

irf cgraph — Combined graphs of IRFs, dynamic-multiplier functions, and FEVDs

Description
Options

Quick start
Remarks and examples

Menu
Stored results

Syntax
Also see

Description

`irf cgraph` makes a graph or a combined graph of IRF results. A graph is drawn for specified combinations of named IRF results, impulse variables, response variables, and statistics. `irf cgraph` combines these graphs into one image, unless separate graphs are requested.

`irf cgraph` operates on the active IRF file; see [\[TS\] irf set](#).

Quick start

Combine graphs of an orthogonalized IRF `myirf` and cumulative IRF `mycirf` for dependent variables `y1` and `y2`

```
irf cgraph (myirf y1 y2 oirf) (mycirf y1 y2 cirf)
```

Same as above, but suppress confidence bands and add a title

```
irf cgraph (myirf y1 y2 oirf) (mycirf y1 y2 cirf), noci ///
    title("My Title")
```

Note: `irf` commands can be used after `var`, `svar`, `vec`, `arma`, `arfima`, `dsge`, or `dsge1`; see [\[TS\] var](#), [\[TS\] var svar](#), [\[TS\] vec](#), [\[TS\] arma](#), [\[TS\] arfima](#), [\[DSGE\] dsge](#), or [\[DSGE\] dsge1](#).

Menu

Statistics > Postestimation

Syntax

```
irf cgraph (spec1) [ (spec2) ... [ (specN) ] ] [ , options ]
```

where (*spec*_{*k*}) is

```
(irfname impulsevar responsevar stat [ , spec_options ])
```

irfname is the name of a set of IRF results in the active IRF file. *impulsevar* should be specified as an endogenous variable for all statistics except *dm* and *cdm*; for those, specify as an exogenous variable. *responsevar* is an endogenous variable name. *stat* is one or more statistics from the list below:

<i>stat</i>	Description
Main	
<i>irf</i>	impulse–response function
<i>oirf</i>	orthogonalized impulse–response function
<i>dm</i>	dynamic-multiplier function
<i>cirf</i>	cumulative impulse–response function
<i>coirf</i>	cumulative orthogonalized impulse–response function
<i>cdm</i>	cumulative dynamic-multiplier function
<i>fevd</i>	Cholesky forecast-error variance decomposition
<i>sirf</i>	structural impulse–response function
<i>sfevd</i>	structural forecast-error variance decomposition

Notes: 1. No statistic may appear more than once.

2. If confidence intervals are included (the default), only two statistics may be included.

3. If confidence intervals are suppressed (option *nocl*), up to four statistics may be included.

<i>spec_options</i>	Description
Main	
<i>nocl</i>	suppress confidence bands
<i>level</i> (#)	set confidence level; default is <i>level</i> (95)
<i>lstep</i> (#)	use # for first step
<i>ustep</i> (#)	use # for maximum step
Plot	
<i>plot</i> # <i>opts</i> (<i>cline_options</i>)	affect rendition of the line plotting the # <i>stat</i>
CI plot	
<i>ci</i> # <i>opts</i> (<i>area_options</i>)	affect rendition of the confidence interval for the # <i>stat</i>

spec_options may be specified within a graph specification, globally, or in both. When specified in a graph specification, *spec_options* affect only the specification in which they are used. When supplied globally, *spec_options* affect all graph specifications. When supplied in both places, options in the graph specification take precedence.

<i>options</i>	Description
<code>set(filename)</code>	make <i>filename</i> active
Graph options	
<code>individual</code>	graph each combination individually
<code>combine_options</code>	affect appearance of combined graph
<code>graph_options</code>	any options other than by() documented in [G-3] <i>twoway_options</i>
collect is allowed; see [U] 11.1.10 Prefix commands.	

Options

`set(filename)` specifies the file to be made active; see [TS] `irf set`. If `set()` is not specified, the active file is used.

spec_options affect the rendition of the specification and include the following:

Main

`noci` suppresses graphing the confidence interval for each statistic. `noci` is assumed when the model was fit by `vec` because no confidence intervals were estimated.

`level(#)` specifies the default confidence level, as a percentage, for confidence intervals, when they are reported. The default is `level(95)` or as set by `set level`; see [U] 20.8 **Specifying the width of confidence intervals**. The value set of an overall `level()` can be overridden by the `level()` inside a (*spec_k*).

`lstep(#)` specifies the first step, or period, to be included in the graph. `lstep(0)` is the default.

`ustep(#)`, $\# \geq 1$, specifies the maximum step, or period, to be included in the graph.

Plot

`plot1opts(cline_options)`, ..., `plot4opts(cline_options)` affect the rendition of the plotted statistics. `plot1opts()` affects the rendition of the first statistic; `plot2opts()`, the second; and so on. *cline_options* are as described in [G-3] *cline_options*.

CI plot

`ci1opts1(area_options)` and `ci2opts2(area_options)` affect the rendition of the confidence intervals for the first (`ci1opts()`) and second (`ci2opts()`) statistics. See [TS] `irf graph` for a description of this option and [G-3] *area_options* for the suboptions that change the look of the CI.

Graph options

`individual` specifies that each graph be displayed individually. By default, `irf cgraph` combines the subgraphs into one image.

combine_options affect the appearance of the combined graph; see [G-2] **graph combine**.

graph_options are any of the options documented in [G-3] *twoway_options*, excluding `by()`. These include options for titling the graph (see [G-3] *title_options*) and for saving the graph to disk (see [G-3] *saving_option*).

Remarks and examples

If you have not read [TS] **irf**, please do so.

The relationship between **irf cgraph** and **irf graph** is syntactically and conceptually the same as that between **irf ctable** and **irf table**; see [TS] **irf ctable** for a description of the syntax.

irf cgraph is much the same as using **irf graph** to make individual graphs and then using **graph combine** to put them together. If you cannot use **irf cgraph** to do what you want, consider the other approach.

► Example 1

You have previously issued the commands:

```

. use https://www.stata-press.com/data/r18/lutkepohl2
. mat a = (., 0, 0\0,.,0\.,.,.)
. mat b = I(3)
. svar dln_inv dln_inc dln_consump, aeq(a) beq(b)
. irf create modela, set(results3) step(8)
. svar dln_inc dln_inv dln_consump, aeq(a) beq(b)
. irf create modelb, step(8)

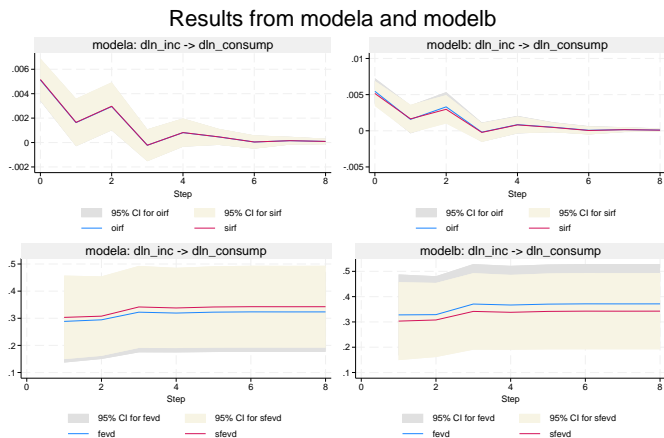
```

You now type

```

. irf cgraph (modela dln_inc dln_consump oirf sirf)
> (modelb dln_inc dln_consump oirf sirf)
> (modela dln_inc dln_consump fevd sfevd, lstep(1))
> (modelb dln_inc dln_consump fevd sfevd, lstep(1)),
> title("Results from modela and modelb")

```



Stored results

`irf cgraph` stores the following in `r()`:

Scalars

`r(k)` number of specific graph commands

Macros

`r(individual)` *individual*, if specified
`r(save)` *filename*, replace from `saving()` option for combined graph
`r(name)` *name*, replace from `name()` option for combined graph
`r(title)` title of the combined graph
`r(save#)` *filename*, replace from `saving()` option for individual graphs
`r(name#)` *name*, replace from `name()` option for individual graphs
`r(title#)` title for the #th graph
`r(ci#)` level applied to the #th confidence interval or `nocl`
`r(response#)` response specified in the #th command
`r(impulse#)` impulse specified in the #th command
`r(irfname#)` IRF name specified in the #th command
`r(stats#)` statistics specified in the #th command

Also see

[TS] [irf](#) — Create and analyze IRFs, dynamic-multiplier functions, and FEVDs

[TS] [var intro](#) — Introduction to vector autoregressive models

[TS] [vec intro](#) — Introduction to vector error-correction models

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

