Package ‘lpirfs’

April 11, 2019

Type Package

Title Local Projections Impulse Response Functions

Version 0.1.5

Maintainer Philipp Adämmer <adaemmer@hsu-hh.de>

BugReports https://github.com/adaemmerp/lpirfs/issues

Description Provides functions to estimate and plot linear as well as nonlinear impulse responses based on local projections by Jordà (2005) <doi:10.1257/0002828053828518>.

License GPL (>= 2)

Depends R (>= 3.1.2)

Encoding UTF-8

LazyData true

NeedsCompilation yes

Imports doParallel(>= 1.0.11), dplyr(>= 0.7.4), foreach(>= 1.4.3), ggplot2(>= 2.2.1), lmtest (>= 0.9-36), plm(>= 1.6-6), sandwich(>= 2.5-0), Rcpp(>= 0.12.17)

RoxygenNote 6.1.1

LinkingTo Rcpp, RcppArmadillo

Suggests testthat, ggpubr, gridExtra, httr, readxl

Author Philipp Adämmer [aut, cre] (<https://orcid.org/0000-0003-3770-0097>), James P. LeSage [ctb], Mehmet Balcilar [ctb]

Repository CRAN

Date/Publication 2019-04-11 12:52:44 UTC

R topics documented:

`lpirfs-package` *Local Projection Impulse Response Functions*

Description

lpirfs provides functions to estimate and plot linear as well as nonlinear impulse responses based on local projections by Jordà (2005) <doi:10.1257/0002828053828518>. Nonlinear impulse responses are estimated for two regimes which can be separated by a smooth transition function as applied in Auerbach and Gorodnichenko (2012) <doi:10.1257/pol.4.2.1>, or by a simple dummy approach.

Author(s)

Philipp Adämmer

`ag_data` *Data to estimate fiscal multipliers*

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

Auerbach, A. J., and Gorodnichenko Y. (2012). "Measuring the Output Responses to Fiscal Policy." *American Economic Journal: Economic Policy*, 4 (2): 1-27.

Jordà, Ò. (2005) "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

Ramey, V.A., Zubairy, S. (2018). "Government Spending Multipliers in Good Times and in Bad: Evidence from US Historical Data." *Journal of Political Economy*, 126(2): 850 - 901.

---

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

References

Hodrick, R.J., and Prescott, E. C. (1997). "Postwar U.S. Business Cycles: An Empirical Investigation." *Journal of Money, Credit and Banking*, 29(1), 1-16.

Ravn, M.O., Uhlig, H. (2002). "On Adjusting the Hodrick-Prescott Filter for the Frequency of Observations." *Review of Economics and Statistics*, 84(2), 371-376.

Examples

```
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.frame
 hp_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 plots
 library(ggplot2)

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

# Plot trend component
 ggplot(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.

Sample: 1955:I - 2003:I

Source

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

References

Jordà, Ò. (2005) "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

`lp_lin` *Compute linear impulse responses*

Description

Compute linear impulse responses with local projections by Jordà (2005).

Usage

```
lp_lin(endog_data, lags_endog_lin = NULL, lags_criterion = NaN,
  max_lags = NaN, trend = NULL, shock_type = NULL, confint = NULL,
  hor = NULL, exog_data = NULL, lags_exog = NULL,
  contemp_data = NULL, num_cores = NULL)
```

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.

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 no be determined via a lag length criterion.

lags_exog
: Integer. Number of lags for the exogenous variables.

contemp_data
: A data.frame, containing exogenous data with contemporaneous impact. 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 *ith* 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*.

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.

Author(s)

Philipp Adämmer

References

Akaike, H. (1974). "A new look at the statistical model identification", *IEEE Transactions on Automatic Control*, 19 (6): 716–723.

Hurvich, C. M., and Tsai, C.-L. (1989), "Regression and time series model selection in small samples", *Biometrika*, 76(2): 297–307

Jordà, Ò. (2005). "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

Newey, W.K., and West, K.D. (1987). "A Simple, Positive-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix." *Econometrica*, 55: 703–708.

Schwarz, Gideon E. (1978). "Estimating the dimension of a model", *Annals of Statistics*, 6 (2): 461–464.

See Also

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

Examples

```
          ## 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)

# Make 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 all plots by using 'ggpubr' and 'gridExtra'
# lpirfs does not depend on those packages so they have to be installed
 library(ggpubr)
 library(gridExtra)

# Compare with Figure 5 in Jordà (2005)
 lin_plots_all <- sapply(linear_plots, ggplotGrob)
 marrangeGrob(lin_plots_all, nrow = ncol(endog_data), ncol = ncol(endog_data), top = NULL)


                      ## 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)

# Make plots
 linear_plots <- plot_lin(results_lin)

# Show all plots
 library(ggpubr)
 library(gridExtra)

 lin_plots_all <- sapply(linear_plots, ggplotGrob)
 marrangeGrob(lin_plots_all, nrow = ncol(endog_data), ncol = ncol(endog_data), top = NULL)
```

`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, instr = NULL, use_twosls = FALSE,
  instrum = NULL, lags_endog_lin = NULL, exog_data = NULL,
  lags_exog = NULL, contemp_data = NULL, lags_criterion = NaN,
  max_lags = NaN, trend = NULL, confint = NULL, hor = NULL,
  num_cores = 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_twosls = TRUE*, this variable will be approximated/regressed on the instrument variable(s) given in *instrum*.
`instr`	Deprecated input name. Use *shock* instead. See *shock* for details.
`use_twosls`	Boolean. Use two stage least squares? TRUE or FALSE.
`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 no be determined via a lag length criterion.
`lags_exog`	NULL or Integer. Integer for the number of lags for the exogenous 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_twosls = 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.
`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_low` 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_lin_mean*.

`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.

Author(s)

Philipp Adämmer

References

Akaike, H. (1974). "A new look at the statistical model identification", *IEEE Transactions on Automatic Control*, 19 (6): 716–723.

Auerbach, A. J., and Gorodnichenko, Y. (2012). "Measuring the Output Responses to Fiscal Policy." *American Economic Journal: Economic Policy*, 4 (2): 1-27.

Blanchard, O., and Perotti, R. (2002). "An Empirical Characterization of the Dynamic Effects of Changes in Government Spending and Taxes on Output." *Quarterly Journal of Economics*, 117(4): 1329–1368.

Hurvich, C. M., and Tsai, C.-L. (1989), "Regression and time series model selection in small samples", *Biometrika*, 76(2): 297–307

Jordà, Ò. (2005). "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

Jordà, Ò, Schularick, M., Taylor, A.M. (2015), "Betting the house", *Journal of International Economics*, 96, S2-S18.

Newey, W.K., and West, K.D. (1987). "A Simple, Positive-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix." *Econometrica*, 55: 703–708.

Ramey, V.A., and Zubairy, S. (2018). "Government Spending Multipliers in Good Times and in Bad: Evidence from US Historical Data." *Journal of Political Economy*, 126(2): 850 - 901.

Schwarz, Gideon E. (1978). "Estimating the dimension of a model", *Annals of Statistics*, 6 (2): 461–464.

See Also

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

Examples

```
# 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)


# 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 all impulse responses by using 'ggpubr' and 'gridExtra'
# lpirfs does not depend on those packages so they have to be installed
 library(ggpubr)
 library(gridExtra)

 lin_plots_all <- sapply(iv_lin_plots, ggplotGrob)
 marrangeGrob(lin_plots_all, nrow = ncol(endog_data), ncol = 1, top = NULL)


## 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)


# Make and save plots
 iv_lin_plots    <- plot_lin(results_lin_iv)
 lin_plots_all   <- sapply(iv_lin_plots, ggplotGrob)
 marrangeGrob(lin_plots_all, nrow = ncol(endog_data), ncol = 1, top = NULL)


##############################################################################
#####                         Use 2SLS                              #########
##############################################################################

# Set seed
 set.seed(007)

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

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

# Estimate linear model via 2SLS
 results_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)

# Create all plots
 iv_lin_plots    <- plot_lin(results_lin_iv)


# Show response of GDP
 iv_lin_plots[[3]]
```

`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, 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 "Vw" (white), "Vcx" (clustered by group and arellano method), "Vcx" (clustered by time and arrellano method), "Vctx" (clustered by group and time). For details see Miller (2017). The other options are "vcovBK", "vcovDC", "vcovG", "vcovHC", "vcovNW", "vcovSCC". For these options see vignette of plm package. If "use_gmm = TRUE", this option has to be NULL.
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_summaries	Regression output for each horizon.
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

Croissant, Y., Millo, G. (2008). "Panel Data Econometrics in R: The plm Package." *Journal of Statistical Software*, 27(2), 1-43. doi: 10.18637/jss.v027.i02 (URL: http://doi.org/10.18637/jss.v027.i02).

Jordà, Ò. (2005). "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

Jordà, Ò., Schualrick, M., Taylor, A.M. (2018). "Large and State-Dependent Effects of Quasi-Random Monetary Experiments", *NBER* working paper 23074, *FRBSF* working paper 2017-02.

Millo G (2017). "Robust Standard Error Estimators for Panel Models: A Unifying Approach." *Journal of Statistical Software*, 82(3), 1-27. doi: 10.18637/jss.v082.i03 (URL: http://doi.org/10.18637/jss.v082.i03).

Examples

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

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

# Retrieve the JST Macrohistory Database
 url_jst <-"http://www.macrohistory.net/JST/JSTdatasetR3.xlsx"
 GET(url_jst, write_disk(jst_link <- tempfile(fileext = ".xlsx")))
 jst_data <- read_excel(jst_link, 2L)

# 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[year==1990][1]) %>%
              mutate(lhpreal  = log(hpreal))                  %>%

              mutate(lhpy     = lhpreal - log(rgdppc))        %>%
              mutate(lhpy     = lhpy - lhpy[year == 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, lrgdp, lcpi, lriy, cay, nmortgdp)


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

# Estimate panel model
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,
```

```
                              confint           = 1.67,
                              hor               = 5)

# Create and plot irfs
 plot_lin_panel <- plot_lin(results_panel)
 plot(plot_lin_panel[[1]])


# Simulate and add instrument to data_set
 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
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
 plot_lin_panel <- plot_lin(results_panel)
 plot(plot_lin_panel[[1]])


##############################################################################
###                             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 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_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
plot_lin_panel <- plot_lin(results_panel)
plot(plot_lin_panel[[1]])
```

---

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

```
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, 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 = NULL)
```

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.
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: $$F_{z_t} = \frac{exp(-\gamma z_t)}{1 + exp(-\gamma z_t)}.$$ Important: F_{z_t} 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-$F(z_{t-1})$)*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 λ 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 no be determined via a lag length criterion.
lags_exog	Integer. Number of lags for the exogenous variables.

`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 *ith* variable, ordered as in *endog_data*. The columns are the horizons. For example, if the results are saved in *results_nl*, results_nl$irf_s1_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 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 *ith* 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.

`fz` A vector containing the values of the transition function F(z_t-1).

Author(s)

Philipp Adämmer

References

Akaike, H. (1974). "A new look at the statistical model identification", *IEEE Transactions on Automatic Control*, 19 (6): 716–723.

Auerbach, A. J., and Gorodnichenko Y. (2012). "Measuring the Output Responses to Fiscal Policy." *American Economic Journal: Economic Policy*, 4 (2): 1-27.

Auerbach, A. J., and Gorodnichenko Y. (2013). "Fiscal Multipliers in Recession and Expansion." *NBER Working Paper Series*. Nr. 17447.

Hurvich, C. M., and Tsai, C.-L. (1989), "Regression and time series model selection in small samples", *Biometrika*, 76(2): 297–307

Jordà, Ò. (2005) "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

Newey, W.K., and West, K.D. (1987). "A Simple, Positive-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix." *Econometrica*, 55, 703–708.

Schwarz, Gideon E. (1978). "Estimating the dimension of a model", *Annals of Statistics*, 6 (2): 461–464.

Ravn, M.O., Uhlig, H. (2002). "On Adjusting the Hodrick-Prescott Filter for the Frequency of Observations." *Review of Economics and Statistics*, 84(2), 371-376.

See Also

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

Examples

```
                 ## Example without exogenous variables ##

# Load package
  library(lpirfs)

# 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)

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

```
# Show all impulse responses by using 'ggpubr' and 'gridExtra'
# lpirfs does not depend on those packages so they have to be installed
  library(ggpubr)
  library(gridExtra)

# 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 all plots
  marrangeGrob(s1_plots, nrow = ncol(endog_data), ncol = ncol(endog_data), top = NULL)
  marrangeGrob(s2_plots, nrow = ncol(endog_data), ncol = ncol(endog_data), top = NULL)


                     ## 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):
                                                  # Anuual data    = 6.25
                                                  # Quarterly data = 1600
                                                  # Monthly data   = 129 600
                          gamma           = 3,
                          exog_data       = exog_data,
                          lags_exog       = 3)
```

```
# 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 all plots
  marrangeGrob(s1_plots, nrow = ncol(endog_data), ncol = ncol(endog_data), top = NULL)
  marrangeGrob(s2_plots, nrow = ncol(endog_data), ncol = ncol(endog_data), top = NULL)
```

`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,
  instr = NULL, exog_data = NULL, lags_exog = NULL,
  contemp_data = NULL, lags_criterion = NaN, max_lags = NaN,
  trend = NULL, confint = NULL, hor = NULL, switching = NULL,
  lag_switching = TRUE, use_logistic = TRUE, use_hp = NULL,
  lambda = NULL, gamma = NULL, num_cores = NULL)
```

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*.
`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 no be determined via a lag length criterion.
`lags_exog`	NULL or Integer. Integer for the number of lags for the exogenous 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.

`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:

$$F_{z_t} = \frac{exp(-\gamma z_t)}{1 + exp(-\gamma z_t)}.$$

Warning: F_{z_t} 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 λ 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:

`irf_s1_mean` A matrix, containing the impulse responses of the first regime. The row in each matrix denotes the responses of the *ith* variable to the shock. The columns are the horizons.

`irf_s1_low` 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*.

`irf_s1_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_s1_mean*.

`irf_s2_mean` A matrix, containing all impulse responses for the second regime. The row in each matrix denotes the responses of the *ith* variable to the shock. The columns denote the horizon.

`irf_s2_low` 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*.

`irf_s2_up` 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*.

`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 estimations of the impulse responses.

`fz` A vector, containing the values of the transition function F(z_t-1).

Author(s)

Philipp Adämmer

References

Akaike, H. (1974). "A new look at the statistical model identification", *IEEE Transactions on Automatic Control*, 19 (6): 716–723.

Auerbach, A. J., and Gorodnichenko Y. (2012). "Measuring the Output Responses to Fiscal Policy." *American Economic Journal: Economic Policy*, 4 (2): 1-27.

Auerbach, A. J., and Gorodnichenko Y. (2013). "Fiscal Multipliers in Recession and Expansion." *NBER Working Paper Series*. Nr 17447.

Blanchard, O., and Perotti, R. (2002). "An Empirical Characterization of the Dynamic Effects of Changes in Government Spending and Taxes on Output." *Quarterly Journal of Economics*, 117(4): 1329–1368.

Hurvich, C. M., and Tsai, C.-L. (1989), "Regression and time series model selection in small samples", *Biometrika*, 76(2): 297–307

Jordà, Ò. (2005) "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

Jordà, Ò, Schularick, M., Taylor, A.M. (2015), "Betting the house", *Journal of International Economics*, 96, S2-S18.

Newey, W.K., and West, K.D. (1987). "A Simple, Positive-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix." *Econometrica*, 55, 703–708.

Ramey, V.A., and Zubairy, S. (2018). "Government Spending Multipliers in Good Times and in Bad: Evidence from US Historical Data." *Journal of Political Economy*, 126(2): 850 - 901.

Schwarz, Gideon E. (1978). "Estimating the dimension of a model", *Annals of Statistics*, 6 (2): 461–464.

See Also

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

Examples

```
# This example replicates results from the Supplementary Appendix
# by Ramey and Zubairy (2018) (RZ-18).

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

# The shock is estimated by RZ-18
 shock             <- ag_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         <- ag_data[sample_start:sample_end, 6]

# Use the 7-quarter moving average growth rate of GDP as switching variable
# and adjust it to have suffiently long recession periods.
 switching_variable <- ag_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)

# Make and save 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 all impulse responses by using 'ggpubr' and 'gridExtra'
# lpirfs does not depend on those packages so they have to be installed
 library(ggpubr)
 library(gridExtra)

 s1_plots <- sapply(plots_nl_iv$gg_s1, ggplotGrob)
 s2_plots <- sapply(plots_nl_iv$gg_s2, ggplotGrob)
```

```
# Show all responses of state 1
 marrangeGrob(s1_plots, nrow = ncol(endog_data), ncol = 1, top = NULL)

# Show all responses of state 2
 marrangeGrob(s2_plots, nrow = ncol(endog_data), ncol = 1, top = NULL)
```

`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

```
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, 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).

`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.

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 "Vw" (white), "Vcx" (clustered by group and arellano method), "Vcx" (clustered by time and arrellano method), "Vctx" (clustered by group and time). For details see Miller (2017). The other options are "vcovBK", "vcovDC", "vcovG", "vcovHC", "vcovNW", "vcovSCC". For these options see vignette of plm package. If "use_gmm = TRUE", this option has to be NULL.

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:

$$F_{z_t} = \frac{exp(-\gamma z_t)}{1 + exp(-\gamma z_t)}.$$

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 λ for the Hodrick-Prescott filter (if "use_hp = TRUE").
gamma	Double. Positive value for γ, 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_summaries	Regression output for each horizon.
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

Croissant, Y., Millo, G. (2008). “Panel Data Econometrics in R: The plm Package.” *Journal of Statistical Software*, 27(2), 1-43. doi: 10.18637/jss.v027.i02 (URL: http://doi.org/10.18637/jss.v027.i02).

Jordà, Ò. (2005). "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

Jordà, Ò., Schualrick, M., Taylor, A.M. (2018). "Large and State-Dependent Effects of Quasi-Random Monetary Experiments", *NBER* working paper 23074, *FRBSF* working paper 2017-02.

Millo, G. (2017). “Robust Standard Error Estimators for Panel Models: A Unifying Approach.” *Journal of Statistical Software*, 82(3), 1-27. doi: 10.18637/jss.v082.i03 (URL: http://doi.org/10.18637/jss.v082.i03).

Examples

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

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

# Retrieve the JST Macrohistory Database
 url_jst <-"http://www.macrohistory.net/JST/JSTdatasetR3.xlsx"
 GET(url_jst, write_disk(jst_link <- tempfile(fileext = ".xlsx")))
 jst_data <- read_excel(jst_link, 2L)

# 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[year==1990][1]) %>%
               mutate(lhpreal  = log(hpreal))                  %>%

               mutate(lhpy     = lhpreal - log(rgdppc))        %>%
               mutate(lhpy     = lhpy - lhpy[year == 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,
                             lrgdp, lcpi, lriy, cay, nmortgdp)


# Use data_sample from 1870 to 2013 and exclude observations from WWI and WWII
  data_sample <-   seq(1870, 2016)[which(!(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",
                                 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)

# Create and plot irfs
 nl_plots <- plot_nl(results_panel)

 plot(nl_plots$gg_s1[[1]])
 plot(nl_plots$gg_s2[[1]])

# Plot values of the transition function for USA between 1950 and 2016
 library(ggplot2)
 library(dplyr)

 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 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,

                              use_gmm            = TRUE,
                              gmm_model          = "onestep",
                              gmm_effect         = "twoways",
                              gmm_transformation = "ld",

                              switching         = "lrgdp",
                              lag_switching     = TRUE,
                              use_hp            = TRUE,
                              lambda            = 6.25,
                              gamma             = 10,

                              l_exog_data       = "mortgdp",
                              lags_exog_data    = 1,

                              confint           = 1.67,
                              hor               = 5)

# Create and plot irfs
 nl_plots <- plot_nl(results_panel)

 plot(nl_plots$gg_s1[[1]])
 plot(nl_plots$gg_s2[[1]])
```

---

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.

Sample: 1960:01 - 2001:02.

Source

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

References

Jordà, Ò. (2005) "Estimation and Inference of Impulse Responses by Local Projections." *American Economic Review*, 95 (1): 161-182.

`plot_lin` *Compute and display plots of linear impulse responses*

Description

Compute and display linear impulse responses, estimated with lp_lin() and lp_lin_iv().

Usage

```
plot_lin(results_lin)
```

Arguments

`results_lin`	A list created with lp_lin() or lp_lin_iv().

Value

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

Author(s)

Philipp Adämmer

Examples

```
# See examples for lp_lin() and lp_lin_iv().
```

`plot_nl` *Compute and display plots of nonlinear impulse responses*

Description

Compute and display (nonlinear) impulse responses, estimated with lp_nl() and lp_nl_iv().

Usage

```
plot_nl(results_nl)
```

Arguments

`results_nl`	A list created with lp_nl() or 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().
```

Index