

Package ‘gets.plm’

December 26, 2021

Type Package

Title General-to-Specific (GETS) Modelling of linear panel data models of class ‘plm’

Version 0.1

Date 2021-12-26

Author Genaro Sucarrat [aut, cre]

Maintainer Genaro Sucarrat <genaro.sucarrat@bi.no>

Description General-to-Specific (GETS) Modelling of linear panel data models of class ‘plm’, see Pretis, Reade and Sucarrat (2018) <doi:10.18637/jss.v086.i03>.

License GPL (>= 2)

Depends R (>= 3.3.0), zoo, gets, plm

Suggests

BugReports <https://github.com/gsucarrat/gets.plm/issues>

URL <https://CRAN.R-project.org/package=gets.plm>,
<http://www.sucarrat.net/R/gets/>

NeedsCompilation no

R topics documented:

gets.plm-package	1
gets.plm	2

Index	5
--------------	----------

gets.plm-package	<i>General-to-Specific (GETS) Modelling of linear panel data models of class ‘plm’</i>
------------------	--

Description

General-to-Specific (GETS) Modelling of linear panel data models of class ‘plm’, see Pretis, Reade and Sucarrat (2018) <doi:10.18637/jss.v086.i03>.

Details

Package: gets
 Type: Package
 Version: 1.0
 Date: 2021-12-26
 License: GPL-2

Author(s)

Genaro Sucarrat <http://www.sucarrat.net/>

Maintainer: Genaro Sucarrat

References

Felix Pretis, James Reade and Genaro Sucarrat (2018): 'Automated General-to-Specific (GETS) Regression Modeling and Indicator Saturation for Outliers and Structural Breaks'. *Journal of Statistical Software* 86, Number 3, pp. 1-44. <https://www.jstatsoft.org/article/view/v086i03>

Genaro Sucarrat (2019): 'User-Specified General-to-Specific and Indicator Saturation Methods'. <https://mpira.ub.uni-muenchen.de/96653/>

See Also

[getsFun](#), [plm](#)

gets.plm	<i>General-to-Specific (GETS) modelling of models of class 'plm' (linear paneldata models)</i>
----------	--

Description

General-to-Specific (GETS) modelling of models of class 'plm' (linear paneldata models), see [plm](#).

Usage

```
## S3 method for class 'plm'
gets(x, t.pval = 0.05, wald.pval = t.pval, do.pet = TRUE,
     keep = NULL, include.gum = FALSE, include.lcut = TRUE,
     include.empty = FALSE, max.paths = NULL, turbo = TRUE, tol = 1e-07,
     print.searchinfo = TRUE, alarm = FALSE, ...)
```

Arguments

x	an object of class 'plm'
t.pval	numeric value between 0 and 1. The significance level used for the two-sided regressor significance t-tests
wald.pval	numeric value between 0 and 1. The significance level used for the Parsimonious Encompassing Tests (PETs)

do.pet	logical. If TRUE, then a Parsimonious Encompassing Test (PET) against the GUM is undertaken at each regressor removal for the joint significance of all the deleted regressors along the current path. If FALSE, then a PET is not undertaken at each regressor removal
keep	NULL or an integer vector that indicates which regressors to be excluded from removal in the search
include.gum	logical. If TRUE, then the GUM (i.e. the starting model) is included among the terminal models. If FALSE, then the GUM is not included
include.1cut	logical. If TRUE, then the 1-cut model is added to the list of terminal models. If FALSE, then the 1-cut is not added, unless it is a terminal model in one of the paths
include.empty	logical. If TRUE, then the empty model is added to the list of terminal models. If FALSE, then the empty model is not added, unless it is a terminal model in one of the paths
max.paths	NULL or an integer greater than 0. If NULL, then there is no limit to the number of paths. If an integer (e.g. 1), then this integer constitutes the maximum number of paths searched (e.g. a single path)
turbo	logical. If TRUE, then (parts of) paths are not searched twice (or more) unnecessarily, thus yielding a significant potential for speed-gain. However, the checking of whether the search has arrived at a point it has already been comes with a slight computational overhead. Accordingly, if turbo=TRUE, then the total search time might in fact be higher than if turbo=FALSE. This happens if estimation is very fast, say, less than quarter of a second
tol	numeric value. The tolerance for detecting linear dependencies in the columns of the variance-covariance matrix when computing the Wald-statistic used in the Parsimonious Encompassing Tests (PETs), see the qr.solve function
print.searchinfo	logical. If TRUE, then a print is returned whenever simplification along a new path is started
alarm	logical. If TRUE, then a sound or beep is emitted (in order to alert the user) when the model selection ends
...	additional arguments (currently ignored)

Value

an object of class [plm](#)

Author(s)

Genaro Sucarrat, <http://www.sucarrat.net/>

See Also

[getsFun](#), [plm](#)

Examples

```
##create some artificial data:
##=====

iN <- 20 #no. of firms
```

```

iT <- 4 #no. of time periods (e.g. year)
iNiT <- iN*iT
set.seed(123)
Z <- rnorm(iNiT)
x <- matrix(rnorm(iNiT*10), iNiT, 10)
colnames(x) <- letters[1:10]
firm <- as.vector( t( 1:iN*matrix(rep(1,iNiT), iN, iT) ) )
year <- rep(2001:2004, iN)
mydata <- data.frame(firm, year, Z, x)
head(mydata)

##delete unnecessary stuff from workspace:
rm(iN, iT, iNiT, Z, x, firm, year)

##estimate gum, do gets:
##=====

mygum <-
  plm(Z ~ a + b + c + d + e + f + g + h + i + j,
    data=mydata)
summary(mygum)

myspecific <- gets(mygum)

myspecific <- gets(mygum, keep=2)

myspecific <- gets(mygum, t.pval=0.4)

##new gum, do gets:
##=====

mygum <-
  plm(Z ~ a + b + c + d + e + f + g + h + i + j,
    data=mydata, effect="twoways")
summary(mygum)

myspecific <- gets(mygum)

myspecific <- gets(mygum, keep=2)

myspecific <- gets(mygum, t.pval=0.4)

```

Index

- * **Climate Econometrics**
 - gets.plm, [2](#)
 - gets.plm-package, [1](#)
- * **Econometrics**
 - gets.plm, [2](#)
 - gets.plm-package, [1](#)
- * **Financial Econometrics**
 - gets.plm, [2](#)
 - gets.plm-package, [1](#)
- * **Panel Data**
 - gets.plm, [2](#)
- * **Statistical Models**
 - gets.plm, [2](#)
 - gets.plm-package, [1](#)
- * **Time Series**
 - gets.plm, [2](#)
 - gets.plm-package, [1](#)

gets.plm, [2](#)
gets.plm-package, [1](#)
getsFun, [2](#), [3](#)

plm, [2](#), [3](#)

qr.solve, [3](#)