Estimating impulse response functions with local projections in Stata 18

David Schenck

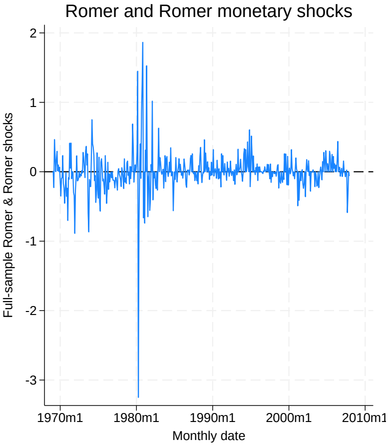
14 November 2023

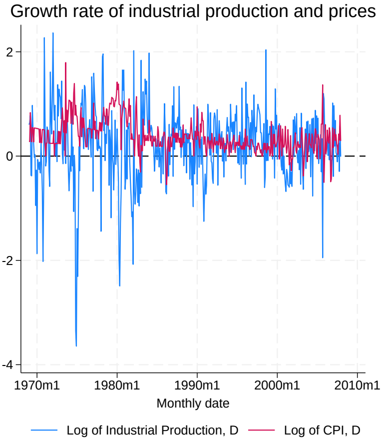
What are impulse response functions?

- Researchers are often interested in the effects of a shock, treatment, or intervention on an outcome.
- Vast literature on causal effects.
- In time-series analysis, we estimate causal effects over time.
- We often call causes *impulses*.
- We call the variables of interest *responses*.
- So, the time path of effects is an *impulse–response function*.
- Stata has many commands for estimating impulse–response functions.
- Today I introduce a new one: local projections, via `lpirf`.

Examples of impulse–response functions I

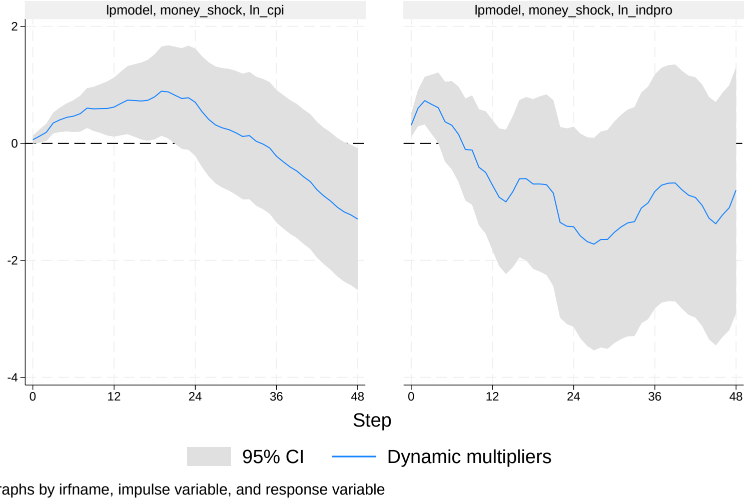
What is the effect of contractionary monetary policy on prices and output?





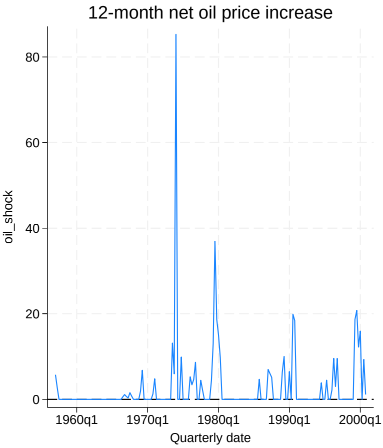
Examples of impulse–response functions I

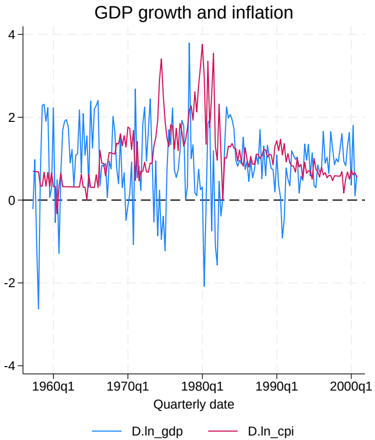
What is the effect of contractionary monetary policy on prices and output?



Examples of impulse–response functions II

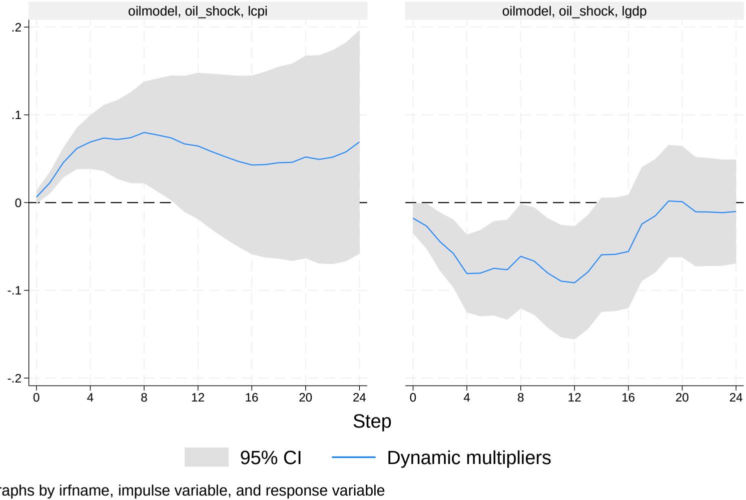
What is the effect of a rise in oil prices on consumer prices and output?





Examples of impulse–response functions II

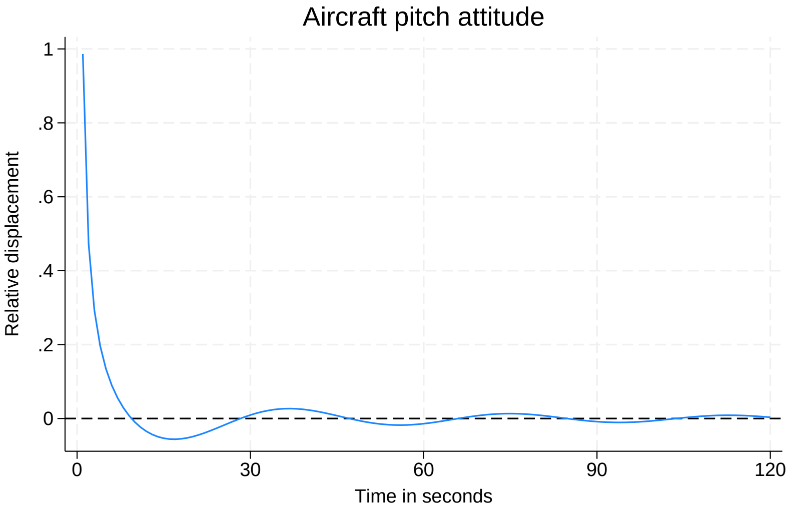
What is the effect of a rise in oil prices on consumer prices and output?



Examples of impulse–response functions III

Examples of impulse–response functions III

Examples of impulse–response functions III



Outline of the talk

1. Computing impulse response functions
2. Impulse responses in Stata
3. The local projection estimator with external shocks
4. The local projection estimator with internal shocks

Computing impulse–response functions

- Impulse response functions are often estimated in the context of a time–series model.
- For example, the ARIMA model:

$$y_t = \alpha_1 y_{t-1} + \cdots + \alpha_p y_{t-p} + u_t + \theta_1 u_{t-1} + \cdots + \theta_q u_{t-q}$$

  in which a variable y_t is modeled as a function of its lags $(y_{t-1}, \ldots, y_{t-p})$ and current and past values of a disturbance term u_t.
- For example, the vector autoregression model:

$$\mathbf{y}_t = \mathbf{B}_1 \mathbf{y}_{t-1} + \cdots + \mathbf{B}_p \mathbf{y}_{t-p} + \mathbf{D}\mathbf{x}_t + \mathbf{u}_t$$

  in which a collection of variables $\mathbf{y}_t$ is modeled as a function of its lags $(\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-p})$, exogenous variables $\mathbf{x}_t$, and disturbances $\mathbf{u}_t$.

Computing impulse–response functions

- After estimating the model, use the model structure to estimate the impulse response function
- Look at a model again:

$$\mathbf{y}_t = \mathbf{B}_1 \mathbf{y}_{t-1} + \cdots + \mathbf{B}_p \mathbf{y}_{t-p} + \mathbf{D}\mathbf{x}_t + \mathbf{u}_t$$

- We are conducting the experiment: what happens to the variables $\mathbf{y}_t$ after an exogenous increase $\mathbf{x}_t$ or an unexpected shock to $\mathbf{u}_t$?
- Perhaps combined with an identification scheme that decomposes $\mathbf{u}_t$ into objects more interpretable by the researcher.
- The impulse–response function is a function of model parameters.

Impulse–response functions in Stata

- Stata provides an `irf` suite of commands to estimate, manage, and display impulse–response functions.
- Impulse–response functions can be computed after many estimation commands with the `irf create` command.
- Results are stored in a file with the `irf set` command.
- Results can be displayed in graphical form with `irf graph` or in tabular form with `irf table`.

Impulse–response functions in Stata

- For example, after an ARIMA model

```
. arima y , ar(1/4) ma(1/4)
. irf create arimamodel
```

- For example, after a VAR model

```
. var y1 y2 y3, lag(1/4)
. irf create varmodel
```

- And now, after a local projection

```
. lpirf y1 y2 y3, lag(1/4) exog(x)
. irf create lpmodel
```

The local projection estimator

- When an exogenous shock is available, the local projection estimator consists of a collection of direct, multistep regressions of the variable of interest y on the shock x, perhaps with controls $\mathbf{z}$:

$$
\begin{aligned}
y_t &= \beta_0 x_t + \gamma' \mathbf{z} + u_t \\
y_{t+1} &= \beta_1 x_t + \gamma' \mathbf{z} + u_{t+1} \\
&\dots \\
y_{t+h} &= \beta_h x_t + \gamma' \mathbf{z} + u_{t+h}
\end{aligned}
$$

- The collection of responses $\{\beta_0, \beta_1, \dots, \beta_h\}$ are the dynamic response of y_t to x_t at each horizon $0, 1, \dots, h$.
- The impulse response function can be read directly off of the regression coefficients.

Local projections and `lpirf`

- New command `lpirf` estimates impulse responses by local projections.
- Allows for exogenous shocks as well as internal shocks.
- Estimates all response coefficients jointly, allowing for easy tests across responses at different horizons, and even for different variables.
- Impulse responses are the coefficients.

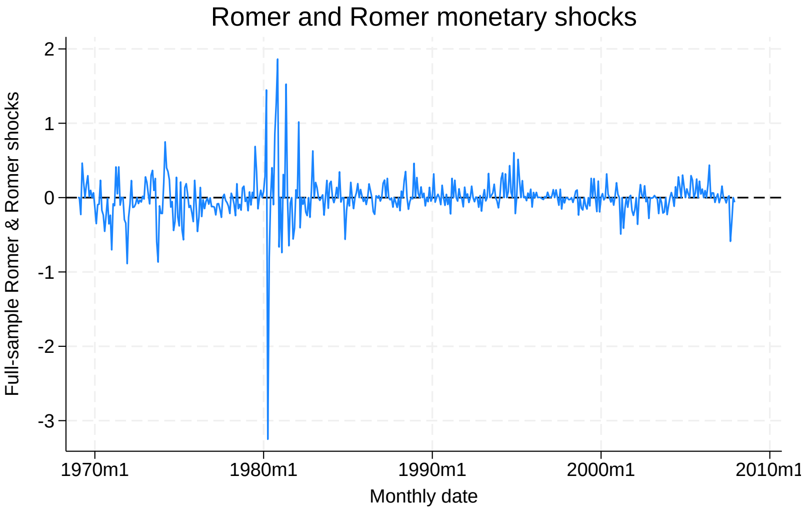
Let's see it work

- With an observed external variable: `lpirf y1 y2 y3, exog(x)`
- Without an observed external variable: `lpirf y1 y2 y3`
- Important options:
  - `exog()` allows for exogenous variables
  - `step()` specifies the maximum horizon
  - `lags()` allows for lags of the endogenous variables as controls
  - `vce()` allows for default, `robust`, and `hac` standard errors

Let's see it work: External variables

```
. use romer2004.dta

. describe

Contains data from romer2004.dta
 Observations:         1,247
    Variables:             4                  13 Nov 2023 21:46
--------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
--------------------------------------------------------------------------------
datem           float   %tm                   Monthly date
ln_indpro       float   %9.0g                 Log of Industrial Production
ln_cpi          float   %9.0g                 Log of CPI
money_shock     double  %9.0g                 Full-sample Romer & Romer shocks
--------------------------------------------------------------------------------

Sorted by: datem
```

Let's see it work: External variables



Let's see it work: External variables

```
. lpirf ln_indpro, lag(1/12) exog(L(0/12).money_shock)
Local-projection impulse-responses
Sample: 1970m1 thru 2007m12                         Number of obs        = 456
                                                    Number of impulses   =   2
                                                    Number of responses  =   1
                                                    Number of controls   =  23
```

	IRF coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
ln_indpro						
ln_indpro						
F1.	1.22147	.0466462	26.19	0.000	1.130045	1.312895
F2.	1.398847	.0737874	18.96	0.000	1.254227	1.543468
F3.	1.589508	.0993041	16.01	0.000	1.394876	1.784141
F4.	1.755352	.1249952	14.04	0.000	1.510366	2.000338
F5.	1.811884	.150196	12.06	0.000	1.517505	2.106263
F6.	1.853374	.1729293	10.72	0.000	1.514439	2.19231
F7.	1.829859	.1935085	9.46	0.000	1.450589	2.209129
F8.	1.857154	.2120404	8.76	0.000	1.441563	2.272746

Let's see it work: External variables

money_shock						
ln_indpro						
--.	.3099914	.1023604	3.03	0.002	.1093687	.5106141
F1.	.595736	.1619189	3.68	0.000	.2783808	.9130913
F2.	.7460005	.2179127	3.42	0.001	.3188995	1.173101
F3.	.681951	.2742891	2.49	0.013	.1443543	1.219548
F4.	.6119531	.3295898	1.86	0.063	-.0340311	1.257937
F5.	.359614	.3794756	0.95	0.343	-.3841444	1.103373
F6.	.2973099	.4246347	0.70	0.484	-.5349588	1.129579
F7.	.1294071	.465301	0.28	0.781	-.7825661	1.04138

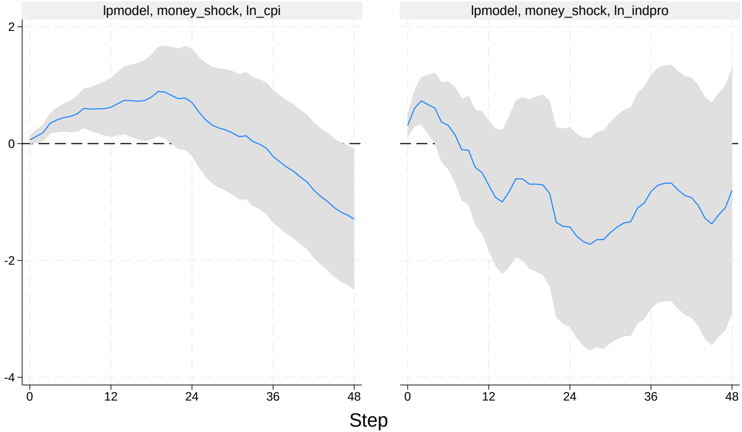
```
Note: IRF coefficients for exogenous variables are dynamic multipliers.
Impulses:  ln_indpro money_shock
Responses: ln_indpro
Controls:  L.money_shock L10.ln_indpro L10.money_shock L11.ln_indpro
           L11.money_shock L12.ln_indpro L12.money_shock L2.ln_indpro
           L2.money_shock L3.ln_indpro L3.money_shock L4.ln_indpro
           L4.money_shock L5.ln_indpro L5.money_shock L6.ln_indpro
           L6.money_shock L7.ln_indpro L7.money_shock L8.ln_indpro
           L8.money_shock L9.ln_indpro L9.money_shock
```

Digression: graphing and tabulating impulse responses

- Graph impulse responses with the `irf graph` *irftype* command.
- The *irftype* available differs across models.
- `lpirf` allows the types `irf` `oirf`, and `dm`.
    - `irf` is used for simple IRFs.
    - `oirf` is used for orthogonalized IRFs.
    - `dm` is used for the responses to exogenous variables.

Let's see it work: External variables

```
. quietly lpirf ln_indpro ln_cpi, lag(1/12) exog(L(0/12).money_shock) step(49)
. irf create lpmodel, replace

. irf graph dm, impulse(money_shock) yline(0) xlabel(0(12)48)
```



Let's see it work: External variables

- Do the responses of prices and output follow the same path?

```
. test [money_shock]ln_indpro = [money_shock]ln_cpi

 ( 1)  [money_shock]ln_indpro - [money_shock]ln_cpi = 0

           chi2(  1) =    5.40
         Prob > chi2 =    0.0202

. test [money_shock]f12.ln_indpro = [money_shock]f12.ln_cpi

 ( 1)  [money_shock]F12.ln_indpro - [money_shock]F12.ln_cpi = 0

           chi2(  1) =    4.85
         Prob > chi2 =    0.0276

. test [money_shock]f24.ln_indpro = [money_shock]f24.ln_cpi

 ( 1)  [money_shock]F24.ln_indpro - [money_shock]F24.ln_cpi = 0

           chi2(  1) =    4.16
         Prob > chi2 =    0.0414

. test [money_shock]f36.ln_indpro = [money_shock]f36.ln_cpi

 ( 1)  [money_shock]F36.ln_indpro - [money_shock]F36.ln_cpi = 0

           chi2(  1) =    0.22
         Prob > chi2 =    0.6388

. test [money_shock]f48.ln_indpro = [money_shock]f48.ln_cpi

 ( 1)  [money_shock]F48.ln_indpro - [money_shock]F48.ln_cpi = 0

           chi2(  1) =    0.14
         Prob > chi2 =    0.7121
```

Internal shocks

- Local projections can also be used if there is no exogenous variable
- We run

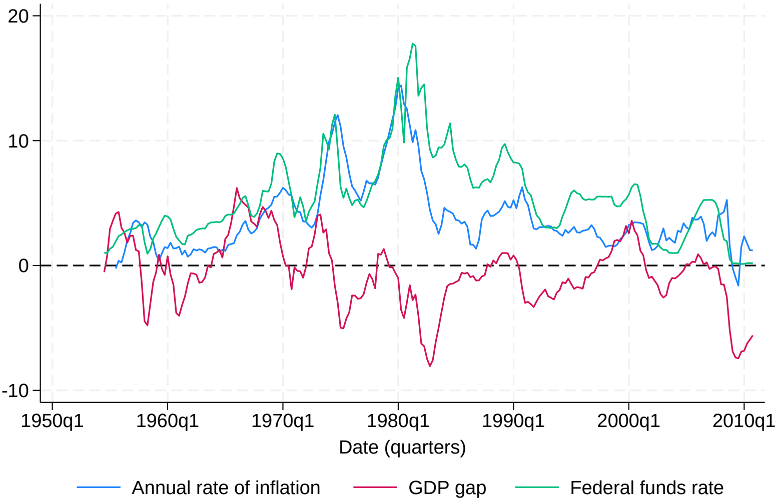
$$
\begin{aligned}
\mathbf{y}_t &= \mathbf{A}_1^1 \mathbf{y}_{t-1} + \cdots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{u}_t \\
\mathbf{y}_{t+1} &= \mathbf{A}_1^2 \mathbf{y}_{t-1} + \cdots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{u}_{t+1} \\
& \dots \\
\mathbf{y}_{t+h-1} &= \mathbf{A}_1^h \mathbf{y}_{t-1} + \cdots + \mathbf{A}_p \mathbf{y}_{t-p} + \mathbf{u}_{t+h-1}
\end{aligned}
$$

- The coefficient matrices $(\mathbf{A}_1^1, \dots, \mathbf{A}_1^h)$ are estimates of the simple impulse response functions to shocks to $\mathbf{u}_t$
- We can combine these coefficients with existing identification schemes, like orthgonalization.

Let's see it work: Internal shocks

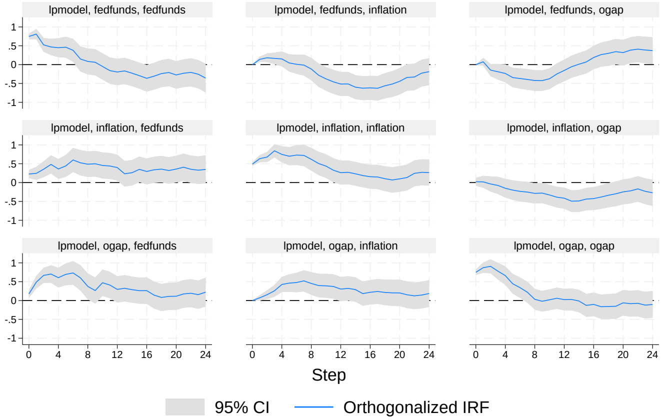
```
. webuse usmacro, clear
(Federal Reserve Economic Data - St. Louis Fed)

. describe

Contains data from https://www.stata-press.com/data/r18/usmacro.dta
 Observations:           226                  Federal Reserve Economic Data -
                                                St. Louis Fed
    Variables:             4                  4 Dec 2022 12:39
--------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
--------------------------------------------------------------------------------
fedfunds        double  %10.0g                Federal funds rate
date            int     %tq                   Date (quarters)
inflation       float   %9.0g                 Annual rate of inflation
ogap            float   %9.0g                 GDP gap
--------------------------------------------------------------------------------
Sorted by: date
```

Let's see it work: Internal shocks



Let's see it work: Internal shocks

```
. lpirf inflation ogap fedfunds, lag(1/4) step(24)
. irf create lpmodel, set(compare.irf)
. irf graph oirf
```

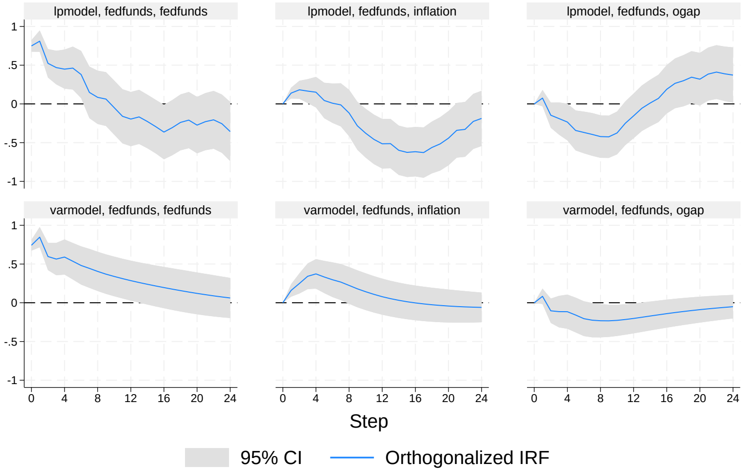


Graphs by irfname, impulse variable, and response variable

Let's see it work: Model comparisons

```
. quietly var inflation ogap fedfunds, lag(1/4)
. irf create varmodel, step(24)
(file compare.irf updated)
. irf graph oirf, impulse(fedfunds) yline(0) xlabel(0(4)24)
```

Let's see it work: Model comparisons



Graphs by irfname, impulse variable, and response variable

Conclusion

- Impulse responses and `irf`
- Using `lpirf` to estimate impulse–response functions
- Internal and external shocks, tests of coefficients, and model comparison

Thank you!