Tutorial:

Automated table generation and reporting with Stata

Ben Jann, ETH Zurich, jannb@ethz.ch

November 2008

Required user packages:

- **mat2txt** *<install>*
- **estwrite** *<install>*
- **estout** *<install>*

Outline

-

Introduction I

Statistical software packages are good at analyzing data, but they are often weak when it comes to reporting.

- Output from statistical routines contains all sorts of details that are valuable to the researcher but are not so important for reporting.

  => you have to select relevant results

- Output from statistical routines sometimes contains results that are not well suited for interpretation or for presentation to a non-expert audience.

  => you have to transform results

Introduction II

- Output from statistical routines is often not well formatted for presentation.

  => you have to rearrange and reformat results

- Various software packages might be used for further processing of results and for reporting.

  => you have to transfer results to specific file formats

- You might need to re-use results for other reports or extract additional results at a later point in time.

  => you have to archive results

<next>   <overview>

Introduction III

TWO MAXIMS

1) Never Copy/Paste results by hand

You will almost surely make tons of mistakes!

2) Do everything only once

It is simply a waste of time to do things more than once.

Introduction IV

- These two goals can be reached by **automation**.

- Automation has its **price**:

  - initial investment of time and effort

  - reduced flexibility

- However, personally I find that automation almost always pays off.

- For example, although you are convinced that you do the tables in your research paper only once, you'll find yourself doing them over, and over, and over, ...

Introduction V

- Furthermore, automation increases **quality**:

    - no copy/paste errors

    - errors and possible improvements are often detected after everything is done; in a non-automated settings there are high barriers against correcting such errors or implementing the improvements

    - the lack of flexibility leads to standardization (which is usually positive, but can sometimes also hinder innovation)

    - automation makes research more replicable

- Moreover, good tools can lower the costs of automation dramatically.

Introduction VI

- Of course, there are also exceptions where automation might not be worth the effort.

- Examples:

  - slides for presentations that are only used once or twice

  - numbers in text body (trick: only cite approximate values)

Part 1: Low-level results processing

- How to access results from Stata routines

- Getting things out of Stata: The **file** command

- Wrappers

Accessing results in Stata I

- A prerequisite for automation is that the results from statistical routines can be accessed by the user.

- In Stata, most commands return their results in **r()** or **e()** (see _return_).

  - **r()** is used by "general" commands such as _summarize_
  - **e()** is used by "estimation" commands such as _regress_

- Returned are:

  - string scalars
  - numeric scalars
  - numeric matrices

  - For example, estimation commands return the number of observations in **e(N)**, the name of the command in **e(cmd)**, and the coefficients vector and the variance matrix in **e(b)** and **e(V)**.

Accessing results in Stata II

- Use <u>return list</u> or <u>ereturn list</u> to find out about available returns. Use <u>matrix list</u> to see the contents of a returned matrix.

```
sysuse auto, clear
summarize price
return list
```
<u>*<run>*</u>

```
regress price mpg weight
ereturn list
```
<u>*<run>*</u>

- Use <u>matrix list</u> to see the contents of a returned matrix.

```
matrix list e(b)
matrix list e(V)
```
<u>*<run>*</u>

Accessing results in Stata III

- You can use the **e()** and **r()** scalars and matrices more or less as you would use any other scalar or matrix, although it is often advisable to first copy the results into regular macros, scalars, or matrices (see **macro**, **scalar**, and **matrix**).

- Examples:

```
display "BIC = " -2 * e(ll) + ln(e(N)) * (e(df_m)+1)
<run>


local BIC = -2 * e(ll) + ln(e(N)) * (e(df_m)+1)
display `BIC'
<run>
```

<next>

Accessing results in Stata IV

- Example with matrices:

```
matrix X = 1                         /*the constant*/
foreach v of varlist weight mpg  {  /*reverse order*/
    summarize `v'
    matrix X = r(mean), X
}
matrix b = e(b)'
matrix Y = X * b
display Y[1,1]
```

```
adjust mpg weight   /*same result*/
```

- Note that coefficients and standard errors can also be accessed as **_b[]** and **_se[]**:

```
display "t value = "_b[mpg] / _se[mpg]
```

Getting things out of Stata: The file command I

- The _file_ command is used in Stata to write to (or read from) a file on disk.

- Use _file_ to produce custom output files.

- _file_ is a low level command. It just writes plain text, line by line. You have to do all formatting yourself.

- _file_ may appear a bit clumsy: You have to

    file open _handle_ **using** _filename_, **write** /*initialize*/

    file write _handle_ ...   /*write*/
    ...

    file close _handle_   /*done*/

- However, _file_ can produce any desired output.

Getting things out of Stata: The file command II

- Example: Write a tab delimited file containing descriptive statistics

```
sysuse auto, clear
file open fh using example.txt, write replace
file write fh "variable" _tab "mean" _tab "sd"
foreach v of varlist price-foreign  {
    summarize `v'
    file write fh _n "`v'" _tab (r(mean)) _tab (r(sd))
}
file close fh
type example.txt
```

_____

Getting things out of Stata: The file command III

- This can easily be turned into a program:

```
capture program drop mysumtab
program define mysumtab
    syntax varlist using [, replace append ]
    tempname fh
    file open `fh' `using', write `replace' `append'
    file write `fh' "variable" _tab "mean" _tab "sd"
    foreach v of local varlist {
        quietly summarize `v'
        file write `fh' _n "`v'" _tab (r(mean)) _tab (r(sd))
    }
    file close `fh'
end

sysuse nlsw88, clear
mysumtab * using example.txt, replace
type example.txt
```

_____

Getting things out of Stata: The file command IV

- Or let's do HTML:

```
capture program drop mysumhtm
program define mysumhtm
    syntax varlist using [, replace append ]
    tempname fh
    file open `fh' `using', write `replace' `append'
    file write `fh' "<html><body><table>"
    file write `fh' _n "<thead><th>variable</th>" ///
        "<th>mean</th><th>sd</th></thead>"
    foreach v of local varlist {
        quietly summarize `v'
        file write `fh' _n "<tr><td>`v'</td><td>" ///
            (r(mean)) "</td><td>" (r(sd)) "</td></tr>"
    }
    file write `fh' _n  "</table></body></html>"
    file close `fh'
end
mysumhtm * using example.html, replace
type example.html
```
<run>   <show>

<next>   <overview>

_____

Wrappers

- Of course you do not have to write a new program for
  everything.

- Check the SSC Archive to find out whether anything already
  exists that serves your needs (see **findit** and **ssc**).

- For example, **mat2txt** can be used to write a matrix to a
  tab-delimited file:

  sysuse auto, clear
  regress price weight mpg for
  mat V = e(V)
  mat2txt, matrix(V) saving(example.txt) replace ///
      title(This is a variance matrix)
  <run>   *<show>*

_____

Part 2: Handling model estimation results

- [Results from "estimation" commands are special](#)

- [Archiving models](#)

- [Tabulating estimation results](#)

Results from "estimation" commands are special

- Results from e-class commands are special because they share a common structure:

  - a coefficients vector: **e(b)**

  - and a variance matrix: **e(V)**

- There is, to some degree, a consensus/common practice of how to design tables containing model estimation results.

- Many models are estimated, usually, and estimation may be computationally intensive so that archiving the results is an issue.

Archiving models I

- A good approach is to keep model estimation and reporting two separate processes.

- This requires that model estimates are stored for later tabulation.

- Estimating a new model replaces the **e()**-returns of a previous model. However, the results can be stored in memory under a specific name using **estimates store** or the **eststo** user command.

- In Stata 10, it is also possible to save the results of a model on disk using **estimates save**.

- A problem with **estimates save** is that it can only store one model at the time (i.e. each model is saved in a separate file). However, the **estwrite** user command overcomes this problem.

Archiving models II

- Example:

```
estimates clear
sysuse auto, clear
bysort foreign: eststo: regress price weight mpg
eststo dir
estwrite * using mymodels, replace
<run>

dir mymodels*
<run>
```

- Two weeks later:

```
estimates clear
sysuse auto, clear
estread mymodels
<run>
```

<next>   <overview>

Tabulating estimation results I

- Various commands exists to compile and export tables of model estimates. **estout** is one of them. Others are **outreg** (John Luke Gallup), **outreg2** (Roy Wada), **xml_tab** (Lokshin & Sajaia), **outtex** (Antoine Terracol), **est2tex** (Marc Muendler), **mktab** (Nicholas Winter), **parmest** (Roger Newson), of which all have their pros and cons.

- The **estout** package contains four commands:

  esttab: User-friendly command to produce publication-style regression tables for screen display or in various export formats such as such as CSV, RTF, HTML, or LaTeX.

  estout: Generic program to compile regression tables (the engine behind **esttab**).

  estadd: Program to add extra results (such as e.g., beta coefficients) to **e()** so that they can be tabulated.

  ststo: Improved version of **estimates store**.

Tabulating estimation results II

- **esttab** and **estout** are very flexible and can produce all sorts of regression tables.

- I will only show a few basic examples here. Many more examples can be found at the following website:

    http://repec.org/bocode/e/estout

- The basic procedure is to store a number of models and then apply **esttab** (or **estout**) to tabulate them:

```
eststo clear
sysuse auto, clear
eststo: regress price weight mpg
eststo: regress price weight mpg foreign
esttab
```
<run>

Tabulating estimation results III

- **esttab** can either display the table in Stata's results window or export it to a file on disk using one of several formats, such as

  - **fixed**: fixed-format ASCII

  - **tab**: tab-delimited ASCII

  - **csv**: CSV (Comma Separated Value format) for use with MS Excel

  - **rtf**: Rich Text Format for use with word processors

  - **tex**: LaTeX format

  - ...

Tabulating estimation results IV

- Use with MS Excel: **csv** or **scsv**

  esttab using example.csv, replace

  esttab using example.csv, replace scsv

  (The **scsv** format uses a semi-colon as delimiter which is appropriate for certain language versions of Excel.)

- Use the **plain** option if you intend to do additional computations in MS Excel:

  esttab using example.csv, replace wide plain

  (No XML support. Sorry.)

Tabulating estimation results V

- Use with MS Word: **rtf**

  esttab using example.rtf, replace
  <run>   *<show>*

- Appending is possible. Furthermore, use **varwidth(#)** and **modelwidth(#)** to change column widths:

  esttab using example.rtf, append wide label modelwidth(8)
  <run>   *<show>*

- Including RTF literals:

  esttab using example.rtf, replace   ///
      title({\b Table 1: This is a bold title})
  <run>   *<show>*

  esttab using example.rtf, replace   ///
      cells(b(fmt(a3)) t(par(\i( ))))
  <run>   *<show>*

Tabulating estimation results VI

- Use with LaTeX: **tex**

  esttab using example.tex, replace   ///
      label nostar page   ///
      title(Regression table\label{tab1})
  <run>

  For a preview, click

      <texify>

  to compile the file (LaTeX required) and then

      <show PDF>

  to view the result. If *<texify>* does not work, click

      <show PDF from web>

  to download and view a precompiled version.

Tabulating estimation results VII

- Improved LaTeX table using the *booktabs* package:

  ```
  esttab using example.tex, replace        ///
      label nostar page booktabs           ///
      title(Regression table\label{tab1})
  ```

- Improved LaTeX table using the *dcolumn* package:

  ```
  esttab using example.tex, replace        ///
      label booktabs                       ///
      page(dcolumn)                        ///
      alignment(D{.}{.}{-1})               ///
      title(Regression table\label{tab1})
  ```

Tabulating estimation results VIII

- Advanced LaTeX example

```
eststo clear

eststo: reg weight mpg
eststo: reg weight mpg foreign

eststo: reg price weight mpg
eststo: reg price weight mpg foreign

esttab using example.tex, replace                        ///
    label booktabs nonumber                              ///
    page(dcolumn)                                        ///
    alignment(D{.}{.}{-1})                               ///
    mgroups(A B, pattern(1 0 1 0)                        ///
        prefix(\multicolumn{@span}{c}{) suffix(})        ///
        span erepeat(\cmidrule(lr){@span}))
```

<run>
<texify>   <show PDF>   <show PDF from web>

<next>   <overview>

_____

Tabulating estimation results IX

- **esttab** can be used to tabulate any results, not just regression models, as long as they are posted in **e()** in an appropriate way.

- Example: descriptives table

  ```
  eststo clear
  sysuse auto, clear
  generate y = uniform()
  quietly regress y price weight mpg foreign, noconstant
  estadd summ
  esttab, cells("mean sd min max") nogap nomtitle nonumber
  ```
  <run>

_____

Part 3: Automatic reporting

- [Automation](#)

- [Example with LaTeX](#)

- [Example with MS Word and Excel](#)

Automation

- Automatic reporting means that results and information on formatting should be separated.

- It has to be possible to replace the data without losing the formatting.

- The usual approach is to maintain a hand-edited master file that structures the document and sets the formatting etc. and then dynamically link the files containing results into this file.

Example with LaTeX I

- Step 1: Set up a master file

```
file open fh using example.tex, replace write
file write fh        "\documentclass{article}"
file write fh _n     "\begin{document}"
file write fh _n     "\section{My Tables}"
file write fh _n(2) "\input{_example1.tex}"
file write fh _n(2) "\input{_example2.tex}"
file write fh _n(2) "\end{document}"
file close fh
type example.tex
```

(Of course you would, usually, set up a master file in a text editor, not in Stata.)

Example with LaTeX II

- Step 2: Generate table files

  ```
  eststo clear
  sysuse auto, clear

  eststo: reg weight mpg
  eststo: reg weight mpg foreign
  esttab using _example1.tex, replace title(Weight)

  eststo clear
  eststo: reg price weight mpg
  eststo: reg price weight mpg foreign
  esttab using _example2.tex, replace title(Price)
  ```
  <run>

- Step 3: Compile the document

  <texify>   *<show PDF>*   *<show PDF from web>*

_____

Example with LaTeX III

- You can now easily replace the tables and recompile the document

  eststo clear
  eststo: reg weight mpg, robust
  eststo: reg weight mpg foreign, robust
  esttab using _example1.tex, replace title(Weight) ///
      se wide label

  eststo clear
  eststo: reg price weight mpg, robust
  eststo: reg price weight mpg foreign, robust
  esttab using _example2.tex, replace title(Price) ///
      se wide label
  <u>*<run>*</u>

      <u>*<texify>*</u>   <u>*<show PDF>*</u>   <u>*<show PDF from web>*</u>

_____

Example with MS Word and Excel I

- Such automation does not seem to be easily possible with MS Word.

- However, you can link data files into Excel and then dynamically link Excel tables into Word.

- Step 1: Generate results files in tab-delimited format

```
eststo clear
sysuse auto, clear
eststo: reg price weight mpg
eststo: reg price weight mpg foreign
esttab using _example.txt, tab replace plain
```
<run>

Example with MS Word and Excel II

- Step 2: Link data into Excel ("Data" > "Import External Data" > "Import Data..."; locate and select the "_example.txt" file; go through the "Text Import Wizard" and click "Finish"; on the last dialog, called "Import Data", click "Properties...", uncheck "Prompt for file name on refresh" and check "Refresh data on file open", possibly uncheck "Adjust column width").

- Step 3: Format the table in Excel.

- Step 4: Mark and copy the table in Excel and, in Word, paste the table as an Excel object ("Edit" > "Paste Special..."; make sure to select "Paste link").

- Save the Excel and Word documents.

Example with MS Word and Excel III

- You can now replace the results files ...

  ```
  eststo clear
  sysuse auto, clear
  eststo: reg price weight mpg, robust
  eststo: reg price weight mpg foreign, robust
  esttab using _example.txt, tab replace plain label
  ```
  <run>

- ... open the Excel file and click "Enable automatic refresh" ...

- ... open the Word document, mark the section containing the table, and hit "F9" ("Edit" > "Update Link") to update the table.

  (The default settings can be changed so that updating occurs automatically when opening the file: "Tools" > "Options" > "General" > check "Update automatic links at open")

<next>   <overview>

End of tutorial

- Clean-up: erase working files

```
capture erase mymodels.sters
capture erase mymodels.dta
capture erase example.txt
capture erase example.html
capture erase example.csv
capture erase example.rtf
capture erase example.tex
capture erase example.pdf
capture erase example.aux
capture erase example.log
capture erase _example.txt
capture erase _example1.tex
capture erase _example2.tex
```
<run>