Custom estimation tables

Jeff Pitblado

StataCorp LLC

2022 Oceania Stata Conference
Virtual

Outline

Estimation results

Example (Simulated data)

```
. describe

Contains data from etable.dta
 Observations:           200
    Variables:             6                  26 Jan 2022 12:01
```

Variable name	Storage type	Display format	Value label	Variable label
x1	float	%9.0g		X_1
x2	float	%9.0g		X_2
f1	byte	%9.0g	f1	F_1
f2	byte	%9.0g	f2	F_2
y1	byte	%9.0g		Binary outcome
y2	float	%9.0g		Continuous outcome

```
Sorted by:

. label list
f1:
           0 Off
           1 On
f2:
           1 First
           2 Second
           3 Third
           4 Fourth
           5 Fifth
```

Estimation results

Estimation commands

Fit models and post their results to **`e()`**.

`ereturn list`

- Scalars
  - **`e(N)`** – sample size
  - **`e(ll)`** – log likelihood
- Macros
  - **`e(cmdline)`** – command as typed
  - **`e(depvar)`** – dependent variable
  - **`e(title)`** – estimation title
- Matrices
  - **`e(b)`** – fitted values
  - **`e(V)`** – estimated variance for **`e(b)`**

Estimation results

Coefficient table

Show fitted values, standard errors, tests, and confidence intervals.

```
matrix list r(table)
```

Columns conform with **e(b)**.
Rows contain the displayed values:

- **b** – fitted values ($\widehat{\beta}$)
- **se** – standard error of **b**
- **z** or **t** – test statistic for $H_0 : \beta = 0$
- **pvalue** – p-value for above test statistic
- **ll** – lower confidence limit
- **ul** – lower confidence limit
- **df** – degrees of freedom

Estimation results

System variables

Provide easy access to elements of **e(b)** and **e(V)**.

- **_b** – elements of **e(b)**
- **_se** – square root of diagonal elements of **e(V)**

Use **_b** with postestimation commands **lincom**, **nlcom**, **test**, and **testnl**.

Example

```
. use etable
. logit y1 x1 x2 i.f1
. logit, coeflegend
. lincom _b[y1:x1] - 2*_b[y1:x2]
. nlcom _b[y1:x1] / _b[y1:x2]
```

Estimation results

System variables new in Stata 17

Provide easy access to elements of **`r(table)`**.

- **`_r_b`** – **`b`** elements of **`r(table)`**
- **`_r_se`** – **`se`** elements of **`r(table)`**
- **`_r_z`** – **`z`** or **`t`** elements of **`r(table)`**
- **`_r_z_abs`** – absolute value of **`_r_z`**
- **`_r_p`** – **`pvalue`** elements of **`r(table)`**
- **`_r_lb`** – **`ll`** elements of **`r(table)`**
- **`_r_ub`** – **`ul`** elements of **`r(table)`**
- **`_r_df`** – **`df`** elements of **`r(table)`**

Estimation results

System variables, continued

- **`r(table)`** is fleeting compared to the results in **`e()`**.
- **`_r_b`** and friends pull from a hidden copy stored with **`e()`**.
  - **`estimates store`**
  - **`estimates restore`**
  - **`estimates save`**
  - **`estimates use`**

Estimation results

Replay estimation results

_r_b and friends automatically update.

Example

```
. quietly logit

. display _b[x1]
1.322184

. display _r_b[x1]
1.322184

. quietly logit, or

. display _b[x1]
1.322184

. display _r_b[x1]
3.7516061
```

Estimation results

Estimation tables

Commands available before Stata 17

- Stata
  - **estimates table**
  - **putexcel**
  - **putdocx table**
  - **putpdf table**
  - **dyndoc**, **dyntext**, and **markdown**
- Community contributed
  - **outreg2** by R. Wada
  - **outreg** and **frmttable** by J. L. Gallup
  - **estout** and **esttab** by B. Jann
  - **asdoc** by A. Shah
  - **markdoc** by E. F. Haghish
  - ...

Collections

`collect` – suite of commands for building custom tables

Basic workflow

- consume results

```
collect get
collect:
```

- arrange items

```
collect layout
```

- manage appearance and behaviors

```
collect style
collect label
```

- publish

```
collect export
```

Collections

collect get *results*

Specify what results to consume.

Example

```
. quietly logit y1 x1 x2 i.f1
. collect get e()
. collect dir

Collections in memory
Current: default
──────────────────────
   Name    No. items
──────────────────────
default    72
──────────────────────
```

Collections

collect [get] [*results*] :

Prefix comand that automatically consumes from **`e()`** or **`r()`**.

Example

```
. collect : logit y1 x1 x2 i.f1
```

Collections

Items and tags

`collect get` and **`collect:`** consume results by adding them as items to a collection.

- An item is a single number or string.
- Items are tagged for reference in style edits and the layout.
- Tags are composed from dimension-level pairs:

$$dim[lev]$$

Collections

Scalar results

An item from **`e(N)`** has tag elements:

- **`result[N]`**
- **`result_type[scalar]`**
- **`program_class[eclass]`**

Macro results

An item from **`e(cmd)`** has tag elements:

- **`result[cmd]`**
- **`result_type[macro]`**
- **`program_class[eclass]`**

Collections

Matrix results

Also have tag elements for row and column names.

Example

```
. matrix list e(rules)

e(rules)[1,4]
    c1  c2  c3  c4
r1   0   0   0   0
```

An item from **`e(rules)["r1","c1"]`** has tag elements:

- **`result[rules]`**
- **`result_type[matrix]`**
- **`program_class[eclass]`**
- **`rowname[r1]`**
- **`colname[c1]`**

Collections

Special matrices

`e(b)` and `e(V)` are not consumed like other matrices.

`r(table)` is consumed using the new system variables.

`r(table)["b","y1:x1"]` is `_r_b[y1:x1]`

An item from this result has tag elements:

- `result[_r_b]`
- `result_type[matrix]`
- `program_class[eclass]`
- `coleq[y1]`
- `colname[x1]`

Collections

collect dims

List the dimensions in a collection

Example

```
. collect dims

Collection dimensions
Collection: default
```

Dimension	No. levels
Layout, style, header, label	
cmdset	1
coleq	1
colname	9
colname_remainder	1
f1	2
program_class	1
result	44
result_type	3
rowname	1
Style only	
border_block	4
cell_type	4

Collections

cmdset

Index for each set of results in the collection.

Example (list the levels/labels for cmdset)

```
. collect levelsof cmdset

Collection: default
 Dimension: cmdset
    Levels: 1

. collect label list cmdset, all

  Collection: default
   Dimension: cmdset
       Label: Command results index
Level labels:
           1

. collect label values cmdset 1 "log(Odds ratio)"
```

Collections

coleq

Column equations taken from matrix results.

Look like a variable? Get the variable's label.

Example

```
. collect levelsof coleq

Collection: default
 Dimension: coleq
    Levels: y1

. collect label list coleq

  Collection: default
   Dimension: coleq
       Label: Depvars, parameters, and column equations
Level labels:
          y1  Binary outcome
```

Collections

colname

Column names taken from matrix results.

Look like a variable? Get the variable's label.

Example

```
. collect levelsof colname

Collection: default
 Dimension: colname
    Levels: x1 x2 0.f1 1.f1 c1 c2 c3 c4 _cons

. collect label list colname

  Collection: default
   Dimension: colname
       Label: Covariate names and column names
Level labels:
       _cons  Intercept
          f1  F_1
          x1  X_1
          x2  X_2
```

Collections

Factor variables

Become dimensions, if found in **`colname`**.

Their value labels are also consumed.

Example

```
. collect levelsof f1

Collection: default
 Dimension: f1
    Levels: 0 1

. collect label list f1

  Collection: default
   Dimension: f1
       Label: F_1
Level labels:
           0  Off
           1  On
```

Collections

result

Filled with the names of results consumed from **`e()`**, and includes the **`r(table)`** system variables.

```
. collect levelsof result
(output too long, omitted)
```

`collect` has command-specific labels for many results.

```
. collect label list result, all
(output too long, omitted)
```

`_r_ci` is a special composite level in **`result`**, it is composed from items tagged with **`_r_lb`** and **`_r_ub`**.

Collections

colname_remainder

Contains the remaining column stripe pieces when factor variables are removed.

Possibly useful for arranging/selecting interaction elements in the layout.

rowname

Similar to **`colname`**.

border_block and cell_type

Are not part of tags.

Used for style targeting.

Layout and appearance

collect layout

Use dimensions to select and arrange items into a table.

Without arguments, show the current layout information and possibly show you a table.

Example (default layout is empty)

```
. collect layout
Your layout specification does not identify any items.
```

Layout and appearance

Syntax

collect layout (*rowspec***) (***colspec***)**

where *rowspec* and *colspec* are composed from dimensions and their interactions (**#**).

Coefficient table layout

rowspec

coleq#colname

colspec

result[_r_b _r_se _r_z _r_p _r_ci]

Layout and appearance

Example (coefficient table layout)

```
. collect layout (coleq#colname) (result[_r_b _r_se _r_z _r_p _r_ci])

Collection: default
      Rows: coleq#colname
   Columns: result[_r_b _r_se _r_z _r_p _r_ci]
   Table 1: 6 x 5
```

	Coefficient	Std. error	z	p-value	95% CI	
Binary outcome						
X_1	1.322184	.5985862	2.21	0.027	.1489766	2.495391
X_2	-1.064822	.5836827	-1.82	0.068	-2.208819	.079175
Off	0	0				
On	2.049094	.3342964	6.13	0.000	1.393885	2.704303
Intercept	-.7792675	.4344927	-1.79	0.073	-1.630858	.0723225

Layout and appearance

collect style header

Controls the appearance of dimensions and their levels in row and column headers.

Example (hide the equation)

```
. collect query header coleq

Dimension header styles
Collection: default
 Dimension: coleq
     Level:
Show title: hide
Show level: label

. collect style header coleq, level(hide)
```

Layout and appearance

collect preview

Shows a preview of your table.

Example (bye bye equation)

```
. collect preview
```

	Coefficient	Std. error	z	p-value	95% CI	
X_1	1.322184	.5985862	2.21	0.027	.1489766	2.495391
X_2	-1.064822	.5836827	-1.82	0.068	-2.208819	.079175
Off	0	0				
On	2.049094	.3342964	6.13	0.000	1.393885	2.704303
Intercept	-.7792675	.4344927	-1.79	0.073	-1.630858	.0723225

Layout and appearance

Example (more polish)

```
. collect style showbase off
. collect style header f1, title(label)
. collect style row stack, nobinder
. collect style cell result[_r_b _r_se _r_ci], nformat(%7.4f)
. collect preview
```

	Coefficient	Std. error	z	p-value	95% CI	
X_1	1.3222	0.5986	2.21	0.027	0.1490	2.4954
X_2	-1.0648	0.5837	-1.82	0.068	-2.2088	0.0792
F_1						
On	2.0491	0.3343	6.13	0.000	1.3939	2.7043
Intercept	-0.7793	0.4345	-1.79	0.073	-1.6309	0.0723

Layout and appearance

Example (estimates table layout)

```
. collect layout (coleq#colname#result[_r_b _r_se]) (cmdset)

Collection: default
      Rows: coleq#colname#result[_r_b _r_se]
   Columns: cmdset
   Table 1: 13 x 1
```

	log(Odds ratio)
X_1	
Coefficient	1.3222
Std. error	0.5986
X_2	
Coefficient	-1.0648
Std. error	0.5837
F_1	
On	
Coefficient	2.0491
Std. error	0.3343
Intercept	
Coefficient	-0.7793
Std. error	0.4345

Layout and appearance

Example (add parenthesis around standard errors)

```
. collect style cell result[_r_se], sformat("(%s)")
. collect preview
```

	log(Odds ratio)
X_1	
Coefficient	1.3222
Std. error	(0.5986)
X_2	
Coefficient	-1.0648
Std. error	(0.5837)
F_1	
On	
Coefficient	2.0491
Std. error	(0.3343)
Intercept	
Coefficient	-0.7793
Std. error	(0.4345)

Layout and appearance

Example (hide result levels)

```
. collect query header result

Dimension header styles
Collection: default
 Dimension: result
     Level:
Show title: hide
Show level: label

. collect style header result , level(hide)

. collect preview
```

	log(Odds ratio)
X_1	1.3222
	(0.5986)
X_2	-1.0648
	(0.5837)
F_1	
On	2.0491
	(0.3343)
Intercept	-0.7793
	(0.4345)

Layout and appearance

collect stars

Label significant results.

Adds items tagged with **`result[stars]`** by default.

Can attach labels to items tagged with a different **`result`**.

Example (commonly used labeling rules)

```
. collect stars _r_p .01 "**" .05 "*" , attach(_r_b)
. collect preview
```

	log(Odds ratio)
X_1	1.3222*
	(0.5986)
X_2	-1.0648
	(0.5837)
F_1	
On	2.0491**
	(0.3343)
Intercept	-0.7793
	(0.4345)

Layout and appearance

Recent update to Stata 17

- **`collect stars`** information is stored in the style.
- **`collect`** will apply the **`stars`** labeling rules to subsequently collected results.
- Option **`dimension`** adds items with **`stars`** as a dimension instead of a level of **`result`**.

Layout and appearance

Example

```
. collect stars, dimension

. collect dims

Collection dimensions
Collection: default
```

Dimension	No. levels
Layout, style, header, label	
cmdset	1
coleq	1
colname	9
colname_remainder	1
f1	2
program_class	1
result	45
result_type	3
rowname	1
stars	2
Style only	
border_block	4
cell_type	4

```
. collect levelsof stars

Collection: default
 Dimension: stars
    Levels: label value
```

Layout and appearance

Example (add stars dimension to the layout)

```
. collect layout (coleq#colname#result[_r_b _r_se]) (cmdset#stars)

Collection: default
      Rows: coleq#colname#result[_r_b _r_se]
   Columns: cmdset#stars
   Table 1: 9 x 2
```

	log(Odds ratio)	log(Odds ratio)
X_1	1.3222	*
	(0.5986)	
X_2	-1.0648	
	(0.5837)	
F_1		
On	2.0491	**
	(0.3343)	
Intercept	-0.7793	
	(0.4345)	

Layout and appearance

collect style column

Controls appearance and arrangement of column headers.

Example (center duplicate column headers)

```
. collect query column

Column header styles
   Collection: default
 No delimiter: on
    Delimiter: " # "
 At delimiter: " @ "
Bar delimiter: " | "
       Binder: "="
   Duplicates: repeat
     Position: top
  Extra space: 0
        Width: asis

. collect style column, dups(center)
```

Layout and appearance

Example (also left align the stars)

```
. collect style cell stars[label], halign(left)
. collect preview
```

	log(Odds ratio)
X_1	1.3222 *
	(0.5986)
X_2	-1.0648
	(0.5837)
F_1	
On	2.0491 **
	(0.3343)
Intercept	-0.7793
	(0.4345)

Layout and appearance

Example (add odds ratios results)

```
. quietly logit, or
. collect get e()
. collect label levels cmdset 2 "Odds ratio"
. collect preview
```

	log(Odds ratio)	Odds ratio
X_1	1.3222 *	3.7516 *
	(0.5986)	(2.2457)
X_2	-1.0648	0.3448
	(0.5837)	(0.2012)
F_1		
On	2.0491 **	7.7609 **
	(0.3343)	(2.5944)
Intercept	-0.7793	0.4587
	(0.4345)	(0.1993)

Estimation tables

Workflow review

- consume results

```
collect get
collect:
```

- arrange items

```
collect layout
```

- manage appearance and behaviors

```
collect style
collect label
```

- publish

```
collect export
```

Estimation tables

Challenges

- Too many commands?
- Learn a new language to specify a layout.
- Some behaviors cannot be coded generically in styles.

Estimation tables

etable

Simple syntax for building estimation tables.

Inspired by community contributed table commands.

Developed using **`collect`**.

Replaces **`estimates table`**.

Estimation tables

etable default behaviors

- Collect from the current estimation results.
- Create a collection named **`ETable`**.
- Show dependent variable in the column header.
- Hide equation names.
- Report coefficients.
- Report standard errors with parenthesis.
- Report the number of observations.

Estimation tables

Example (default etable)

```
. quietly logit

. etable
```

	y1
X_1	1.322
	(0.599)
X_2	-1.065
	(0.584)
F_1	
On	2.049
	(0.334)
Intercept	-0.779
	(0.434)
Number of observations	200

```
. estimates store m1

. etable, estimates(m1)
(same table)
```

Estimation tables

Example (show stars and note)

```
. etable, showstars showstarsnote
```

	y1
X_1	1.322 *
	(0.599)
X_2	-1.065
	(0.584)
F_1	
On	2.049 **
	(0.334)
Intercept	-0.779
	(0.434)
Number of observations	200

** p<.01, * p<.05

Estimation tables

stars() option

Manage labels for significant results.

Example (etable default stars properties)

```
. collect query stars

Stars styles
    Collection: ETable
          Type: dimension
       Results: _r_p
        Attach: _r_b
      For tags:
        Rule 1: .01 **
        Rule 2: .05 *
        Rule 3:
        Rule 4:
        Rule 5:
     Show note: on
Name of p-value: "p"
Numeric format: "%9.0g"
Note delimiter: ","
   Note prefix: ""
   Note suffix: ""
          Note: "** p<.01, * p<.05"
```

Estimation tables

etable replaces its collection and consumes from the current estimation results.

Example

```
. quietly logit, or
. etable
```

	y1
X_1	3.752
	(2.246)
X_2	0.345
	(0.201)
F_1	
On	7.761
	(2.594)
Intercept	0.459
	(0.199)
Number of observations	200

Use option **append** to add results to collection **ETable**.

Estimation tables

Example (quick model comparisons)

```
. quietly regress y2 x1 x2 i.f1
. estimates store m2
. etable, estimates(m1 m2) showstars
```

	y1	y2
X_1	1.322 *	
	(0.599)	
X_2	-1.065	
	(0.584)	
F_1		
On	2.049 **	
	(0.334)	
Intercept	-0.779	
	(0.434)	
X_1		0.691
		(0.363)
X_2		-0.920 *
		(0.355)
F_1		
On		3.079 **
		(0.206)
Intercept		-0.879 **
		(0.276)
Number of observations	200	200

Estimation tables

Example (show equations)

```
. etable, estimates(m1 m2) showstars showeq
```

	y1	y2
Binary outcome		
X_1	1.322 *	
	(0.599)	
X_2	-1.065	
	(0.584)	
F_1		
On	2.049 **	
	(0.334)	
Intercept	-0.779	
	(0.434)	
Continuous outcome		
X_1		0.691
		(0.363)
X_2		-0.920 *
		(0.355)
F_1		
On		3.079 **
		(0.206)
Intercept		-0.879 **
		(0.276)
Number of observations	200	200

Estimation tables

Use option **`replay`** to apply edits to collection **`ETable`**.

Use option **`eqrecode()`** to recode equations.

Example

```
. etable, replay eqrecode(y1 = xb y2 = xb) noshoweq
```

	y1		y2	
X_1	1.322	*	0.691	
	(0.599)		(0.363)	
X_2	-1.065		-0.920	*
	(0.584)		(0.355)	
F_1				
On	2.049	**	3.079	**
	(0.334)		(0.206)	
Intercept	-0.779		-0.879	**
	(0.434)		(0.276)	
Number of observations	200		200	

Estimation tables

Example (change column header to command names)

```
. etable, replay column(command)
```

	logit	regress
X_1	1.322 *	0.691
	(0.599)	(0.363)
X_2	-1.065	-0.920 *
	(0.584)	(0.355)
F_1		
On	2.049 **	3.079 **
	(0.334)	(0.206)
Intercept	-0.779	-0.879 **
	(0.434)	(0.276)
Number of observations	200	200

Estimation tables

Example (select your model statistics)

```
. etable, replay mstat(ll) mstat(N)
```

	logit	regress
X_1	1.322 *	0.691
	(0.599)	(0.363)
X_2	-1.065	-0.920 *
	(0.584)	(0.355)
F_1		
On	2.049 **	3.079 **
	(0.334)	(0.206)
Intercept	-0.779	-0.879 **
	(0.434)	(0.276)
Log likelihood	-109.82	-354.82
Number of observations	200	200

Estimation tables

Example (show variable names and factor values)

```
. etable, replay novarlabel nofvlabel
```

	logit	regress
x1	1.322 *	0.691
	(0.599)	(0.363)
x2	-1.065	-0.920 *
	(0.584)	(0.355)
f1		
1	2.049 **	3.079 **
	(0.334)	(0.206)
_cons	-0.779	-0.879 **
	(0.434)	(0.276)
Log likelihood	-109.82	-354.82
Number of observations	200	200

Estimation tables

Example (add title and note)

```
. etable, replay ///
>         title("Model comparison") ///
>         titlestyles(font(, bold)) ///
>         showstarsnote ///
>         notestyles(font(, italic))
```

Model comparison

	logit	regress
x1	1.322 *	0.691
	(0.599)	(0.363)
x2	-1.065	-0.920 *
	(0.584)	(0.355)
f1		
1	2.049 **	3.079 **
	(0.334)	(0.206)
_cons	-0.779	-0.879 **
	(0.434)	(0.276)
Log likelihood	-109.82	-354.82
Number of observations	200	200

*** p<.01, * p<.05*

Estimation tables

collect style save

Save your style for use as a starting point in a future analysis.

Example

```
. collect style save my-et-style, replace
(style from ETable saved to file my-et-style.stjson)
```

set etable_style

You will be able to set a custom default style for **`etable`**.

Estimation tables

Rebuild the table using our new style.

Example

```
. collect clear
. etable, estimates(m1 m2) style(my-et-style)
```

Model comparison

	logit	regress
x1	1.322 *	0.691
	(0.599)	(0.363)
x2	-1.065	-0.920 *
	(0.584)	(0.355)
f1		
1	2.049 **	3.079 **
	(0.334)	(0.206)
_cons	-0.779	-0.879 **
	(0.434)	(0.276)
Log likelihood	-109.82	-354.82
Number of observations	200	200

*** p<.01, * p<.05*

Estimation tables

What's next?

- Custom composite results, beyond **`_r_ci`**
- Alternate text when a value goes out of a specified range
  - Report **`<.01`** instead of **`0.00`** for very small p-values
- Alternate text for empty cells
  - Option for **`table`** to show **`0`** counts in empty cells
- Document the Mata code that implements **`collect`**
- ...