

Package ‘feVAR’

August 17, 2023

Type Package

Title Fixed Effects Vector Autoregressive Models

Version 0.1.1

Date 2023-08-17

Author Alessandro Magrini

Maintainer Alessandro Magrini <alessandro.magrini@unifi.it>

Description Estimation and analysis of fixed effects vector autoregressive models. The package implements the bias-corrected ordinary least squares estimation proposed by Dhaene & Jochmans (2016 <[doi:10.1016/j.econlet.2016.06.010](https://doi.org/10.1016/j.econlet.2016.06.010)>). In case of missing values, estimation is performed through the Expectation-Maximization (EM) algorithm. Several numerical and graphical tools to analyse the estimated models are available. Both panel, even unbalanced, and non-panel data are allowed.

Depends R (>= 3.5.0)

Imports graphics, stats, MASS

License GPL-2

NeedsCompilation no

R topics documented:

feVAR-package	2
agrisus2020	3
cooks.distance.feVAR	4
djstock	5
feVAR	6
forecastError	10
IRF	11
LAG	13
plot.feVAR	13
plot.IRF.feVAR	15
plot.predict.feVAR	17
predict.feVAR	19
residBPtest	20
unirootTest	21

Index	24
--------------	-----------

Description

Estimation and analysis of fixed effects vector autoregressive models. The package implements the bias-corrected ordinary least squares estimation proposed by Dhaene & Jochmans (2016) and the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to perform estimation in case of missing values. Several numerical and graphical tools to analyse the estimated models are available. Both panel, even unbalanced, and non-panel data are allowed.

Details

Package: feVAR
Type: Package
Version: 0.1.1
Date: 2023-08-17
License: GPL-2

Fixed effects vector autoregressive models (feVARs) are an extension of vector autoregressive models to panel data, firstly introduced by Holtz-Eakin et al. (1988). The main problem of feVARs is that ordinary least squares (OLS) estimation is biased even with a large number of units (Nickell, 1981), and, although the bias diminishes as the number of time points gets larger, simulations show significant bias when the time series has moderate size (e.g. 30 time points, see Judson & Owen, 1999). For this reason, the generalized method of moments (GMM) is often employed in place of OLS estimation. Unfortunately, the GMM may be highly inefficient when the time series are short. In this case, a simple and effective solution is represented by the bias-corrected OLS estimation proposed by Dhaene & Jochmans (2016).

The main functions of the package are:

- `unirootTest`: to check stationarity of the time series;
- `feVAR`: to estimate a feVAR model through bias-corrected OLS estimation (Dhaene & Jochmans, 2016);
- `plot.feVAR`: to display residual diagnostics for an estimated feVAR model;
- `IRF` and `plot.IRF.feVAR`: to compute and display impulse response functions for an estimated feVAR model;
- `predict.feVAR` and `plot.predict.feVAR`, to perform and display in-sample predictions or h-step ahead forecasts based on an estimated feVAR model;
- `forecastError`, to estimate the h-step ahead forecast error for an estimated feVAR model.

Author(s)

Alessandro Magrini <alessandro.magrini@unifi.it>

References

- Dempster A.P., Laird N.M., Rubin D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1): 1-38.
- Dhaene G., Jochmans K. (2016). Bias-corrected estimation of panel vector autoregressions. *Economics Letters*, 145: 98-103. DOI: 10.1016/j.econlet.2016.06.010
- Holtz-Eakin D., Newey W., Rosen H.S. (1988). Estimating vector autoregressions with panel data. *Econometrica*, 56(6), 1371-1395. DOI: 10.2307/1913103
- Judson R.A., Owen A.L. (1999). Estimating dynamic panel data models: a guide for macroeconomists. *Economics Letters*, 65: 9-15. DOI: 10.1016/S0165-1765(99)00130-5
- Nickell S. (1981). Biases in dynamic models with fixed effects. *Econometrica*, 49: 1417-1426.

agrisus2020

EU agricultural sustainability data

Description

Data on several indicators covering the economic, social and environmental dimensions of agricultural sustainability for 27 EU countries plus United Kingdom in the period 2004-2020.

Usage

```
data(agrisus2020)
```

Format

Object of class `data.frame` with a total of 476 observations on the following 16 variables:

`Country` Country name.

`Country_code` Country code.

`Year` Time of measurement (year).

`TFP_2015` Total Factor Productivity (TFP) index of agriculture (2015=100). Source: USDA, Economic Research Service (ERS).

`NetCapital_GVA` Net capital stocks in agriculture (2015 US dollars) to gross value added of agriculture (2015 US dollars). Source: Faostat.

`Manager_ratio` Ratio young/elderly for farm managers (number of managers with less than 35 years by 100 managers with 55 years and over). Source: CMEF.

`FactorIncome_paid_2010` Real income of agricultural factors per paid annual work unit (index 2010=100). Source: Eurostat.

`EntrIncome_unpaid_2010` Net entrepreneurial income of agriculture per unpaid annual work unit (index 2010=100). Source: Eurostat.

`Income_rur` Median equivalised net income in rural areas (power purchasing standard). Source: Eurostat.

`Unempl_rur` At-risk-of-poverty rate in rural areas (%). Source: Eurostat.

`Poverty_rur` Unemployment rate in rural areas (%). Source: Eurostat.

`RenewProd_UAA` Production of renewable energy from agriculture (share of utilized agricultural area, %). Source: CMEF.

Organic_p Area under organic cultivation (share of utilized agricultural area, %). Source: Faostat.

GHG_UAA Greenhouse gas emissions due to agriculture (million CO2 equivalent grams per hectare of utilized agricultural area). Source: Faostat.

GNB_UAA Gross nitrogen balance (tonnes of nutrient per hectare of utilized agricultural area). Source: Eurostat.

GDP Gross domestic product (million 2015 international US dollars). Source: World Bank.

Note

This dataset was employed in Magrini & Giambona (2022) and in Magrini (2022).

References

European Commission (2022). Eurostat database. <https://ec.europa.eu/eurostat/data/database>

European Commission (2020). Common Monitoring and Evaluation Framework (CMEF) for the CAP 2014-2020. https://agridata.ec.europa.eu/extensions/DataPortal/cmef_indicators.html

Food and Agriculture Organization (2022). Faostat statistical database. <https://www.fao.org/faostat/en/#home>

Magrini A., Giambona F. (2022). A composite indicator to assess sustainability of agriculture in European Union countries. *Social Indicators Research*, 163: 1003-10362. DOI: 10.1007/s11205-022-02925-6

Magrini A. (2022). Assessment of agricultural sustainability in European Union countries: A group-based multivariate trajectory approach. *Advances in Statistical Analysis*, 106: 673-703. DOI: 10.1007/s10182-022-00437-9

United States Department of Agriculture (2022). International agricultural productivity. USDA, Economic Research Service (ERS), Washington, DC, US.

<https://www.ers.usda.gov/data-products/international-agricultural-productivity>

cooks.distance.feVAR *Cook's distance method for class feVAR*

Description

Compute Cook's distances for an estimated fixed effects vector autoregressive model.

Usage

```
## S3 method for class 'feVAR'
cooks.distance(model, ...)
```

Arguments

model	Object of class feVAR.
...	Further parameters to be passed to the generic method cooks.distance.

Details

The Cook's distance (Cook, 1979) measures the leverage of each observation, i.e. how much its deletion impacts on the estimated regression hyperplane.

Value

An object of class `data.frame` including the Cook's distance of each observation (by row) in each equation (by column), plus the maximal one across all equations (last column).

References

Cook R.D. (1979). Influential observations in linear regression. *Journal of the American Statistical Association*, 74(365): 169-174. DOI: 10.2307/2286747

See Also

[feVAR](#).

Examples

```
## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)

m_agr_cook <- cooks.distance(m_agr) ## cook's distances in each equation
plot(m_agr_cook$(maximum'))        ## display maximal distance per observation

## End(Not run)
```

djstock

Data on Dow Jones stock

Description

Closing price of Dow Jones companies from 20 March 2019 to 9 June 2022.

Usage

```
data(djstock)
```

Format

Object of class `data.frame` with a total of 814 observations on 31 variables. The first variable represents the date of observation, while the subsequent variables represent the closing price of each Dow Jones company.

Description

Estimation of a fixed effects vector autoregressive model through bias-corrected ordinary least squares (Dhaene & Jochmans, 2016). If there are missing values in the endogenous variables, estimation is performed through the EM algorithm. Both panel, even unbalanced, and non-panel data are allowed.

Usage

```
feVAR(var.names, unit=NULL, time=NULL, exogenous=NULL, data, max.nlags=NULL,
      nlags=NULL, trend=c("none", "global", "unit"), ic=c("bic", "aic", "aicc", "hqic"),
      box.cox=1, ndiff=0, max.ndiff=2, auto.restrict=FALSE, imputation=TRUE,
      em.tol=1e-4, em.maxiter=100, quiet=FALSE)
```

Arguments

<code>var.names</code>	Character vector of length 2 or greater including the names of the endogenous variables, which must be quantitative variables.
<code>unit</code>	Character containing the name of the variable that identifies the units of observation. If NULL (the default), a single unit of observation is assumed.
<code>time</code>	Character containing the name of the time variable, which must be in numeric or date format. If NULL (the default), data are assumed to be temporally ordered.
<code>exogenous</code>	Character vector including the names of the exogenous variables, which can be either quantitative or qualitative variables. If NULL (the default), no exogenous variable is included in the model.
<code>data</code>	Object of class <code>data.frame</code> containing the variables in <code>var.names</code> , <code>exogenous</code> , <code>unit</code> and <code>time</code> .
<code>max.nlags</code>	The maximum lag order p of the model, which must be no greater than $T-5$, where T is the minimum time length across all units of observations. If an invalid value is provided, it is automatically replaced by the nearest valid one. If NULL (the default) and argument <code>nlags</code> is also NULL, it is set to \sqrt{T} . If NULL and argument <code>nlags</code> is not NULL, the best lag order is selected automatically based on information criteria. Ignored if there are missing values for the endogenous variables. See 'Details'.
<code>nlags</code>	The lag order p of the model. If NULL (the default), the best lag order is selected based on information criteria (see argument <code>max.nlags</code>), unless there are missing values for the endogenous variables. In this case, it is set equal to argument <code>max.nlags</code> . See 'Details'.
<code>trend</code>	The type of trend, which may be one among 'none' (no trend, i.e. $\delta_i = \mathbf{0}$), 'global' (same linear trend across all units of observation, i.e. $\delta_i = \delta \forall i$), and 'unit' (unit-specific linear trend, i.e. unconstrained δ_i). If 'unit' and there is only one unit of observation, it is set to 'global'. See 'Details'.
<code>ic</code>	The information criterion to be used in the automated selection of the lag order p of the model, which should be one among 'bic' (Bayesian information criterion), 'aic' (Akaike information criterion), 'aicc' (corrected Akaike information criterion), and 'hqic' (Hannan & Quinn information criterion). Ignored if <code>nlags</code> is not NULL. Default is 'bic'.

<code>box.cox</code>	Named vector including the lambda values of the Box-Cox transformation (Box & Cox, 1964) for variables in <code>var.names</code> and in <code>exogenous</code> . If <code>box.cox</code> has no names and length greater than one, the same ordering as in <code>c(var.names, exogenous)</code> is assumed. If <code>box.cox</code> has no names and length equal to one, the same lambda value is used for all variables in <code>var.names</code> and in <code>exogenous</code> . Valid lambda values are in the interval $[-2, 2]$. All invalid values are automatically replaced with the nearest valid ones. Default is 1, meaning no transformation for all variables. See the function <code>unirootTest</code> for more details.
<code>ndiff</code>	Named vector including the number of differences for variables in <code>var.names</code> and in <code>exogenous</code> . If <code>ndiff</code> has no names and length greater than one, the same ordering as in <code>c(var.names, exogenous)</code> is assumed. If <code>ndiff</code> has no names and length equal to one, the same number of differences is used for all variables in <code>var.names</code> and in <code>exogenous</code> . The number of differences cannot exceed $T-5$, where T is the minimum time length across all units of observations. All invalid values are automatically replaced with the nearest valid ones. Default is 0, meaning no differencing for all variables. See the function <code>unirootTest</code> for more details.
<code>max.ndiff</code>	Maximum order of differencing. Default is 2. Ignored if <code>ndiff</code> is not NULL. See 'Note'.
<code>auto.restrict</code>	Logical value indicating whether automated parameter restriction based on information criteria must be performed. Default is FALSE.
<code>imputation</code>	Logical value indicating whether imputation of missing values must be performed. Default is TRUE.
<code>em.tol</code>	Tolerance of the EM algorithm, which must be a non-negative value. If negative, it is set to $1e-4$ (the default).
<code>em.maxiter</code>	Maximum number of iterations for the EM algorithm, which must be an integer value no lower than 1. If not integer, it is rounded to the nearest integer. Default is 100.
<code>quiet</code>	Logical value indicating whether prompt messages must be suppressed. Default is FALSE.

Details

Fixed effects vector autoregressive models are an extension of vector autoregressive models to panel data, firstly introduced by Holtz-Eakin et al. (1988). Let Y_1, \dots, Y_m be m quantitative endogenous variables and X_1, \dots, X_q be q , either quantitative or qualitative, exogenous variables. Also, let $\mathbf{y}_{i,t}$ be an m -dimensional vector including the values of the endogenous variables observed on unit i at time t , and $\mathbf{x}_{i,t}$ be a q -dimensional vector including the values of the exogenous variables observed on unit i at time t . A fixed effect Vector Autoregressive Model (feVAR) of order $p \in \mathbb{N}_0$ is defined as:

$$\mathbf{y}_{i,t} = \boldsymbol{\alpha}_i + \boldsymbol{\delta}_i t + \sum_{j=1}^p \mathbf{B}_j \mathbf{y}_{i,t-j} + \mathbf{G} \mathbf{x}_{i,t} + \boldsymbol{\varepsilon}_{i,t}$$

where:

- $\boldsymbol{\alpha}_i$ and $\boldsymbol{\delta}_i$ are p -dimensional vectors of parameters specific for unit i ;
- \mathbf{B}_j is a $m \times m$ matrix of parameters common to all units, $j = 1, \dots, p$;
- \mathbf{G} is a $q \times q$ matrix of parameters common to all units;
- $\boldsymbol{\varepsilon}_{i,t}$ is a p -dimensional vector of random errors for unit i at time t such that:

$$\mathbb{E}(\boldsymbol{\varepsilon}_{i,t}) = \mathbf{0} \quad \forall i, t \quad \mathbb{E}(\boldsymbol{\varepsilon}_{i,t} \boldsymbol{\varepsilon}_{i,t}) = \boldsymbol{\Sigma} \quad \forall i, t \quad \mathbb{E}(\boldsymbol{\varepsilon}_{i,s} \boldsymbol{\varepsilon}_{i,t}) = \mathbf{0} \quad \forall i, s \neq t$$

A feVAR model can be viewed as a system of regression equations, one for each endogenous variable, where the regressors include the lags up to p of all endogenous variables and the exogenous variables. In particular:

- parameter α_i contains the intercepts for unit i across all equations;
- parameter δ_i contains the coefficients of the linear trend for unit i across all equations;
- parameter B_j contains, by column, the coefficients at lag j of the endogenous variables across all equations;
- parameter G contains, by column, the coefficients of the exogenous variables across all equations.

The function feVAR allows both panel, even unbalanced, and non-panel data (single unit of observation). The bias-corrected ordinary least squares (Dhaene & Jochmans, 2016) is applied only in the former case, while ordinary least squares is used in the latter case.

In case of missing values in the endogenous variables, provided that argument `imputation` is set to `TRUE`, estimation is performed through the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) as follows:

1. *Initialization*: missing values for each endogenous variable are filled with the sample mean;
2. *Expectation step*: Box-Cox transformation and/or differencing are eventually applied and the feVAR model is estimated;
3. *Maximization step*: in-sample predictions are used to fill missing values internal to the time series, and backward h -steps ahead forecasts are used to fill missing values at the beginning of the time series;
4. Repeat from step 2) until convergence of the likelihood.

During this procedure, the lag order of the feVAR model is held constant without possibility to select it automatically based on information criteria. However, it is possible to apply automated parameter restriction (if argument `auto.restrict` is set to `TRUE`) after convergence of the algorithm. Estimation resulting from the EM algorithm has the property to maximize the expected value of the likelihood with respect to the distribution of missing data. Currently, the EM algorithm only handles missing values for the endogenous variables, while missing values in the exogenous variables are imputed with the most recent value before estimating the feVAR model.

The following S3 methods are available for class feVAR: `summary`, `coef`, `confint`, `nobs`, `residuals`, `fitted.values`, `logLik`, `extractAIC`, `AIC`, `BIC`, `cooks.distance.feVAR`, `predict.feVAR`.

Value

An object of class feVAR including the following components:

- `equations`: list of objects of class `lm`, including parameter estimation for each equation;
- `call`: list including the main arguments passed to the function feVAR;
- `intercepts`: matrix including the estimated intercepts for each unit (by row) and endogenous variable (by column);
- `Beta`: point estimation of regression coefficients among endogenous variables (matrix B);
- `Sigma`: estimated covariance matrix of random errors;
- `data.orig`: original data (before Box-Cox transformation and differencing);
- `data.used`: data used for model estimation (after Box-Cox transformation and differencing);
- `companion`: the companion matrix of the model;
- `eigen.module`: module of eigenvalues of the companion matrix. If all of them are lower than one, then the vector autoregressive model is stable.

Note

Second-order stationarity (expected value and autocorrelation function independent of time) of all time series is a basic assumption of the model, that is guaranteed if no time series contains unit roots. Before calling the function `feVAR`, the user is strongly recommended to check the absence of unit roots in each time series through the function `unirootTest`.

If argument `box.cox` is NULL, lambda values are determined automatically through maximum likelihood estimation. Similarly, if argument `ndiff` is NULL, the order of differencing is determined automatically based on sequential unit root tests. See the function `unirootTest` for more details.

A popular combination of Box-Cox transformation and differencing is the first order difference of logarithmic values (`box.cox=0` and `ndiff=1`), which approximates the relative changes with respect to the previous time point. In this case, the feVAR model is formulated as:

$$\Delta \log \mathbf{y}_{i,t} = \boldsymbol{\delta}_i + \sum_{j=1}^p \mathbf{B}_j \Delta \log \mathbf{y}_{i,t-j} + \mathbf{G} \Delta \log \mathbf{x}_{i,t} + \boldsymbol{\varepsilon}_{i,t}$$

$$E(\boldsymbol{\varepsilon}_{i,t}) = 0 \quad \forall i, t \quad E(\boldsymbol{\varepsilon}_{i,t} \boldsymbol{\varepsilon}_{i,t}) = \boldsymbol{\Sigma} \quad \forall i, t \quad E(\boldsymbol{\varepsilon}_{i,s} \boldsymbol{\varepsilon}_{i,t}) = 0 \quad \forall i, s \neq t$$

In general, it is easy to see that, when an endogenous variable is differenced one time, the original intercepts α_i are canceled out and the new intercepts become δ_i , i.e. the unit-specific coefficients of a linear trend.

References

- H. Akaike (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6): 716-723. DOI: 10.1109/TAC.1974.1100705
- Cavanaugh, J.E. (1997), Unifying the derivations of the Akaike and corrected Akaike information criteria. *Statistics & Probability Letters*, 31(2): 201-208. DOI: 10.1016/s0167-7152(96)00128-9
- Dempster A.P., Laird N.M., Rubin D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1): 1-38.
- Dhaene G., Jochmans K. (2016). Bias-corrected estimation of panel vector autoregressions. *Economics Letters*, 145: 98-103. DOI: 10.1016/j.econlet.2016.06.010
- Hannan E.J., Quinn B.G. (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society, Series B*, 41: 190-195.
- Holtz-Eakin D., Newey W., Rosen H.S. (1988). Estimating vector autoregressions with panel data. *Econometrica*, 56(6), 1371-1395. DOI: 10.2307/1913103
- Schwarz G.E. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2): 461-464. DOI: 10.1214/aos/1176344136

See Also

`unirootTest`; `plot.feVAR`; `cooks.distance.feVAR`; `IRF`; `predict.feVAR`; `forecastError`.

Examples

```
## Not run:
## EXAMPLE WITH A SINGLE UNIT OF OBSERVATION ##

data(djstock)
x_dj <- colnames(djstock)[-1]

# fit a model with p=1 on data in logarithmic differences ('box.cox'=0 and 'ndiff'=1)
```

```

m_dj <- feVAR(var.names=x_dj, time="Date", data=djstock,
  box.cox=0, ndiff=1, nlags=1)
summary(m_dj) ## summary of parameter estimation

# automated selection of the lag order
m_dj_auto <- feVAR(var.names=x_dj, time="Date", data=djstock,
  box.cox=0, ndiff=1, ic="bic", max.nlags=5)
summary(m_dj_auto) ## summary of parameter estimation

## End(Not run)

## EXAMPLE WITH PANEL DATA ##

## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]

# fit a model with p=1 on data in logarithmic differences ('box.cox'=0 and 'ndiff'=1)
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)
summary(m_agr) ## summary of parameter estimation

# automated selection of the lag order
m_agr_auto <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, ic="bic", max.nlags=4)
summary(m_agr_auto) ## summary of parameter estimation

## End(Not run)

```

forecastError

Forecast error for a feVAR model

Description

Estimate h-step ahead forecast error for an estimated fixed effects vector autoregressive model.

Usage

```
forecastError(model, n.ahead=1, quiet=FALSE)
```

Arguments

model	Object of class feVAR.
n.ahead	The number of steps ahead, which must be no lower than 1 (the default). If an invalid value is provided, it is automatically replaced by the nearest valid one.
quiet	Logical value indicating whether prompt messages must be suppressed. Default is FALSE.

Value

A list with one component for each step ahead. Each component of the list is a matrix including the value of forecast error metrics (by column) for each endogenous variable (by row). Let $\hat{y}_t(h)$ be the (pseudo) prediction at h steps ahead performed based on the information up to time t . Available metrics include:

- 'rmse', the root mean squared error (RMSE):

$$RMSE(h) = \sqrt{\frac{1}{T-h} \sum_{t=1}^{T-h} (y_{t+h} - \hat{y}_t(h))^2}$$

- 'mae', the mean absolute error (MAE):

$$MAE(h) = \frac{1}{T-h} \sum_{t=1}^{T-h} |y_{t+h} - \hat{y}_t(h)|$$

- 'rmsse', the root mean squared scaled error (RMSSE), equal to the RMSE divided by the RMSE of the naive predictor:

$$RMSSE(h) = \sqrt{\frac{\frac{1}{T-h} \sum_{t=1}^{T-h} (y_{t+h} - \hat{y}_t(h))^2}{\frac{1}{T-h} \sum_{t=1}^{T-h} (y_{t+h} - y_t)^2}}$$

- 'mase', the mean absolute scaled error (MAE), equal to the MAE divided by the MAE of the naive predictor:

$$MASE(h) = \frac{\frac{1}{T-h} \sum_{t=1}^{T-h} |y_{t+h} - \hat{y}_t(h)|}{\frac{1}{T-h} \sum_{t=1}^{T-h} |y_{t+h} - y_t|}$$

- 'mape', the mean absolute percentage error (MAPE), computed only for strictly positive endogenous variables:

$$MAPE(h) = \frac{100}{T-h} \sum_{t=1}^{T-h} \frac{|y_{t+h} - \hat{y}_t(h)|}{y_{t+h}}$$

See Also

[feVAR](#).

Examples

```
## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)

forecastError(m_agr, n.ahead=3) ## forecast error at 3 steps ahead

## End(Not run)
```

Description

Compute impulse response functions (IRFs) for an estimated fixed effects vector autoregressive model.

Usage

```
IRF(model, n.ahead=10, nboot=100, level=0.95, quiet=FALSE)
```

Arguments

<code>model</code>	Object of class <code>feVAR</code> .
<code>n.ahead</code>	The number of steps ahead, which must be no lower than 1. Default is 10. If an invalid value is provided, it is automatically replaced by the nearest valid one.
<code>nboot</code>	The number of bootstrap resamples to estimate confidence intervals. Default is 100. If 0, confidence intervals are not estimated. If an invalid value is provided, it is automatically replaced by the nearest valid one.
<code>level</code>	The confidence level. Default is 0.95. If outside the interval $[0, 1]$, it is set to 0.95.
<code>quiet</code>	Logical value indicating whether prompt messages must be suppressed. Default is <code>FALSE</code> .

Value

Object of class `IRF.feVAR`, which is a list with the following three components:

- `irf`, including the computed IRFs,
- `irf_sx`, including the lower bound of confidence intervals for IRFs,
- `irf_dx`, including the upper bound of confidence intervals for IRFs.

Each of these three components is an array with three dimensions that correspond, respectively, to the number of steps ahead, to the endogenous variable originating the impulse, and to the endogenous variables receiving the impulse.

Note

The class `IRF.feVAR` has its own `plot` method. See [plot.IRF.feVAR](#) for details.

See Also

[feVAR](#), [plot.IRF.feVAR](#).

Examples

```
## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)

m_agr_irf <- IRF(m_agr, nboot=500)      ## compute IRFs (500 bootstrap resamples
                                       ##   for more accurate confidence intervals)
m_agr_irf$irf[, "TFP_2015", ]          ## IRFs generated by 'TFP_2015'
m_agr_irf$irf[, "NetCapital_GVA", ]    ## IRFs received by 'NetCapital_GVA'
m_agr_irf$irf[, "TFP_2015", "Poverty_rur"] ## IRFs from 'TFP_2015' to 'Poverty_rur'

## End(Not run)
```

LAG	<i>Generate lagged values</i>
-----	-------------------------------

Description

Generate lagged values of a quantitative variable.

Usage

```
LAG(x, lag, unit=NULL, del=0)
```

Arguments

x	Numerical vector representing temporally ordered data.
lag	The lag to generate. If 0, x is returned with no changes. If an invalid value is provided, it is automatically replaced by the nearest valid one.
unit	Numerical or character vector of the same length of x containing the identifiers of the units of observation. If NULL (the default), a single unit of observation is assumed.
del	The number of initial observations to be set to missing. Default is 0. Note that the first lag observations are inevitably lost and therefore set to missing.

Value

Numerical vector of the same length of x.

Examples

```
# single unit of observation
x <- rnorm(10) ## simulate a white noise on 10 time points
LAG(x, 3)      ## lag 3

# panel data
x <- rnorm(30)
u <- rep(1:3, each=10) ## 3 units, each with 10 time points
LAG(x, 3, unit=u)    ## lag 3
```

plot.feVAR	<i>Plot method for class feVAR</i>
------------	------------------------------------

Description

Display residual diagnostics for an estimated fixed effect vector autoregressive model.

Usage

```
## S3 method for class 'feVAR'
plot(x, type=c("ts", "acf", "acf2", "fitVSres", "qqnorm"), var.names=NULL, max.nlags=NULL,
     signif=0.05, ylim=NULL, cex.points=0.6, add.grid=TRUE, xlab=NULL, ylab=NULL,
     titles=NULL, las=0, mfrow=NULL, mar=c(3.5, 3.5, 2, 2), mgp=c(2.3, 0.8, 0), ...)
```

Arguments

<code>x</code>	Object of class feVAR.
<code>type</code>	Type of diagnostic, which should be one among 'ts' (1): time series plot of residuals, 'acf' (2): autocorrelograms of residuals, 'acf2' (3): autocorrelograms of squared residuals, 'fitVSres' (4): fitted values versus residuals, and 'qqnorm' (5): normal quantile plot of residuals.
<code>var.names</code>	Character vector indicating the name of endogenous variables for which the diagnostic must be displayed.
<code>max.nlags</code>	The number of lags to be displayed in the autocorrelograms. If NULL (the default), it is determined automatically. If an invalid value is provided, it is automatically replaced by the nearest valid one. Ignored if type is not equal to 'acf' or 'acf2'.
<code>signif</code>	The significance level of the test for no autocorrelation in the autocorrelograms. Default is 0.05. If outside the interval $[0, 1]$, it is set to 0.05. Ignored if type is not equal to 'acf' or 'acf2'.
<code>ylim</code>	Numerical vector of length 2 indicating the limits of the y-axis.
<code>cex.points</code>	Expansion factor for points. If equal to 0, points are suppressed. Ignored if type is not equal to 'qqnorm' or 'fitVSres'. Default is 0.6.
<code>add.grid</code>	Logical value indicating whether the grid must be added. Ignored if type is equal to 'acf' or 'acf2'. Default is TRUE.
<code>xlab</code>	Label for the x-axis.
<code>ylab</code>	Label for the y-axis.
<code>titles</code>	Character vector containing the title for the graphics. If the vector has names, titles are assigned based on them, otherwise they are assigned in the same order of the endogenous variables. If NULL (the default), the names of endogenous variables are used as titles.
<code>las</code>	Orientation of the tick mark labels. Possible values are: parallel to the axis (0), horizontal (1), perpendicular to the axis (2), vertical (3). If NULL (the default), it is set automatically based on the length of axis labels.
<code>mfrow</code>	Numerical vector of the form $c(nr, nc)$: graphics are displayed in an nr -by- nc array on the device by rows. If NULL (the default), it is determined automatically based on the argument <code>var.names</code> .
<code>mar</code>	Numerical vector of length 4 indicating the margin sizes in the following order: bottom, left, top, right. Default is $c(3.5, 3.5, 2, 2)$.
<code>mgp</code>	Numerical vector of length 3 indicating the location of axis labels, tick mark labels and tick marks, respectively, relative to the plot window. Default is $c(2.3, 0.8, 0)$.
<code>...</code>	Further parameters to be passed to the generic method plot.

Details

In order to meet the hypothesis of serially uncorrelated errors, autocorrelograms of residuals must show no significant spikes. Note that the violation of the hypothesis of serially uncorrelated errors leads to inconsistent estimates.

In order to meet the hypothesis of homoschedastic errors, autocorrelograms of squared residuals must show no significant spikes, and the graphics of fitted values versus residuals must show no patterns. Note that the violation of the hypothesis of homoschedasticity leads to inefficient but still

consistent estimates. The Box-Cox transformation may solve heteroschedasticity (see argument 'box.cox' in the function [feVAR](#)).

In order to meet the hypothesis of normal errors, the normal quantile plot of residuals must look approximatively linear. Note that the violation of the hypothesis of normal errors does not affect neither consistency nor efficiency of the estimates, but significance tests and confidence intervals are no more exact and are valid only asymptotically.

See Also

[feVAR](#).

Examples

```
## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)

# time series plots of residuals
plot(m_agr, type="ts", cex.main=1.1)

# autocorrelograms of residuals
plot(m_agr, type="acf", cex.main=1.1)

# autocorrelograms of squared residuals
plot(m_agr, type="acf2", cex.main=1.1)

# fitted versus residuals
plot(m_agr, type="fitVSres", cex.main=1.1)

# normal quantile plots of residuals
plot(m_agr, type="qqnorm", cex.main=1.1)

## End(Not run)
```

plot.IRF.feVAR

Plot method for class IRF.feVAR

Description

Display impulse response functions (IRFs) for an estimated fixed effects vector autoregressive model.

Usage

```
## S3 method for class 'IRF.feVAR'
plot(x, from=NULL, to=NULL, n.ahead=NULL, labels=NULL, ylim=NULL, add.grid=TRUE,
  add.titles=FALSE, cex.points=0, mfrow=NULL, mar=c(3.5,3.5,2,2), mgp=c(2.3,0.8,0), ...)
```

Arguments

<code>x</code>	Object of class <code>IRF.feVAR</code> .
<code>from</code>	Vector containing the names or the numerical identifiers of the endogenous variables originating the IRFs. If <code>NULL</code> (the default), all the endogenous variables are considered.
<code>to</code>	Character vector containing the names of the endogenous variables receiving the IRFs. Numerical identifiers can be provided instead of names. If <code>NULL</code> (the default), all the endogenous variables are considered.
<code>n.ahead</code>	The number of steps ahead, which must be no lower than 1 and no greater than the number of steps ahead in <code>x</code> (the default). If an invalid value is provided, it is automatically replaced by the nearest valid one.
<code>labels</code>	Character vector containing the labels for the endogenous variables. If the vector has names, labels are assigned based on them, otherwise they are assigned in the same order of the endogenous variables. If <code>NULL</code> (the default), the names of endogenous variables are used as labels.
<code>ylim</code>	Numerical vector of length 2 indicating the limits of the y-axis.
<code>add.grid</code>	Logical value indicating whether the grid must be added. Default is <code>TRUE</code> .
<code>add.titles</code>	Logical value indicating whether a title must be added to each graphic. Default is <code>FALSE</code> .
<code>cex.points</code>	Expansion factor for points. If 0, points are suppressed. Default is 0.
<code>mfrow</code>	Numerical vector of the form <code>c(nr, nc)</code> : graphics are displayed in an <code>nr</code> -by- <code>nc</code> array on the device by rows. If <code>NULL</code> (the default), it is determined automatically based on arguments <code>from</code> and <code>to</code> .
<code>mar</code>	Numerical vector of length 4 indicating the margin sizes in the following order: bottom, left, top, right. Default is <code>c(3.5, 3.5, 2, 2)</code> .
<code>mgp</code>	Numerical vector of length 3 indicating the location of axis labels, tick mark labels and tick marks, respectively, relative to the plot window. Default is <code>c(2.3, 0.8, 0)</code> .
<code>...</code>	Further parameters to be passed to the generic method <code>plot</code> .

See Also

[feVAR](#); [IRF](#).

Examples

```
## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)
m_agr_irf <- IRF(m_agr, nboot=500) ## compute IRFs

# display IRFs generated by 'TFP_2015'
plot(m_agr_irf, from="TFP_2015")
plot(m_agr_irf, from=1) ## equivalent

# display IRFs received by 'NetCapital_GVA'
plot(m_agr_irf, to="NetCapital_GVA")
plot(m_agr_irf, from=2) ## equivalent
```



```
# display IRFs generated by 'TFP_2015' and 'NetCapital_GVA' and
# received by 'Poverty_rur', 'Organic_p' and 'GHG_UAA'
plot(m_agr_irf, from=c("TFP_2015","NetCapital_GVA"),
     to=c("Poverty_rur", "Organic_p" and "GHG_UAA"))
plot(m_agr_irf, from=c(1,2), to=c(8,10,11)) ## equivalent

## End(Not run)
```

plot.predict.feVAR *Plot method for class predict.feVAR*

Description

Display in-sample predictions or h-step ahead forecasts based on an estimated fixed effects vector autoregressive model.

Usage

```
## S3 method for class 'predict.feVAR'
plot(x, var.names=NULL, unit.id, newdata=NULL, start=NULL, ylim=NULL, add.grid=TRUE,
     las=0, cex.axis=c(1,1), cex.lab=c(1,1), xlab=NULL, ylab=NULL, titles=NULL,
     obs.col="grey40", fit.col="dodgerblue", out.col="red", new.col="grey40",
     interval.col="grey70", obs.lty=1, fit.lty=5, out.lty=1, new.lty=2, obs.cex=0.4,
     new.cex=0.4, mfrow=NULL, mar=c(3.5,3.5,2,2), mgp=c(2.3,0.8,0), ...)
```

Arguments

x	Object of class predict.feVAR.
var.names	Vector indicating the names or the numerical identifiers of the endogenous variables for which predictions must be displayed.
unit.id	Name or identificative number of one unit of observation for which predictions must be displayed. Ignored in case of a single unit of observation.
newdata	Object of class data.frame containing new observations to be added to the graphics. If NULL (the default), no further observations are added besides the existing ones. Ignored if x contains in-sample predictions.
start	Time point from which the graphic must start. If NULL (the default), the whole observed time series is displayed.
ylim	Numeric vector of length 2 indicating the limits of the y-axis.
add.grid	Logical value indicating whether the grid must be added. Default is TRUE.
las	Orientation of the tick mark labels. Possible values are: parallel to the axis (0), horizontal (1), perpendicular to the axis (2), vertical (3). If NULL (the default), it is set automatically based on the length of axis labels.
cex.axis	Numeric vector of length 2 indicating the expansion factor for x-axis and y-axis labels, respectively. Default is (1,1).
cex.lab	Numeric vector of length 2 indicating the expansion factor for x and y labels, respectively. Default is (1,1).
xlab	Label for the x-axis.

<code>ylab</code>	Label for the y-axis.
<code>titles</code>	Character vector containing the title for each graphic. If NULL (the default), the names of endogenous variables are used as titles.
<code>obs.col</code>	Color for the time series of observed data. Default is black.
<code>fit.col</code>	Color for the time series of fitted values. Default is black.
<code>out.col</code>	Color for the time series of predictions. Default is red.
<code>new.col</code>	Color for the time series of new data. Default is black.
<code>interval.col</code>	Color of prediction intervals. Default is grey40.
<code>obs.lty</code>	Line type for the time series of observed data. Default is 1.
<code>fit.lty</code>	Line type for the time series of fitted values. Default is 1.
<code>out.lty</code>	Line type for the time series of predictions. Default is 1.
<code>new.lty</code>	Line type for the time series of new data. Default is 1.
<code>obs.cex</code>	Expansion factor for points in the time series of observed data. If 0, points are suppressed. Default is 0.4.
<code>new.cex</code>	Expansion factor for points in the time series of new data. If 0, points are suppressed. Default is 0.4.
<code>mfrow</code>	Numerical vector of the form <code>c(nr, nc)</code> : graphics are displayed in an <code>nr</code> -by- <code>nc</code> array on the device by rows. If NULL (the default), it is determined automatically based on the argument <code>var.names</code> .
<code>mar</code>	Numeric vector of length 4 indicating the margin sizes in the following order: bottom, left, top, right. Default is <code>c(3.5, 3.5, 2, 2)</code> .
<code>mgp</code>	Numeric vector of length 3 indicating the location of axis labels, tick mark labels and tick marks, respectively, relative to the plot window. Default is <code>c(2.3, 0.8, 0)</code> .
<code>...</code>	Further parameters to be passed to the generic method <code>plot</code> .

See Also

[feVAR](#); [predict.feVAR](#).

Examples

```
## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)

# predictions for the first four variables on unit 'Italy'
pr1 <- predict(m_agr, unit.id="Italy", var.names=varNames[1:4]) ## in-sample
plot(pr1)
pr2 <- predict(m_agr, unit.id=15, var.names=1:4) ## equivalent
plot(pr2)
pr3 <- predict(m_agr, unit.id=15, var.names=1:4, n.ahead=3) ## 3 steps ahead forecast
plot(pr3)

## End(Not run)
```

predict.feVAR	<i>Predict method for class feVAR</i>
---------------	---------------------------------------

Description

Perform in-sample prediction or h-step ahead forecasts based on an estimated fixed effects vector autoregressive model.

Usage

```
## S3 method for class 'feVAR'
predict(object, n.ahead=0, unit.id=NULL, newdata=NULL, level=0.95, subset=NULL, ...)
```

Arguments

object	Object of class feVAR.
n.ahead	The number of steps ahead of predictions. If equal to 0 (the default), in-sample predictions are returned. If an invalid value is provided, it is automatically replaced by the nearest valid one.
unit.id	Name or numerical identifier of the units of observation for which predictions must be made. Ignored in case of a single unit of observation. If NULL (the default), predictions are performed for all units of observation.
newdata	Object of class data.frame where to look at future values of exogenous variables. Ignored if n.ahead is equal to 0. If NULL (the default) and n.ahead is greater than 0, exogenous variables are assumed to keep the latest observed value.
level	Confidence level of prediction intervals. Default is 0.95. If outside the interval [0, 1], it is set to 0.95.
subset	For internal use only.
...	Further parameters to be passed to the generic method predict.

Value

An object of class predict.feVAR, which is a list with the following four components:

- predicted: a named list of objects of class data.frame, one for each endogenous variable. Each data frame includes one record for each observation, and the following variables: unit name (only in case of panel data), time of observation, point predictions, lower and upper bound of prediction intervals;
- observed: an object of class data.frame containing the observed data;
- n.ahead: the number of steps ahead of the predictions;
- call: list including the main characteristics of the argument object.

Note

The class predict.feVAR has its own plot method. See [plot.predict.feVAR](#) for details.

See Also

[feVAR](#); [plot.predict.feVAR](#).

Examples

```
## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)

predict(mod_agr)          ## in-sample prediction
predict(mod_agr, n.ahead=3) ## 3 steps ahead forecast

## End(Not run)
```

residBPtest

Box-Pierce test for a feVAR model

Description

Box-Pierce test on residuals for an estimated fixed effects vector autoregressive model.

Usage

```
residBPtest(model, max.lag=NULL)
```

Arguments

<code>model</code>	Object of class <code>feVAR</code> .
<code>max.lag</code>	Maximum lag at which to perform the test, which must be no lower than 1 and no greater than $T-5$, where T is the minimum time length across all units of observations. If <code>NULL</code> (the default), it is set to \sqrt{T} . If an invalid value is provided, it is automatically replaced by the nearest valid one.

Details

The Box-Pierce test (Box & Pierce, 1970) checks the hypothesis of no autocorrelation at specific lags. Performing the test on the residuals provides an indication about the hypothesis of serially uncorrelated errors, while performing the test on the squared residuals provides an indication about the hypothesis of homoschedastic errors.

Value

An object of class `residBPtest.feVAR`, which is a list with two components: `residual`, containing the results of the test on residuals, and `sq.residual`, containing the results of the test on squared residuals. Each of these two components is a list with the following two components:

- `statistic`: matrix containing the test statistic for each lag (by row) and each endogenous variable (by column);
- `p.value`: matrix containing the p-value for each lag (by row) and each endogenous variable (by column).

Note

The class `residBPtest.feVAR` has its own `print` method that reports, for each endogenous variable, the minimum lag with significant autocorrelation in the residuals and in the squared residuals.

References

Box G.E.P., Pierce D.A. (1970). Distribution of residual autocorrelations in autoregressive integrated moving average time series models. *Journal of the American Statistical Association*, 65(332): 1509-1526. DOI: 10.1080/01621459.1970.10481180

See Also

[feVAR](#).

Examples

```
## Not run:
data(agrisus2020)
x_agr <- colnames(agrisus2020)[4:15]
m_agr <- feVAR(var.names=x_agr, unit="Country", time="Year", exogenous="GDP",
  data=agrisus2020, box.cox=0, ndiff=1, nlags=1)

# perform the box-pierce test on residuals
m_agr_bpt <- residBPtest(m_agr)
m_agr_bpt ## summary of the test

# p-value of the test at each lag
m_agr_bpt$residual$p.value ## on residuals
m_agr_bpt$sq.residual$p.value ## on squared residuals

## End(Not run)
```

unirootTest

Unit root tests for panel data

Description

Perform augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests for unit roots on a multivariate time series, which may have a panel structure.

Usage

```
unirootTest(var.names, unit=NULL, time=NULL, data, box.cox=1, ndiff=0, max.nlags=NULL)
```

Arguments

<code>var.names</code>	Character vector including the name of the variables to undergo the tests.
<code>unit</code>	Character containing the name of the variable that identifies the units of observation. If NULL (the default), a single unit of observation is assumed.
<code>time</code>	Character containing the name of the time variable, which must be in numeric or date format. If NULL (the default), data are assumed to be temporally ordered.
<code>data</code>	Object of class <code>data.frame</code> containing the variables in <code>var.names</code> , <code>unit</code> and <code>time</code> .

<code>box.cox</code>	Named vector including the lambda values of the Box-Cox transformation (Box & Cox, 1964) for variables in <code>var.names</code> . If <code>box.cox</code> has no names and length greater than one, the same ordering as in <code>var.names</code> is assumed. If <code>box.cox</code> has no names and length equal to one, the same lambda value is used for all variables in <code>var.names</code> . Valid lambda values are in the interval $[-2, 2]$. All invalid values are automatically replaced with the nearest valid ones. Default is 1, meaning no transformation for all variables. See 'Details'.
<code>ndiff</code>	Named vector including the number of differences for variables in <code>var.names</code> . If <code>ndiff</code> has no names and length greater than one, the same ordering as in <code>var.names</code> is assumed. If <code>ndiff</code> has no names and length equal to one, the same number of differences is used for all variables in <code>var.names</code> . The number of differences cannot exceed $T-5$, where T is the minimum time length across all units of observations. All invalid values are automatically replaced with the nearest valid ones. Default is 0, meaning no differencing for all variables.
<code>max.lags</code>	The maximum lag to consider in the tests, which must be no greater than $T-5$, where T is the minimum time length across all units of observations. If NULL (the default), it is set to \sqrt{T} . If an invalid value is provided, it is automatically replaced by the nearest valid one. See 'Details'.

Details

The variables subjected to the tests must be quantitative. In order to keep statistical power as high as possible in presence of missing values, they are replaced by the most recent observed value (Ryan & Giles, 1998).

The Box-Cox transformation is defined as:

$$f(y; \lambda, \eta) = \begin{cases} \frac{(y + \eta)^{\lambda-1}}{\lambda}, & \lambda \neq 0 \\ \log(y + \eta), & \lambda = 0 \end{cases}$$

where η is an offset equal to zero if the variable contains no negative values, otherwise equal to $-\min(y)$. If $\lambda = 0$ (logarithmic transformation), then η is further increased by half the lowest non-zero value of $y - \min(y)$. If $\lambda = 1$, no transformation is applied.

In case of panel data, the ADF test is performed by combining the individual p-values according to the method by Demetrescu *et al.* (2006), while the KPSS test is performed by combining the individual test statistics as proposed by Hadri (2000).

The tests account for a deterministic linear trend, unless the order of differencing is greater than 0. The lag length at which to perform the tests is selected through AIC-based backward elimination starting from the lag length specified in argument `max.lag`.

The null hypothesis of the ADF test (Dickey & Fuller, 1981) is the presence of unit roots, while the null hypothesis of the KPSS test (Kwiatkowski *et al.*, 1992) is the absence of unit roots. Therefore, p-value higher than 0.05 for the ADF test or p-value lower than 0.05 for the KPSS test suggest the presence of unit roots and the need of further differencing.

Value

An object of class `unirootTest`, which is a list including a list for each variable in `var.names`, each with the following components:

- `statistic`: test statistic for each test;

- lag.selected: lag selected for each test;
- p.value: p-value for each test;
- box.cox: offset and lambda values of the Box-Cox transformation for each variable subjected to the tests;
- ndiff: order of differencing for each variable subjected to the tests.

Note

The class unirootTest has its own print method that reports the p-values of the tests for each endogenous variable.

References

- Demetrescu M., Hassler U., Tarcolea A. (2006). Combining significance of correlated statistics with application to panel data. *Oxford Bulletin of Economics and Statistics*, 68(5), 647-663. DOI: 10.1111/j.1468-0084.2006.00181.x
- Dickey D.A., Fuller W.A. (1981). Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica*, 49(4): 1057-1072. DOI: 10.2307/1912517
- Hadri K. (2000). Testing for stationarity in heterogeneous panel data. *Econometrics Journal*, 3: 148-161. DOI: 10.1111/1368-423X.00043
- Kwiatkowski D., Phillips P.C.B., Schmidt P., Shin Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54(1-3): 159-178. DOI: 10.1016/0304-4076(92)90104-Y
- Ryan K.F., Giles D.E.A (1998). Testing for unit roots in economic time-series with missing observations. In: T. B. Fomby & R. C. Hill (eds.) *Messy Data (Advances in Econometrics, Vol. 13)*, pages 203-242. DOI: 10.1108/S0731-9053(1999)0000013010

See Also

[feVAR](#).

Examples

```
data(agrisus2020)
x_agr <- names(agrisus2020)[4:15]

# tests on variables in level
unirootTest(x_agr, unit="Country", time="Year", data=agrisus2020)

# tests on variables in logarithmic differences
unirootTest(x_agr, unit="Country", time="Year", data=agrisus2020,
  box.cox=0, ndiff=1)
```

Index

`agrisus2020`, [3](#)

`cooks.distance.feVAR`, [4](#), [8](#), [9](#)

`djstock`, [5](#)

`feVAR`, [2](#), [5](#), [6](#), [9](#), [11](#), [12](#), [15](#), [16](#), [18](#), [19](#), [21](#), [23](#)

`feVAR-package`, [2](#)

`forecastError`, [2](#), [9](#), [10](#)

`IRF`, [2](#), [9](#), [11](#), [16](#)

`LAG`, [13](#)

`plot.feVAR`, [2](#), [9](#), [13](#)

`plot.IRF.feVAR`, [2](#), [12](#), [15](#)

`plot.predict.feVAR`, [2](#), [17](#), [19](#)

`predict.feVAR`, [2](#), [8](#), [9](#), [18](#), [19](#)

`residBPtest`, [20](#)

`unirootTest`, [2](#), [7](#), [9](#), [21](#)