ivlewbel: Uses heteroscedasticity to estimate mismeasured and endogenous regressor models

Fernihough, A. ivlewbel: Uses heteroscedasticity to estimate mismeasured and endogenous regressor models.

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person’s rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.
Package ‘ivlewbel’

July 2, 2014

Type Package

Title Uses heteroscedasticity to estimate mismeasured and endogenous regressor models

Version 1.1

Date 2014-05-28

Author Alan Fernihough

Maintainer Alan Fernihough <alan.fernihough@gmail.com>

Description GMM estimation of triangular systems using heteroscedasticity based instrumental variables as in Lewbel (2012)

License GPL-2 | GPL-3

Depends stats, gmm, plyr, lmtest

NeedsCompilation no

Repository CRAN

Date/Publication 2014-05-28 18:37:46

R topics documented:

lewbel .................................................. 2

Index 4
lewbel

Identification using heteroscedasticity

Description

This function estimates the model parameters and associated standard errors for a linear regression model with one or more endogenous regressors. Identification is achieved through heteroscedastic covariance restrictions within the triangular system.

Usage

lewbel(formula, data, clustervar = NULL, robust = TRUE)

Arguments

- formula: an object of class “formula” (or one that can be coerced to that class).
- data: the data frame containing these data. This argument must be used.
- clustervar: a character value naming the cluster on which to adjust the standard errors and test statistics.
- robust: if TRUE the function reports standard errors and test statistics that have been corrected for the presence heteroscedasticity using White’s method.

Details

The formula follows a four-part specification. Each part is separated by a vertical bar character “|”. The following formula is an example: y2 ~ y1 | x1 + x2 + x3 | x1 + x2 | z1. Here, y2 is the dependent variable and y1 is the endogenous regressor. The code x1 + x2 + x3 represents the exogenous regressors whereas the third part x1 + x2 specifies the exogenous heteroscedastic variables from which the instruments are derived. The final part z1 is optional, allowing the user to include traditional instrumental variables. If both robust=TRUE and !is.null(clustervar) the function overrides the robust command and computes clustered standard errors and test statistics adjusted to account for clustering. This function computes partial F-statistics that indicate potentially weak identification. In cases where there is more than one endogenous regressor the Angrist-Pischke (2009) method for multivariate first-stage F-statistics is employed.

Value

- coef.est: a coefficient matrix with columns containing the estimates, associated standard errors, test statistics and p-values.
- call: the matched call.
- num.obs: the number of observations.
- j.test: J-test for overidentifying restrictions.
- f.test.stats: Partial F-test statistics for weak IV detection.
References


Examples

```r
set.seed(1234)
n = 1000
x1 = rnorm(n, 0, 1)
xR = rnorm(n, 0, 1)
u = rnorm(n, 0, 1)
s1 = rnorm(n, 0, 1)
sR = rnorm(n, 0, 1)
ove = rnorm(n, 0, 1)
z1 = rnorm(n, 0, 1)
e1 = u + exp(x1)*s1 + exp(xR)*s1
eR = u + exp(-x1)*sR + exp(-xR)*sR
y1 = 1 + x1 + xR + ove + eR + 2*z1
y2 = 1 + x1 + xR + y1 + 2*ove + e1
data = data.frame(y2, y1, x1, xR, z1)

lewbel(formula = y2 ~ y1 | x1 + xR | x1 + xR, data = data)
lewbel(formula = y2 ~ y1 | x1 + xR | x1 + xR | z1, data = data)
```
Index

lewbel, 2

print.lewbel.model(lewbel), 2