Title

stata.com

putexcel — Export results to an Excel file

Description

`putexcel` writes Stata expressions, matrices, images, and returned results to an Excel file. It may also be used to format cells in an Excel worksheet. This allows you to automate exporting and formatting of, for example, Stata estimation results. Excel 1997/2003 (`.xls`) files and Excel 2007/2010 and newer (`.xlsx`) files are supported.

`putexcel set` sets the Excel file to create, modify, or replace in subsequent `putexcel` commands. You must set the destination file before using any other `putexcel` commands. `putexcel clear` clears the file information set by `putexcel set`. `putexcel describe` displays the file information set by `putexcel set`.

For the advanced syntax that lets you simultaneously write multiple output types, see [P] **putexcel advanced**.

Quick start

Declare the first sheet of `myresults.xlsx` to be the destination workbook for subsequent `putexcel` commands

```
putexcel set myresults
```

As above, but use a new sheet named Estimation Results and replace the existing workbook

```
putexcel set myresults, replace sheet("Estimation Results")
```

Write the text "Coefficients" to cell B1

```
putexcel B1 = "Coefficients"
```

Add variable names and estimated coefficients in the column under "Coefficients" after `regress`, and format coefficients with two decimal places

```
matrix b = e(b)'
putexcel A2 = matrix(b), rownames nformat(number_d2)
```

Format the header row of the table with a bottom border and bold text

```
putexcel (A1:B1), bold border(bottom)
```

Add PNG of a `margins` plot saved to disk as `mymargins.png` where the upper-left corner is aligned with the upper-left corner of cell D2

```
marginsplot, name(mymargins)
graph export mymargins.png, name(mymargins)
putexcel D2 = picture(mymargins.png)
```

Menu

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

Syntax

Set workbook for export

```
putexcel set filename [, set_options]
```

Write expression to Excel

```
putexcel ul_cell = exp [, export_options format_options]
```

Export Stata matrix to Excel

```
putexcel ul_cell = matrix(name) [, export_options format_options]
```

Export Stata graph, path diagram, or other picture to Excel

```
putexcel ul_cell = picture(filename)
```

Export returned results to Excel

```
putexcel ul_cell = returnset [, export_options]
```

Write formula to Excel

```
putexcel ul_cell = formula(formula) [, export_options]
```

Format cells

```
putexcel cellrange, format_options
```

Describe current export settings

```
putexcel describe
```

Clear current export settings

```
putexcel clear
```

ul_cell is a valid Excel upper-left cell specified using standard Excel notation, for example, `A1` or `D4`.

cellrange is *ul_cell* or *ul_cell*`:`*lr_cell*, where *lr_cell* is a valid Excel lower-right cell, for example, `A1`, `A1:D1`, `A1:A4`, or `A1:D4`.

set_options	Description
`sheet(`*sheetname*`[, replace])`	specify the worksheet to use; default is the first worksheet
`modify`	modify Excel file
`replace`	overwrite Excel file

export_options	Description
Main	
overwritefmt	overwrite existing cell formatting when exporting new content
asdate	convert Stata date (%td-formatted) *exp* to an Excel date
asdatetime	convert Stata datetime (%tc-formatted) *exp* to an Excel datetime
asdatenum	convert Stata date *exp* to an Excel date number, preserving the cell's format
asdatetimenum	convert Stata datetime *exp* to an Excel datetime number, preserving the cell's format
names	also write row names and column names for matrix *name*; may not be combined with rownames or colnames
rownames	also write matrix row names for matrix *name*; may not be combined with names or colnames
colnames	also write matrix column names for matrix *name*; may not be combined with names or rownames
colwise	write results in *returnset* to consecutive columns instead of rows

format_options	Description
Number	
nformat(*excelnfmt*)	specify format for numbers
Alignment	
left	left-align text
hcenter	center text horizontally
right	right-align text
top	vertically align text with the top
vcenter	center text vertically
bottom	vertically align text with the bottom
txtindent(#)	indent text by # spaces; default is 0
txtrotate(#)	rotate text by # degrees; default is 0
[no]txtwrap	wrap text within each cell
[no]shrinkfit	shrink text to fit the cell width
merge	merge cells in *cellrange*
unmerge	separate merged cells identified by *ul_cell*
Font	
font(*fontname* [, *size* [, *color*]])	specify font, font size, and font color
[no]italic	format text as italic
[no]bold	format text as bold
[no]underline	underline text in the specified cells
[no]strikeout	strikeout text in the specified cells
script(sub|super|none)	specify subscript or superscript formatting
Border	
border(*border* [, *style* [, *color*]])	specify horizontal and vertical cell border style
dborder(*direction* [, *style* [, *color*]])	specify diagonal cell border style
Fill	
fpattern(*pattern* [, *fgcolor* [, *bgcolor*]])	specify fill pattern for cells

Output types

exp writes a valid Stata expression to a cell. See **[U] 13 Functions and expressions**. Stata dates and datetimes differ from Excel dates and datetimes. To properly export date and datetime values, use `asdate` and `asdatetime`.

`matrix(`*name*`)` writes the values from a Stata matrix to Excel. Stata determines where to place the data in Excel by default from the size of the matrix (the number of rows and columns) and the location you specified in *ul_cell*. By default, *ul_cell* contains the first element of *name*, and matrix row names and column names are not written.

`picture(`*filename*`)` writes a portable network graphics (`.png`), JPEG (`.jpg`), Windows metafile (`.wmf`), device-independent bitmap (`.dib`), enhanced metafile (`.emf`), or bitmap (`.bmp`) file to an Excel worksheet. The upper-left corner of the image is aligned with the upper-left corner of the specified *ul_cell*. The image is not resized. If *filename* contains spaces, it must be enclosed in double quotes.

returnset is a shortcut name that is used to identify a group of return values. It is intended primarily for use by programmers and by those who intend to do further processing of their exported results in Excel. *returnset* may be any one of the following:

returnset	
escalars	escalarnames
rscalars	rscalarnames
emacros	emacronames
rmacros	rmacronames
ematrices	ematrixnames
rmatrices	rmatrixnames
e*	enames
r*	rnames

`formula(`*formula*`)` writes an Excel formula to the cell specified in *ul_cell*. *formula* may be any valid Excel formula. Stata does not validate formulas; the text is passed literally to Excel.

Options

Set

`sheet(`*sheetname* [`, replace`]`)` saves to the worksheet named *sheetname*. If there is no worksheet named *sheetname* in the workbook, then a new sheet named *sheetname* is created. If this option is not specified, the first worksheet of the workbook is used.

`replace` permits `putexcel set` to overwrite *sheetname* if it exists in the specified *filename*.

`modify` permits `putexcel set` to modify an Excel file.

`replace` permits `putexcel set` to overwrite an existing Excel workbook. The workbook is overwritten when the first `putexcel` command is issued.

Main

`overwritefmt` causes `putexcel` to remove any existing cell formatting in the cell or cells to which it is writing new output. By default, all existing cell formatting is preserved.

`asdate` tells `putexcel` that the specified *exp* is a Stata `%td`-formatted date that should be converted to an Excel date with *m/d/yyyy* Excel date format.

This option has no effect if an *exp* is not specified as the output type.

`asdatetime` tells `putexcel` that the specified *exp* is a Stata `%tc`-formatted datetime that should be converted to an Excel datetime with *m/d/yyyy h:mm* Excel datetime format.

This option has no effect if an *exp* is not specified as the output type.

`asdatenum` tells `putexcel` that the specified *exp* is a Stata `%td`-formatted date that should be converted to an Excel date number, preserving the cell's format.

This option has no effect if an *exp* is not specified as the output type.

`asdatetimenum` tells `putexcel` that the specified *exp* is a Stata `%tc`-formatted datetime that should be converted to an Excel datetime number, preserving the cell's format.

This option has no effect if an *exp* is not specified as the output type.

`names` specifies that matrix row names and column names be written into the Excel worksheet along with the matrix values. If you specify `names`, then *ul_cell* will be blank, the cell to the right of it will contain the name of the first column, and the cell below it will contain the name of the first row. `names` may not be specified with `rownames` or `colnames`.

This option has no effect if `matrix()` is not specified as the output type.

`rownames` specifies that matrix row names be written into the Excel worksheet along with the matrix values. If you specify `rownames`, then *ul_cell* will contain the name of the first row. `rownames` may not be specified with `names` or `colnames`.

This option has no effect if `matrix()` is not specified as the output type.

`colnames` specifies that matrix column names be written into the Excel worksheet along with the matrix values. If you specify `colnames`, then *ul_cell* will contain the name of the first column. `colnames` may not be specified with `names` or `rownames`.

This option has no effect if `matrix()` is not specified as the output type.

`colwise` specifies that if a *returnset* is used, the values written to the Excel worksheet be written in consecutive columns. By default, the values are written in consecutive rows.

This option has no effect if a *returnset* is not specified as the output type.

Number

`nformat(`*excelnfmt*`)` changes the numeric format of a cell range. Codes for commonly used formats are shown in the table of numeric formats in the *Appendix*. However, any valid Excel format is permitted. For information about creating your own formats, see the description of `nformat()` in *Options* of [P] **putexcel advanced**.

Alignment

`left` sets the specified cells to have contents left-aligned within the cell. `left` may not be combined with `right` or `hcenter`. Right-alignment is the Excel default for numeric values and need not be specified when outputting numbers.

`hcenter` sets the specified cells to have contents horizontally centered within the cell. `hcenter` may not be combined with `left` or `right`.

`right` sets the specified cells to have contents right-aligned within the cell. `right` may not be combined with `left` or `hcenter`. Left-alignment is the Excel default for text and need not be specified when outputting strings.

`top` sets the specified cells to have contents vertically aligned with the top of the cell. `top` may not be combined with `bottom` or `vcenter`.

`vcenter` sets the specified cells to have contents vertically aligned with the center of the cell. `vcenter` may not be combined with `top` or `bottom`.

`bottom` sets the specified cells to have contents vertically aligned with the bottom of the cell. `bottom` may not be combined with `top` or `vcenter`.

`txtindent(#)` sets the text indention in each cell in a cell range. # must be an integer between 0 and 15.

`txtrotate(#)` sets the text rotation in each cell in a cell range. # must be an integer between 0 and 180 or equal to 255. `txtrotate(0)` is equal to no rotation and is the default. `txtrotate(255)` specifies vertical text. Values `1`–`90` rotate the text counterclockwise 1 to 90 degrees. Values `91`–`180` rotate the text clockwise 1 to 90 degrees.

`txtwrap` and `notxtwrap` specify whether or not the text is to be wrapped in a cell or within each cell in a range of cells. The default is no wrapping. `notxtwrap` has an effect only if the cell or cells were previously formatted to wrap. `txtwrap` may not be specified with `shrinkfit`.

`shrinkfit` and `noshrinkfit` specify whether or not the text is to be shrunk to fit in the cell width of a cell or in each cell of a range of cells. The default is no shrinking. `noshrinkfit` has an

effect only if the cell or cells were previously formatted to shrink text to fit. `shrinkfit` may not be specified with `txtwrap`.

`merge` tells Excel that cells in the specified cell range should be merged. `merge` may be combined with `left`, `right`, `hcenter`, `top`, `bottom`, and `vcenter` to format the merged cell. Merging cells that contain data in each cell will result in the upper-leftmost data being kept.

Once you have merged cells, you can refer to the merged cell by using any single cell from the specified *cellrange*. For example, if you specified a *cellrange* of `A1:B2`, you could refer to the merged cell using `A1`, `B1`, `A2`, or `B2`.

`unmerge` tells Excel to unmerge previously merged cells. When using `unmerge`, you only need to use a single cell from the merged cell in the previously specified *cellrange*.

Font

`font(`*fontname* [`,` *size* [`,` *color*]]`)` sets the font, font size, and font color for each cell in a cell range. If `font()` is not specified, the Excel defaults are preserved.

fontname may be any valid Excel font. If *fontname* includes spaces, then it must be enclosed in double quotes. What constitutes a valid Excel font is determined by the version of Excel that is installed on the user's computer.

size is a numeric value that represents any valid Excel font size. The default is `12`.

color may be one of the colors listed in the table of colors in the *Appendix* or may be a valid RGB value in the form `"### ### ###"`. If no *color* is specified, then Excel workbook defaults are used.

`italic` and `noitalic` specify whether to italicize or unitalicize the text in a cell or range of cells. The default is for text to be unitalicized. `noitalic` has an effect only if the cell or cells were previously italicized.

`bold` and `nobold` specify whether to bold or unbold the text in a cell or range of cells. The default is for text to be unbold. `nobold` has an effect only if the cell or cells were previously formatted as bold.

`underline` and `nounderline` specify whether to underline the text or remove the underline from the text in a cell or range of cells. The default is for text not to be underlined. `nounderline` has an effect only if the cell or cells previously contained underlined text.

`strikeout` and `nostrikeout` specify whether to strikeout the text or remove the strikeout from the text in a cell or range of cells. The default is for text not to have a strikeout mark. `nostrikeout` has an effect only if the cell or cells previously had a strikeout mark.

`script(sub | super | none)` changes the script style of the cell. `script(sub)` makes all text in a cell or range of cells a subscript. `script(super)` makes all text in a cell or range of cells a superscript. `script(none)` removes all subscript or superscript formatting from a cell or range of cells. Specifying `script(none)` has an effect only if the cell or cells were previously formatted as subscript or superscript.

Border

`border(`*border* [`,` *style* [`,` *color*]]`)` sets the cell border, style, and color for a cell or range of cells.

border may be `all`, `left`, `right`, `top`, or `bottom`.

style is a keyword specifying the look of the border. The most common styles are `thin`, `medium`, `thick`, and `double`. The default is `thin`. For a complete list of border styles, see the *Appendix*. To remove an existing border, specify `none` as the *style*.

color may be one of the colors listed in the table of colors in the *Appendix* or may be a valid RGB value in the form "### ### ###". If no *color* is specified, then Excel workbook defaults are used.

`dborder(`*direction* [`,` *style* [`,` *color*]]`)` sets the cell diagonal border direction, style, and color for a cell or range of cells.

direction may be `down`, `up`, or `both`. `down` draws a line from the upper-left corner of the cell to the lower-right corner of the cell or, for a range of cells, from the upper-left corner of *ul_cell* to the lower-right corner of *lr_cell*. `up` draws a line from the lower-left corner of the cell to the upper-right corner of the cell or, for a range of cells, from the lower-left corner of the area defined by *ul_cell*`:`*lr_cell* to the upper-right corner.

style is a keyword specifying the look of the border. The most common styles are `thin`, `medium`, `thick`, and `double`. The default is `thin`. For a complete list of border styles, see the *Appendix*. To remove an existing border, specify `none` as the *style*.

color may be one of the colors listed in the table of colors in the *Appendix* or may be a valid RGB value in the form "### ### ###". If no *color* is specified, then Excel workbook defaults are used.

Fill

`fpattern(`*pattern* [`,` *fgcolor* [`,` *bgcolor*]]`)` sets the fill pattern, foreground color, and background color for a cell or range of cells.

pattern is a keyword specifying the fill pattern. The most common fill patterns are `solid` for a solid color (determined by *fgcolor*), `gray25` for 25% gray scale, `gray50` for 50% gray scale, and `gray75` for 75% gray scale. A complete list of fill patterns is shown in the *Appendix*. To remove an existing fill pattern from the cell or cells, specify `none` as the *pattern*.

fgcolor specifies the foreground color. The default foreground color is `black`. *fgcolor* may be any of the colors listed in the table of colors in the *Appendix* or may be a valid RGB value in the form "### ### ###".

bgcolor specifies the background color. *bgcolor* may be any of the colors listed in the table of colors in the *Appendix* or may be a valid RGB value in the form "### ### ###". If no *bgcolor* is specified, then Excel workbook defaults are used.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction

The `putexcel` command is a means of directly controlling the layout of an Excel file. As such, `putexcel` is designed to mimic the options and functionality of Excel, and many options of `putexcel` are simply pass-through arguments to Excel itself. In what follows, we provide documentation of how to use the `putexcel` command. However, for many options, such as the specification of valid numeric formats, font names, and font sizes, users are encouraged to also consult the help for their specific version of Excel.

`putexcel` may be used with Excel 1997/2003 (`.xls`) and Excel 2007/2010 and newer (`.xlsx`). `putexcel` looks at the file extension `.xls` or `.xlsx` to determine which Excel format to write. It is supported on Windows, Mac, and Linux.

`putexcel` also has an advanced syntax that allows you to write multiple types of output to different cells or cell ranges at a time; see [P] **putexcel advanced**.

▷ Example 1: Setting the workbook and sheet

Before we can write to an Excel workbook using `putexcel`, we need to tell Stata what the destination is. We do this using the `putexcel set` command. For the next several examples, we will use an Excel file named `myresults.xlsx` and a sheet named `Descriptive`.

```
. putexcel set myresults.xlsx, sheet(Descriptive)
```

If we had not specified the sheet name, `putexcel set` would have defaulted to using the first sheet in the workbook.

◁

Writing expressions and formatting cells

Although there are many uses for `putexcel`, one of the primary reasons to use the command is to keep track of the results of analyses in a single location for later use. The next several examples show how to export results to a single location for easy sharing and to create formatted tables that can be placed in a paper or poster.

Suppose we are analyzing data about the number of times young adults visited a news website. These data are contained in `website.dta` and described in detail in example 1 of [R] **ivpoisson**.

```
. use http://www.stata-press.com/data/r15/website
(Visits to website)
. describe
Contains data from http://www.stata-press.com/data/r15/website.dta
  obs:           500                          Visits to website
 vars:             6                          11 Feb 2016 14:45
 size:        13,500
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
visits          byte    %8.0g                 Visits to website
female          byte    %8.0g                 Female
ad              byte    %8.0g                 Advertisements
time            double  %10.0g                Time on internet (hrs.)
phone           double  %10.0g                Time on phone (hrs.)
frfam           double  %10.0g                Time with friends and out of town
                                                family (hrs.)
-------------------------------------------------------------------------------
Sorted by:
```

We want to create a formatted table of summary statistics for men and women.

Example 2: Write expression to cell

We start by writing column headers to the first row of our worksheet.

```
. putexcel A1 = "Variable"
file myresults.xlsx saved
. putexcel B1 = "Men"
file myresults.xlsx saved
. putexcel C1 = "Women"
file myresults.xlsx saved
```

Each of these commands opens the Excel workbook, writes the text to the specified cell, and then closes the workbook. We also use expressions to write out specific results (see example 4).

Example 3: Format a range of cells

We may also want to make the column headings bold and add a border underneath. We can format a range of cells by specifying a cell range and the appropriate formatting options.

```
. putexcel A1:C1, bold border(bottom)
file myresults.xlsx saved
```

We also could have specified the `bold` and `border()` options each time we exported the column heading in example 2.

```
. putexcel A1 = "Variable", bold border(bottom)
file myresults.xlsx saved
. putexcel B1 = "Men", bold border(bottom)
file myresults.xlsx saved
. putexcel C1 = "Women", bold border(bottom)
file myresults.xlsx saved
```

Whether you apply formatting at the time you write to Excel or for a range will likely depend on the number of options you have and the number of cells affected by common formatting options.

Exporting summary statistics to Excel

We can use putexcel to write summary statistics to an Excel worksheet. The available summary statistics are determined by returned results. The returned results are documented in the *Stored results* section of the command's manual entry or help. You can also see what is returned by typing return list or ereturn list.

▷ Example 4: Export selected statistics

To export a specific statistic, we use the expression output type. For example, we might want a table that has the number of observations for men and women followed by means for each variable, which we show in example 5.

We can obtain counts many ways in Stata. Here we use the summarize command and restrict the command to males (female==0). We use visits, but any continuous variable without missing values would have worked just as well. r(N) stores the number of observations.

```
. summarize visits if female==0
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      visits |        257    5.105058    3.893185          1         27

. putexcel B2 = `r(N)'
file myresults.xlsx saved
```

A more direct way to obtain the number of observations is with the count command.

```
. count if female==1
  243

. putexcel C2 = `r(N)'
file myresults.xlsx saved
```

Notice that we typed `r(N)' instead of "r(N)". The `' tell Stata to fill in the numeric value associated with r(N) instead of exporting the text, known as macro substitution; see [P] **macro**. If we wanted to treat the contents of r(N), or any other return value, like a string, we could have typed "`r(N)'".

◁

▷ Example 5: Export frequency tables

You can use putexcel in Stata to create tables in Excel by using the matrix() output type. Suppose we want create a table of means for each variable for each value of female. We can use tabstat with the save option and then check the return values with return list to determine what values to output.

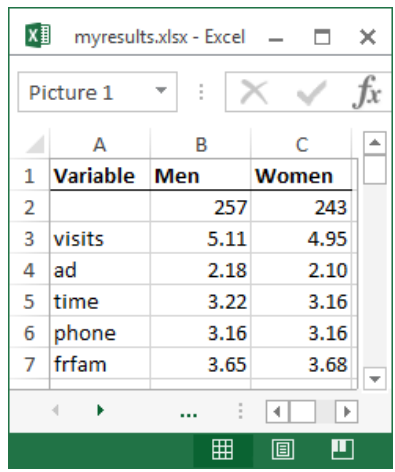
```
. tabstat visits ad time phone frfam, by(female) save
Summary statistics: mean
  by categories of: female (Female)
  female |    visits        ad      time     phone     frfam
---------+--------------------------------------------------
       0 |  5.105058  2.175097  3.222412  3.164086   3.65428
       1 |  4.946502  2.098765  3.161276  3.155391  3.676708
---------+--------------------------------------------------
   Total |     5.028     2.138    3.1927   3.15986   3.66518
------------------------------------------------------------
```

```
. return list
macros:
              r(name2) : "1"
              r(name1) : "0"
matrices:
              r(Stat2) :  1 x 5
              r(Stat1) :  1 x 5
          r(StatTotal) :  1 x 5
```

First, we transpose the row vectors `r(Stat1)` and `r(Stat2)`, which contain the means, so that the values are written in a column under each heading. We want the variable names to be included, so for males, which is the first matrix we output, we include the `rownames` option. Because we want the values in B3 and the row names of the matrix in A3, we specify A3. We use `nformat(number_d2)` to format the means with two decimal places. For females, we do not need to specify the names again, so we just specify the cell where we want the data to be written.

```
. matrix male = r(Stat1)'
. matrix female = r(Stat2)'
. putexcel A3 = matrix(male), rownames nformat(number_d2)
file myresults.xlsx saved
. putexcel C3 = matrix(female), nformat(number_d2)
file myresults.xlsx saved
```

The above commands give a final table of frequencies and means that looks like this:



	A	B	C
1	**Variable**	**Men**	**Women**
2		257	243
3	visits	5.11	4.95
4	ad	2.18	2.10
5	time	3.22	3.16
6	phone	3.16	3.16
7	frfam	3.65	3.68

Export estimation results

A similar approach to that used in example 5 can be used to export estimation results. Suppose we want to fit a linear regression model of `visits` as a function of `ad`, `female`, and `time` using `regress` and we want to export formatted results.

```
. regress visits ad female time
      Source |       SS           df       MS      Number of obs   =       500
-------------+----------------------------------   F(3, 496)       =    346.75
       Model |  5710.6792         3  1903.55973   Prob > F        =    0.0000
    Residual |  2722.9288       496   5.4897758   R-squared       =    0.6771
-------------+----------------------------------   Adj R-squared   =    0.6752
       Total |   8433.608       499   16.901018   Root MSE        =     2.343

------------------------------------------------------------------------------
      visits |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          ad |   .7996179   .0516591    15.48   0.000     .6981203    .9011156
      female |  -.0467997   .2096816    -0.22   0.823    -.4587734    .3651739
        time |     .82962   .0436601    19.00   0.000     .7438385    .9154014
       _cons |   .6924339   .2007914     3.45   0.001     .2979273    1.086941
------------------------------------------------------------------------------
```

▷ Example 6: Export point estimates and sample size

We start by using `putexcel set` again to create a new worksheet for our regression results.

```
. putexcel set myresults.xlsx, sheet(Estimation)
```

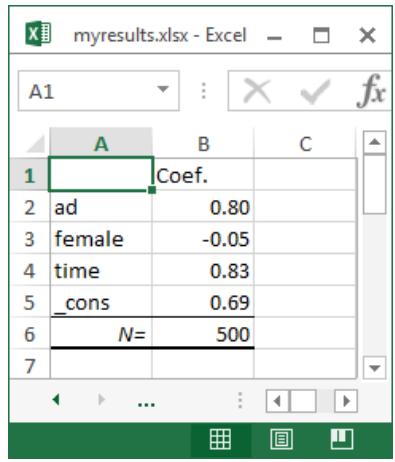
We want to give our coefficients the title "Coef." in the table we create, so we use an expression to write this to cell `B1`. We create a new matrix for our coefficients as the transpose of the values of e-class matrix `e(b)`. We use column A row 2 as the starting location for the matrix row labels, and we use column B row 2 as the starting location for the coefficients; to do this, we only need to specify the upper-left cell (A2) and the `rownames` option. To make our table more readable, we format the coefficient estimates with two decimal places by using `nformat()` and add a bottom border under the estimates.

```
. putexcel B1 = "Coef."
file myresults.xlsx saved
. matrix b = e(b)'
. putexcel A2 = matrix(b), rownames nformat(number_d2)
file myresults.xlsx saved
. putexcel A5:B5, border(bottom)
file myresults.xlsx saved
```

We then add a right-aligned and italicized "N=" in column A row 6 and the sample size from e-class scalar `e(N)` in column B row 6. We use a medium border under this row instead of the default thin border to indicate that the table is complete.

```
. putexcel A6 = "N=", italic right border(bottom, medium)
file myresults.xlsx saved
. putexcel B6 = matrix(e(N)), nformat(number_sep) border(bottom, medium)
file myresults.xlsx saved
```

The above commands create a table that looks like this:



If you are going to export more complex tables or many objects, you should use the advanced syntax of `putexcel`; see [P] **putexcel advanced**.

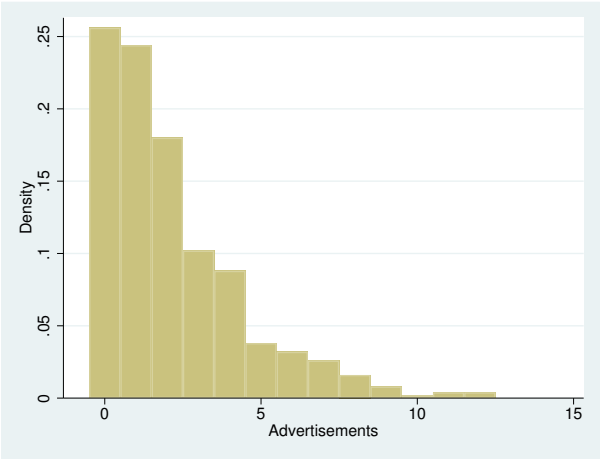
Export graphs and other images

You can export PNG, JPEG, and other image formats to Excel with `putexcel`. To export a Stata graph, you must first use `graph export` to convert the Stata graph to one of the supported image formats.

Example 7: Export Stata graph

We may want to add a histogram about the number of advertisements viewed to our set of descriptive results. We can use the `histogram` command and then `graph export` to create a PNG file of our histogram.

```
. histogram ad, discrete
(start=0, width=1)
. graph export ads.png
(file ads.png written in PNG format)
```



Next, we use `putexcel set` to switch back to the Excel sheet named `Descriptive`, and then we output the graph to Excel with the `picture()` output type.

```
. putexcel set myresults.xlsx, sheet(Descriptive)
. putexcel E2 = picture(ads.png)
file myresults.xlsx saved
```

This adds the graph with the upper-left corner of the graph aligned in the upper-left corner of cell E2. Graphs are not resized by `putexcel`, but you can change the size with the `graph export` command; see [G-2] **graph export**.

◁

❑ Technical note

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

❑

Appendix

Codes for numeric formats

Code	Example
number	1000
number_d2	1000.00
number_sep	100,000
number_sep_d2	100,000.00
number_sep_negbra	(1,000)
number_sep_negbrared	(1,000)
number_d2_sep_negbra	(1,000.00)
number_d2_sep_negbrared	(1,000.00)
currency_negbra	($4000)
currency_negbrared	($4000)
currency_d2_negbra	($4000.00)
currency_d2_negbrared	($4000.00)
account	5,000
accountcur	$ 5,000
account_d2	5,000.00
account_d2_cur	$ 5,000.00
percent	75%
percent_d2	75.00%
scientific_d2	10.00E+1
fraction_onedig	10 1/2
fraction_twodig	10 23/95
date	3/18/2007
date_d_mon_yy	18-Mar-07
date_d_mon	18-Mar
date_mon_yy	Mar-07
time_hmm_AM	8:30 AM
time_HMMSS_AM	8:30:00 AM
time_HMM	8:30
time_HMMSS	8:30:00
time_MMSS	30:55
time_HOMMSS	20:30:55
time_MMSS0	30:55.0
date_time	3/18/2007 8:30
text	this is text

Colors

color	
aliceblue	deeppink
antiquewhite	deepskyblue
aqua	dimgray
aquamarine	dodgerblue
azure	firebrick
beige	floralwhite
bisque	forestgreen
black	fuchsia
blanchedalmond	gainsboro
blue	ghostwhite
blueviolet	gold
brown	goldenrod
burlywood	gray
cadetblue	green
chartreuse	greenyellow
chocolate	honeydew
coral	hotpink
cornflowerblue	indianred
cornsilk	indigo
crimson	ivory
cyan	khaki
darkblue	lavender
darkcyan	lavenderblush
darkgoldenrod	lawngreen
darkgray	lemonchiffon
darkgreen	lightblue
darkkhaki	lightcoral
darkmagenta	lightcyan
darkolivegreen	lightgoldenrodyellow
darkorange	lightgray
darkorchid	lightgreen
darkred	lightpink
darksalmon	lightsalmon
darkseagreen	lightseagreen
darkslateblue	lightskyblue
darkslategray	lightslategray
darkturquoise	lightsteelblue
darkviolet	lightyellow

color, continued

lime	peru
limegreen	pink
linen	plum
magenta	powerblue
maroon	purple
mediumaquamarine	red
mediumblue	rosybrown
mediumorchid	royalblue
mediumpurple	saddlebrown
mediumseagreen	salmon
mediumslateblue	sandybrown
mediumspringgreen	seagreen
mediumturquoise	seashell
mediumvioletred	sienna
midnightblue	silver
mintcream	skyblue
mistyrose	slateblue
moccasin	snow
navajowhite	springgreen
navy	steelblue
oldlace	tan
olive	teal
olivedrab	thistle
orange	tomato
orangered	turquoise
orchid	violet
palegoldenrod	wheat
palegreen	white
paleturquoise	whitesmoke
palevioletred	yellow
papayawhip	yellowgreen
peachpuff	

Border styles

style
none
thin
medium
dashed
dotted
thick
double
hair
medium_dashed
dash_dot
medium_dash_dot
dash_dot_dot
medium_dash_dot_dot
slant_dash_dot

Background patterns

pattern
none
solid
gray50
gray75
gray25
horstripe
verstripe
diagstripe
revdiagstripe
diagcrosshatch
thinhorstripe
thinverstripe
thindiagstripe
thinrevdiagstripe
thinhorcrosshatch
thindiagcrosshatch
thickdiagcrosshatch
gray12p5
gray6p25

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/.

——. 2014. Retaining an Excel cell's format when using putexcel. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2014/02/04/retaining-an-excel-cells-format-when-using-putexcel/.

Gallup, J. L. 2012. A new system for formatting estimation tables. *Stata Journal* 12: 3–28.

Huber, C. 2017a. Creating Excel tables with putexcel, part 1: Introduction and formatting. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2017/01/10/creating-excel-tables-with-putexcel-part-1-introduction-and-formatting/.

——. 2017b. Creating Excel tables with putexcel, part 2: Macro, picture, matrix, and formula expressions. *The Stata Blog: Not Elsewhere Classified.* http://blog.stata.com/2017/01/24/creating-excel-tables-with-putexcel-part-2-macro-picture-matrix-and-formula-expressions/.

Quintó, L. 2012. HTML output in Stata. *Stata Journal* 12: 702–717.

Also see

[P] **putexcel advanced** — Export results to an Excel file using advanced syntax

[P] **putdocx** — Generate Office Open XML (.docx) file

[P] **putpdf** — Create a PDF file

[D] **import excel** — Import and export Excel files

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

[M-5] **Pdf*( )** — Create a PDF file

[M-5] **xl( )** — Excel file I/O class