The Stata Journal publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go “beyond the Stata manual” in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the Stata Journal, including information for authors, see the web page

http://www.stata-journal.com

The Stata Journal is indexed and abstracted in the following:

- Science Citation Index Expanded (also known as SciSearch®)
- CompuMath Citation Index®

Copyright Statement: The Stata Journal and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the Stata Journal.

The articles appearing in the Stata Journal may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the Stata Journal.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the Stata Journal, in whole or in part, on publicly accessible web sites, file servers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the Stata Journal or the supporting files understand that such use is made without warranty of any kind, by either the Stata Journal, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the Stata Journal is to promote free communication among Stata users.

The Stata Journal, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata and Mata are registered trademarks of StataCorp LP.
Creating print-ready tables in Stata

Michael Lokshin  
The World Bank  
Washington, DC  
mlokshin@worldbank.org

Zurab Sajaia  
The World Bank  
Washington, DC  
zsajaia@worldbank.org

Abstract. This article describes the new Stata command xml_tab, which outputs the results of estimation commands and Stata matrices directly into tables in XML format. The XML files can be opened with Microsoft Excel or OpenOffice Calc, or they can be linked with Microsoft Word files. By using XML, xml_tab allows Stata users to apply a rich set of formatting options to the elements of output tables.

Keywords: dm0037, xml_tab, estimates, regression, matrices, xml, Excel, Word

1 Introduction

Stata output results, while convenient for interactive work, are not well suited for presentation. Several user-written routines provide Stata users with the capability of saving Stata results in a more convenient format for presentation. Among these are outreg by John Gallup (1998, 1999, 2000); its modification, outreg2, by Roy Wada (2008); estout by Ben Jann (2005, 2007); tabout by Ian Watson (2007); est2tex by Marc Muendler (2005); mktab by Nicholas Winter (2005); and a set of programs developed by Roger Newson (2003). These routines are designed to compile one or more sets of Stata estimates or other results into tables that could be inserted into word processors or exported into \TeX, \LaTeX, and HTML formats. Gini and Pasquini (2006) describe the interaction between Stata and markup languages, and they suggest an approach to automatically generate and update reports, presentations, and web sites.

This article describes the new Stata command xml_tab, which outputs the results of estimation commands and Stata matrices directly into tables in XML format. The XML files can be opened with Microsoft Excel or OpenOffice Calc. By using XML, xml_tab allows Stata users to apply a rich set of formatting options to the elements of output tables.

2 Syntax and options

2.1 Syntax

\begin{verbatim}
xml_tab [ namelist ] [ , options ]
\end{verbatim}

where namelist comprises one or more specifications, separated by spaces. A specification can be the name of a stored estimation or a matrix name. xml_tab will output the estimation coefficients and one of the three statistics (standard errors, t ratio, or p-values).
For estimation results, `xml_tab` has enough information to calculate significance levels itself, but if a matrix is to be outputted, `xml_tab` looks also for `matname_STARS`.

The stored estimation also could be specified in an extended form with parameters:

```
xml_tab [ estname1(stat11 stat12, eform_option) [ estname2(stat21 stat22, 
  eform_option) [... ]]] [ , options ]
```

where `estname1` and `estname2` are names of stored estimations, and `stat11`, `stat12`, `stat21`, and `stat22` are matrices stored in `e()`. `eform_option` is one of the following: `eform`, `hr`, `irr`, or `rrr`, which will display exponentiated coefficients, hazard ratios, incidence-rate ratios, odds ratios, or relative-risk ratios, respectively. For more detail, see [SVY] `eform_option`.

### 2.2 Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output</strong></td>
<td></td>
</tr>
<tr>
<td><code>save(filename)</code></td>
<td>name and path for the output XML file/workbook</td>
</tr>
<tr>
<td><code>replace</code></td>
<td>overwrite existing <code>filename</code></td>
</tr>
<tr>
<td><code>append</code></td>
<td>add a new sheet to the existing workbook, or create a new workbook</td>
</tr>
<tr>
<td><code>sheet(name [, sh_opts])</code></td>
<td>name a worksheet where the table is outputted</td>
</tr>
<tr>
<td></td>
<td>where <code>sh_opts</code> are</td>
</tr>
<tr>
<td></td>
<td><code>color[#]</code></td>
</tr>
<tr>
<td></td>
<td><code>nogridlines</code></td>
</tr>
<tr>
<td><code>savemat(name [, sm_opts])</code></td>
<td>save estimates to a matrix</td>
</tr>
<tr>
<td></td>
<td>where <code>sm_opts</code> are</td>
</tr>
<tr>
<td></td>
<td><code>replace</code></td>
</tr>
<tr>
<td></td>
<td><code>exit</code></td>
</tr>
<tr>
<td><code>mv(numspec)</code></td>
<td>change missing values to string or numeric values; see <code>help xml_tab</code> and <code>mvencode</code> for more detail</td>
</tr>
<tr>
<td><strong>Statistics</strong></td>
<td></td>
</tr>
<tr>
<td><code>sd</code></td>
<td>show estimated coefficients and standard deviations; the default</td>
</tr>
<tr>
<td><code>tstat</code></td>
<td>show estimated coefficients and <code>t</code> statistics</td>
</tr>
</tbody>
</table>
Creating print-ready tables

*value* show estimated coefficients and *p*-values

*stats(scalarlist)* report *scalarlist* statistics at the bottom of the table

*stars(starspec)* control significance levels and symbols

where *starspec* is

\[
(symbol_1) \#_1 [(symbol_2) \#_2 [(symbol_3) \#_3]]
\]

or

\[
\#_1 [\#_2 [\#_3]]
\]


*nogradjust* report unadjusted *t* statistics

*eform* display exponentiated coefficients

*hr* display hazard ratios

*irr* display incidence-rate ratios

*or* display odds ratios

*rrr* display relative-risk ratios

Table layout

*bellow* show standard deviations (*t* statistics or *p*-values) under the estimates

*right* show standard deviations (*t* statistics or *p*-values) next to the estimates; the default

*no brackets* suppress brackets around standard deviations (*t* statistics or *p*-values) if *bellow* is specified

*long* output the table in long form

*wide* output the table in wide form; the default

*keep(keeplist)* report *keeplist* rows

*drop(droplist)* drop *droplist* rows from the table

*equations(matchlist)* match the equations of the models according to *matchlist*

*keep(), drop(), and equations()* work like the options documented in [R] *estimates table*

Table formatting

*format(flist)* define the format of the output table

where *flist* is

\[
((F_{11} F_{12} \ldots F_{1n}) (F_{21} F_{22} \ldots F_{2n}) \ldots (F_{m1} F_{m2} \ldots F_{mn}))
\]

and *F* *ij*'s are strings of five alphanumerical symbols defining, correspondingly, cell type, vertical alignment, horizontal alignment, font style, and number of digits after the decimal point. For more detail, see help *xml_tab*. 

lines(llist) underline rows

where llist is
row linestyle [row linestyle [...]]
linestyle can be one of 14 line styles, defined in Excel

nolabel display variable names instead of labels
constant(string) specify label for the constant
rblanks(rblist) add rows to the table

where rblist is
[varname [text] [format]], [varname [text] [format]], [...]

cblanks(equations | numlist) add blank columns to the table
cwidth(width) modify column widths; see help xml_tab for more information
tblanks(#) add # blank rows at the top of the table	
title(string) title the table
rnames(strlist) define custom row names
cnames(strlist) define custom column names
ceq(strlist) define custom column equation names (supertitles)
notes(string) add notes to the end of the table
font(font) specify font for a worksheet

where font is
fontname [fontsize]

style(stylename) apply predefined formatting styles to the table

System
exceelpath(filename) specify the location of the Excel executable
calcpath(filename) specify the location of the Calc executable
noisily display the complete list of options applied to the table
updateopts update the options file

See help xml_tab for further information about any of these options.

3 Basic usage

xml_tab creates formatted tables of coefficients, standard errors, t statistics and p-values, summary statistics, etc., after any Stata estimation command that saves its results in e(). The results of multiple estimations can be combined into one table if these results are stored by estimates store. xml_tab can also generate formatted
Creating print-ready tables

tables from Stata matrices and combine several matrices/estimations into one table. The program can output several tables into different sheets of an XML workbook; this is useful for storing the results of multiple estimations into one file.

To illustrate the rich functionality of xml_tab, we start from a simple example. Suppose we want to output the results of a Stata estimation command (in our case, regress) into a formatted table:

```
. sysuse auto
   (1978 Automobile Data)
. regress price mpg headroom
   (output omitted)
. xml_tab, replace
   note: results saved to c:\temp\stata_out.xml
   click here to open with Excel
```

xml_tab without arguments saves the results of the last estimation into the default file (stata_out.xml) located in the current Stata working directory (see pwd in [D] cd). The note after the xml_tab command indicates that Excel is installed on the computer. The program automatically searches the system to locate Excel or Calc. If either of these programs is installed, clicking on the highlighted link opens the table in Excel or Calc.\footnote{1. All tables in this paper are shown in Excel.}

![Figure 1. Regression results saved in default format](image)

The table in figure 1 is saved in the default format. The results of the regression are represented by two columns of regression coefficients and standard errors. Significance of the coefficients is shown by the stars: by default, xml_tab displays *** for $p < 0.01$, ** for $p < 0.05$, and * for $p < 0.1$. The significance thresholds and symbols can be
customized (see \texttt{stars()} in section 2.2). By default, \texttt{xml\_tab} shows variable labels in the first column of the table (see \texttt{nolabel} in section 2.2). To control the label on the constant (default \texttt{cons}), use \texttt{constant()}.

Suppose we want to combine the results of two regressions into one table. For that, the estimated results should be saved by using the \texttt{estimates store} command (see \texttt{[R] estimates}):

\begin{verbatim}
. sysuse auto
(1978 Automotive Data)
. regress price mpg headroom
(output omitted)
. estimates store r1, title(Regression 1)
. regress price mpg headroom weight length
(output omitted)
. estimates store r2, title(Regression 2)
. xml\_tab *, replace save(c:\temp\example1.xml)
\end{verbatim}

We store results from the first regression in Stata memory under the name \texttt{r1} and the title “Regression 1”. We then store the results from the second regression under the name \texttt{r2} and the title “Regression 2”. \texttt{*} instructs \texttt{xml\_tab} to combine all estimated results currently stored in memory into one table. Option \texttt{save()} instructs \texttt{xml\_tab} to save the table into the file \texttt{c:\temp\example1.xml}. Alternatively, we can specify the names of the stored estimated results directly:

\begin{verbatim}
. xml\_tab r1 r2, replace save(c:\temp\example1.xml)
\end{verbatim}

By clicking on the highlighted link, the generated table is opened in Excel; see figure 2.

\begin{verbatim}
(Continued on next page)
\end{verbatim}
Creating print-ready tables

Figure 2. Results of two regressions saved in c:\temp\example1.xml

We can instruct `xml_tab` to output the results of the above estimations into different sheets in the XML workbook by using the `sheet()` option:

```
.xml_tab r1, replace sheet(Regression 1) save(c:\temp\example1.xml)
(output omitted)
.xml_tab r2, append sheet(Regression 2) save(c:\temp\example1.xml)
(output omitted)
.xml_tab r1 r2, append save(c:\temp\example1.xml)
> sheet(Two regressions) font("Arial Narrow" 12)
note: results saved to c:\temp\example1.xml
```

The first statement of the code replaces file c:\temp\example1.xml with an empty workbook and places the results of the first regression into the sheet “Regression 1”. The next statement uses `append` to append a new sheet, “Regression 2”, to this workbook and outputs the coefficients and the standard errors of the second regression to it. The last statement of the code saves the table, which combines the results of the two regressions into the sheet “Two regressions”. The option `font()` specifies the font “Arial Narrow”, size 12, to be used for the table saved in the sheet “Two regressions”.

`xml_tab` can be invoked in a loop to output the results of multiple estimations into the sheets of a workbook.

```
. foreach var of varlist mpg rep78 headroom trunk weight gear_ratio {
    .    regress price `var'
    .    estimates store r_`var'
    .    xml_tab r_`var', append sheet(`var') save(c:\temp\example2.xml)
    . }
```
3.1 Parameter statistics and table layout

Several options of xml_tab specify the parameter statistics to be tabulated and control the table layout. Options sd, tstat, and pvalue control which statistic will be outputted together with the estimated coefficients. If option sd is specified, the standard errors of the estimated coefficients are outputted (this is the default). Specifying tstat or pvalue produces a table of coefficients and corresponding t statistics or p-values. The position of the parameter statistics relative to the parameter estimate is determined by the options right and below. By default, xml_tab places the parameter statistics to the right of the estimated coefficient. If option below is specified, the parameter statistics are outputted in parentheses (unless option nobrackets is used) below the estimated parameter.

Option stats() specifies one or more scalar statistics to be displayed at the end of the table. The statistics specified in stats() could be any statistic saved in e() scalars after estimation routines ([i] 26 Overview of Stata estimation commands). For example, stats(N) displays the number of observations used for the estimation; stats(r2_a) shows the adjusted $R^2$ for a regression; stats(ll) outputs the value of the log likelihood, etc. When several estimations are combined in one table, the specified statistics will be displayed for each estimation. The user-written command estadd can be used to add additional statistics to the estimation results stored in Stata memory (Jann 2005, 2007). These additional statistics could be outputted in the tables by using the standard syntax of the stats() option.

Options keep(), drop(), and equation() select the subset of parameters (or equations in multiple-equation estimations) to be displayed in the table. The variables to be shown or excluded from the table can be referred to by a simple name or by the full name (equation name and variable name). See [R] estimates table for further details. Specifying variables by their full names allows users to exclude several variables from one of the equations in the multiple-equation estimation.

```
. xml_tab r1 r2, replace drop(weight) below pvalue stats(N r2)
> sheet(Two regressions)
    note: results saved to c:\temp\stata_out.xml
    click here to open with Excel
```

The above example generates the table in figure 3 by combining the results of estimation r1 and r2. The table shows the estimated coefficients and their corresponding p-values (option pvalue). The p-values are placed under the estimated coefficients (option below). The coefficient and p-value of weight are not displayed (option drop()). The number of observations and the $R^2$ are outputted at the bottom of the table (option stats()).

(Continued on next page)
Creating print-ready tables

Figure 3. Results of estimation with $p$-values

3.2 Table formatting

The tables generated by the simple syntax described in the previous section could be improved by adding xml_tab formatting options. xml_tab allows you to apply user-specified formats to individual cells, to columns, and to the table as a whole. The format of each element is a string of five alphanumerical symbols (see the description of the formatting symbols in help xml_tab). The most practical use of the format() option is to define formats for columns in the table. In tables where coefficients and statistics are outputted side-by-side, it is usually sufficient to specify just three formatting strings. The example below illustrates this feature of xml_tab; see figure 4.

```
. xml_tab r1 r2, append save(c:\temp\example1.xml)
> sheet(Two regressions)
> format(sclb0 ncrr3 ncci3)
note: results saved to c:\temp\example1.xml
```
Figure 4. Coefficient and statistics table with three formatting strings

The first string in the `format()` option describes the formatting of the column containing variable names—the variable names are outputted as vertically centered (C|c|2), left-justified (L|l|1) strings, in bold (B|b|1) font. The column of coefficients is vertically centered, right-justified (R|r|3), with three digits after the decimal point. The column of standard errors is vertically and horizontally centered, and outputted in italics (I|i|2) with three digits after the decimal point. This formatting is applied to all columns of coefficients and standard errors in the table. In other words, option `format(sclb0 ncrr3 ncci3)` is equivalent to `format(sclb0 ncrr3 ncci3 ncrr3 ncci3 ncrr3 ncci3 ...)`.

The next example demonstrates the extended formatting options of `xml_tab`; see figure 5.

```
  . xml_tab r1 r2, append format(sclb0 ncrr3 ncci3) stats(N r2)
    > sheet(Two regressions Extended Syntax, color(3) nogridlines)
    > cblocks(2) cwidth(0 125, 3 4) rblocks(mpg "Dimensions" scci0)
    > lines(_cons 2 LAST_ROW 13 COL_NAMES 2 EST_NAMES 2)
    > title("Example of extended formatting syntax")
  note: results saved to c:\temp\stata_out.xml
  click here to open with Excel
```

(Continued on next page)
Creating print-ready tables

Figure 5. Output using extended formatting options

Here, in addition to the formatting options discussed in the previous examples, `title()` titles the table; `lines()` draws lines under the top row (COL NAMES), the row containing estimate names (EST NAMES), the row containing the constant (_cons), and the last row of the table (LAST ROW); `rblanks()` inserts a row with the word “Dimensions” after the mpg variable; `cblanks()` inserts an empty column that separates the estimates of Regression 1 and Regression 2; `cwidth()` changes the widths of the first column and the newly added separator column; and `sheet(..., color() nogridlines)` colors the sheet tab and suppresses the gridlines. See `help xml_tab` for more detail on the formatting options.

All the formatting and layout options in `xml_tab` can also be provided via the `style()` option. The predefined styles are stored in a file named `xml_tab_options.txt`; user-defined styles can be added to this file. The `style()` option is useful if you want to produce multiple tables in a similar format.

4 Advanced applications

4.1 Multiple-equation models

`xml_tab` works both with single-equation and multiple-equation models. Examples of multiple-equation models in Stata include `ivreg, heckman, mlogit, probit, asmprobit`, and others. With the default `wide` option, `xml_tab` arranges the different equations of multiple-equation models into separate columns. Summary statistics for the multiple-equation estimations, specified by using the `stats()` option, are reported under the first equation. For example, if a dependent variable with three categories is fitted with `mlogit` using 10 exogenous variables, specifying the `wide` option would result in a 12 × 5
table. The first column of this table contains variable labels/names, the second and third
columns contain the estimated coefficients and standard errors for the first equation,
and the fourth and fifth columns contain the estimated coefficients and standard errors
for the second equation.

Alternatively, the different equations in the multiple-equation models can be outputted one after another in vertical order by using the \texttt{long} option.

Sometimes it might be convenient to suppress the output of one or more equations
in the multiple-equation models. This can be done by using an extended syntax of the
option \texttt{drop()}, \texttt{keep()}, or \texttt{equations()}. See \texttt{[R] estimates table} for details. For
example, when fitting Heckman-type models, one might want to suppress the output of
the coefficients in the selection equation. The code below illustrates this feature:

\begin{verbatim}
. heckman price mpg rep78, select(foreign = length weight mpg rep78) nolog
(output omitted)
. xml_tab, replace drop(foreign:)
\end{verbatim}

\texttt{click here to open with Excel}

### 4.2 Marginal effects

\texttt{xml} \texttt{tab} supports Stata commands for calculating marginal effects and elasticities: \texttt{mfx},
\texttt{dprobit}, \texttt{dlogit}, and others. To create tables with marginal effects, we need to instruct
\texttt{xml} \texttt{tab} which matrices stored in \texttt{e()} to use. The stored estimations can be specified
in an extended form with parameters:

\begin{verbatim}
\texttt{xml} \texttt{tab} [ \texttt{estname1} (\texttt{stat11} \texttt{stat12}, \texttt{eform\_option}) ] [ \texttt{estname2} (\texttt{stat21} \texttt{stat22},
\texttt{eform\_option}) ] ]
\end{verbatim}

where \texttt{estname1} and \texttt{estname2} are the names of stored estimation, and \texttt{stat11}, \texttt{stat12},
\texttt{stat21}, and \texttt{stat22} are the names of matrices stored in \texttt{e()}. By default, \texttt{xml} \texttt{tab} uses
matrices \texttt{e(b)} and \texttt{e(V)}, which are equivalent to the extended syntax \texttt{estname(b V)}.

Suppose we want to generate a table of marginal effects after \texttt{heckman} estimation:

\begin{verbatim}
. probit foreign mpg trunk length
(output omitted)
. mfx
(output omitted)
. estimates store prob_mfx
\end{verbatim}

To output the marginal effects from this estimation, we specify the names of the
matrices stored after the \texttt{mfx} command that contain the estimates of the marginal
effects and their standard errors:

\begin{verbatim}
. xml_tab prob_mfx(Xmfx_dydx Xmfx_se_dydx), replace
\end{verbatim}

Similarly, to output the marginal effects and the standard errors after \texttt{dprobit}, we
would specify \texttt{(dfdx se_dfdx)} statistics. One can check the names of the matrices with
the stored statistics using \texttt{ereturn list}. 

4.3 Appending several estimation results/matrices

Sometimes it might be convenient to stack the results of multiple estimations vertically and output several estimations into one sheet of the workbook. The option `savemat()` can be used for this purpose. When `savemat()` is specified, `xml_tab` stores estimation results into a matrix (appending new rows if that matrix already exists). After several estimations have been appended by using this option, `xml_tab` can be called to output the resulting table into an XML file:

```
. regress price mpg
    (output omitted)
. xml_tab, savemat(A, replace exit)
. regress price trunk
    (output omitted)
. xml_tab, savemat(A, exit)
```

`xml_tab` saves the results of regressing `price` on `mpg` to matrix `A` (replacing the existing matrix) without generating the XML table (option `exit`). The results of the regression of `price` on `trunk` are appended to matrix `A`. The combined table is outputted with the command (see figure 6):

```
. xml_tab A, replace rblanks(0 "First regression", 2 "Second regression")
```

![Figure 6. A combined table for first regression and second regression](image)

4.4 Exporting Stata matrices

`xml_tab` can output any matrix stored in Stata memory. Most of the options of `xml_tab` to control the layout and the formatting can be used for the matrix output. Custom tables can be created by generating matrices of results and outputting them with `xml_tab`. 
tabstatmat (Nichols 2007) is a useful, user-written command to save various summary statistics into matrices. The following example demonstrates how to create and output a simple table of means after the tabstat command:

```
. tabstat price mpg rep78 headroom trunk weight length, by(foreign) save
   (output omitted)
. tabstatmat A
   (output omitted)
. matrix TAB = A`
   (output omitted)
. xml_tab TAB, replace
   note: results saved to c:\temp\stata_out.xml
```

![Figure 7. Table of the means after using tabstat](image)

In the above code, tabstat generates a table of means for the list of variables, categorized by foreign. tabstatmat saves the results to matrix A (see figure 7). This matrix has three columns: “Domestic”, “Foreign”, and “Total”. The columns of matrix A contain the means for the listed variables. We save the transposed matrix A into another matrix, TAB. xml_tab outputs this matrix into the default XML file.

Custom significance levels for the elements of a matrix could be specified by constructing matrix matname_STARS. When outputting matrices, xml_tab checks if a matrix matname_STARS exists in memory. matname_STARS should have the same dimensions as the matrix to be outputted. The elements of matname_STARS should contain values 0, 1, 2, or 3, corresponding to levels of statistical significance. See help xml_tab for more details.

Use the rnames(), cnames(), and ceq() options to modify the names of rows and columns in the outputted matrices and estimation results.
4.5 Dynamically updated tables in Microsoft Word

Stata results saved in XML workbooks can be linked with Microsoft Word to create dynamically updated documents. Suppose we want to create a Word document containing multiple tables that must be frequently updated. To do that, we would

- Output tables of results with `xml_tab` into an XML workbook,
- Open the resulting workbook in Excel or Calc, and then
- Copy each table of the workbook by using `Menu > Paste special > Paste link`. Select `as HTML Format` from the list of possible formats.

When the new tables are generated, open Word and the XML files, right-click on a table in the Word document, and select `Update Link`. Tables in the Word document linked with the tables in the XML workbook will be updated. The changes in layout, formatting, fonts, etc., in the XML workbook will be carried over to the Word document.

5 References


Muendler, M. 2005. est2tex: Stata module to create LaTeX tables from estimation results. Statistical Software Components, Boston College Department of Economics. Downloadable from `http://fmwww.bc.edu/repec/bocode/e`.


About the authors

Michael Lokshin is a senior economist in the research department of the World Bank.
Zurab Sajaia is an economist in the research department of the World Bank.