Running, Storing, and Plotting Regression in *Stata*

Andrew Taeho Kim

Department of Sociology
University of Kansas

Demography & Labor Market Research Group
April 22, 2022

Necessary Packages

```
. ssc install estout
. ssc install estwrite
. ssc install coefplot
```

Data

```
. use "../workingdata/wrkdt", clear
. desc
Contains data from ../workingdata/wrkdt.dta
 Observations:     1,086,199
    Variables:             9                  22 Apr 2022 15:53
```

Variable name	Storage type	Display format	Value label	Variable label
year	int	%9.0g		year
wt	float	%9.0g		survey weight
fem	float	%9.0g	fem	women
mst	float	%9.0g	mst	marital status
red	float	%9.0g	red	level of education
rra	float	%9.0g	rra	race/ethnicity
baa	float	%9.0g		bachelor´s degree
age	float	%9.0g		age
wage	float	%9.0g		log hourly wage

```
Sorted by:
```

Macros

Typing all

```
. reg wage i.fem c.age##c.age i.red i.rra i.mst i.year [pw=wt]
. eststo m1
```

Set up globals & locals

```
. global dv wage
. local ctr1 c.age##c.age i.red i.rra i.mst i.year
.
. reg $dv i.fem `ctr1´ [pw=wt]
. eststo m1
```

- *global* can be called again as long as you don't exit out of Stata
- *local* needs to be executed with the part of the code you call on it
- *eststo* stores the regression result on memory until you exit out of Stata

Utilize loops #1

Different models

```
. local ctr1 c.age##c.age i.rra i.mst i.year
. local ctr2 c.age##c.age i.rra i.mst i.year i.red
.
. reg $dv i.fem `ctr1´ [pw=wt]
. eststo m1
.
. reg $dv i.fem `ctr2´ [pw=wt]
. eststo m2
```

Loop over models

```
. local ctr1 c.age##c.age i.rra i.mst i.year
. local ctr2 c.age##c.age i.rra i.mst i.year i.red
.
. foreach md in 1 2 {
.   reg $dv i.fem `ctr`md´´ [pw=wt]
.   eststo m`md´
. }
```

Results

```
. esttab m1 m2, ///
>   mtitle(m1 m2) ///
>   b(3) se(3) r2(3) ar2(3) keep(*.fem) lab
```

	(1) m1	(2) m2
men	0.000	0.000
	(.)	(.)
women	-0.228***	-0.266***
	(0.002)	(0.002)
Observations	1086199	1086199
R-squared	0.083	0.213
Adjusted R-squared	0.083	0.213

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

Utilize loops #2

Same model for multiple years

```
. local ctr3 c.age##c.age i.rra i.mst
.
. reg $dv i.fem `ctr3´ [pw=wt] if year == 2000
. eststo m3_y2000
.
. reg $dv i.fem `ctr3´ [pw=wt] if year == 2010
. eststo m3_y2010
.
. reg $dv i.fem `ctr3´ [pw=wt] if year == 2019
. eststo m3_y2019
```

Loop same model over years

```
. local ctr3 c.age##c.age i.rra i.mst
.
. foreach y in 2000 2010 2019 {
.   reg $dv i.fem `ctr3´ [pw=wt] if year == `y´
.   eststo m3_y`y´
. }
```

Results

```
. esttab m3_y2000 m3_y2010 m3_y2019, ///
>   mtitle(y2000 y2010 y2019) ///
>   b(3) se(3) r2(3) ar2(3) keep(*.fem) lab
```

	(1) y2000	(2) y2010	(3) y2019
men	0.000	0.000	0.000
	(.)	(.)	(.)
women	-0.296***	-0.231***	-0.183***
	(0.008)	(0.007)	(0.008)
Observations	40023	56033	44917
R-squared	0.085	0.082	0.090
Adjusted R-squared	0.085	0.082	0.090

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

Loops can be nested

Loop over each model & each year

```
. local ctr3 c.age##c.age i.rra i.mst
. local ctr4 c.age##c.age i.rra i.mst i.red
.
. foreach md in 3 4 {
.   foreach y in 2000 2010 2019 {
.     reg $dv i.fem `ctr`md´´ [pw=wt] if year == `y´
.     eststo m`md´_y`y´
.   }
. }
```

- above codes run 6 regressions (3 years $\times$ 2 models)

What if there are lots of years?

Put values of variable *year* into local *yrs* and loop over each

```
. levelsof year, local(yrs)
.
. local ctr3 c.age##c.age i.rra i.mst
.
. foreach y in `yrs´ {
.   reg $dv i.fem `ctr3´ [pw=wt] if year == `y´
.   eststo m3_y`y´
. }
```

Same results, different way to loop

```
. levelsof year, local(yrs)
. local yrsn : word count `yrs´
.
. local ctr3 c.age##c.age i.rra i.mst
.
. forvalues i = 1(1)`yrsn´ {
.   local y : word `i´ of `yrs´
.
.   reg $dv i.fem `ctr3´ [pw=wt] if year == `y´
.   eststo m3_y`y´
. }
```

Loop over models and years

```
. levelsof year, local(yrs)
. local yrsn : word count `yrs´
.
. local ctr3 c.age##c.age i.rra i.mst
. local ctr4 c.age##c.age i.rra i.mst i.red
.
. foreach md in 3 4 {
.   forvalues i = 1(1)`yrsn´ {
.         local y : word `i´ of `yrs´
.     reg $dv i.fem `ctr`md´´ [pw=wt] if year == `y´
.     eststo m`md´_y`y´
.   }
. }
```

Save results

Save all stored estimates through *eststo*

```
. estwrite * using "../estimates/results", replace
```

Save the results with names that starts with m4

```
. estwrite m4_* using "../estimates/results_m4", replace
```

- do this so we can use stored regression results later without having to run everything again

Load/Read

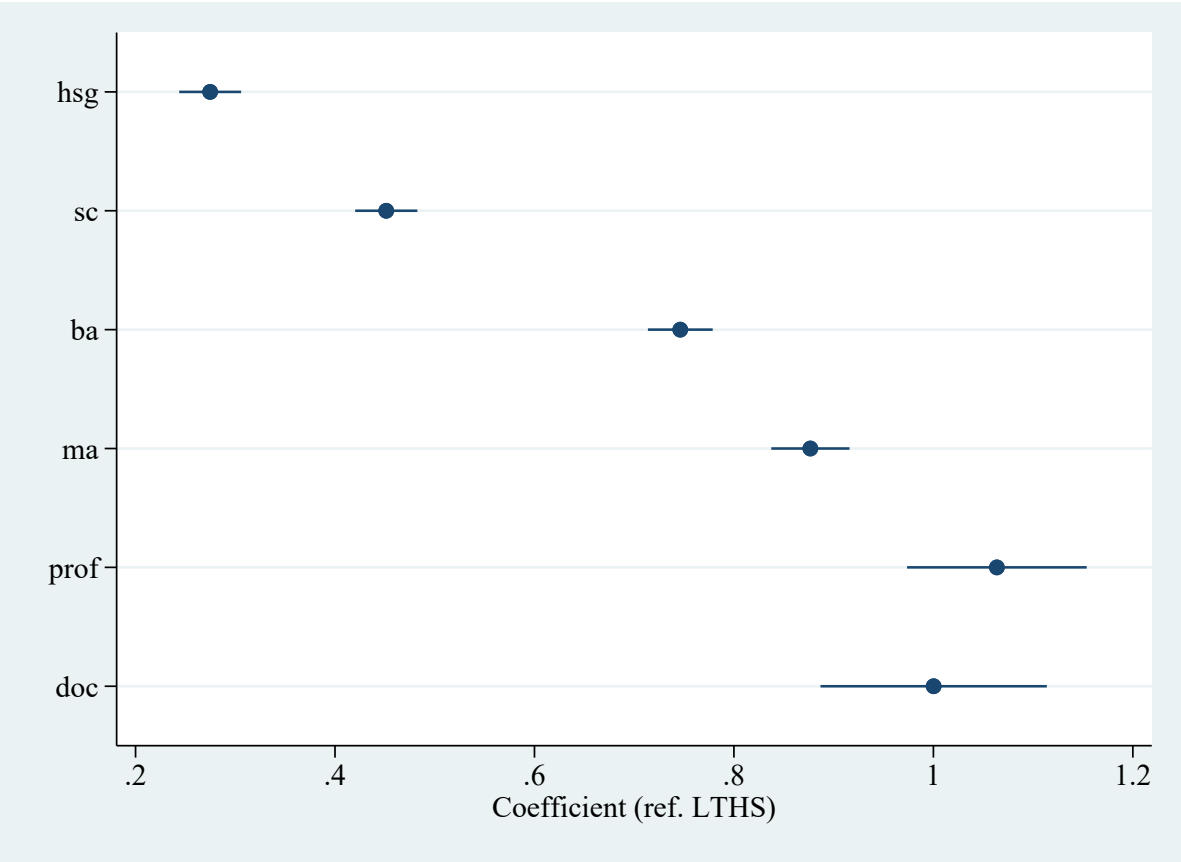
Load all estimates

```
. estread using "../estimates/results"
```

- if you already have an estimate stored with *eststo* and it happens to have the same name as one of the estimates in the file you just loaded, it will be overwritten

coefplot: typical usage

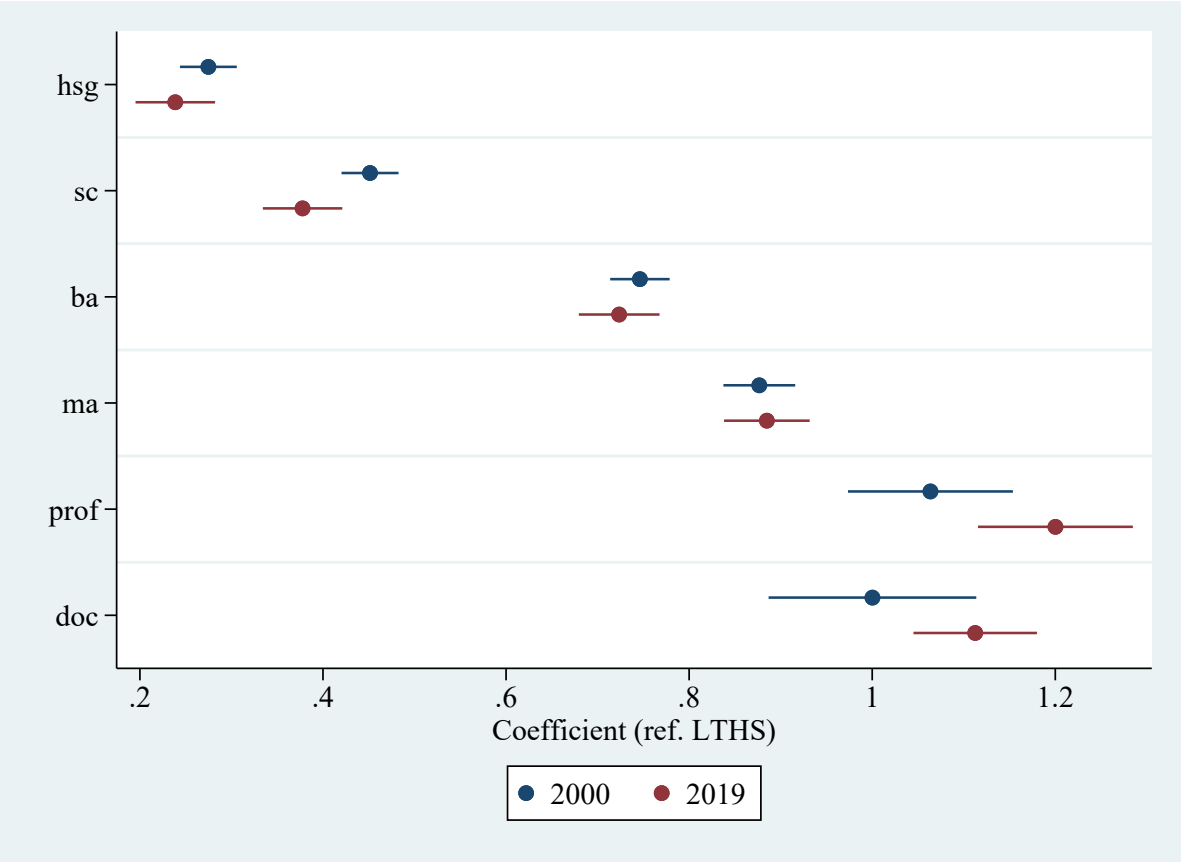
```
. coefplot m4_y2000, ///
>   keep(*.red) lab xtitle("Coefficient (ref. LTHS)") name(fig1, replace)
.
```



- returns to education in year 2000

coefplot: typical usage, multiple years

```
. coefplot m4_y2000 m4_y2019, ///
>   keep(*.red) lab xtitle("Coefficient (ref. LTHS)") ///
>   legend(label(2 "2000") label(4 "2019")) name(fig2, replace)
```



- returns to education in year 2000 and 2019

Plot coefficient for each year (2000 to 2019), Prep

- Create a matrix with a name *m3*
  - # of rows = number of years
  - # of columns = 4 : year, coefficient, CI lower, CI upper

```
. levelsof year, local(yrs)
. local row : word count `yrs´
. local col = 4
.
. mat define m3 = J(`row´, `col´, .)
. mat colnames m3 = year b ci_l ci_u
```

Plot coefficient for each year (2000 to 2019), Prep

```
. mat list m3
m3[20,4]
     year      b   ci_l   ci_u
 r1     .      .      .      .
 r2     .      .      .      .
 r3     .      .      .      .
 r4     .      .      .      .
 r5     .      .      .      .
 r6     .      .      .      .
 r7     .      .      .      .
 r8     .      .      .      .
 r9     .      .      .      .
r10     .      .      .      .
r11     .      .      .      .
r12     .      .      .      .
r13     .      .      .      .
r14     .      .      .      .
r15     .      .      .      .
r16     .      .      .      .
r17     .      .      .      .
r18     .      .      .      .
r19     .      .      .      .
r20     .      .      .      .
```

Plot coefficient for each year (2000 to 2019), Prep

```
. levelsof year, local(yrs)
. local row : word count `yrs´
.
. forvalues i = 1(1)`row´ {
.         local y : word `i´ of `yrs´
.
.         est res m3_y`y´
.
.         mat m3[`i´, 1] = `y´
.         mat m3[`i´, 2] = _b[1.fem]
.         mat m3[`i´, 3] = _b[1.fem] - invttail(e(df_r),0.025)*_se[1.fem]
.         mat m3[`i´, 4] = _b[1.fem] + invttail(e(df_r),0.025)*_se[1.fem]
. }
```

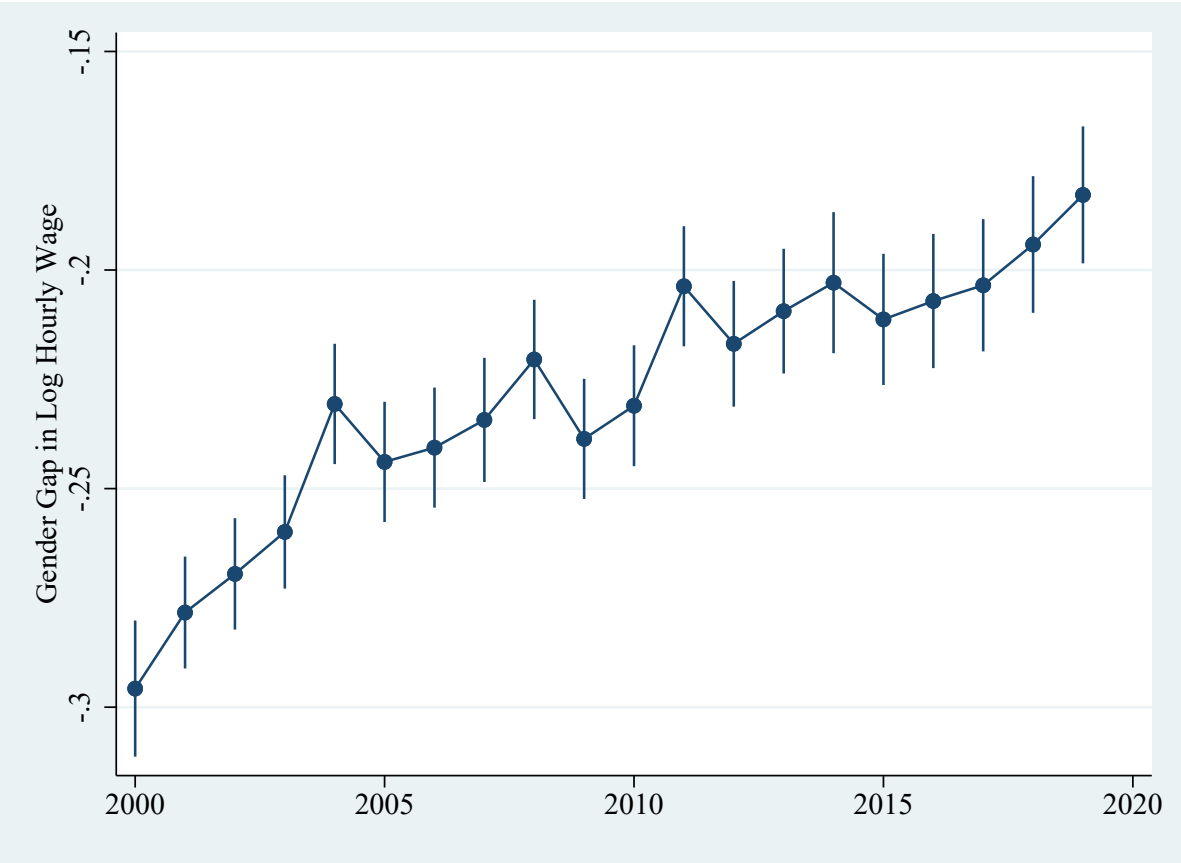
- *est res* call estimation result to memory
- model 3 : m3
  - _b[1.fem] : coefficient for *fem*
  - _se[1.fem] : standard error for *fem*

Plot coefficient gap for each year (2000 to 2019), Prep

```
. mat list m3
m3[20,4]
            year           b        ci_l        ci_u
 r1         2000  -.29574664  -.31131913  -.28017415
 r2         2001  -.27834477  -.29114098  -.26554855
 r3         2002  -.26951339  -.28225537   -.2567714
 r4         2003  -.25991102  -.27287517  -.24694688
 r5         2004  -.23063205  -.24439363  -.21687048
 r6         2005  -.24389231  -.25764426  -.23014035
 r7         2006  -.24062516  -.25436089  -.22688943
 r8         2007  -.23429088  -.24849467  -.22008709
 r9         2008  -.22045402  -.23410958  -.20679847
r10         2009  -.23862355  -.25236239  -.22488471
r11         2010  -.23104325  -.24487143  -.21721507
r12         2011  -.20372199  -.21746175  -.18998223
r13         2012  -.21687375  -.23125467  -.20249284
r14         2013  -.20940939  -.22368305  -.19513572
r15         2014  -.20289215  -.21902764  -.18675666
r16         2015  -.21129402  -.22629525  -.19629279
r17         2016   -.2070912  -.22244277  -.19173964
r18         2017  -.20347991   -.2186064  -.18835341
r19         2018  -.19416118  -.20978985  -.17853252
r20         2019  -.18281315  -.19847676  -.16714955
```

Plot coefficient for each year (2000 to 2019)

```
. coefplot (matrix(m3[, 2]), ci((m3[, 3] m3[, 4])) ), ///
>   at(matrix(m3[, 1])) ///
>   vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") name(fig3, replace)
```



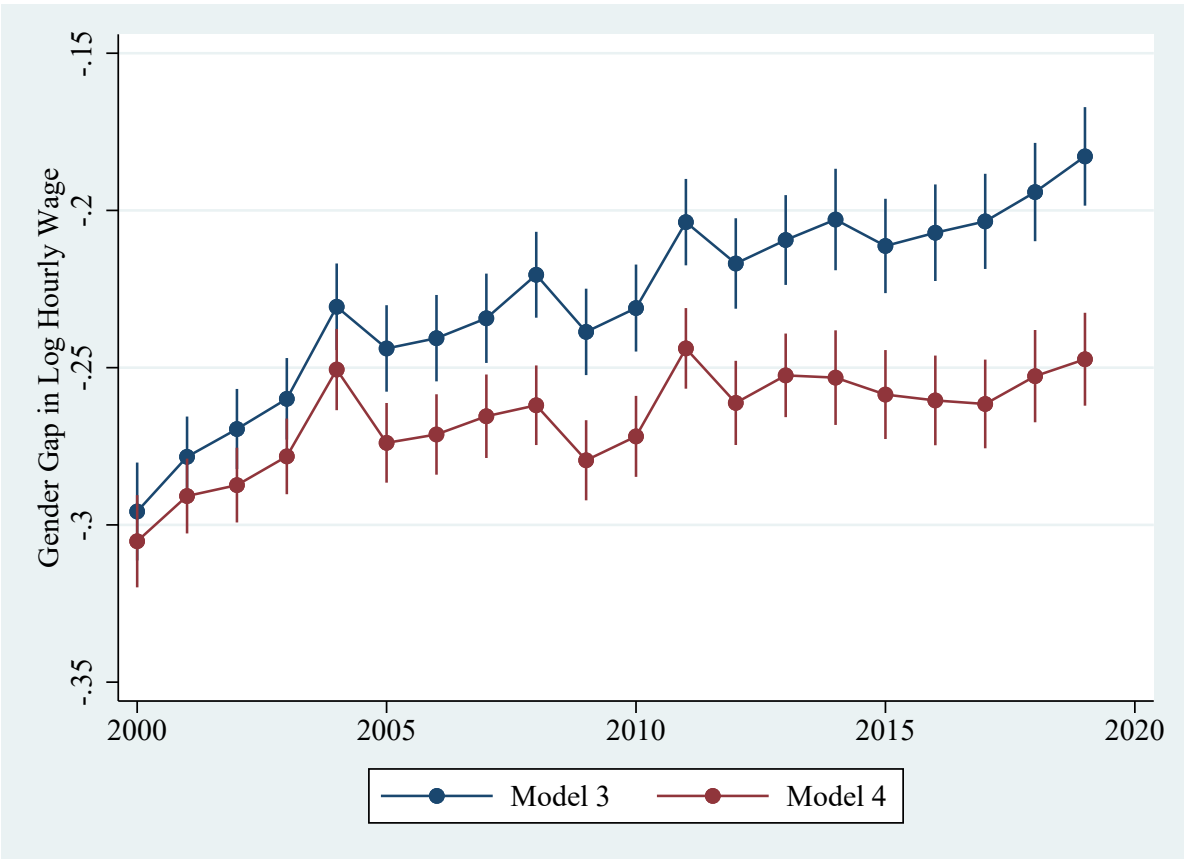
Plot multiple models

- create a matrix for model 4 results on gender gap

```
. levelsof year, local(yrs)
. local row : word count `yrs´
. local col = 4
.
. mat define m4 = J(`row´, `col´, .)
. mat colnames m4 = year b ci_l ci_u
.
. forvalues i = 1(1)`row´ {
.         local y : word `i´ of `yrs´
.
.         est res m4_y`y´
.
.         mat m4[`i´, 1] = `y´
.         mat m4[`i´, 2] = _b[1.fem]
.         mat m4[`i´, 3] = _b[1.fem] - invttail(e(df_r),0.025)*_se[1.fem]
.         mat m4[`i´, 4] = _b[1.fem] + invttail(e(df_r),0.025)*_se[1.fem]
. }
```

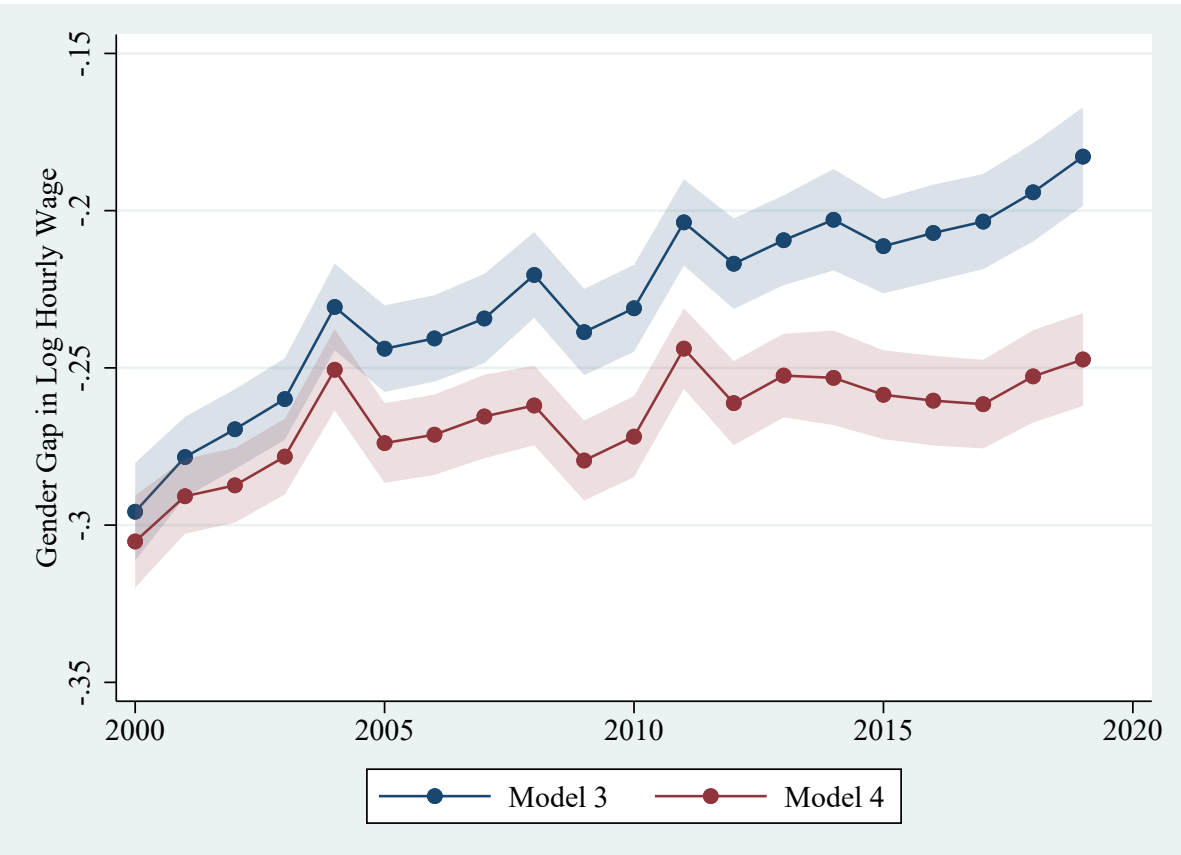
Plot coefficient for each year (2000 to 2019), m3 & m4

```
. coefplot (matrix(m3[, 2]), ci((m3[, 3] m3[, 4])) ) ///
>   (matrix(m4[, 2]), ci((m4[, 3] m4[, 4])) ), ///
>   at(matrix(m3[, 1])) ///
>   vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") ///
>   legend(label(2 "Model 3") label(4 "Model 4")) name(fig4, replace)
```



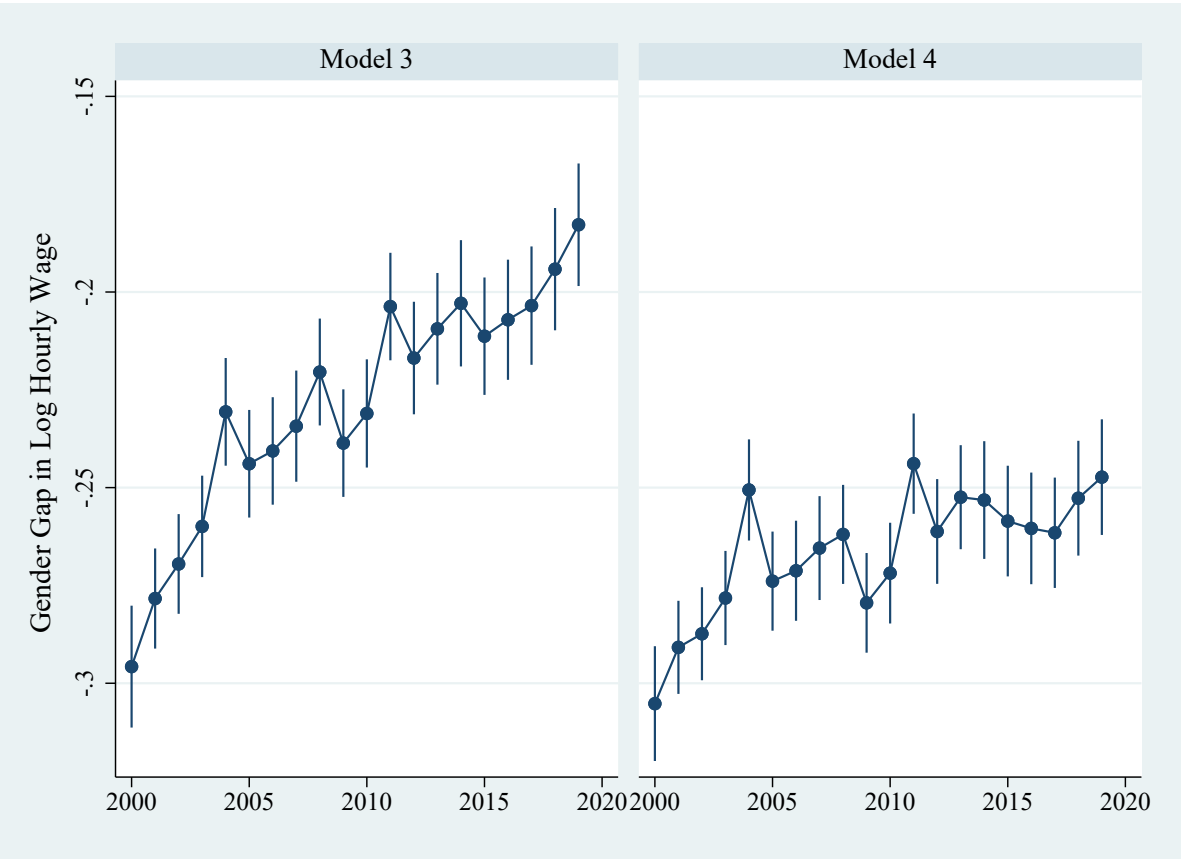
Plot coefficient for each year (2000 to 2019), m3 & m4

```
. coefplot (matrix(m3[, 2]), ci((m3[, 3] m3[, 4])) ) ///
>   (matrix(m4[, 2]), ci((m4[, 3] m4[, 4])) ), ///
>   at(matrix(m3[, 1])) ///
>   vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") ///
>   legend(label(2 "Model 3") label(4 "Model 4")) ///
>   ciopts(recast(rarea) fcolor(%20) lcolor(%0)) name(fig5, replace)
```



Plot coefficient for each year (2000 to 2019), m3 & m4

```
. coefplot (matrix(m3[, 2]), ci((m3[, 3] m3[, 4])) ), bylabel("Model 3") ///
>   || (matrix(m4[, 2]), ci((m4[, 3] m4[, 4])) ), bylabel("Model 4") ///
>   at(matrix(m3[, 1])) ///
>   vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") ///
>   byopts(rows(1)) name(fig6, replace)
```



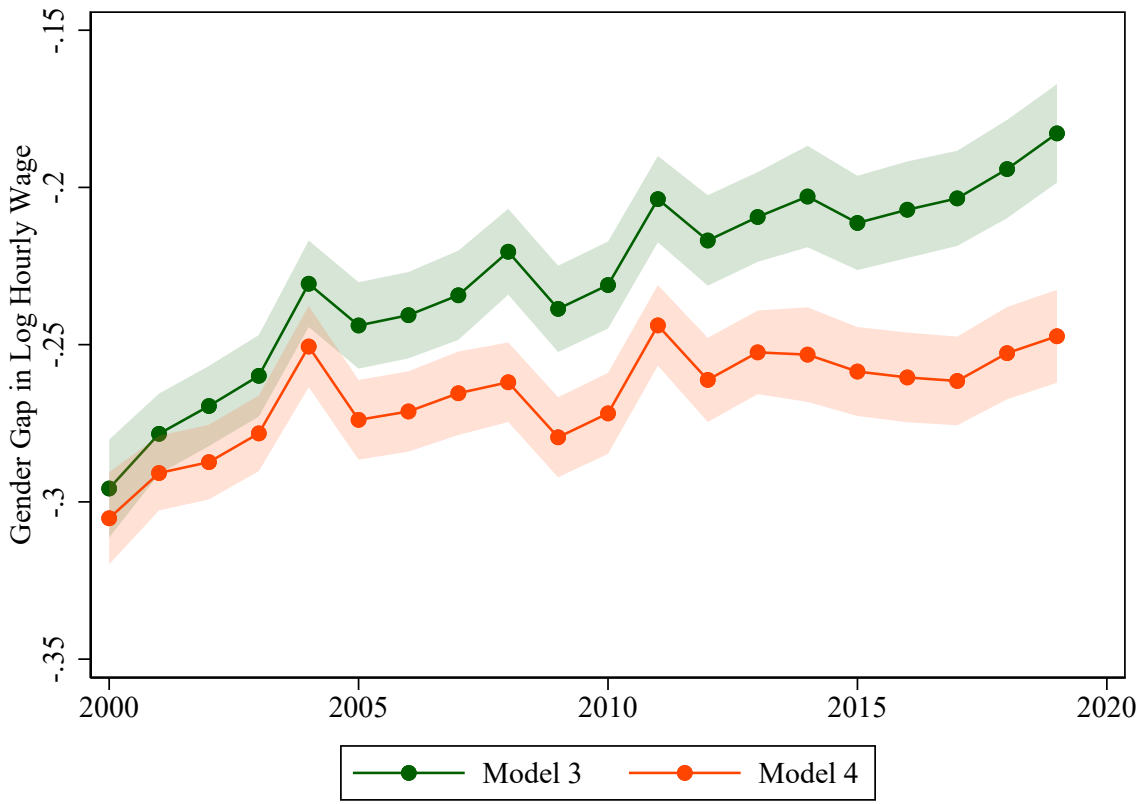
Scheme

- *Stata* use graphic schemes
  - default is `s2color`

```
. help scheme
. net install cleanplots, from("https://tdmize.github.io/data/cleanplots")
```

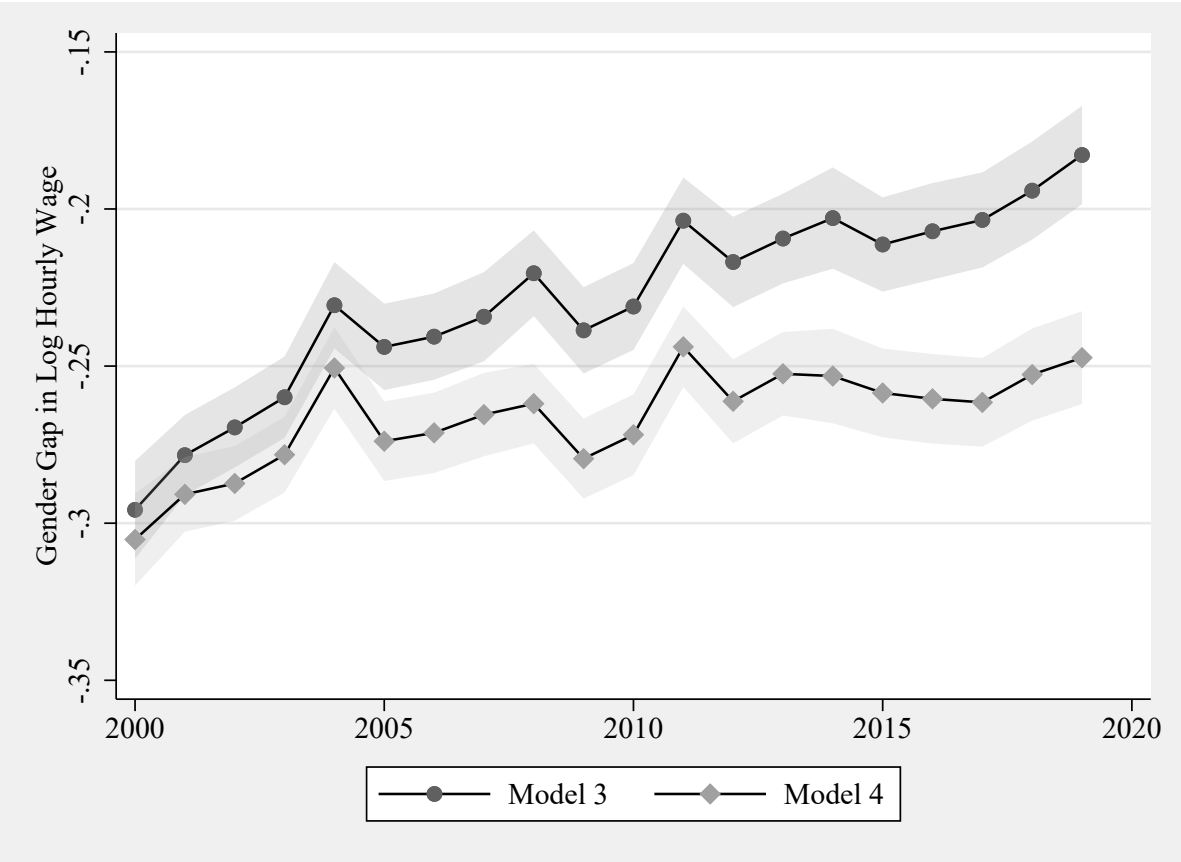
Scheme: s1color

```
. set scheme s1color
. coefplot (matrix(m3[, 2]), ci((m3[, 3] m3[, 4])) ) ///
>   (matrix(m4[, 2]), ci((m4[, 3] m4[, 4])) ), ///
>   at(matrix(m3[, 1])) ///
>   vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") ///
>   legend(label(2 "Model 3") label(4 "Model 4")) ///
>   ciopts(recast(rarea) fcolor(%20) lcolor(%0)) name(fig7, replace)
```



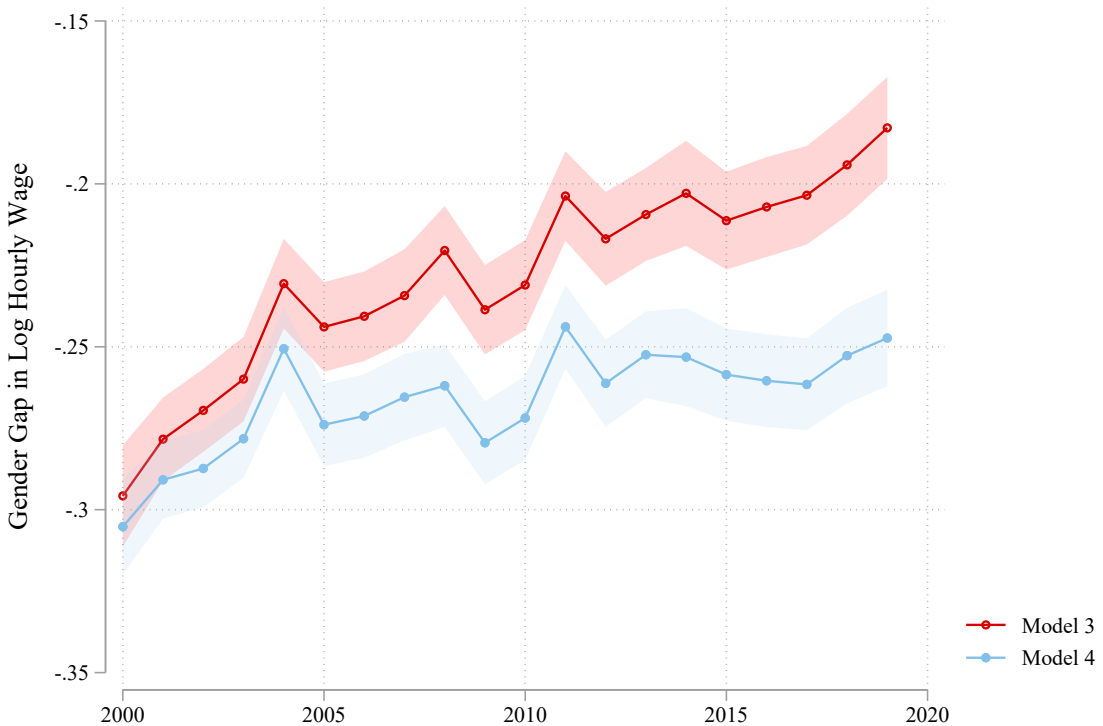
Scheme: sj → *Stata Journal*

```
. set scheme sj
. coefplot (matrix(m3[, 2]), ci((m3[, 3] m3[, 4])) ) ///
>   (matrix(m4[, 2]), ci((m4[, 3] m4[, 4])) ), ///
>   at(matrix(m3[, 1])) ///
>   vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") ///
>   legend(label(2 "Model 3") label(4 "Model 4")) ///
>   ciopts(recast(rarea) fcolor(%20) lcolor(%0)) name(fig8, replace)
```



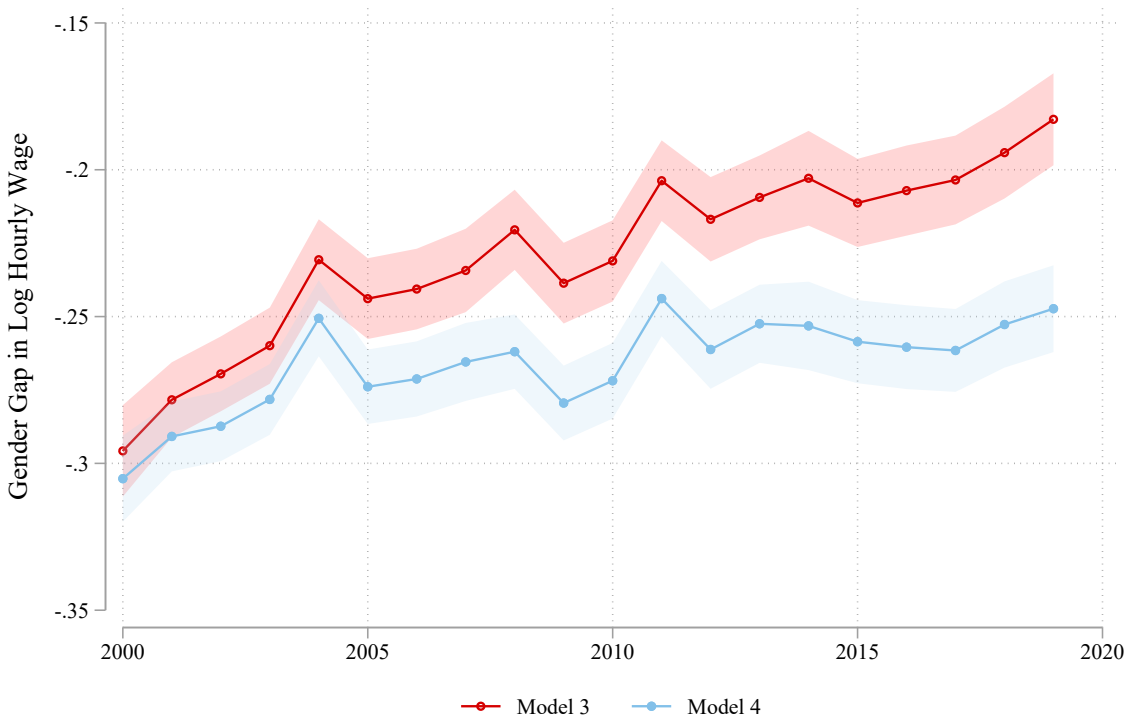
Scheme: cleanplots

```
. set scheme cleanplots
. coefplot (matrix(m3[, 2]), ci((m3[, 3] m3[, 4])) ) ///
>   (matrix(m4[, 2]), ci((m4[, 3] m4[, 4])) ), ///
>   at(matrix(m3[, 1])) ///
>   vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") ///
>   legend(label(2 "Model 3") label(4 "Model 4")) ///
>   ciopts(recast(rarea) fcolor(%20) lcolor(%0)) name(fig9, replace)
```



Scheme: cleanplots

```
. set scheme cleanplots
. coefplot (matrix(m3[, 2]), ci((m3[, 3] m3[, 4])) ) ///
>   (matrix(m4[, 2]), ci((m4[, 3] m4[, 4])) ), ///
>   at(matrix(m3[, 1])) ///
>   vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") ///
>   legend(label(2 "Model 3") label(4 "Model 4") pos(6) rows(1)) ///
>   ciopts(recast(rarea) fcolor(%20) lcolor(%0)) name(fig10, replace)
```



Saving Graphs

```
. forvalues i = 1(1)10 {
  2.         gr export "../table_figure/fig`i´.pdf", name(fig`i´) replace
  3. }
file ../table_figure/fig1.pdf saved as PDF format
file ../table_figure/fig2.pdf saved as PDF format
file ../table_figure/fig3.pdf saved as PDF format
file ../table_figure/fig4.pdf saved as PDF format
file ../table_figure/fig5.pdf saved as PDF format
file ../table_figure/fig6.pdf saved as PDF format
file ../table_figure/fig7.pdf saved as PDF format
file ../table_figure/fig8.pdf saved as PDF format
file ../table_figure/fig9.pdf saved as PDF format
file ../table_figure/fig10.pdf saved as PDF format
```

Put coefficient from multiple years into a single model

```
. capt prog drop appendmd
. prog appendmd, eclass
  1.          version 8
  2.          syntax namelist
  3.          tempname b V tmp
  4.          foreach name of local namelist {
  5.                  qui est restore `name´
  6.                  mat `b´ = nullmat(`b´) , e(b)
  7.                  mat `b´ = `b´[1,1..colsof(`b´)]
  8.                  mat `tmp´ = e(V)
  9.                  mat `tmp´ = `tmp´[1..rowsof(`tmp´),1..colsof(`tmp´)]
 10.                  capt confirm matrix `V´
 11.                  if _rc {
 12.                          mat `V´ = `tmp´
 13.                  }
 14.                  else {
 15.                          mat `V´ = ///
>                                  ( `V´ , J(rowsof(`V´),colsof(`tmp´),0) ) \ ///
>                                  ( J(rowsof(`tmp´),colsof(`V´),0) , `tmp´ )
 16.                  }
 17.          }
 18.          local names: colfullnames `b´
 19.          mat coln `V´ = `names´
 20.          mat rown `V´ = `names´
 21.          eret post `b´ `V´
 22.          eret local cmd "whatever"
 23. end
```

Put coefficient from multiple years into a single model

```
. levelsof year, local(yrs)
.
. foreach y in `yrs´ {
.         est res m3_y`y´
.
.         local fem_coln = colnumb(e(b), "1.fem")
.         local fem_rown = colnumb(e(V), "1.fem")
.
.         mat b = e(b)[1, `fem_coln´]
.         mat v = e(V)[`fem_rown´, `fem_coln´]
.
.         mat colnames b = "y`y´"
.         mat colnames v = "y`y´"
.         mat rownames v = "y`y´"
.
.         ereturn post b v
.         eststo y`y´
. }
```

- extract coefficient `1.fem` from models on each year and save it as estimate name y[year]

Put coefficient from multiple years into a single model

```
. levelsof year, local(yrs)
. local yrsn : word count `yrs´
.
. local mods "y2000"
. forvalues i = 2(1)`yrsn´ {
.         local y : word `i´ of `yrs´
.         local mods "`mods´ y`y´"
. }
.
. appendmd `mods´
. eststo m3_fem
```

- combine y[year] estimates with only `1.fem` into a single model

Plot the combined estimation

```
. coefplot m3_fem, ///
>         vert recast(connected) ytitle("Gender Gap in Log Hourly Wage") ///
>         xlabel(1 "2000" 6 "2005" 11 "2010" 16 "2015" 21 "2020", grid) ///
>         name(fig11, replace)
```