable names or variable labels to first row
`nolabel`	export values instead of value labels
`keepcellfmt`	when writing data, preserve the cell style and format of existing worksheet
`replace`	overwrite Excel file
Advanced	
`datestring("`*datetime_format*`")`	save dates as strings with a *datetime_format*
`missing("`*repval*`")`	save missing values as *repval*
`locale("`*locale*`")`	specify the locale used by the workbook; has no effect on Microsoft Windows

`collect` is allowed with `import excel`; see [U] **11.1.10 Prefix commands**.

`locale()` does not appear in the dialog box.

extvarlist specifies variable names of imported columns. An *extvarlist* is one or more of any of the following:

varname
varname=*columnname*

Example: `import excel make mpg weight price using auto.xlsx, clear` imports columns A, B, C, and D from the Excel file `auto.xlsx`.

Example: `import excel make=A mpg=B price=D using auto.xlsx, clear` imports columns A, B, and D from the Excel file `auto.xlsx`. Column C and any columns after D are skipped.

Options for import excel

`sheet("`*sheetname*`")` imports the worksheet named *sheetname* in the workbook. The default is to import the first worksheet.

`cellrange(`[*start*][`:`*end*]`)` specifies a range of cells within the worksheet to load. *start* and *end* are specified using standard Excel cell notation, for example, `A1`, `BC2000`, and `C23`.

`firstrow` specifies that the first row of data in the Excel worksheet consists of variable names. This option cannot be used with *extvarlist*. `firstrow` uses the first row of the cell range for variable names if `cellrange()` is specified. `import excel` translates the names in the first row to valid Stata variable names. The original names in the first row are stored unmodified as variable labels.

`case(preserve|lower|upper)` specifies the case of the variable names read when using the `firstrow` option. The default is `case(preserve)`, meaning to preserve the variable name case. Only the ASCII letters in names are changed to lowercase or uppercase. Unicode characters beyond ASCII range are not changed.

`allstring`[`("`*format*`")`] forces `import excel` to import all Excel data as string data. You can specify the numeric display format used to convert the numeric data to string using the optional argument *format*. See [D] **format**.

`clear` clears data in memory before loading data from the Excel workbook.

The following option is available with import excel but is not shown in the dialog box:

locale("*locale*") specifies the locale used by the workbook. You might need this option when working with extended ASCII character sets. This option has no effect on Microsoft Windows. The default locale is UTF-8.

Options for export excel

Main

sheet("*sheetname*"[, modify | replace]) saves to the worksheet named *sheetname*. If there is no worksheet named *sheetname* in the workbook, a new sheet named *sheetname* is created. If this option is not specified, the first worksheet of the workbook is used. If *sheetname* does exist in the workbook, you can either modify or replace the worksheet.

modify exports data to the worksheet without changing the cells outside the exported range. This option cannot be specified with replace, nor when overwriting the Excel workbook.

replace clears the worksheet before the data are exported to it. replace cannot be specified with modify, nor when overwriting the Excel workbook.

cell(*start*) specifies the start (upper-left) cell in the Excel worksheet to begin saving to. By default, export excel saves starting in the first row and first column of the worksheet.

firstrow(variables | varlabels) specifies that the variable names or the variable labels be saved in the first row in the Excel worksheet. The variable name is used if there is no variable label for a given variable.

nolabel exports the underlying numeric values instead of the value labels.

keepcellfmt specifies that, when writing data, export excel should preserve the existing worksheet's cell style and format. By default, export excel does not preserve a cell's style or format.

replace overwrites an existing Excel workbook. replace cannot be specified when modifying or replacing a given worksheet: export excel ..., sheet("", modify) or export excel ... sheet("", replace).

Advanced

datestring("*datetime_format*") exports all datetime variables as strings formatted by *datetime_format*. See [D] **Datetime display formats**.

missing("*repval*") exports missing values as *repval*. *repval* can be either string or numeric. Without specifying this option, export excel exports the missing values as empty cells.

The following option is available with export excel but is not shown in the dialog box:

locale("*locale*") specifies the locale used by the workbook. You might need this option when working with extended ASCII character sets. The default locale is UTF-8.

Remarks and examples

stata.com

To demonstrate the use of import excel and export excel, we will first load auto.dta and export it as an Excel file named auto.xlsx:

```
. use https://www.stata-press.com/data/r18/auto
(1978 automobile data)
. export excel auto, firstrow(variables)
file auto.xlsx saved
```

Now we can import from the auto.xlsx file we just created, telling Stata to clear the current data from memory and to treat the first row of the worksheet in the Excel file as variable names:

```
. import excel auto.xlsx, firstrow clear
(12 vars, 74 obs)

. describe

Contains data
 Observations:            74
    Variables:            12
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
make            str17   %17s                  make
price           int     %10.0gc               price
mpg             byte    %10.0g                mpg
rep78           byte    %10.0g                rep78
headroom        double  %10.0g                headroom
trunk           byte    %10.0g                trunk
weight          int     %10.0gc               weight
length          int     %10.0g                length
turn            byte    %10.0g                turn
displacement    int     %10.0g                displacement
gear_ratio      double  %14.2f                gear_ratio
foreign         str8    %9s                   foreign
-------------------------------------------------------------------------------
Sorted by:
     Note: Dataset has changed since last saved.
```

We can also import a subrange of the cells in the Excel file:

```
. import excel auto.xlsx, cellrange(:D70) firstrow clear
(4 vars, 69 obs)

. describe

Contains data
 Observations:            69
    Variables:             4
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
make            str17   %17s                  make
price           int     %10.0gc               price
mpg             byte    %10.0g                mpg
rep78           byte    %10.0g                rep78
-------------------------------------------------------------------------------
Sorted by:
     Note: Dataset has changed since last saved.
```

Both .xls and .xlsx files are supported by import excel and export excel. If a file extension is not specified with export excel, .xlsx is assumed, because this format is more common and is compatible with more applications that also can read from Excel files. To save the data in memory as a .xls file, specify the extension:

```
. use https://www.stata-press.com/data/r18/auto, clear
(1978 automobile data)

. export excel auto.xls
file auto.xls saved
```

To export a subset of variables and overwrite the existing `auto.xlsx` Excel file, specify a variable list and the `replace` option:

```
. export excel make mpg weight using auto, replace
file auto.xlsx saved
```

For additional examples illustrating `import excel` and `export excel`, see Mitchell (2020, chap. 2–3).

❑ Technical note: Excel data size limits

For an Excel `.xls`-type workbook, the worksheet size limits are 65,536 rows by 256 columns. The string size limit is 255 characters.

For an Excel `.xlsx`-type workbook, the worksheet size limits are 1,048,576 rows by 16,384 columns. The string size limit is 32,767 characters.

❑

❑ Technical note: Dates and times

Excel has two different date systems, the "1900 Date System" and the "1904 Date System". Excel stores a date and time as an integer representing the number of days since a start date plus a fraction of a 24-hour day.

In the 1900 Date System, the start date is 00Jan1900; in the 1904 Date System, the start date is 01Jan1904. In the 1900 Date System, there is another artificial date, 29feb1900, besides 00Jan1900. `import excel` translates 29feb1900 to 28feb1900 and 00Jan1900 to 31dec1899.

See *Converting Excel dates* in [D] **Datetime values from other software** for a discussion of the relationship between Stata datetimes and Excel datetimes.

❑

❑ Technical note: Mixed data types

Because Excel's data type is cell based, `import excel` may encounter a column of cells with mixed data types. In such a case, the following rules are used to determine the variable type in Stata of the imported column.

- If the column contains at least one cell with nonnumerical text, the entire column is imported as a string variable.
- If an all-numerical column contains at least one cell formatted as a date or time, the entire column is imported as a Stata date or datetime variable. `import excel` imports the column as a Stata date if all date cells in Excel are dates only; otherwise, a datetime is used.

❑

Video example

Import Excel data into Stata

Stored results

`import excel` *filename*`, describe` stores the following in `r()`:

Scalars

`r(N_worksheet)`	number of worksheets in the Excel workbook

Macros

`r(worksheet_#)`	name of worksheet # in the Excel workbook
`r(range_#)`	available cell range for worksheet # in the Excel workbook

References

Crow, K. 2012. Using import excel with real world data. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2012/06/25/using-import-excel-with-real-world-data/.

Jeanty, P. W. 2013. Dealing with identifier variables in data management and analysis. *Stata Journal* 13: 699–718.

Mitchell, M. N. 2020. *Data Management Using Stata: A Practical Handbook*. 2nd ed. College Station, TX: Stata Press.

Also see

[D] **Datetime** — Date and time values and variables

[D] **export** — Overview of exporting data from Stata

[D] **import** — Overview of importing data into Stata

[M-5] **_docx*()** — Generate Office Open XML (.docx) file

[M-5] **xl()** — Excel file I/O class

[RPT] **putexcel** — Export results to an Excel file