

**putexcel** — Export results to an Excel file

Syntax                      Menu                      Description                      Options  
Remarks and examples      References                      Also see

## Syntax

### Basic syntax

```
putexcel cellexplist using filename [, options]
```

### Advanced syntax

```
putexcel set filename [, putexcel_set_options]
```

```
putexcel describe
```

```
putexcel clear
```

```
putexcel cellexplist [, sheet("sheetname", ...) colwise]
```

<i>options</i>	Description
<u>modify</u>	modify Excel file
<u>replace</u>	overwrite Excel file
<u>sheet</u> ("sheetname" [, <u>replace</u> ])	write to Excel worksheet <i>sheetname</i>
<u>colwise</u>	write <i>resultset</i> values in consecutive columns instead of consecutive rows
<u>keepcellformat</u>	when writing <i>resultset</i> , preserve the cell style and format of an existing worksheet

<i>putexcel_set_options</i>	Description
<u>modify</u>	modify Excel file
<u>replace</u>	overwrite Excel file
<u>sheet</u> ("sheetname" [, <u>replace</u> ])	write to Excel worksheet <i>sheetname</i>
<u>keepcellformat</u>	when writing <i>resultset</i> , preserve the cell style and format of an existing worksheet

*cellexplist* is one or more of any of the following:

```
cell=(exp [, asdate | asdatetime])  
cell=matrix(name [, names | rownames | colnames])  
cell=resultset
```

*cell* is a valid Excel cell specified using standard Excel notation. For matrix(*name*), *cell* is where the first value of the matrix will be written. If you specify a cell multiple times in a `putexcel` command, the rightmost *cell=value* is the one that is written to the Excel file.

If your expression evaluates to a Stata date and you want that date to be written as an Excel date, use (*exp*, *asdate*). If your expression evaluates to a Stata datetime and you want that datetime to be written as an Excel datetime, use (*exp*, *asdatetime*).

Use `matrix(name)` when working with any Stata matrix. `matrix(name, names)` specifies that matrix row and column names, row and column equation names, and the matrix values be written to the Excel worksheet. By default, matrix row and column names are not written. `matrix(name, rownames)` writes the matrix row names and values to the Excel worksheet, and `matrix(name, colnames)` writes the column names and values to the Excel worksheet.

*resultset* is a shortcut name used to identify a group of [return](#) values that are stored by a Stata command. *resultset* can be

<code>rscalars</code>	<code>rscalarnames</code>
<code>escalars</code>	<code>escalarnames</code>
<code>emacros</code>	<code>emacronames</code>
<code>rmacros</code>	<code>rmacronames</code>
<code>ematrices</code>	<code>ematrixnames</code>
<code>rmatrices</code>	<code>rmatrixnames</code>
<code>e*</code>	<code>enames</code>
<code>r*</code>	<code>rnames</code>

Example: `putexcel A1=(2+2)` using file

Write the result of the expression `2+2` into Excel column A row 1 in the file `file.xlsx`.

Example: `putexcel A1=("Mean of mpg") A2=(r(mean))` using file

Write "Mean of mpg" in Excel column A row 1, and write the r-class result `r(mean)` in Excel column A row 2.

Example: `putexcel D14=matrix(A)` using file

Take the Stata matrix A, and write it into Excel using column D row 14 as the upper-left cell for matrix A.

Example: `putexcel D4=("Coefficients") B5=matrix(e(b))` using file

Write "Coefficients" in Excel column D row 4, and write the values of e-class matrix `e(b)` into Excel using column B row 5 as the upper-left cell.

Example: `putexcel A1=rscalars` using file, `sheet("Results")`

Write all r-class scalars in memory to sheet `Results` in `file.xlsx`. The first scalar value in memory will be written in Excel column A row 1, the next value in column A row 2, etc.

Example: `putexcel A1=rscalarnames B1=rscalars` using file, `sheet("Results")`

Write all r-class scalar names and values in memory to sheet `Results` in `file.xlsx`. The first scalar name will be written in Excel column A row 1, the next in column A row 2, and so on, and the first scalar value will be written in column B row 1, the next in column B row 2, and so on.

Example: `putexcel A1=r* B1=e*` using file

Write all r-class scalars, macros, and matrices and all e-class scalars, macros, and matrices in memory to file `file.xlsx`. The first r-class value in memory will be written in Excel column A row 1, the next value in column A row 2, and so on. The first e-class value in memory will be written in Excel column B row 1, the next value in column B row 2, and so on.

## Menu

## Description

`putexcel` writes Stata [expressions](#), [matrices](#), and [stored results](#) to an Excel file. `putexcel` is supported on Windows, Mac, and Linux. Excel 1997/2003 (.xls) files and Excel 2007/2010 (.xlsx) files are supported. `putexcel` looks at the file extension .xls or .xlsx to determine which Excel format to write.

`putexcel set filename` sets the Excel file to create, modify, or replace subsequent `putexcel cellexplist` commands. If *filename* does not exist, `putexcel set` will create the file. If *filename* exists, it will not be modified unless you specify the `modify` or `replace` options.

`putexcel describe` displays the file information set by `putexcel set`.

`putexcel clear` clears the file information set by `putexcel set`.

`putexcel cellexplist` writes Stata [expressions](#), [matrices](#), and [stored results](#) to an Excel file.

The default file extension for `putexcel` is .xlsx.

## Options

`modify` in the basic syntax modifies an existing Excel file.

In the advanced syntax, `modify` allows you to modify the file specified with `putexcel set` using subsequent `putexcel cellexplist` commands.

`replace` in the basic syntax overwrites an existing Excel file.

In the advanced syntax, `replace` replaces the file specified with `putexcel set` and then allows you to modify the file using subsequent `putexcel cellexplist` commands.

`sheet("sheetname")` writes 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.

`sheet("sheetname", replace)` clears the worksheet before values are written to it.

`colwise` specifies that if a [resultset](#) is used, the values written to the Excel worksheet are written in consecutive columns. By default, the values are written in consecutive rows.

`keepcellformat` specifies that when writing the [resultset](#), `putexcel` should preserve the existing worksheet's cell style and format. By default, `putexcel` does not preserve a cell's style or format.

## Remarks and examples

[stata.com](http://www.stata.com)

To demonstrate the use of `putexcel`, we will first load `auto.dta` and export the results of the `summarize` command (see [\[R\] summarize](#)) to an Excel file named `results.xlsx`:

```
. use http://www.stata-press.com/data/r13/auto
(1978 Automobile Data)
. summarize mpg
```

Variable	Obs	Mean	Std. Dev.	Min	Max
mpg	74	21.2973	5.785503	12	41

```

. return list
scalars:
      r(N) = 74
r(sum_w) = 74
r(mean) = 21.2972972972973
r(Var) = 33.47204738985561
r(sd) = 5.785503209735141
r(min) = 12
r(max) = 41
r(sum) = 1576

. putexcel A30=rscalars using results, sheet("June 3") modify
file results.xlsx saved

```

The above command modifies Excel workbook `results.xlsx` sheet June 3 with the following cell values:

```

A30 = 74
A31 = 74
A32 = 21.2972972972973
A33 = 33.47204738985561
A34 = 5.785503209735141
A35 = 12
A36 = 41
A37 = 1576

```

You can write out specific results by using the following command:

```

. putexcel A30=(r(min)) A31=(r(N)) using results, sheet("June 3", replace) modify

```

The above command would write over sheet June 3 in `results.xls` so that just cell A30 and A31 contained values 12 and 74.

You can use `putexcel` to create tables in Excel using Stata `return` results. To create a `tabulate oneway` table of the variable `foreign` in Excel format, type

```

. tabulate foreign, matcell(cell) matrow(rows)

```

Car type	Freq.	Percent	Cum.
Domestic	52	70.27	70.27
Foreign	22	29.73	100.00
Total	74	100.00	

```

. putexcel A1=("Car type") B1=("Freq.") using results,
> sheet("tabulate oneway") replace
file results.xlsx saved

. putexcel A2=matrix(rows) B2=matrix(cell) using results,
> sheet("tabulate oneway") modify
file results.xlsx saved

. putexcel A4=("Total") B4=(r(N)) using results,
> sheet("tabulate oneway") modify
file results.xlsx saved

```

If you are going to export complex tables or export numerous objects, you should use the advanced syntax of `putexcel`. For example, to create a regression table in Excel format using returned results from `regress`, type

```

. use http://www.stata-press.com/data/r13/auto, clear
. regress price turn gear
. putexcel set "results.xls", sheet("regress results")
. putexcel F1=("Number of obs") G1=(e(N))

```

```

. putexcel F2=("F")           G2=(e(F))
. putexcel F3=("Prob > F")   G3=(Ftail(e(df_m), e(df_r), e(F)))
. putexcel F4=("R-squared")  G4=(e(r2))
. putexcel F5=("Adj R-squared") G5=(e(r2_a))
. putexcel F6=("Root MSE")  G6=(e(rmse))
. matrix a = r(table)'
. matrix a = a[., 1..6]
. putexcel A8=matrix(a, names)

```

## □ Technical note

See the technical notes [Excel data size limits](#) and [Dates and times](#) in [D] [import excel](#).



## References

- Crow, K. 2013. Export tables to Excel. The Stata Blog: Not Elsewhere Classified. <http://blog.stata.com/2013/09/25/export-tables-to-excel/>.
- Gallup, J. L. 2012. A new system for formatting estimation tables. *Stata Journal* 12: 3–28.
- Quintó, L. 2012. HTML output in Stata. *Stata Journal* 12: 702–717.

## Also see

- [P] [postfile](#) — Post results in Stata dataset
- [P] [return](#) — Return stored results
- [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