Package 'dynamac'

May 8, 2018

Title Dynamic Simulation and Testing for Single-Equation ARDL Models

Version 0.1.4

Maintainer Soren Jordan <sorenjordanpols@gmail.com>

Description While autoregressive distributed lag models allow for extremely flexible dynamics, interpreting substantive significance of complex lag structures remains difficult. This package is designed to assist users in dynamically simulating and plotting the results of various autoregressive distributed lag models. It also contains post-estimation diagnostics, including a test for cointegration when estimating the error-correction variant of the autoregressive distributed lag model (Pesaran, Shin, and Smith 2001 <doi:10.1002/jae.616>).

Imports MASS, lmtest

Suggests urca, knitr, rmarkdown

Depends R (>= 3.0.1)

License GPL (>=2)

Encoding UTF-8

LazyData true

BuildManual yes

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Soren Jordan [aut, cre, cph],
Andrew Q. Philips [aut]

R topics documented:

`area.simulation.graph` *Create an area graph of a simulated response in a dynardl model*

Description

Create an area graph of a simulated response in a dynardl model

Usage

```
area.simulation.graph(x, changes = FALSE, bw = FALSE)
```

Arguments

x	a dynardl model with a simulation to be graphed
changes	whether the graph should be shown in levels of the dependent variable or in changes in levels. The default is FALSE
bw	should the colors be in black and white (for publication)? The default is FALSE

Details

When running dynardl, simulate must be true so that there is a simulation to graph.

Value

an area graph

Author(s)

Soren Jordan and Andrew Q. Philips

Examples

```
# Using the ineq data in dynamac
# Shocking Income Top 10
ardl.model <- dynardl(concern ~ incshare10 + urate, data = ineq,
       lags = list("concern" = 1, "incshare10" = 1),
       diffs = c("incshare10", "urate"),
       lagdiffs = list("concern" = 1),
       ec = TRUE, simulate = TRUE, range = 30,
       shockvar = "incshare10", graph = FALSE)
area.simulation.graph(ardl.model) # Shows absolute levels
area.simulation.graph(ardl.model, changes = TRUE)  # Shows changes from mean level
area.simulation.graph(ardl.model, bw = TRUE)  # Grayscale plots
```

dshift *Take first difference of a series.*

Description

Take first difference of a series.

Usage

```
dshift(x)
```

Arguments

x	a series to be differenced

Details

dshift assumes that the series are ordered, that there is no missing data, and that the time intervals are even.

Value

the differenced series

Author(s)

Soren Jordan and Andrew Q. Philips

Examples

```
x.var <- seq(0, 50, 5)
d.x.var <- dshift(x.var)
head(x.var)
head(d.x.var)
```

dynardl *Estimate and Simulate ARDL Model*

Description

Estimate autoregressive distributed lag model, simulate interesting values, and plot predictions

Usage

```
dynardl(formula, data = list(), lags = list(), diffs = list(),
  lagdiffs = list(), levels = list(), ec = FALSE, range = 20,
  sig = 95, time = 10, shockvar = list(),
  shockval = sd(data[[shockvar]]), sims = 1000, forceset = NULL,
  burnin = 20, expectedval = FALSE, trend = FALSE, constant = TRUE,
  graph = FALSE, rarea = TRUE, modelout = FALSE, simulate = TRUE)
```

Arguments

formula	a symbolic description of the model to be estimated. ARDL models are estimated using linear regression.
data	an optional data frame or list containing the the variables in the model.
lags	a list of variables and their corresponding lags to be estimated.
diffs	a vector of variables to be differenced. Only first differences are supported.
lagdiffs	a list of variables to be included in lagged differences.
levels	a vector of variables to be included in levels.
ec	estimate model in error-correction form, (i.e., y appears in first-differences). By default, ec is set to FALSE, meaning y will appear in levels.
range	the range of the simulation to be conducted
sig	the significance level (1 - p) that the user wants for the simulations. The default level is 95% significance (sig = 95).
time	the time period in the simulation for the variable to be shocked.
shockvar	the variable to be shocked. There is no default.
shockval	the amount by which the shockvar should be shocked. The default is one standard deviation of the shocked variable.
sims	the number of simulations to use in creating the quantities of interest. The default is 1000.
forceset	by default, in the simulations, variables in levels will be set to their means; variables in differences will be set to 0. Alternatively, users can set any variable in the model to a different value using a list in forceset.
burnin	the number of time periods to disregard before recording the values. These do not include the range; in other words, they take place before the range specified above. Users can increase the number of burnin periods, but probably should not decrease them. The default is 20.
expectedval	if this is TRUE, the simulation will record the expected values of across the sims by averaging errors. We recommend setting it to FALSE, since expected values do not account for stochastic error present in the model itself.
trend	include a linear time trend. The default is FALSE.
constant	include a constant. The default is TRUE.
graph	create a plot of the simulated response. The default is FALSE.
rarea	if graph = TRUE, create an area plot. If graph = TRUE and rarea = FALSE, a spike plot will be created.
modelout	print the regression estimates in the console
simulate	simulate the reponse. Otherwise, just the regression model will be estimated.

Details

Estimate an auto-regressive distributed lag model. Moreover, provide a graphical interpretation of the results by simulating the response of the dependent variable to shocks in one of the regressors.

Value

dynardl should always return an estimated model. It may or may not be simulated, according to the user. But the relevant regression output, model residuals (which can be tested for autocorrelation), and simulated response (if created) are stored in a list if the model is assigned to an object.

Author(s)

Soren Jordan and Andrew Q. Philips

Examples

```
# Using the inequality data from dynamac
ardl.model <- dynardl(concern ~ incshare10 + urate, data = ineq,
       lags = list("concern" = 1, "incshare10" = 1),
       diffs = c("incshare10", "urate"),
       ec = TRUE, simulate = FALSE)
summary(ardl.model$model)

# Adding a lagged difference of the dependent variable
ardl.model.2 <- dynardl(concern ~ incshare10 + urate, data = ineq,
       lags = list("concern" = 1, "incshare10" = 1),
       diffs = c("incshare10", "urate"),
       lagdiffs = list("concern" = 1),
       ec = TRUE, simulate = FALSE)
summary(ardl.model.2$model)

# Does not work: levels must appear as a vector
ardl.model.3 <- dynardl(concern ~ incshare10 + urate, data = ineq,
       lags = list("concern" = 1, "incshare10" = 1),
       levels = list("urate" = 1),
       diffs = c("incshare10", "urate"),
       lagdiffs = list("concern" = 1),
       ec = TRUE, simulate = FALSE)

ardl.model.3 <- dynardl(concern ~ incshare10 + urate, data = ineq,
       lags = list("concern" = 1, "incshare10" = 1),
       levels = c("urate"),
       diffs = c("incshare10", "urate"),
       lagdiffs = list("concern" = 1),
       ec = TRUE, simulate = FALSE)
```

---

dynardl.auto.correlated

Run a variety of autocorrelation tests on the residuals from a dynardl *model.*

Description

Run a variety of autocorrelation tests on the residuals from a dynardl model.

Usage

```
dynardl.auto.correlated(x, bg.type = "Chisq", digits = 3, order = NULL,
  object.out = FALSE)
```

Arguments

x	a dynardl model

bg.type	a character string for the type of Breusch-Godfrey test to run. The default is Chisq: the Chisq test statistic. The other option is F: the F-test statistic.
digits	the number of digits to round to when showing output. We recommend three.
order	the maximum order of serial autocorrelation to test when executing the Breusch-Godfrey test.
object.out	if TRUE, and dynardl.auto.correlated is assigned to an object, the AIC, BIC, and results will be stored for the user's convenience.

Details

This is a simple and convenient way to test whether the residuals from the dynardl model are white noise. As an aside, this is also why dynardl has a simulate = FALSE argument: users can ensure the model is white noise residuals before estimating a potentially time-intensive simulation. The output also reminds the user of the null hypotheses for the autocorrelation tests.

Value

The results of autocorrelation tests.

Author(s)

Soren Jordan and Andrew Q. Philips

Examples

```
# Using the ineq data from dynamac
ardl.model <- dynardl(concern ~ incshare10 + urate, data = ineq,
       lags = list("concern" = 1, "incshare10" = 1),
       diffs = c("incshare10", "urate"),
       lagdiffs = list("concern" = 1),
       ec = TRUE, simulate = FALSE)
dynardl.auto.correlated(ardl.model)
```

---

ineq *Data on public concern about economic inequality*

Description

A dataset from: Wright, Graham. 2017. "The political implications of American concerns about economic inequality." Political Behavior: 1-23. Online first.

Usage

```
data(ineq)
```

Format

A data frame with 49 rows and 9 variables:

year Year

mood Public mood liberalism

urate Unemployment rate

concern Concern about economic inequality

demcontrol Democratic control of congress

incshare10 Proportion of income of top 10 percent

csentiment Consumer sentiment

incshare01 Proportion of income of top 1 percent

Source

http://dx.doi.org/10.7910/DVN/UYUU9G

---

ldshift *Take the lagged first difference of a series.*

Description

Take the lagged first difference of a series.

Usage

```
ldshift(x, l)
```

Arguments

x	a series to be differenced
l	the number of lags

Details

`ldshift` assumes that the series are ordered, that there is no missing data, and that the time intervals are even.

Value

the lagged differenced series

Author(s)

Soren Jordan and Andrew Q. Philips

Examples

```
x.var <- runif(50)
ld.1.x.var <- ldshift(x.var, 1)
ld.2.x.var <- ldshift(x.var, 2)
head(x.var)
head(ld.1.x.var)
head(ld.2.x.var)
```

`lshift` *Take lag transformation of a series.*

Description

Take lag transformation of a series.

Usage

```
lshift(x, l)
```

Arguments

`x`	a series to be lagged
`l`	the number of lags

Details

`lshift` assumes that the series are ordered, that there is no missing data, and that the time intervals are even.

Value

the lagged series

Author(s)

Soren Jordan and Andrew Q. Philips

Examples

```
x.var <- runif(50)
l.1.x.var <- lshift(x.var, 1)
l.1.x.var <- lshift(x.var, 2)
head(x.var)
head(l.1.x.var)
head(l.1.x.var)
```

pssbounds *Perform Pesaran, Shin and Smith (2001) cointegration test*

Description

Perform Pesaran, Shin and Smith (2001) cointegration test

Usage

```
pssbounds(data = list(), obs = NULL, fstat = NULL, tstat = NULL,
  case = NULL, k = NULL, digits = 3, object.out = FALSE)
```

Arguments

data	an optional dynardl model. We highly recommend this option. Users are of course welcome to determine their own Case, t-statistic, F-statistic, and observations, but it is easier to have the model determine these quantities.
obs	number of observations
fstat	F-statistic of the joint test that variables in levels (except the lagged dependent variable) are equal to zero: `l.y = l.x1 + l.x2 +...+l.xk = 0`
tstat	t-statistic of the lagged dependent variable
case	specify certain restrictions on the constant and trend terms, since critical values differ by case. Case I: no intercept or trend, Case II: restricted intercept, no trend, Case III: unrestricted intercept with no trend, Case IV: unrestricted intercept and restricted trend, Case V: unrestricted intercept and trend. Case III is most frequently specified
k	number of regressors appearing in levels in the estimated model
digits	the number of digits to round to when showing output. We recommend three.
object.out	if TRUE, and dynardl.auto.correlated is assigned to an object, the AIC, BIC, and results will be stored for the user's convenience.

Details

pssbounds performs post-estimation cointegration testing using the bounds testing procedure from Pesaran, Shin, and Smith (2001). Since test statistics vary based on the number of k regressors, length of the series, these are required, in addition to F- and t-statistics.

Author(s)

Soren Jordan and Andrew Q. Philips

Examples

```
# Using the ineq data from dynamac
# We can get all the values by hand
ardl.model <- dynardl(concern ~ incshare10 + urate, data = ineq,
        lags = list("concern" = 1, "incshare10" = 1),
        diffs = c("incshare10", "urate"),
        lagdiffs = list("concern" = 1),
       ec = TRUE, simulate = FALSE)
```

```
summary(ardl.model)
pssbounds(obs = 47, fstat = 7.01578, tstat = -3.223, case = 3, k = 1)

# Or just pass a dynardl model.
pssbounds(ardl.model)
```

`spike.simulation.graph` *Create a spike graph of a simulated response in a dynardl model*

Description

Create a spike graph of a simulated response in a dynardl model

Usage

```
spike.simulation.graph(x, changes = FALSE, bw = FALSE)
```

Arguments

`x`	a dynardl model with a simulation to be graphed
`changes`	whether the graph should be shown in levels of the dependent variable or in changes in levels. The default is `FALSE`
`bw`	should the colors be in black and white (for publication)? The default is `FALSE`

Details

When running `dynardl`, `simulate` must be true so that there is a simulation to graph.

Value

a spike graph

Author(s)

Soren Jordan and Andrew Q. Philips

Examples

```
# Using the ineq data in dynamac
# Shocking Income Top 10
ardl.model <- dynardl(concern ~ incshare10 + urate, data = ineq,
       lags = list("concern" = 1, "incshare10" = 1),
       diffs = c("incshare10", "urate"),
       lagdiffs = list("concern" = 1),
       ec = TRUE, simulate = TRUE, range = 30,
       shockvar = "incshare10", graph = FALSE)
spike.simulation.graph(ardl.model) # Shows absolute levels
spike.simulation.graph(ardl.model, changes = TRUE)  # Shows changes from mean level
spike.simulation.graph(ardl.model, bw = TRUE)  # Grayscale plots
```

supreme.sup *Data on US Supreme Court Approval*

Description

A dataset from: Durr, Robert H., Andrew D. Martin, and Christina Wolbrecht. 2000. "Ideological divergence and public support for the Supreme Court." American Journal of Policial Science 44(4): 768-776.

Usage

```
data(supreme.sup)
```

Format

A data frame with 42 rows and 9 variables:

dcalc Supreme Court support

l_dcalc Lagged Supreme Court spport

iddiv Ideological divergence

mooddev Mean deviation of Mood

dirdev Mean deviation of percent liberal decisions

sg Rulings against Solicitor General's amicus briefs

laws Laws declared unconstitutional

presapp Approval of president

congapp Approval of Congress

Source

https://sites.lsa.umich.edu/admart/replication/

Index