Package ‘lpirfs’

July 6, 2023

Type Package
Title Local Projections Impulse Response Functions
Version 0.2.3
Maintainer Philipp Adämmer <philipp.adaemmer@uni-greifswald.de>
BugReports https://github.com/adaemmerp/lpirfs/issues
License GPL (>= 2)
Depends R (>= 3.1.2)
Encoding UTF-8
LazyData true
NeedsCompilation yes
Imports doParallel(>= 1.0.15), dplyr(>= 1.0.0), foreach(>= 1.5.0), ggplot2(>= 3.3.0), gridExtra(>= 2.3), lmtest(>= 0.9-36), plm(>= 2.2-3), sandwich(>= 2.5-1), Rcpp(>= 1.0.4.6)
RoxygenNote 7.2.3
LinkingTo Rcpp, RcppArmadillo
Suggests testthat (>= 3.0.0), readxl, knitr, ggpubr, rmarkdown, zoo, vars
VignetteBuilder knitr
Config/testthat/edition 3
Author Philipp Adämmer [aut, cre] (<https://orcid.org/0000-0003-3770-0097>), James P. LeSage [ctb], Mehmet Balcilar [ctb], Jon Danielsson [ctb]
Repository CRAN
Date/Publication 2023-07-06 07:50:03 UTC
**Description**


**Author(s)**

Philipp Adämmer
Description

A tibble, containing data to estimate fiscal multipliers. This data was originally used by Auerbach and Gorodnichenko (2012). Sarah and Zubairy (2018) use this data to re-evaluate their results with local projections.

Usage

ag_data

Format

A tibble with 248 quarterly observations (rows) and 7 variables (columns):

Year  Year of observation.
Quarter Quarter of observation.
Gov   Logs of real government (federal, state, and local) purchases (consumption and investment).
Tax   Logs of real government receipts of direct and indirect taxes net of transfers to businesses and individuals.
GDP   Logs of real gross domestic product.
GDP_MA 7-quarter moving average growth rate of GDP.
Gov_shock_mean Identified government spending shock. For details see Supplementary Appendix of Ramey and Zubairy (2018).

Sample: 1948:IV - 2008:IV

Source

https://www.journals.uchicago.edu/doi/10.1086/696277

References


get_robust_cov_panel  
Function to get robust covariance matrix for panel data

Description
Function to get robust covariance matrix for panel data

Usage
get_robust_cov_panel(panel_results, specs)

Arguments
panel_results  Plm object from estimation
specs  List with specifications

Value
Object with robust covariance matrix

hp_filter  
Decompose a times series via the Hodrick-Prescott filter

Description
Estimate cyclical and trend component with filter by Hodrick and Prescott (1997). The function is based on the function hpfilter from the archived mFilter-package.

Usage
hp_filter(x, lambda)

Arguments
x  One column matrix with numeric values.
lambda  Numeric value.

Value
A list. The first element contains the cyclical component and the second element the trend component.

Author(s)
Philipp Adämmer
interest_rules_var_data

References


Examples

```r
library(lpirfs)

# Decompose the Federal Funds Rate
data_set <- as.matrix(interest_rules_var_data$FF)
hp_results <- hp_filter(data_set, 1600)

# Extract results and save as data.framehp_cyc <- as.data.frame(hp_results[[1]])hp_trend <- as.data.frame(hp_results[[2]])

# Make data.frames for plots
cyc_df <- data.frame(yy = hp_cyc$V1, xx = seq(as.Date('1955-01-01'), as.Date('2003-01-01'), "quarter"))trend_df <- data.frame(yy = hp_trend$V1, xx = seq(as.Date('1955-01-01'), as.Date('2003-01-01'), "quarter"))

# Make plotslibrary(ggplot2)

# Plot cyclical partggplot(data = cyc_df) +geom_line(aes(y = yy, x = xx))

# Plot trend componentggplot(trend_df) +geom_line(aes(y = yy, x = xx))
```

---

**interest_rules_var_data**

*Data to estimate the effects of interest rate rules for monetary policy*

**Description**

A tibble, containing data to estimate the effects of interest rate rules for monetary policy. The data are used by Jordà (2005).
Usage

interest_rules_var_data

Format

A tibble with 193 quarterly observations (rows) and 3 variables (columns):

- **GDP_gap**  Percentage difference between real GDP and potential GDP (Congressional Budget Office).
- **Infl**  Inflation: Percentage change in the GDP, chain weighted price index at annual rate.
- **FF**  Federal funds rate: quarterly average of daily rates.


Source

https://www.aeaweb.org/articles?id=10.1257/0002828053828518

References


---

**lpirfs_obj-methods-base**

*Base methods for lpirfs_obj objects*

---

**Description**

Base methods for lpirfs_obj objects

---

**lp_lin**

*Compute linear impulse responses*

---

**Description**

lp_lin

Usage

lp_lin(
  endog_data, 
  lags_endog_lin = NULL, 
  lags_criterion = NaN, 
  max_lags = NaN, 
  trend = NULL, 
  shock_type = NULL, 
  confint = NULL, 
  use_nw = TRUE, 
  nw_lag = NULL, 
  nw_prewite = FALSE, 
  adjust_se = FALSE, 
  hor = NULL, 
  exog_data = NULL, 
  lags_exog = NULL, 
  contemp_data = NULL, 
  num_cores = 1
)

Arguments

endog_data A data.frame, containing the endogenous variables for the VAR. The Cholesky decomposition is based on the column order.
lags_endog_lin NaN or integer. NaN if lag length criterion is used. Integer for number of lags for endog_data.
lags_criterion NaN or character. NaN (default) means that the number of lags has to be given at lags_endog_lin. The character specifies the lag length criterion (’AICc’, ’AIC’ or ’BIC’).
max_lags NaN or integer. Maximum number of lags if lags_criterion is given. NaN (default) otherwise.
trend Integer. No trend = 0, include trend = 1, include trend and quadratic trend = 2.
shock_type Integer. Standard deviation shock = 0, unit shock = 1.
confint Double. Width of confidence bands. 68% = 1; 90% = 1.65; 95% = 1.96.
use_nw Boolean. Use Newey-West (1987) standard errors for impulse responses? TRUE (default) or FALSE.
nw_lag Integer. Specifies the maximum lag with positive weight for the Newey-West estimator. If set to NULL (default), the lag increases with with the number of horizon.
nw_prewite Boolean. Should the estimators be pre-whitened? TRUE or FALSE (default).
adjust_se Boolean. Should a finite sample adjustment be made to the covariance matrix estimators? TRUE or FALSE (default).
hor Integer. Number of horizons for impulse responses.
exog_data  A data.frame, containing exogenous variables for the VAR. The row length has to be the same as endog_data. Lag lengths for exogenous variables have to be given and will not be determined via a lag length criterion.

lags_exog  NULL or Integer. Integer for the number of lags for the exogenous data. The value cannot be 0. If you want to to include exogenous data with contemporaneous impact use contemp_data.

contemp_data  A data.frame, containing exogenous data with contemporaneous impact. This data will not be lagged. The row length has to be the same as endog_data.

num_cores  NULL or Integer. The number of cores to use for the estimation. If NULL, the function will use the maximum number of cores minus one.

Value

A list containing:

irf_lin_mean  A three 3D array, containing all impulse responses for all endogenous variables. The last dimension denotes the shock variable. The row in each matrix gives the responses of the \( i \)th variable, ordered as in endog_data. The columns denote the horizons. For example, if results_lin contains the list with results, results_lin$irf_lin_mean[, , 1] returns a KXH matrix, where K is the number of variables and H the number of horizons. '1' is the shock variable, corresponding to the first variable in endog_data.

irf_lin_low  A three 3D array containing all lower confidence bands of the responses, based on robust standard errors by Newey and West (1987). Properties are equal to irf_lin_mean.

irf_lin_up  A three 3D array containing all upper confidence bands of the responses, based on robust standard errors by Newey and West (1987). Properties are equal to irf_lin_mean.

diagnostic_list  A list OLS diagnostics. To see everything you can simply use summary() or results$diagnostic_list. The first entry the shock variable. The rows of each shown matrix then denotes the endogenous variable that reacts to the shock.

specs  A list with properties of endog_data for the plot function. It also contains lagged data (y_lin and x_lin) used for the irf estimations, and the selected lag lengths when an information criterion has been used.

Author(s)

Philipp Adämmer

References


**See Also**

[https://adaemmerp.github.io/lpirfs/README_docs.html](https://adaemmerp.github.io/lpirfs/README_docs.html)

**Examples**

```r
## Example without exogenous variables

# Load package
library(lpirfs)

# Load (endogenous) data
endog_data <- interest_rules_var_data

# Estimate linear model
results_lin <- lp_lin(endog_data,
    lags_endog_lin = 4,
    trend = 0,
    shock_type = 1,
    confint = 1.96,
    hor = 12)

# Show all impulse responses
# Compare with Figure 5 in Jordà (2005)
plot(results_lin)

# Make individual plots
linear_plots <- plot_lin(results_lin)

# Show single plots
# * The first element of 'linear_plots' shows the response of the first
#   variable (GDP_gap) to a shock in the first variable (GDP_gap).
# * The second element of 'linear_plots' shows the response of the first
#   variable (GDP_gap) to a shock in the second variable (inflation).
# * ...

linear_plots[[1]]
linear_plots[[2]]

# Show diagnostics. The first element correponds to the first shock variable.
summary(results_lin)
```
## Example with exogenous variables ##

# Load (endogenous) data
endog_data <- interest_rules_var_data

# Create exogenous data and data with contemporaneous impact (for illustration purposes only)
exog_data <- endog_data$GDP_gap*endog_data$Infl*endog_data$FF + rnorm(dim(endog_data)[1])
contemp_data <- endog_data$GDP_gap*endog_data$Infl*endog_data$FF + rnorm(dim(endog_data)[1])

# Exogenous data has to be a data.frame
exog_data <- data.frame(xx = exog_data)
contemp_data <- data.frame(cc = contemp_data)

# Estimate linear model
results_lin <- lp_lin(endog_data,
                       lags_endog_lin = 4,
                       trend = 0,
                       shock_type = 1,
                       confint = 1.96,
                       hor = 12,
                       exog_data = exog_data,
                       lags_exog = 4,
                       contemp_data = contemp_data)

# Show all impulse responses
plot(results_lin)

# Show diagnostics. The first element corresponds to the first shock variable.
summary(results_lin)

---

**lp_lin_iv**

*Compute linear impulse responses with identified shock and/or with 2SLS*

### Description###

Compute linear impulse responses with identified shock and/or with 2SLS.

### Usage###

```
lp_lin_iv(
  endog_data,
  shock = NULL,
  cumul_mult = FALSE,
  instr = NULL,
  use_twosls = FALSE,
  instrum = NULL,
  lags_endog_lin = NULL,
)```

Arguments

endog_data  A data.frame, containing the values of the dependent variable(s).

shock       A one column data.frame, including the variable to shock with. The row length has to be the same as endog_data. When use_twosl = TRUE, this variable will be approximated/regressed on the instrument variable(s) given in instrum.

cumul_mult  Boolean. Estimate cumulative multipliers? TRUE or FALSE (default). If TRUE, cumulative responses are estimated via:

\[ y(t+h) - y(t-1) \]

where \( h = 0, \ldots, H-1 \). This option is only available for lags_criterion = NaN.

instr       Deprecated input name. Use shock instead. See shock for details.

use_twosl   Boolean. Use two stage least squares? TRUE or FALSE (default).

instrum     A data.frame, containing the instrument(s) to use for 2SLS. This instrument will be used for the variable in shock.

lags_endog_lin  NaN or integer. NaN if lags are chosen by a lag length criterion. Integer for number of lags for endog_data.

exog_data   A data.frame, containing exogenous variables. The row length has to be the same as endog_data. Lag lengths for exogenous variables have to be given and will not be determined via a lag length criterion.

lags_exog   NULL or integer. Integer for the number of lags for the exogenous data. The value cannot be 0. If you want to to include exogenous data with contemporaneous impact use 'contemp_data'.

contemp_data   A data.frame, containing exogenous data with contemporaneous impact. The row length has to be the same as endog_data.

lags_criterion  NaN or character. NaN means that the number of lags will be given at lags_endog_lin. Possible lag length criteria are 'AICc', 'AIC' or 'BIC'. Note that when use_twosl = TRUE, the lag lengths are chosen based on normal OLS regressions, without using the instruments.
max_lags: NaN or integer. Maximum number of lags if lags_criterion is a character denoting the lag length criterion. NaN otherwise.

trend: Integer. No trend = 0, include trend = 1, include trend and quadratic trend = 2.

confint: Double. Width of confidence bands. 68% = 1; 90% = 1.65; 95% = 1.96.

use_nw: Boolean. Use Newey-West (1987) standard errors for impulse responses? TRUE (default) or FALSE.

nw_lag: Integer. Specifies the maximum lag with positive weight for the Newey-West estimator. If set to NULL (default), the lag increases with with the number of horizon.

nw_prewhtie: Boolean. Should the estimators be pre-whitened? TRUE of FALSE (default).

adjust_se: Boolean. Should a finite sample adjustment be made to the covariance matrix estimators? TRUE or FALSE (default).

hor: Integer. Number of horizons for impulse responses.

num_cores: NULL or Integer. The number of cores to use for the estimation. If NULL, the function will use the maximum number of cores minus one.

Value:

A list containing:

irf_lin_mean: A matrix, containing the impulse responses. The row in each matrix denotes the response of the ith variable to the shock. The columns are the horizons.


irf_lin_up: A matrix, containing all upper confidence bands of the impulse responses, based on robust standard errors by Newey and West (1987). Properties are equal to irf_lin_mean.

specs: A list with properties of endog_data for the plot function. It also contains lagged data (y_lin and x_lin) used for the estimations of the impulse responses, and the selected lag lengths when an information criterion has been used.

Author(s):

Philipp Adämmer

References:


See Also
https://adaemmerp.github.io/lpirfs/README_docs.html

Examples

```r
# This example replicates a result from the Supplementary Appendix
# by Ramey and Zubairy (2018) (RZ-18)

# Load data
ag_data <- ag_data
sample_start <- 7
sample_end <- dim(ag_data)[1]

# Endogenous data
endog_data <- ag_data[sample_start:sample_end,3:5]

# Variable to shock with. Here government spending due to
# Blanchard and Perotti (2002) framework
shock <- ag_data[sample_start:sample_end, 3]

# Estimate linear model
results_lin_iv <- lp_lin_iv(endog_data,
                          lags_endog_lin = 4,
                          shock = shock,
                          trend = 0,
                          confint = 1.96,
                          hor = 20)

# Show all impulse responses
plot(results_lin_iv)

# Make and save plots
iv_lin_plots <- plot_lin(results_lin_iv)
```
# The first element of 'iv_lin_plots' shows the response of the first variable (Gov) to the shock (Gov).
# The second element of 'iv_lin_plots' shows the response of the second variable (Tax) to the shock (Gov).
# ...

This plot replicates the left plot in the mid-panel of Figure 12 in the Supplementary Appendix by RZ-18.

iv_lin_plots[[1]]

Show diagnostics. The first element shows the reaction of the first endogenous variable.
summary(results_lin_iv)

## Add lags of the identified shock ##

Endogenous data but now exclude government spending
endog_data <- ag_data[sample_start:sample_end, 4:5]

Variable to shock with (government spending)
shock <- ag_data[sample_start:sample_end, 3]

Add the shock variable to exogenous data
exog_data <- shock

Estimate linear model with lagged shock variable
results_lin_iv <- lp_lin_iv(endog_data,
    lags_endog_lin = 4,
    shock = shock,
    exog_data = exog_data,
    lags_exog = 2,
    trend = 0,
    confint = 1.96,
    hor = 20)

Show all responses
plot(results_lin_iv)

Show diagnostics. The first element shows the reaction of the first endogenous variable.
summary(results_lin_iv)

Use 2SLS

Set seed
set.seed(007)

Load data
ag_data <- ag_data
sample_start <- 7
sample_end <- dim(ag_data)[1]

# Endogenous data
derg_data <- ag_data[sample_start:sample_end,3:5]

# Variable to shock with (government spending)
dshock <- ag_data[sample_start:sample_end, 3]

# Generate instrument variable that is correlated with government spending
dinstrum <- data.frame(0.9*shock$Gov + rnorm(length(shock$Gov), 0, 0.02) )

# Estimate linear model via 2SLS
dresults_lin_iv <- lp_lin_iv(endog_data,
  lags_endog_lin = 4,
  shock = shock,
  instrum = instrum,
  use_twosls = TRUE,
  trend = 0,
  confint = 1.96,
  hor = 20)

# Show all responses
dplot(results_lin_iv)

lp_lin_panel

Compute linear impulse responses with local projections for panel data

Description

This function estimates impulse responses with local projections for panel data, either with an identified shock or by an instrument variable approach.

Usage

lp_lin_panel(
  data_set = NULL,
  data_sample = "Full",
  endog_data = NULL,
  cumul_mult = TRUE,
  shock = NULL,
  diff_shock = TRUE,
  iv_reg = FALSE,
  instrum = NULL,
panel_model = "within",
panel_effect = "individual",
robust_cov = NULL,
robust_method = NULL,
robust_type = NULL,
robust_cluster = NULL,
robust_maxlag = NULL,
use_gmm = FALSE,
gmm_model = "onestep",
gmm_effect = "twoways",
gmm_transformation = "d",
c_exog_data = NULL,
l_exog_data = NULL,
lags_exog_data = NaN,
c_fd_exog_data = NULL,
l_fd_exog_data = NULL,
lags_fd_exog_data = NaN,
confint = NULL,
hor = NULL
)

Arguments

data_set A data.frame, containing the panel data set. The first column has to be the variable denoting the cross section. The second column has to be the variable denoting the time section.
data_sample Character or numeric. To use the full sample set value to "Full" (default). To estimate a subset, you have to provide a sequence of dates. This sequence has to be in the same format as the second column (time-section).
endog_data Character. The column name of the endogenous variable. You can only provide one endogenous variable at a time.
cumul_mult Boolean. Estimate cumulative multipliers? TRUE (default) or FALSE. If TRUE, cumulative responses are estimated via:

\[ y(t+h) - y(t-1), \]

where h = 0,..., H-1.

shock Character. The column name of the variable to shock with.
diff_shock Boolean. Take first differences of the shock variable? TRUE (default) or FALSE.
iv_reg Boolean. Use instrument variable approach? TRUE or FALSE.
instrum NULL or Character. The name(s) of the instrument variable(s) if iv_reg = TRUE.
panel_model Character. Type of panel model. The default is "within" (fixed effects). Other options are "random", "ht", "between", "pooling" or "fd". See vignette of the plm package for details.
panel_effect Character. The effects introduced in the model. Options are "individual" (default), "time", "twoways", or "nested". See the vignette of the plm-package for details.
robust_cov NULL or Character. The character specifies the method how to estimate robust standard errors: Options are "vcovBK", "vcovDC", "vcovG", "vcovHC", "vcovNW", "vcovSCC". For these options see vignette of plm package. Another option is "Vcxt". For details see Miller (2017) If "use_gmm = TRUE", this option has to be NULL.

robust_method NULL (default) or Character. The character is an option when robust_cov = "vcovHC". See vignette of the plm package for details.

robust_type NULL (default) or Character. The character is an option when robust_cov = "vcovBK", "vcovDC", "vcovHC", "vcovNW" or "vcovSCC". See vignette of the plm package for details.

robust_cluster NULL (default) or Character. The character is an option when robust_cov = "vcovBK", "vcovG" or "vcovHC". See vignette of the plm package for details.

robust_maxlag NULL (default) or Character. The character is an option when robust_cov = "vcovNW" or "vcovSCC". See vignette of the plm package for details.

use_gmm Boolean. Use GMM for estimation? TRUE or FALSE (default). See vignette of plm package for details. If TRUE, the option "robust_cov" has to be set to NULL.

gmm_model Character. Either "onestep" (default) or "twosteps". See vignette of the plm package for details.

gmm_effect Character. The effects introduced in the model: "twoways" (default) or "individual". See vignette of the plm-package for details.

gmm_transformation Character. Either "d" (default) for the "difference GMM" model or "ld" for the "system GMM". See vignette of the plm package for details.

c_exog_data NULL or Character. Name(s) of the exogenous variable(s) with contemporaneous impact.

l_exog_data NULL or Character. Name(s) of the exogenous variable(s) with lagged impact.

lags_exog_data Integer. Lag length for the exogenous variable(s) with lagged impact.

c_fd_exog_data NULL or Character. Name(s) of the exogenous variable(s) with contemporaneous impact of first differences.

l_fd_exog_data NULL or Character. Name(s) of exogenous variable(s) with lagged impact of first differences.

lags_fd_exog_data NaN or Integer. Number of lags for variable(s) with impact of first differences.

confint Double. Width of confidence bands. 68% = 1; 90% = 1.65; 95% = 1.96.

hor Integer. Number of horizons for impulse responses.

Value

A list containing:

irf_lin_mean A matrix, containing the impulse responses. The columns are the horizons.
irf_lin_low A matrix, containing all lower confidence bands. The columns are the horizons.
irf_lin_up A matrix, containing all upper confidence bands. The columns are the horizons.
reg_outputs  Full regression output (plm object) for each horizon.
reg_summaries  Summary of regression output for each horizon. In case of robust covariance
               estimators, this only includes the t-tests.
xy_data_sets  Data sets with endogenous and exogenous variables for each horizon.
specs  A list with data properties for e.g. the plot function.

Author(s)
Philipp Adämmer

References


Examples

```r
#--- Info
# This example is based on a STATA code that has been provided on
# Òscar Jordà's website (https://sites.google.com/site/oscarjorda/home/local-projections)
# It estimates impuse responses of the ratio of (mortgage lending/GDP) to a
# +1% change in the short term interest rate

#--- Get data
# Go to the website of the 'The MacroFinance and MacroHistory Lab'
# Download the Excel-Sheet of the 'Jordà-Schularick-Taylor Macrohistory Database':
# URL: https://www.macrohistory.net/database/
# Then uncomment and run the code below...

#--- Code

# Load libraries to download and read excel file from the website
# library(lpirfs)
# library(readxl)
# library(dplyr)

# Load JST Macrohistory Database
# jst_data <- read_excel("JSTdatasetR5.xlsx", sheet = "Data")

# Choose years <= 2013. Swap the first two columns so that 'country' is the
## Prepare variables

```r
# data_set <- jst_data %>%
#          mutate(stir = stir) %>%
#          mutate(mortgdp = 100*(tmort/gdp)) %>%
#          mutate(hpreal = hpnom/cpi) %>%
#          group_by(country) %>%
#          mutate(hpreal = hpreal/hpreal[year==1990][1]) %>%
#          mutate(lhpreal = log(hpreal)) %>%
#          mutate(lhpy = lhpreal - log(rgdppc)) %>%
#          mutate(lhpy = lhpy - lhpy[year == 1990][1]) %>%
#          mutate(lhpy = 100*lhpy) %>%
#          ungroup() %>%
#          mutate(lrgdp = 100*log(rgdppc)) %>%
#          mutate(lcpi = 100*log(cpi)) %>%
#          mutate(lriy = 100*log(iy*rgdppc)) %>%
#          mutate(cay = 100*(ca/gdp)) %>%
#          mutate(tnmort = tloans - tmort) %>%
#          mutate(nmortgdp = 100*(tnmort/gdp)) %>%
#          dplyr::select(country, year, mortgdp, stir, ltrate, lhpy, lrgdp, lcpi, lriy, cay, nmortgdp)
```

## Use data from 1870 to 2013 and exclude observations during WWI and WWII

```r
# data_sample <- seq(1870, 2013)[!(seq(1870, 2016) %in%
#                                   c(seq(1914, 1918), seq(1939, 1947)))]
```

## Estimate panel model

```r
# results_panel <- lp_lin_panel(data_set = data_set, data_sample = data_sample, endog_data = "mortgdp", cumul_mult = TRUE, shock = "stir", diff_shock = TRUE, panel_model = "within", panel_effect = "individual", robust_cov = "vcovSCC", c_exog_data = "cay", l_exog_data = "cay", lags_exog_data = 2, c_fd_exog_data = colnames(data_set)[c(seq(4,9),11)], l_fd_exog_data = colnames(data_set)[c(seq(3,9),11)], lags_fd_exog_data = 2,)
```
### Plot irfs

```r
# plot(results_panel)
```

### Simulate and add instrument to data_set

```r
# set.seed(123)
# data_set <- data_set %>%
#   group_by(country) %>%
#   mutate(instrument = 0.8*stir + rnorm(length(stir), 0, sd(na.omit(stir))/10)) %>%
# ungroup()
```

### Estimate panel model with iv approach

```r
# results_panel <- lp_lin_panel(data_set = data_set,
# data_sample = data_sample,
# endog_data = "mortgdp",
# cumul_mult = TRUE,
# shock = "stir",
# diff_shock = TRUE,
# iv_reg = TRUE,
# instrum = "instrument",
# panel_model = "within",
# panel_effect = "individual",
# robust_cov = "vcovSCC",
#
# c_exog_data = "cay",
# l_exog_data = "cay",
# lags_exog_data = 2,
# c_fd_exog_data = colnames(data_set)[c(seq(4,9),11)],
# l_fd_exog_data = colnames(data_set)[c(seq(3,9),11)],
# lags_fd_exog_data = 2,
#
# confint = 1.67,
# hor = 5)
```

### Create and plot irfs

```r
# plot(results_panel)
```

#### Use GMM

```r
### Use GMM
```

```r
Use a much smaller sample to have fewer T than N
```

```r
# data_sample <- seq(2000, 2012)
```

```r
# Estimate panel model with gmm
```

```r
# This example (please uncomment) gives a warning at each iteration.
```
## The data set is not well suited for GMM as GMM is based on N-asymptotics and the data set only contains 27 countries

```r
# results_panel <- lp_lin_panel(data_set = data_set, 
# data_sample = data_sample, 
# endog_data = "mortgdp", 
# cumul_mult = TRUE, 
# 
# shock = "stir", 
# diff_shock = TRUE, 
# 
# use_gmm = TRUE, 
# gmm_model = "onestep", 
# gmm_effect = "twoways", 
# gmm_transformation = "ld", 
# 
# l_exog_data = "mortgdp", 
# lags_exog_data = 2, 
# l_fd_exog_data = colnames(data_set)[c(4, 6)], 
# lags_fd_exog_data = 1, 
# 
# confint = 1.67, 
# hor = 5)
```

Create and plot irfs

```r
# plot(results_panel)
#```

---

**lp_nl**

*Compute nonlinear impulse responses*

### Description

Compute nonlinear impulse responses with local projections by Jordà (2005). The data can be separated into two states by a smooth transition function as applied in Auerbach and Gorodnichenko (2012), or by a simple dummy approach.

### Usage

```r
lp_nl(
  endog_data, 
  lags_endog_lin = NULL, 
  lags_endog_nl = NULL, 
  lags_criterion = NaN, 
  max_lags = NaN, 
  trend = NULL,
```
shock_type = NULL,
confint = NULL,
use_nw = TRUE,
nw_lag = NULL,
nw_prewhtie = FALSE,
adjust_se = FALSE,
hor = NULL,
switching = NULL,
lag_switching = TRUE,
use_logistic = TRUE,
use_hp = NULL,
lambda = NULL,
gamma = NULL,
exog_data = NULL,
lags_exog = NULL,
contemp_data = NULL,
num_cores = 1
)

Arguments

endog_data A data.frame, containing all endogenous variables for the VAR. The Cholesky
decomposition is based on the column order.

lags_endog_lin NaN or integer. NaN if lag length criterion is used. Integer for number of lags
for linear VAR to identify shock.

lags_endog_nl NaN or integer. Number of lags for nonlinear VAR. NaN if lag length criterion
is given.

lags_criterion NaN or character. NaN (default) means that the number of lags will be given
at lags_endog_nl and lags_endog_lin. The lag length criteria are ‘AICc’, ‘AIC’
and ‘BIC’.

max_lags NaN or integer. Maximum number of lags (if lags_criterion = ‘AICc’, ‘AIC’,
‘BIC’). NaN (default) otherwise.

trend Integer. Include no trend = 0 , include trend = 1, include trend and quadratic
trend = 2.

shock_type Integer. Standard deviation shock = 0, unit shock = 1.

confint Double. Width of confidence bands. 68% = 1; 90% = 1.65; 95% = 1.96.

(default) or FALSE.

nw_lag Integer. Specifies the maximum lag with positive weight for the Newey-West
estimator. If set to NULL (default), the lag increases with with the number of
horizon.

nw_prewhtie Boolean. Should the estimators be pre-whitened? TRUE of FALSE (default).

adjust_se Boolean. Should a finite sample adjsutment be made to the covariance matrix
estimators? TRUE or FALSE (default).

hor Integer. Number of horizons for impulse responses.
switching
Numeric vector. A column vector with the same length as endog_data. If 'use_logistic = TRUE', this series can either be decomposed via the Hodrick-Prescott filter (see Auerbach and Gorodnichenko, 2013) or directly plugged into the following logistic function:

\[ F_{zt} = \frac{\exp(-\gamma z_t)}{1 + \exp(-\gamma z_t)}. \]

Important: \( F_{zt} \) will be lagged by one and then multiplied with the data. If the variable shall not be lagged, use 'lag_switching = FALSE':
Regime 1 = \((1-F(z_{t-1}))^\times_y_{(t-p)}\),
Regime 2 = \(F(z_{t-1})^\times_y_{(t-p)}\).

lag_switching
Boolean. Use the first lag of the values of the transition function? TRUE (default) or FALSE.

use_logistic
Boolean. Use logistic function to separate states? TRUE (default) or FALSE. If FALSE, the values of the switching variable have to be binary (0/1).

use_hp
Boolean. Use HP-filter? TRUE or FALSE.

lambda
Double. Value of \( \lambda \) for the Hodrick-Prescott filter (if use_hp = TRUE).

gamma
Double. Positive number which is used in the transition function.

exog_data
A data.frame, containing exogenous variables for the VAR. The row length has to be the same as endog_data. Lag lengths for exogenous variables have to be given and will not be determined via a lag length criterion.

lags_exog
NULL or Integer. Integer for the number of lags for the exogenous data. The value cannot be 0. If you want to include exogenous data with contemporaneous impact use contemp_data.

contemp_data
A data.frame, containing exogenous data with contemporaneous impact. This data will not be lagged. The row length has to be the same as endog_data.

num_cores
Integer. The number of cores to use for the estimation. If NULL, the function will use the maximum number of cores minus one.

Value
A list containing:

irf_s1_mean
A three 3D array, containing all impulse responses for all endogenous variables of the first state. The last dimension denotes the shock variable. The row in each matrix denotes the responses of the \( i \)th variable, ordered as in endog_data. The columns are the horizons. For example, if the results are saved in results_nl, results_nl$sirf_s1_mean[,] returns a KXH matrix, where K is the number of variables and H the number of horizons. '1' is the shock variable, corresponding to the variable in the first column of endog_data.

irf_s1_low
A three 3D array, containing all lower confidence bands of the impulse responses, based on robust standard errors by Newey and West (1987). Properties are equal to irf_s1_mean.

irf_s1_up
A three 3D array, containing all upper confidence bands of the impulse responses, based on robust standard errors by Newey and West (1987). Properties are equal to irf_s1_mean.
irf_s2_mean A three 3D array, containing all impulse responses for all endogenous variables of the second state. The last dimension denotes the shock variable. The row in each matrix denotes the responses of the \( i \)th variable, ordered as in endog_data. The columns denote the horizon. For example, if the results are saved in results_nl, results_nl$irf_s2_mean[, , 1] returns a KXH matrix, where K is the number of variables and H the number of horizons. '1' is the first shock variable corresponding to the variable in the first column of endog_data.

irf_s2_low A three 3D array, containing all lower confidence bands of the responses, based on robust standard errors by Newey and West (1987). Properties are equal to irf_s2_mean.

irf_s2_up A three 3D array, containing all upper confidence bands of the responses, based on robust standard errors by Newey and West (1987). Properties are equal to irf_s2_mean.

specs A list with properties of endog_data for the plot function. It also contains lagged data (y_nl and x_nl) used for the irf estimations, and the selected lag lengths when an information criterion has been used.

fz A vector containing the values of the transition function \( F(z_{t-1}) \).

Author(s)
Philipp Adämmer

References


See Also

https://adaemmerp.github.io/lpirfs/README_docs.html
Examples

## Example without exogenous variables ##

# Load package
library(lpirfs)
library(gridExtra)
library(ggpubr)

# Load (endogenous) data
endog_data <- interest_rules_var_data

# Choose data for switching variable (here Federal Funds Rate)
# Important: The switching variable does not have to be used within the VAR!
switching_data <- endog_data$Infl

# Estimate model and save results
results_nl <- lp_nl(endog_data,
                     lags_endog_lin = 4,
                     lags_endog_nl = 3,
                     trend = 0,
                     shock_type = 1,
                     confint = 1.96,
                     hor = 24,
                     switching = switching_data,
                     use_hp = TRUE,
                     lambda = 1600,
                     gamma = 3)

# Show all plots
plot(results_nl)

# Make and save all plots
nl_plots <- plot_nl(results_nl)

# Save plots based on states
s1_plots <- sapply(nl_plots$gg_s1, ggplotGrob)
s2_plots <- sapply(nl_plots$gg_s2, ggplotGrob)

# Show first irf of each state
plot(s1_plots[[1]])
plot(s2_plots[[1]])

# Show diagnostics. The first element correponds to the first shock variable.
summary(results_nl)

## Example with exogenous variables ##

# Load (endogenous) data
endog_data <- interest_rules_var_data
# Choose data for switching variable (here Federal Funds Rate)
# Important: The switching variable does not have to be used within the VAR!
switching_data <- endog_data$FF

# Create exogenous data and data with contemporaneous impact (for illustration purposes only)
exog_data <- endog_data$GDP_gap*endog_data$Infl*endog_data$FF + rnorm(dim(endog_data)[1])
contemp_data <- endog_data$GDP_gap*endog_data$Infl*endog_data$FF + rnorm(dim(endog_data)[1])

# Exogenous data has to be a data.frame
exog_data <- data.frame(xx = exog_data)
contemp_data <- data.frame(cc = contemp_data)

# Estimate model and save results
results_nl <- lp_nl(endog_data,
                     lags_endog_lin = 4,
lags_endog_nl = 3,
trend = 0,
shock_type = 1,
confint = 1.96,
hor = 24,
switching = switching_data,
use_hp = TRUE,
lambda = 1600, # Ravn and Uhlig (2002):
            # Annual data = 6.25
            # Quarterly data = 1600
            # Monthly data = 129 600
gamma = 3,
exog_data = exog_data,
lags_exog = 3)

# Show all plots
plot(results_nl)

# Show diagnostics. The first element correponds to the first shock variable.
summary(results_nl)

---

**lp_nl Iv**

**Compute nonlinear impulse responses with identified shock**

**Description**

Compute nonlinear impulse responses with local projections and identified shock. The data can be separated into two states by a smooth transition function as applied in Auerbach and Gorodnichenko (2012), or by a simple dummy approach.
Usage

```
lp_nl_iv(
  endog_data,
  lags_endog_nl = NULL,
  shock = NULL,
  cumul_mult = FALSE,
  instr = NULL,
  exog_data = NULL,
  lags_exog = NULL,
  contemp_data = NULL,
  lags_criterion = NaN,
  max_lags = NaN,
  trend = NULL,
  confint = NULL,
  use_nw = TRUE,
  nw_lag = NULL,
  nw_prewight = FALSE,
  adjust_se = FALSE,
  hor = NULL,
  switching = NULL,
  lag_switching = TRUE,
  use_logistic = TRUE,
  use_hp = NULL,
  lambda = NULL,
  gamma = NULL,
  num_cores = 1
)
```

Arguments

- **endog_data**: A data.frame, containing all endogenous variables for the VAR.
- **lags_endog_nl**: NaN or integer. NaN if lags are chosen by a lag length criterion. Integer for number of lags for `endog_data`.
- **shock**: One column data.frame, including the instrument to shock with. The row length has to be the same as `endog_data`.
- **cumul_mult**: Boolean. Estimate cumulative multipliers? TRUE or FALSE (default). If TRUE, cumulative responses are estimated via:
  \[ y(t+h) - y(t-1), \]
  where \( h = 0,..., H-1 \). This option is only available for `lags_criterion = NaN`.
- **instr**: Deprecated input name. Use `shock` instead. See `shock` for details.
- **exog_data**: A data.frame, containing exogenous variables. The row length has to be the same as `endog_data`. Lag lengths for exogenous variables have to be given and will not be determined via a lag length criterion.
- **lags_exog**: NULL or Integer. Integer for the number of lags for the exogenous data. The value cannot be 0. If you want to to include exogenous data with contemporaneous impact use `contemp_data`. 
contemp_data A data.frame, containing exogenous data with contemporaneous impact. This data will not be lagged. The row length has to be the same as endog_data.

lags_criterion NaN or character. NaN means that the number of lags will be given at lags_endog_nl. Possible lag length criteria are 'AICc', 'AIC' or 'BIC'.

max_lags NaN or integer. Maximum number of lags (if lags_criterion = 'AICc', 'AIC', 'BIC'). NaN otherwise.

trend Integer. Include no trend = 0, include trend = 1, include trend and quadratic trend = 2.

confint Double. Width of confidence bands. 68% = 1; 90% = 1.65; 95% = 1.96.

use_nw Boolean. Use Newey-West (1987) standard errors for impulse responses? TRUE (default) or FALSE.

nw_lag Integer. Specifies the maximum lag with positive weight for the Newey-West estimator. If set to NULL (default), the lag increases with with the number of horizon.

nw_prewhite Boolean. Should the estimators be pre-whitened? TRUE of FALSE (default).

adjust_se Boolean. Should a finite sample adjustment be made to the covariance matrix estimators? TRUE or FALSE (default).

hor Integer. Number of horizons for impulse responses.

switching Numeric vector. A column vector with the same length as endog_data. This series can either be decomposed via the Hodrick-Prescott filter (see Auerbach and Gorodnichenko, 2013) or directly plugged into the following smooth transition function:

\[ F_{zt} = \frac{\exp(-\gamma z_{t-1})}{1 + \exp(-\gamma z_{t-1})}. \]

Warning: \( F_{zt} \) will be lagged by one and then multiplied with the data. If the variable shall not be lagged, the vector has to be given with a lead of one. The data for the two regimes are:

- Regime 1 = \((1-F(z_{t-1}))y_{(t-p)}\),
- Regime 2 = \(F(z_{t-1})y_{(t-p)}\).

lag_switching Boolean. Use the first lag of the values of the transition function? TRUE (default) or FALSE.

use_logistic Boolean. Use logistic function to separate states? TRUE (default) or FALSE. If FALSE, the values of the switching variable have to be binary (0/1).

use_hp Boolean. Use HP-filter? TRUE or FALSE.

lambda Double. Value of \( \lambda \) for the Hodrick-Prescott filter (if use_hp = TRUE).

gamma Double. Positive number which is used in the transition function.

num_cores Integer. The number of cores to use for the estimation. If NULL, the function will use the maximum number of cores minus one.

**Value**

A list containing:
A matrix, containing the impulse responses of the first regime. The row in each matrix denotes the responses of the \( i \)th variable to the shock. The columns are the horizons.

A matrix, containing all lower confidence bands of the impulse responses, based on robust standard errors by Newey and West (1987). Properties are equal to `irf_s1_mean`.

A matrix, containing all upper confidence bands of the impulse responses, based on robust standard errors by Newey and West (1987). Properties are equal to `irf_s1_mean`.

A matrix, containing all impulse responses for the second regime. The row in each matrix denotes the responses of the \( i \)th variable to the shock. The columns denote the horizon.

A matrix, containing all lower confidence bands of the responses, based on robust standard errors by Newey and West (1987). Properties are equal to `irf_s2_mean`.

A matrix, containing all upper confidence bands of the responses, based on robust standard errors by Newey and West (1987). Properties are equal to `irf_s2_mean`.

A list with properties of `endog_data` for the plot function. It also contains lagged data (y_nl and x_nl) used for the estimations of the impulse responses, and the selected lag lengths when an information criterion has been used.

A vector, containing the values of the transition function \( F(z_{t-1}) \).

**Author(s)**

Philipp Adämmer

**References**


See Also

https://adaemmerp.github.io/lpirfs/README_docs.html

Examples

```r
# This example replicates results from the Supplementary Appendix

# Load and prepare data
ag_data <- ag_data
capital_data <- capital_data
sample_start <- 7
sample_start <- dim(capital_data)[1]
endog_data <- capital_data[sample_start:sample_end, 3:5]

# The shock is estimated by RZ-18
shock <- capital_data[sample_start:sample_end, 7]

# Include four lags of the 7-quarter moving average growth rate of GDP
# as exogenous variables (see RZ-18)
exog_data <- capital_data[sample_start:sample_end, 6]

# Use the 7-quarter moving average growth rate of GDP as switching variable
# and adjust it to have sufficiently long recession periods.
switching_variable <- capital_data$GDP_MA[sample_start:sample_end] - 0.8

# Estimate local projections
results_nl_iv <- lp_nl_iv(endog_data,
                          lags_endog_nl = 3,
                          shock = shock,
                          exog_data = exog_data,
                          lags_exog = 4,
                          trend = 0,
                          confint = 1.96,
                          hor = 20,
                          switching = switching_variable,
                          use_hp = FALSE,
                          gamma = 3)

# Show all impulse responses
plot(results_nl_iv)

# Make and save individual plots
```

plots_nl_iv <- plot_nl(results_nl_iv)

# Show single impulse responses
# Compare with red line of left plot (lower panel) in Figure 12 in Supplementary Appendix of RZ-18.
plot(plots_nl_iv$gg_s1[[1]])
# Compare with blue line of left plot (lower panel) in Figure 12 in Supplementary Appendix of RZ-18.
plot(plots_nl_iv$gg_s2[[1]])

# Show diagnostics. The first element shows the reaction of the first endogenous variable.
summary(results_nl_iv)

**lp_nl_panel**

**Compute nonlinear impulse responses for panel data**

**Description**

This function estimates nonlinear impulse responses by using local projections for panel data with an identified shock. The data can be separated into two states by a smooth transition function as applied in Auerbach and Gorodnichenko (2012), or by a simple dummy approach.

**Usage**

```r
lp_nl_panel(
  data_set = NULL,
  data_sample = "Full",
  endog_data = NULL,
  cumul_mult = TRUE,
  shock = NULL,
  diff_shock = TRUE,
  panel_model = "within",
  panel_effect = "individual",
  robust_cov = NULL,
  robust_method = NULL,
  robust_type = NULL,
  robust_cluster = NULL,
  robust_maxlag = NULL,
  use_gmm = FALSE,
  gmm_model = "onestep",
  gmm_effect = "twoways",
  gmm_transformation = "d",
  c_exog_data = NULL,
  l_exog_data = NULL,
  lags_exog_data = NaN,
  c_fd_exog_data = NULL,
  l_fd_exog_data = NULL,
  lags_fd_exog_data = NaN,
)```

switching = NULL,
use_logistic = TRUE,
use_hp = FALSE,
lag_switching = TRUE,
lambda = NULL,
gamma = NULL,
confint = NULL,
hor = NULL
)

Arguments

data_set A data.frame, containing the panel data set. The first column has to be the variable denoting the cross section. The second column has to be the variable denoting the time section.
data_sample Character or numeric. To use the full sample set value to "Full" (default). To estimate a subset, you have to provide a sequence of dates. This sequence has to be in the same format as the second column (time-section).
dendog_data Character. The column name of the endogenous variable. You can only provide one endogenous variable at a time.
cumul_mult Boolean. Estimate cumulative multipliers? TRUE (default) or FALSE. If TRUE, cumulative responses are estimated via:
\[ y(t+h) - y(t-1), \]
where h = 0, ..., H-1.
shock Character. The column name of the variable to shock with.
diff_shock Boolean. Take first differences of the shock variable? TRUE (default) or FALSE.
panel_model Character. Type of panel model. The default is "within" (fixed effects). Other options are "random", "ht", "between", "pooling" or "fd". See vignette of the plm package for details.
panel_effect Character. The effects introduced in the model. Options are "individual" (default), "time", "twoways", or "nested". See the vignette of the plm-package for details.
robust_cov NULL or Character. The character specifies the method how to estimate robust standard errors: Options are "vcovBK", "vcovDC", "vcovG", "vcovHC", "vcovNW", "vcovSCC". For these options see vignette of plm package. Another option is "Vcxt". For details see Miller (2017) If "use_gmm = TRUE", this option has to be NULL.
robust_method NULL (default) or Character. The character is an option when robust_cov = "vcovHC". See vignette of the plm package for details.
robust_type NULL (default) or Character. The character is an option when robust_cov = "vcovBK", "vcovDC", "vcovHC", "vcovNW" or "vcovSCC". See vignette of the plm package for details.
robust_cluster NULL (default) or Character. The character is an option when robust_cov = "vcovBK", "vcovG" or "vcovHC". See vignette of the plm package for details.
robust_maxlag  NULL (default) or Character. The character is an option when robust_cov = "vcovNW" or "vcovSCC". See vignette of the plm package for details.

use_gmm  Boolean. Use GMM for estimation? TRUE or FALSE (default). See vignette of plm package for details. If TRUE, the option "robust_cov" has to be set to NULL.

gmm_model  Character. Either "onestep" (default) or "twosteps". See vignette of the plm package for details.

gmm_effect  Character. The effects introduced in the model: "twoways" (default) or "individual". See vignette of the plm-package for details.

gmm_transformation  Character. Either "d" (default) for the "difference GMM" model or "ld" for the "system GMM". See vignette of the plm package for details.

c_exog_data  NULL or Character. Name(s) of the exogenous variable(s) with contemporaneous impact.

l_exog_data  NULL or Character. Name(s) of the exogenous variable(s) with lagged impact.

lags_exog_data  Integer. Lag length for the exogenous variable(s) with lagged impact.

c_fd_exog_data  NULL or Character. Name(s) of the exogenous variable(s) with contemporaneous impact of first differences.

l_fd_exog_data  NULL or Character. Name(s) of exogenous variable(s) with lagged impact of first differences.

lags_fd_exog_data  NaN or Integer. Number of lags for variable(s) with impact of first differences.

switching  Character. Column name of the switching variable. If "use_logistic = TRUE", this series can either be decomposed by the Hodrick-Prescott filter (see Auerbach and Gorodnichenko, 2013) or directly plugged into the following smooth transition function:

\[ F_{zt} = \frac{exp(-\gamma z_{t})}{1 + exp(-\gamma z_{t})}. \]

The data for the two regimes are lagged by default:
Regime 1 = (1-F(z_{t-1}))*y_{(t-p)},
Regime 2 = F(z_{t-1})*y_{(t-p)}. This option can be suppressed with "lag_switching = FALSE".

use_logistic  Boolean. Use logistic function to separate states? TRUE (default) or FALSE. If FALSE, the values of the switching variable have to be binary (0/1).

use_hp  Boolean. Use HP-filter? TRUE or FALSE (default).

lag_switching  Boolean. Use the first lag of the values of the transition function? TRUE (default) or FALSE.

lambda  Double. Value of \( \lambda \) for the Hodrick-Prescott filter (if "use_hp = TRUE").

gamma  Double. Positive value for \( \gamma \), used in the transition function.

confint  Double. Width of confidence bands. 68% = 1; 90% = 1.65; 95% = 1.96.

hor  Integer. Number of horizons for impulse responses.
Value

A list containing:

- `irf_lin_mean`: A matrix, containing the impulse responses. The columns are the horizons.
- `irf_lin_low`: A matrix, containing all lower confidence bands. The columns are the horizons.
- `irf_lin_up`: A matrix, containing all upper confidence bands. The columns are the horizons.
- `reg_outputs`: Full regression output (plm object) for each horizon.
- `reg_summaries`: Summary of regression output for each horizon. In case of robust covariance estimators, this only includes the t-tests.
- `xy_data_sets`: Data sets with endogenous and exogenous variables for each horizon.
- `specs`: A list with data properties for e.g. the plot function.

Author(s)

Philipp Adämmer

References


Examples

```r
#--- Info
# This example is based on a STATA code that has been provided on
# Òscar Jordà's website (https://sites.google.com/site/oscarjorda/home/local-projections)
# It estimates impulse responses of the ratio of (mortgage lending/GDP) to a
# +1% change in the short term interest rate

#--- Get data
# Go to the website of the 'The MacroFinance and MacroHistory Lab'
# Download the Excel-Sheet of the 'Jordà–Schularick–Taylor Macrohistory Database':
# URL: https://www.macrohistory.net/database/
# Then uncomment and run the code below...

#--- Code

# Load libraries to download and read excel file from the website
```
# library(lpirfs)
# library(readxl)
# library(dplyr)

# Load JST Macrohistory Database
# jst_data <- read_excel("JSTdatasetR5.xlsx", sheet = "Data")

## Choose years <= 2013. Swap the first two columns so that 'country' is the
## first (cross section) and 'year' the second (time section) column
# jst_data <- jst_data %>%
#   dplyr::filter(year <= 2013) %>%
#   dplyr::select(country, year, everything())

## Prepare variables. This is based on the 'data.do' file
# data_set <- jst_data %>%
#   mutate(stir = stir) %>%
#   mutate(mortgdp = 100*(tmort/gdp)) %>%
#   mutate(hpreal = hpnom/cpi) %>%
#   group_by(country) %>%
#   mutate(hpreal = hpreal/hpreal[length==1990][1]) %>%
#   mutate(lhpreal = log(hpreal)) %>%
#   mutate(lhpy = lhpreal - log(rgdppc)) %>%
#   mutate(lhpy = lhpy[length == 1990][1]) %>%
#   mutate(lhpreal = 100*lhpreal) %>%
#   mutate(lhpy = 100*lhpy) %>%
#   ungroup() %>%
#   mutate(lrgdp = 100*log(rgdppc)) %>%
#   mutate(lcpi = 100*log(cpi)) %>%
#   mutate(lriy = 100*log(iy*rgdppc)) %>%
#   mutate(cay = 100*(ca/gdp)) %>%
#   mutate(tnmort = tloans - tmort) %>%
#   mutate(nmortgdp = 100*(tnmort/gdp)) %>%
#   dplyr::select(country, year, mortgdp, stir, ltrate, lhpy, lhpreal, lrgdp, lcpi, lriy, cay, nmortgdp)

## Use data_sample from 1870 to 2013 and exclude observations from WWI and WWII
# data_sample <- seq(1870, 2016)[!(seq(1870, 2016) %in%
# c(seq(1914, 1918),
# seq(1939, 1947)))]

## Estimate panel model
# results_panel <- lp_nl_panel(data_set = data_set,
#   data_sample = data_sample,
#   endog_data = "mortgdp",
#   cumul_mult = TRUE,
#   shock = "stir",
#   diff_shock = TRUE,
#   panel_model = "within",
#   panel_effect = "individual",
#   panel = FALSE,
#   demean = FALSE,
#   trend = FALSE)
# robust_cov = "vcovSCC",
#
# switching = "lrgdp",
# lag_switching = TRUE,
# use_hp = TRUE,
# lambda = 6.25,
# gamma = 10,
#
# c_exog_data = "cay",
# c_fd_exog_data = colnames(data_set)[c(seq(4,9),11)],
# l_fd_exog_data = colnames(data_set)[c(seq(3,9),11)],
# lags_fd_exog_data = 2,
#
# confint = 1.67,
# hor = 5)
#
## Plot irfs
# plot(results_panel)
#
## Plot values of the transition function for USA between 1950 and 2016
# library(ggplot2)
#
# data_set %>%
# mutate(fz = results_panel$fz$fz) %>%
# select(country, year, fz) %>%
# filter(country == "USA" & year > 1950 & year <= 2016) %>%
# ggplot()+
# geom_line(aes(x = year, y = fz)) +
# scale_x_continuous(breaks = seq(1950, 2016, 5))
#
#
##############################################################################
### Use GMM ###
##############################################################################
#
## Use a much smaller sample to have fewer T than N
# data_sample <- seq(2000, 2012)
#
#
## Estimate panel model with gmm
## This example (please uncomment) gives a warning at each iteration.
## The data set is not well suited for
## GMM as GMM is based on N-asymptotics and the data set only contains 27 countries
#
# results_panel <- lp_nl_panel(data_set = data_set,
# data_sample = data_sample,
# endog_data = "mortgdp",
# cumul_mult = TRUE,
#
# shock = "stir",
# diff_shock = TRUE,
monetary_var_data

Data to estimate a standard monetary VAR

Description

A tibble, containing data to estimate a standard monetary VAR.

Usage

monetary_var_data

Format

A tibble with 494 monthly observations (rows) and 6 variables (columns):

EM Log of non-agricultural payroll employment.
P Log of personal consumption expenditures deflator (1996 = 100).
POCM Annual growth rate of the index of sensitive materials prices issued by the Conference Board.
FF Federal funds rate.
NBRX Ratio of nonborrowed reserves plus extended credit to total reserves.
M2 Annual growth rate of M2 stock.

plot.lpirfs_lin_obj

Base print() function to plot all impulse responses from linear lpirfs object

Description

Base print() function to plot all impulse responses from linear lpirfs object

Usage

```r
## S3 method for class 'lpirfs_lin_iv_obj'
plot(x, ...)
```

Arguments

- **x**: An object of type `lpirfs_lin_obj`
- **...**: Additional arguments to be consistent with S3 print() function
plot.lpirfs_lin_panel_obj

*Base print() function to plot all impulse responses from linear lpirfs object*

---

**Description**

Base print() function to plot all impulse responses from linear lpirfs object

**Usage**

```r
## S3 method for class 'lpirfs_lin_panel_obj'
plot(x, ...)
```

**Arguments**

- `x`: An object of type 'lpirfs_lin_panel_obj'
- `...`: Additional arguments to be consistent with S3 print() function

---

plot.lpirfs_nl_iv_obj

*Base print() function to plot all impulse responses from linear lpirfs object*

---

**Description**

Base print() function to plot all impulse responses from linear lpirfs object

**Usage**

```r
## S3 method for class 'lpirfs_nl_iv_obj'
plot(x, ...)
```

**Arguments**

- `x`: An object of type 'lpirfs_nl_iv_obj'
- `...`: Additional arguments to be consistent with S3 print() function
plot.lpirfs_nl_obj
Base print() function to plot all impulse responses from nonlinear lpirfs object

Description
Base print() function to plot all impulse responses from nonlinear lpirfs object

Usage
## S3 method for class 'lpirfs_nl_obj'
plot(x, ...)

Arguments
x An object of type 'lpirfs_nl_obj'
... Additional arguments to be consistent with S3 print() function

plot.lpirfs_nl_panel_obj
Base print() function to plot all impulse responses from linear lpirfs object

Description
Base print() function to plot all impulse responses from linear lpirfs object

Usage
## S3 method for class 'lpirfs_nl_panel_obj'
plot(x, ...)

Arguments
x An object of type 'lpirfs_lin_panel_obj'
... Additional arguments to be consistent with S3 print() function
plot_lin

Compute and display plots of linear impulse responses

Description

Compute and display linear impulse responses, estimated with \texttt{lp_lin()} and \texttt{lp_lin_iv()}. 

Usage

\texttt{plot_lin(results_lin)}

Arguments

\texttt{results_lin} \hspace{1em} A list created with \texttt{lp_lin()} or \texttt{lp_lin_iv()}. 

Value

A list with (gg-)plots for linear impulse responses.

Author(s)

Philipp Adämmer

Examples

\begin{verbatim}
# See examples for \texttt{lp_lin()} and \texttt{lp_lin_iv()}. 
\end{verbatim}

plot_nl

Compute and display plots of nonlinear impulse responses

Description

Compute and display (nonlinear) impulse responses, estimated with \texttt{lp_nl()} and \texttt{lp_nl_iv()}. 

Usage

\texttt{plot_nl(results_nl)}

Arguments

\texttt{results_nl} \hspace{1em} A list created with \texttt{lp_nl()} or \texttt{lp_nl_iv()}. 

Value

A list with (gg-)plots for nonlinear impulse responses.

Author(s)

Philipp Adämmer

Examples

# Load package

# See examples for lp_nl() and lp_nl_iv().

summary.lpirfs_lin_iv_obj

Summary for nonlinear lpirfs object

Description

Summary for nonlinear lpirfs object

Usage

## S3 method for class 'lpirfs_lin_iv_obj'
summary(object, ...)

Arguments

object An object of type 'lpirfs_lin_iv_obj'
... Additional arguments to be consistent with S3 print() function

summary.lpirfs_lin_obj

Summary for linear lpirfs object

Description

Summary for linear lpirfs object

Usage

## S3 method for class 'lpirfs_lin_obj'
summary(object, ...)

# Load package

# See examples for lp_nl() and lp_nl_iv().
### summary.lpirfs_lin_panel_obj

#### Description
Summary for nonlinear lpirfs object

#### Usage
```r
## S3 method for class 'lpirfs_lin_panel_obj'
summary(object, ...)
```

#### Arguments
- `object`: An object of type 'lpirfs_lin_obj'
- `...`: Additional arguments to be consistent with S3 print() function

### summary.lpirfs_nl_iv_obj

#### Description
Summary for nonlinear lpirfs object

#### Usage
```r
## S3 method for class 'lpirfs_nl_iv_obj'
summary(object, ...)
```

#### Arguments
- `object`: An object of type 'lpirfs_nl_iv_obj'
- `...`: Additional arguments to be consistent with S3 print() function
**summary.lpirfs_nl_obj**  
*Summary for nonlinear lpirfs object*

**Description**

Summary for nonlinear lpirfs object

**Usage**

```r
## S3 method for class 'lpirfs_nl_obj'
summary(object, ...)
```

**Arguments**

- `object`  
  An object of type `lpirfs_nl_obj`
- `...`  
  Additional arguments to be consistent with S3 print() function

**summary.lpirfs_nl_panel_obj**  
*Summary for nonlinear lpirfs object*

**Description**

Summary for nonlinear lpirfs object

**Usage**

```r
## S3 method for class 'lpirfs_nl_panel_obj'
summary(object, ...)
```

**Arguments**

- `object`  
  An object of type `lpirfs_lin_panel_obj`
- `...`  
  Additional arguments to be consistent with S3 print() function
# Index

## datasets
- **ag_data**, 3
- **interest_rules_var_data**, 5
- **monetary_var_data**, 37

* **ag_data**, 3
  * array, 8, 23, 24

* **data.frame**, 7, 8, 11, 16, 22, 23, 27, 28, 32

* **get_robust_cov_panel**, 4

* **hp_filter**, 4

* **interest_rules_var_data**, 5

* **list**, 41

* **lp_lin**, 6, 41

* **lp_lin_iv**, 10, 41

* **lp_lin_panel**, 15

* **lp_nl**, 21, 41

* **lp_nl_iv**, 26, 41

* **lp_nl_panel**, 31

* **lpirfs-package**, 2

* **lpirfs_obj-methods-base**, 6

* **matrix**, 12, 17, 29, 34

* **monetary_var_data**, 37

* **plot.lpirfs_lin_iv_obj**, 38

* **plot.lpirfs_lin_obj**, 38

* **plot.lpirfs_lin_panel_obj**, 39

* **plot.lpirfs_nl_iv_obj**, 39

* **plot.lpirfs_nl_obj**, 40

* **plot.lpirfs_nl_panel_obj**, 40

* **plot_lin**, 41

* **plot_nl**, 41

* summary.lpirfs_lin_iv_obj, 42

* summary.lpirfs_nl_obj, 44

* summary.lpirfs_nl_panel_obj, 44

* tibble, 3, 6, 37

45