Package ‘tseffects’

October 9, 2025

Title Dynamic (Causal) Inferences from Time Series (with Interactions)

Version 0.1.4

Maintainer Soren Jordan <sorenjordanpols@gmail.com>

Description

Autoregressive distributed lag (A[R]DL) models (and their reparameterized equivalent, the Generalized Error-Correction Model [GECM]) (see De Boef and Keele 2008 <doi:10.1111/j.1540-5907.2007.00307.x>) are the workhorse models in uncovering dynamic inferences. ADL models are simple to estimate; this is what makes them attractive. Once these models are estimated, what is less clear is how to uncover a rich set of dynamic inferences from these models. We provide tools for recovering those inferences in three forms: causal inferences from ADL models, traditional time series quantities of interest (short- and long-run effects), and dynamic conditional relationships.

URL https://sorenjordan.github.io/tseffects/,
https://github.com/sorenjordan/tseffects

BugReports https://github.com/sorenjordan/tseffects/issues

Imports mpoly, car, ggplot2, sandwich, stats, utils

Suggests knitr, rmarkdown, vdiffr, testthat (>= 3.0.0)

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

BuildManual yes

RoxygenNote 7.3.2

NeedsCompilation no

Author Soren Jordan [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0003-4201-1085>),
Garrett N. Vande Kamp [aut],
Reshi Rajan [aut]

Repository CRAN

Date/Publication 2025-10-09 12:10:02 UTC

Contents

`approval` *Data on US Presidential Approval*

Description

A dataset from: Cavari, Amnon. 2019. "Evaluating the President on Your Priorities: Issue Priorities, Policy Performance, and Presidential Approval, 1981–2016." Presidential Studies Quarterly 49(4): 798-826.

Usage

```
data(approval)
```

Format

A data frame with 140 rows and 14 variables:

APPROVE Presidential approval

APPROVE_ECONOMY Presidential approval: economy

APPROVE_FOREIGN Presidential approval: foreign affairs

MIP_MACROECONOMICS Salience (Most Important Problem): economy

MIP_FOREIGN Salience (Most Important Problem): foreign affairs

PARTY_IN Macropartisanship (in-party)

PARTY_OUT Macropartisanship (out-party)

PRESIDENT Numeric indicator for president

DIVIDEDGOV Dummy variable for divided government

ELECTION Dummy variable for election years

HONEYMOON Dummy variable for honeymoon period

UMCSENT Consumer sentiment

UNRATE Unemployment rate

APPROVE_L1 Lagged presidential approval

Source

doi:10.1111/psq.12594

GDTE.adl.plot — *Evaluate (and possibly plot) the General Dynamic Treatment Effect (GDTE) for an autoregressive distributed lag (ADL) model*

Description

Evaluate (and possibly plot) the General Dynamic Treatment Effect (GDTE) for an autoregressive distributed lag (ADL) model

Usage

```
GDTE.adl.plot(
  model = NULL,
  x.vrbl = NULL,
  y.vrbl = NULL,
  d.x = NULL,
  d.y = NULL,
  te.type = "pte",
  inferences.y = "levels",
  inferences.x = "levels",
  dM.level = 0.95,
  s.limit = 20,
  se.type = "const",
  return.data = FALSE,
  return.plot = TRUE,
  return.formulae = FALSE,
  ...
)
```

Arguments

`model`	the `lm` model containing the ADL estimates
`x.vrbl`	a named vector of the x variables and corresponding lag orders in the ADL model
`y.vrbl`	a named vector of the (lagged) y variables and corresponding lag orders in the ADL model
`d.x`	the order of differencing of the x variable in the ADL model
`d.y`	the order of differencing of the y variable in the ADL model
`te.type`	the desired treatment history. `te.type` determines the counterfactual series (h) that will be applied to the independent variable. -1 represents a Pulse Treatment Effect (PTE). 0 represents a Step Treatment Effect (STE). These can also be specified via `pte`, `pulse`, `ste`, and `step`. For others, see Vande Kamp, Jordan, and Rajan. The default is `pte`
`inferences.y`	does the user want resulting inferences about the dependent variable in levels or in differences? (For y variables where `d.y` is 0, this is automatically levels.) The default is `levels`

inferences.x	does the user want to apply the counterfactual treatment to the independent variable in levels or in differences? (For x variables where d.x is 0, this is automatically levels.) The default is levels
dM.level	level of significance of the GDTE, calculated by the delta method. The default is 0.95
s.limit	an integer for the number of periods to determine the GDTE (beginning at s = 0)
se.type	the type of standard error to extract from the ADL model. The default is const, but any argument to vcovHC from the sandwich package is accepted
return.data	return the raw calculated GDTEs as a list element under estimates. The default is FALSE
return.plot	return the visualized GDTEs as a list element under plot. The default is TRUE
return.formulae	return the formulae for the GDTEs as a list element under formulae (for the GDTEs) and binomials (for the treatment history). The default is FALSE
...	other arguments to be passed to the call to plot

Details

We assume that the ADL model estimated is well specified, free of residual autocorrelation, balanced, and meets other standard time-series qualities. Given that, to obtain causal inferences for the specified treatment history, the user only needs a named vector of the x and y variables, as well as the order of the differencing

Value

depending on return.data, return.plot, and return.formulae, a list of elements relating to the GDTE

Author(s)

Soren Jordan, Garrett N. Vande Kamp, and Reshi Rajan

Examples

```
# ADL(1,1)
# Use the toy data to run an ADL. No argument is made this is well specified; it is just expository
model <- lm(y ~ l_1_y + x + l_1_x, data = toy.ts.interaction.data)
test.pulse <- GDTE.adl.plot(model = model,
                                  x.vrbl = c("x" = 0, "l_1_x" = 1),
                                  y.vrbl = c("l_1_y" = 1),
                                  d.x = 0,
                                  d.y = 0,
                                  te.type = "pulse",
                                  inferences.y = "levels",
                                  inferences.x = "levels",
                                  s.limit = 20,
                                  return.plot = TRUE,
                                  return.formulae = TRUE)
names(test.pulse)
```

```
# Using Cavari's (2019) approval model (without interactions)
# Cavari's original model: APPROVE ~ APPROVE_ECONOMY + APPROVE_FOREIGN +
#     APPROVE_L1 + PARTY_IN + PARTY_OUT + UNRATE +
#      MIP_MACROECONOMICS + MIP_FOREIGN +
#     DIVIDEDGOV + ELECTION + HONEYMOON + as.factor(PRESIDENT)

cavari.model <- lm(APPROVE ~ APPROVE_ECONOMY + APPROVE_FOREIGN + MIP_MACROECONOMICS + MIP_FOREIGN +
     APPROVE_L1 + PARTY_IN + PARTY_OUT + UNRATE +
     DIVIDEDGOV + ELECTION + HONEYMOON + as.factor(PRESIDENT), data = approval)

# What if there was a permanent, one-unit change in the salience of foreign affairs?
cavari.step <- GDTE.adl.plot(model = cavari.model,
                                  x.vrbl = c("MIP_FOREIGN" = 0),
                                  y.vrbl = c("APPROVE_L1" = 1),
                                  d.x = 0,
                                  d.y = 0,
                                  te.type = "ste",
                                  inferences.y = "levels",
                                  inferences.x = "levels",
                                  s.limit = 10,
                                  return.plot = TRUE,
                                  return.formulae = TRUE)
```

`GDTE.calculator` *Generate the General Dynamic Treatment Effect (GDTE) for an autoregressive distributed lag (ADL) model, given Pulse Treatment Effects (PTEs)*

Description

Generate the General Dynamic Treatment Effect (GDTE) for an autoregressive distributed lag (ADL) model, given Pulse Treatment Effects (PTEs)

Usage

```
GDTE.calculator(d.x, d.y, h, limit, pte)
```

Arguments

`d.x`	the order of differencing of the x variable in the ADL model. (Generally, this is the same x variable used in `pte.calculator`)
`d.y`	the order of differencing of the y variable in the ADL model. (Generally, this is the same y variable used in `pte.calculator`)
`h`	an integer for the treatment history. `h` determines the counterfactual series that will be applied to the independent variable. -1 represents a Pulse Treatment Effect (PTE). 0 represents a Step Treatment Effect (STE). For others, see Vande Kamp, Jordan, and Rajan

`limit`	an integer for the number of periods (s) to determine the GDTE (beginning at 0)
`pte`	a list of PTEs used to construct the GDTE. We expect this will be provided by `pte.calculator`

Details

`GDTE.calculator` does no calculation. It generates a list of `mpoly` formulae that contain variable names that represent the GDTE in each period. The expectation is that these will be evaluated using coefficients from an object containing an ADL model with corresponding variables. It is used as a subfunction in both `GDTE.adl.plot` and `GDTE.gecm.plot`. Note: `mpoly` does not allow variable names with a .; variables passed to `GDTE.calculator` should not include this character

Value

a list of `limit` + 1 `mpoly` formulae containing the GDTE in each period

Author(s)

Soren Jordan, Garrett N. Vande Kamp, and Reshi Rajan

Examples

```
# ADL(1,1)
x.lags <- c("x" = 0, "l_1_x" = 1) # lags of x
y.lags <- c("l_1_y" = 1)
s <- 5
PTEs <- pte.calculator(x.vrbl = x.lags, y.vrbl = y.lags, limit = s)
# Assume that both x and y are in levels and we want a pulse treatment
GDTEs.pte <- GDTE.calculator(d.x = 0, d.y = 0, h = -1, limit = s, pte = PTEs)
GDTEs.pte
# Apply a step treatment
GDTEs.ste <- GDTE.calculator(d.x = 0, d.y = 0, h = 0, limit = s, pte = PTEs)
GDTEs.ste
```

`GDTE.gecm.plot` *Evaluate (and possibly plot) the General Dynamic Treatment Effect (GDTE) for a Generalized Error Correction Model (GECM)*

Description

Evaluate (and possibly plot) the General Dynamic Treatment Effect (GDTE) for a Generalized Error Correction Model (GECM)

Usage

```
GDTE.gecm.plot(
  model = NULL,
  x.vrbl = NULL,
  y.vrbl = NULL,
  x.vrbl.d.x = NULL,
  y.vrbl.d.y = NULL,
  x.d.vrbl = NULL,
  y.d.vrbl = NULL,
  x.d.vrbl.d.x = NULL,
  y.d.vrbl.d.y = NULL,
  te.type = "pte",
  inferences.y = "levels",
  inferences.x = "levels",
  dM.level = 0.95,
  s.limit = 20,
  se.type = "const",
  return.data = FALSE,
  return.plot = TRUE,
  return.formulae = FALSE,
  ...
)
```

Arguments

`model`	the `lm` model containing the GECM estimates
`x.vrbl`	a named vector of the x variables (of the lower level of differencing, usually in levels d = 0) and corresponding lag orders in the GECM model
`y.vrbl`	a named vector of the (lagged) y variables (of the lower level of differencing, usually in levels d = 0) and corresponding lag orders in the GECM model
`x.vrbl.d.x`	the order of differencing of the x variable (of the lower level of differencing, usually in levels d = 0) in the GECM model
`y.vrbl.d.y`	the order of differencing of the y variable (of the lower level of differencing, usually in levels d = 0) in the GECM model
`x.d.vrbl`	a named vector of the x variables (of the higher level of differencing, usually first differences d = 1) and corresponding lag orders in the GECM model
`y.d.vrbl`	a named vector of the y variables (of the higher level of differencing, usually first differences d = 1) and corresponding lag orders in the GECM model
`x.d.vrbl.d.x`	the order of differencing of the x variable (of the higher level of differencing, usually first differences d = 1) in the GECM model
`y.d.vrbl.d.y`	the order of differencing of the y variable (of the higher level of differencing, usually first differences d = 1) in the GECM model
`te.type`	the desired treatment history. `te.type` determines the counterfactual series (h) that will be applied to the independent variable. -1 represents a Pulse Treatment Effect (PTE). 0 represents a Step Treatment Effect (STE). These can also be

	specified via pte, pulse, ste, and step. For others, see Vande Kamp, Jordan, and Rajan. The default is pte
inferences.y	does the user want resulting inferences about the dependent variable in levels or in differences? The default is levels
inferences.x	does the user want to apply the counterfactual treatment to the independent variable in levels or in differences? The default is levels
dM.level	level of significance of the GDTE, calculated by the delta method. The default is 0.95
s.limit	an integer for the number of periods to determine the GDTE (beginning at s = 0)
se.type	the type of standard error to extract from the GECM model. The default is const, but any argument to vcovHC from the sandwich package is accepted
return.data	return the raw calculated GDTEs as a list element under estimates. The default is FALSE
return.plot	return the visualized GDTEs as a list element under plot. The default is TRUE
return.formulae	return the formulae for the GDTEs as a list element under formulae (for the GDTEs) and binomials (for the treatment history). The default is FALSE
...	other arguments to be passed to the call to plot

Details

We assume that the GECM model estimated is well specified, free of residual autocorrelation, balanced, and meets other standard time-series qualities. Given that, to obtain causal inferences for the specified treatment history, the user only needs a named vector of the x and y variables, as well as the order of the differencing. Internally, the GECM to ADL equivalences are used to calculate the GDTEs from the GECM

Value

depending on return.data, return.plot, and return.formulae, a list of elements relating to the GDTE

Author(s)

Soren Jordan, Garrett N. Vande Kamp, and Reshi Rajan

Examples

```
# ADL(1,1)
# Use the toy data to run a GECM. No argument is made this
#  is well specified or even sensible; it is just expository
model <- lm(d_y ~ l_1_y + l_1_x + l_1_d_y + d_x + l_1_d_x, data = toy.ts.interaction.data)
test.pulse <- GDTE.gecm.plot(model = model,
                                x.vrbl = c("l_1_x" = 1),
                                y.vrbl = c("l_1_y" = 1),
                                x.vrbl.d.x = 0,
                                y.vrbl.d.y = 0,
```

```
                                  x.d.vrbl = c("d_x" = 0, "l_1_d_x" = 1),
                                  y.d.vrbl = c("l_1_d_y" = 1),
                                  x.d.vrbl.d.x = 1,
                                  y.d.vrbl.d.y = 1,
                                  te.type = "pulse",
                                  inferences.y = "levels",
                                  inferences.x = "levels",
                                  s.limit = 10,
                                  return.plot = TRUE,
                                  return.formulae = TRUE)
names(test.pulse)
```

`pte.calculator` *Generate the Pulse Treatment Effect (PTE) for a given autoregressive distributed lag (ADL) model*

Description

Generate the Pulse Treatment Effect (PTE) for a given autoregressive distributed lag (ADL) model

Usage

```
pte.calculator(x.vrbl, y.vrbl, limit)
```

Arguments

`x.vrbl`	a named vector of the x variables and corresponding lag orders in an ADL model
`y.vrbl`	a named vector of the (lagged) y variables and corresponding lag orders in an ADL model
`limit`	an integer for the number of periods (s) to determine the PTE (beginning at 0)

Details

`pte.calculator` does no calculation. It generates a list of `mpoly` formulae that contain variable names that represent the PTE in each period. The expectation is that these will be evaluated using coefficients from an object containing an ADL model with corresponding variables. It is used as a subfunction in both `GDTE.adl.plot` and `GDTE.gecm.plot`. Note: `mpoly` does not allow variable names with a .; variables passed to `GDTE.calculator` should not include this character

Value

a list of limit + 1 `mpoly` formulae containing the PTE in each period

Author(s)

Soren Jordan, Garrett N. Vande Kamp, and Reshi Rajan

Examples

```
# ADL(1,1)
x.lags <- c("x" = 0, "l_1_x" = 1) # lags of x
y.lags <- c("l_1_y" = 1)
s <- 5
PTEs <- pte.calculator(x.vrbl = x.lags, y.vrbl = y.lags, limit = s)
PTEs
```

---

toy.ts.interaction.data

Simulated interactive time series data

Description

A simulated, well-behaved dataset of interactive time series data

Usage

```
data(toy.ts.interaction.data)
```

Format

A data frame with 50 rows and 23 variables:

time Indicator for time period

x Contemporaneous x

l_1_x First lag of x

l_2_x Second lag of x

l_3_x Third lag of x

l_4_x Fourth lag of x

l_5_x Fifth lag of x

d_x First difference of x

l_1_d_x First lag of first difference of x

l_2_d_x Second lag of first difference of x

l_3_d_x Third lag of first difference of x

z Contemporaneous z

l_1_z First lag of z

l_2_z Second lag of z

l_3_z Third lag of z

l_4_z Fourth lag of z

l_5_z Fifth lag of z

y Contemporaneous y

l_1_y First lag of y

l_2_y Second lag of y

l_3_y Third lag of y

l_4_y Fourth lag of y

l_5_y Fifth lag of y

d_y First difference of y

l_1_d_y First lag of first difference of y

l_2_d_y Second lag of first difference of y

d_2_y Second difference of y

l_1_d_2_y First lag of second difference of y

x_z Interaction of contemporaneous x and z

x_l_1_z Interaction of contemporaneous x and lagged z

z_l_1_x Interaction of lagged x and contemporaneous z

l_1_x_l_1_z Interaction of lagged x and lagged z

Index