```
  ___  ____  ____  ____  ____(R)
 /__    /   ____/   /   ____/
___/   /   /___/   /   /___/
  Statistics/Data analysis
```

with our shock variable. We also need to use the explicit option to define which variable is our shock without any intera

display remote help file

Note: Some links below may not work because you have not yet chosen to install this package.

http://fmwww.bc.edu/RePEc/bocode/l/locproj.sthlp:

Help for **locproj**

Description

locproj estimates linear and non-linear Impulse Response Functions (IRF) based on the local projections methodology first
projections.

It generates temporary variables with the necessary transformations of the response variable in order to estimate the IRF
log-differences. For every option, the procedure also generates temporary variables with the corresponding transformatio
chosen transformation.

locproj allows the user to choose different estimation methods for both time series and panel data, including some instru

locproj reports the IRF, together with its standard error and confidence interval, as an output matrix and through an IRF

The options allow defining the desired specification in a fully automatic or in a more explicit way, with many alternative

If the user chooses the automatic specification, the syntax is very close to a typical regression command in Stata, with t
variable or its lagged terms, and only that one variable represents the shock.

Alternatively, the user can choose to explicitly define the shock variable (or variables), the number of lags of the shock
between the fully automatic or the fully explicit, depending on which option is easier or more convenient to use.

The explicit option is recommended when the shock should include more than one variable, for instance, an additional non-l

locproj uses the Stata command **lincom** to estimate the response to the shock variable or variables, allowing to estimate re
the use of marginal effects instead of regression coefficients, which is highly convenient when the response variable is b
margins, which could further facilitate the estimation of responses when the shock corresponds to an interaction of variab

locproj also allows different options regarding the horizon and the response starting period. For instance, it automatical
which case the response at *hor = 0* would be equal to 0 and an extra period will be added to final horizon period.

Remark: If the shock includes an interaction with a categorical variable, then we must use one of the options **lcs()** or

Syntax

Automatic Specification (Shock and Lags)

locproj *depvar* *shock* [*depvar lagged-terms*] [*shock lagged-terms*] [*controls*] [*if*] [*in*] **,** [**hor(***numlist integer***) lcs(**
noisily **stats** **saveirf** **irfname(***string***)** **fact(***real***)** **margins** **mrfvar(***varlist***)** **mrpredict(***string***)** **mropt(***string***)** **nogr**

Explicit Specification (Shock and Lags)

locproj *depvar* [*if*] [*in*] **,** [**hor(***numlist integer***)** **shock(***varlist***)** **controls(***varlist***)** **ylags(***integer***)** **slags(***integer***)** l
*integer***)** **noisily** **stats** **saveirf** **irfname(***string***)** **fact(***real***)** **margins** **mrfvar(***varlist***)** **mrpredict(***string***)** **mropt(***str*
]

See help lpgraph for using the post-estimation command **lpgraph** that allows plotting together results of different estimati

Options	Description

Model Specification:

hor(*numlist/integer*) Specifies the number of steps or horizon length for the IRF. It can be specified e
the horizon starts at period 0 and ends in period 6. The default horizon range is

shock(*varlist*) Allows to explicitly define the variable or variables that represent the shock or i
variable that is inmmediately after the *depvar* and its lagged terms if they are i
non-linear term or an interaction term.

lcs(*string*) Specifies an expression, usually an addition of variables, that defines a linear co
more than one variable and the name of one of them is not explictily included in
terms, e.g. *12.code#c.xvar*. The expression that should go inside the parenthesis

slags(*integer*) Specifies explicitly the number of lags of the shock variable or variables that sho
the first variable that represents the shock. If more than one variable is speci

ylags(*integer*) Specifies explicitly the number of lags of the *depvar* that should be included in th
defined by the user through the option **transf()**.

controls(*varlist*) Allows to explicitly define the variable or variables that represent the control va
variable(s) and its lagged terms if they are included in the main *varlist*. The c

fcontrols(*varlist*) Specifies any control variable(s) that should be included at the same horizon as th

lcopt(*string*) Specifies any option available in the command **lincom**. See lincom for specific help

Marginal Effect Options:

margins Specifies that the marginal effect of the shock variable is used instead of the reg
about the command **margins**

mrfvar(*varlist*) Specifies the factor or continuous variable that it is interacted with the shock va

mrpred(*string*) Specifies the option to be used with the predict command to produce the variable th
default option of the estimation method being used.

mropt(*string*) Allows to specify other options available in the command **margins** that have not beer

Transformation Options:

transf(*string*) Specifies the type of transformation that should be applied to the dependent variat
are the ones in the following list, and they should be written exactly as they ar

1. **(level)**: *Levels:* It keeps the dependent variable as originally specified and use
transformation is specified. When the option **ylags()** is specified, it includes

2. **(diff)**: *Differences:* It uses forecasts of the dependent variable in simple "dif
differences, i.e. *y(t)-y(t-l)* with *l = 1,...,ylags*.

3. **(cmlt)**: *Cumulative differences:* It uses forecasts of the dependent varible in cu
in differences, i.e. *y(t)-y(t-l)* with *l = 1,...,ylags*.

4. **(logs)**: *Logs:* It uses forecasts of the logarithm of the dependent varible, i.e.
ln(y(t-l)) with *l = 1,...,ylags*.

5. **(logs diff)**: *Log-differences:* It uses forecasts of the dependent variable in di
includes lags of the variable in log-differences, i.e. *ln(y(t))-ln(y(t-l))* with

6. **(logs cmlt)**: *Cumulative log-differences:* It uses forecasts of the dependent vari
specified, it includes lags of the variable in log-differences, i.e. *ln(y(t))-l*

Estimation Method:

<u>**m**</u>**et(***string***)** Specifies the estimation method. The default is **xtreg** when using panel data and **reg** instrumental variable commands **ivregress** and **xtivreg** and other IV methods with a following way: **met(ivregress** *estimator***)**, where *estimator* could be either on of 2

<u>**ins**</u>**tr(***varlist***)** Specifies the variables to use as instruments for the impulse (shock) variable when specification available options.

model_options Specifies any other estimation options specific to the method used and not defined to enter them alongside the rest of **locproj** options.

hopt(*string***)** Specifies any methodological option that depends directly on the horizon of the IRF

<u>**noi**</u>**sily** If this option is specified, the command displays a regression output for each one confidence bands.

<u>**st**</u>**ats** If this option is specified, the command displays a table with the summary statisti pseudo-R-squared, the F-statistic or Chi2-statistic, and the p-value (prob) of th

IRF Options:

conf(*numlist***)** Specifies one or (max) two confidence levels for calculating the confidence bands.

<u>**save**</u>**irf** If this option is specified, the IRF, its standard error and the confidence bands a to the new generated variables.

<u>**irfn**</u>**ame(***string***)** Specifies a name/prefix for the new IRF variable and the other new generated variab

<u>**f**</u>**act(***real***)** Specifies a factor for scaling the IRF, for instance, if the user wants to express

Graph Options:

nograph	If this option is specified a graph is not displayed.
zero	If this option is specified the graph includes a dashed line for the value 0.
title(*string*)	Specifies a title for the IRF graph.
lcolor(*string*)	Specifies a color for the IRF line and the confidence bands.
label(*string*)	Specifies a label for the IRF line in the IRF graph.
ttitle(*string*)	Specifies a name for the time axis in the IRF graph.
grname(*string*)	Specifies a graph name that could be used, for instance, when combining various gra
grsave(*string*)	Specifies a file name and path that should be used to save the IRF graph on the dis
as(*string*)	Specifies the desired file format of the saved graph.
gropt(*string*)	Specifies any other graph options not defined elsewhere.

Stored results

locproj stores the following in **e()**:

Matrices

e(irf)	A matrix including the Impulse Response Function (IRF), its standard error and its confidence inte
e(stats)	A matrix including the regression statistics by each step/horizon.

Name of Generated Variables if no name is defined:

_birf	estimated impulse response function (IRF).
_seirf	IRF's standar error
_irfup	IRF's upper confidence interval
_irflo	IRF's lower confidence interval
_irfup2	second IRF's upper confidence interval
_irflo2	second IRF's lower confidence interval

Name of Generated Variables if the name given to the IRF is "irfname":

irfname	estimated impulse response function (IRF).
irfname_se	IRF's standar error
irfname_lo	IRF's lower confidence interval
irfname_up	IRF's upper confidence interval
irfname_up2	second IRF's upper confidence interval
irfname_lo2	second IRF's lower confidence interval

Example 1. Use of locproj to replicate the IRFs in the "simple time series example: lp-example" do-file in Jordà website

(https://sites.google.com/site/oscarjorda/home/local-projections?pli=1)

```
. use AED_INTERESTRATES.dta
```

Example 1.1 Defining the basic specific options

What the automatic vs. explicit specification means is that the user can let **locproj** interpret in an automatic way which v
also which ones are just lags of each type of variable. Alternatively, in more complicated cases, the user can specify al

The simplest local projection specification in which the response variable is gs10 and the shock variable is gs1 would be

. locproj gs10 gs1

Which would also be equivalent to the following (explicit specification):

. locproj gs10, shock(gs1)

A more interesting specification would include lags of the dependent variable and of the shock, and would specify a respon
All the next six examples do those things and are exactly equivalent:

Automatic specification

. locproj gs10 l(0/4).gs1 l(1/3).gs10, hor(12)
. locproj l(0/3).gs10 l(0/4).gs1, hor(12)
. locproj gs10 l(1/3).gs10 l(0/4).gs1, hor(12)

Explicit or intermmediate specification

. locproj gs10, shock(gs1) ylags(3) slags(4) hor(12)
. locproj gs10 l(1/3).gs10, shock(gs1) slags(4) hor(12)
. locproj gs10 l(0/4).gs1, ylags(3) hor(12)

Example 1.2. A simple non-linear example

If we want to specify a very simple non-linear shock, for instance, a quadratic term of the variable **gs1**, we only need to

. gen gs1_2 = gs1^2

. locproj gs10, shock(gs1 gs1_2) ylags(3) slags(4) hor(12)

Alternatively we can include an additional interaction term of two continuous variables in the **shock()** option:

```
. locproj gs10, shock(gs1 c.gs1#c.gs1) ylags(3) slags(4) hor(12)
```

locproj will take all the variables that are defined in the **shock()** option and the resulting IRF will correspond to the addition of the individual effect of those variables, for instance, in this example, the IRF would be the addition of the coefficients of the variables **gs1** and **gs1_2**, or the variables **gs1** and **c.gs1#c.gs1**.

Example 1.3. Estimation method options

The Jordà example requires using Newey-West as the estimation method, which consequently requires specifying that the opti "lag" in the Newey-West command should depend on the horizon of the IRF in the following way:

```
. locproj gs10 l(0/4).gs1 l(1/3).gs10, h(12) met(newey) hopt(lag)
```

Example 1.4. Displaying all the regression outputs

If we want to take a look at the regression output for each one of the horizons of the IRF we can use the options noisily The regression outputs displayed are not the direct outputs from whatever estimation method we are using, but a simplifie The reason for this is that **locproj** uses temporary variables whose given names do not have any meaning and would be diff understand. **locproj** generates a new output table with variable names related to the variable list defined by the user.

```
. locproj gs10 l(0/4).gs1, h(12) m(newey) hopt(lag) yl(3) noisily
```

The **stats** option generates a table with each regression statistics for every horizon, i.e. number of observations, R-squa or pseudo-R-squared, F-statistic or Chi2-statistic and their respective p-values.

```
. locproj gs10 l(0/4).gs1, h(12) m(newey) hopt(lag) yl(3) stats
```

Example 1.5. Use of the transformation options

If we want to run the LP in differences (without using the already existing variables) we can use the option **transf()** toge
differentiating the shock variable gs1):

```
. locproj gs10 l(0/4).d.gs1, h(12) m(newey) hopt(lag) transf(diff) yl(3)
```

Alternatively, the model in differences using the already differentiated variables in the dataset without using the transf

```
. locproj dgs10 l(0/4).dgs1 l(1/3).dgs10, hor(12) met(newey) hopt(lag)
```

In the case of running the LP in cumulative differences we would have to use the option **transf(cmlt)**, again, together with

```
. locproj gs10 l(0/4).d.gs1, h(12) m(newey) hopt(lag) tr(cmlt) yl(3)
```

In the next examples we use the transformation in difference of logarithm and cumulative log-differences. In order to exp
to estimate the IRF of a shock of the variable **aaa** into the variable **baa**:

```
. gen lnaaa = ln(aaa)
```

log-differences:

```
. locproj baa l(0/4).lnaaa, h(12) m(newey) hopt(lag) transf(logs diff) yl(3) fact(100)
```

cumulative log-differences:

```
. locproj baa l(0/4).lnaaa, h(12) m(newey) hopt(lag) transf(logs cmlt) yl(3) fact(100)
```

<u>Example 1.6. Changing the confidence level or using more than one level</u>

By default the confidence level for the confidence bands is 95%. If we want to change it, we can use the **conf()** option
which admits a maximum of two levels and only admits integer values:

```
. locproj gs10 l(0/4).dgs1, h(12) m(newey) hopt(lag) tr(diff) yl(3) conf(90)
```

```
. locproj gs10 l(0/4).dgs1, h(12) m(newey) hopt(lag) tr(diff) yl(3) conf(66 99)
```

Example 1.7. Saving the IRF results into new variables

If we want to save the estimated IRF into a new variable that can be used later, we can use it through the options **saveir**
confidence bands. **locproj** uses some predetermined default names to save the corresponding variables (_irf, _seirf, _irf_

```
. locproj gs10 l(0/4).dgs1, h(12) m(newey) hopt(lag) tr(diff) yl(3) saveirf
```

```
. locproj gs10 l(0/4).dgs1, h(12) m(newey) hopt(lag) tr(diff) yl(3) save irfname(newirf)
```

Example 1.8. Some graph options

If we do not want **locproj** to produce a graph, we just have to type **nograph**:

```
. locproj gs10 l(0/4).dgs1, h(12) m(newey) hopt(lag) tr(diff) yl(3) nograph
```

In the following example we are going to produce a graph in which a dashed-line with the value of zero is included, we are
confidence interval to red instead of blue, and define the time axis as "Number of Days":

```
. locproj gs10 l(0/4).dgs1, h(12) m(newey) hopt(lag) tr(diff) yl(3) zero title("LP Example") label("1 Year Treasury I
```

Next, we are going to give the graph a name, we are going to save it in a folder in our disk as a png file named "example

```
. locproj gs10 l(0/4).dgs1, h(12) m(newey) hopt(lag) tr(diff) yl(3) zero title(LP Example) grname(Example1) grsave(C:
```

We can also add other graph options inside the **gropt()** option, for instance, we can define the labels of the y-axis and c

```
. locproj gs10 l(0/4).dgs1, h(12) m(newey) hopt(lag) tr(diff) yl(3) zero title(LP Example) gropt(graphregion(fcolor(w
```

We can also combine the results of different IRFs into a single graph using the post-estimation command **lpgraph**. See lpgr

Example 2. Use of locproj to replicate the IRFs in the "panel data example: lp example panel" do-file in Jordà website

```
. use "http://data.macrohistory.net/JST/JSTdatasetR5.dta"
```

We need to declare the panel data variables:

```
. xtset ifs year
```

For reproducing the first example in the Panel data example in Jordà website, we first need to generate some variables:

```
. gen lgdpr = 100*ln(rgdppc*pop)
. gen lcpi = 100*ln(cpi)
. gen dlgdpr = d.lgdpr
. gen dstir = d.stir
```

Throughout this example we assume that the real GDP (GDPR) responds with a lag to shocks in the short-term interest rate (
responds contemporaneously to shocks in GDPR and CPI.

Now, we can reproduce the response of real GDP to a 1pp shock to the real short-term interest rate. The next five examples
all of them, the estimation method is **xtreg**, they use the "fixed-effect" estimator and a cluster-robust covariance matrix
graph:

Using the transformation option transf() and the fully explicit options:

```
. locproj lgdpr, s(l.d.stir) c(l(1/3).d.lcpi) tr(diff) h(4) yl(3) sl(2) fe cluster(iso) z conf(90) gropt(ylabel(-0.4(0
```

Using the existing differentiated variables and a fully automatic specification:

```
. locproj l(0/3).dlgdpr l(1/3).dstir l(1/3).d.lcpi, fe cluster(iso) h(4) z conf(90) gropt(ylabel(-0.4(0.2)0.2) ytitle(
```

Using the transformation option transf() and the fully automatic options:

```
. locproj l(0/3).lgdpr l(1/3).d.stir l(1/3).d.lcpi, fe cluster(iso) tr(diff) h(4) z conf(90) gropt(ylabel(-0.4(0.2)0.2
```

Using the transformation option transf() and a combination of automatic and explicit options:

```
. locproj lgdpr l.d.stir l(1/3).d.lcpi, h(4) yl(3) sl(2) fe cluster(iso) tr(diff) z conf(90) gropt(ylabel(-0.4(0.2)0.2
```

In all the previous equivalent five examples, the shock variable is included with a lag and no contemporaneous term (L.D.s

Alternatively, in the next example, when estimating the response of CPI to the short-term interest rate, we just need to c
the control variable, which in this case is the GDP, from (0/3) to (1/3) in order to specify a Cholesky decomposition.

```
. locproj lcpi, s(l.d.stir) c(l(0/3).d.lgdpr) tr(diff) h(4) yl(3) sl(2) fe cluster(iso) z conf(90) gropt(ytitle(Percen
```

Finally, in the final exercise, the shock variable is the same as the dependent variable, so it is more convenient to use
the shock are the same at that horizon period.

```
. locproj d.stir l(0/3).d.lcpi l(0/3).d.lgdpr, s(d.stir) sl(3) h(4) fe cluster(iso) z conf(90) gropt(ytitle(Percent))
```

<u>Example 3. Use of locproj to replicate the IRFs in the "LPIV example: lpiv example" do-file in Jordà website</u>

For this example we use the databases RR_monetary_shock_quarterly.dta and lpiv_15Mar2022.dta from Jordà website.

```
. use RR_monetary_shock_quarterly.dta
. merge 1:1 date using lpiv_15Mar2022.dta, nogen
```

Next, we keep only nonmissing observations in resid_full i.e. 1969m1 - 2007m12

```
. keep if resid_full != .
```

We need to declare the time series variable

```
. tsset date
```

For estimating the IRF using OLS we need to take into consideration that in the example, the response horizon starts at *h*
instead of UNRATE. We also use Newey-West as the variance-covariance estimation method, which requires defining the option

```
. locproj f.UNRATE DFF l(1/4).DFF l(1/4).UNRATE, h(0/15) hopt(lag) m(newey) z
```

For replicating the instrumental variable IRF example, we use the option **met()** specifying that the method is *ivregress gmm*
for the shock, which in this case corresponds to the variable *resid_full*. Moreover, we define as an estimation method opti

```
. locproj f.UNRATE l(0/4).DFF l(1/4).UNRATE, h(0/15) met(ivregress gmm) instr(resid_full) vce(hac nwest) z
```

In this particular case, if we want to replicate the Jordà example using the **ylags()** option would be a bit tricky, since t
specify **ylags(***4***)** then we would actually be including lags **(0/3)** given the forecast of the dependent variable. Therefore, i
with the option **controls()**:

```
. locproj f.UNRATE l(1/4).UNRATE, s(DFF) sl(4) h(0/15) met(ivregress gmm) instr(resid_full) vce(hac nwest) z

. locproj f.UNRATE, s(DFF) sl(4) controls(l(1/4).UNRATE) h(0/15) met(ivregress gmm) instr(resid_full) vce(hac nwest) z
```

Example 4. Non-linear effects and interactions: Using the option lcs()

For Examples 4 and 5 we are going to use need the JST dataset and the "RecessionDummies" dataset that contains data on rec

```
. use "http://data.macrohistory.net/JST/JSTdatasetR5.dta"

. merge 1:1 year iso using "RecessionDummies.dta", nogen
```

We need to declare the panel data variables:

. xtset ifs year

We also need to drop WWI and WWII years from JST dataset:

. drop if year >=1914 & year <=1919
. drop if year >=1939 & year <=1947

In the "RecessionDummies" database, the variable **F** is a Financial Crisis dummy variable, while the variable **N** is a Normal
controling for the effect of normal recessions, and viceversa. Thus, we need to use the option **lcs()** since the effect we a

Therefore, in the following examples we include the expression "_cons + l.F" and "_cons + l.N" inside the option **lcs()**. No
that **locproj** evaluates is **only determined** by what it is defined by the option **lcs()**:

. locproj rgdppc l.F l.N, fe robust tr(logs cmlt) h(4) z f(100) lcs(_cons + l.F)

. locproj rgdppc l.F l.N, fe robust tr(logs cmlt) h(4) z f(100) lcs(_cons + l.N)

We can include more complicated interactions, for instance we can interact the effect of Financial Crisis (F) or Normal Re

We first estimate the mean and standard deviation of the Public Debt-to-GDP ratio:

. sum debtgdp
. sca dm=r(mean)
. sca dsd=r(sd)

In the command syntax we include the interaction of the public debt ratio with the financial crises and the normal recessi
multiplied by the mean and std. deviaton of the ratio **(dm+dsd)**:

. locproj rgdppc l.N l.F l.(N#c.debtgdp F#c.debtgdp), fe robust tr(logs cmlt) nograph f(100) lcs(_cons+l.F+1.l.F#c.l.d

In a similar way for normal recessions:

. locproj rgdppc l.N l.F l.(N#c.debtgdp F#c.debtgdp), fe robust tr(logs cmlt) nograph f(100) lcs(_cons+l.N+1.l.N#c.l.d

Example 5. Non-linear effects, interactions and binary dependent variable: Using the option margins

We need again the JST and the "RecessionDummies" datasets:

```
. use "http://data.macrohistory.net/JST/JSTdatasetR5.dta"
. merge 1:1 year iso using "RecessionDummies.dta", nogen
. xtset ifs year
```

We also need to drop WWI and WWII years from JST dataset:

```
. drop if year >=1914 & year <=1919
. drop if year >=1939 & year <=1947
```

In our first example, we will estimate the IRF of the probability of a banking crisis to an increase in the USA short-terr

We need to generate a new variable **stir_us** with the US interest rate as a common variable for all the countries in the sar

```
. gen stir_us0=stir if iso=="USA"
. egen stir_us=mean(stir_us0), by(year)
```

Now we are going to estimate the IRF using the option **margins**. The option **margins** estimates the marginal effect of a unit
estimation method the command **xtlogit** with fixed effects:

```
. locproj crisisJST l(0/2).stir_us, margins m(xtlogit) fe
```

We can also interact the shock variable with a dummy variable, for instance, whether a country has a "PEG" foreign exchang

The option margins allow us to estimate a separate IRF for each category of the dummy variable PEG. For doing that we need
with our shock variable. We also need to use the explicit option to define which variable is our shock without any intera

```
. locproj crisisJST peg#c.l(0/2).stir_us, s(stir_us) margins m(xtlogit) fe mrfvar(1.peg)
. locproj crisisJST peg#c.l(0/2).stir_us, s(stir_us) margins m(xtlogit) fe mrfvar(0.peg)
```

Alternatively, instead of entering the shock variable as **peg#c.l(0/2).stir_us** in the main syntax, we can enter the express
the need to specify it through the **locproj** option **shock()**:

```
. locproj crisisJST l(0/2).stir_us peg#c.l(0/2).stir_us, margins m(xtlogit) fe mrfvar(1.peg)
. locproj crisisJST l(0/2).stir_us peg#c.l(0/2).stir_us, margins m(xtlogit) fe mrfvar(0.peg)
```

References

"Jordà, Òscar. "Estimation and inference of impulse responses by local projections." American Economic Review 95, no. 1 (2
https://sites.google.com/site/oscarjorda/home/local-projections?pli=1

Author

Alfonso Ugarte-Ruiz
alfonso.ugarte@bbva.com
(remote file ends)

(click here to return to the previous screen)