linearFeedbackModel

May 18, 2016

R topics documented:

`lfm` *Estimate the linear feedback model in Blundell, Griffith and Windmeijer (2002)*

Description

Estimate the linear feedback model in Blundell, Griffith and Windmeijer "Individual effects and dynamics in count data models", Journal of Econometrics 108 (2002) 113-131

Usage

```
lfm(formula, data, effect = "individual", model = "onestep")
```

Arguments

`formula`	Similar to the `pgmm()` function in package `plm`. A symbolic description for the model to be estimated. Indicate a multi-part formula, the first two parts describing the covariates and the gmm instruments and, if any, the third part the 'normal' instruments. The first independent variable must be the lag of the dependent variable.
`data`	A `pdata.frame`.
`effect`	Either `"individual"` or `"twoways"`. The former only includes individual fixed effects while the latter also includes time fixed effects.
`effect`	Either `"onestep"` or `"twosteps"`. Whether to do one-step GMM or two-step GMM.

Value

`call` The matched call

`coefficients` The estimated coefficient

`D` The average of the Jacobian of the sample moment conditions over each individual

`fitted.values` `data.frame` of fitted values

`first` The first stage estimates

`fixed.effects` Estimates of the individual fixed effects

`model` The variables used for estimation for each individual

`residuals` `data.frame` of residuals

`vcov` The covariance matrix of the coefficients

`W1` The first-stage weight matrix used

`W2` The second-stage (efficient) weight matrix used (only returned if `model = "twosteps"` is used)

`Z` The instrument matrix for each individual

Examples

```
library(linearFeedbackModel)
library(plm)
set.seed(123)

detach("package:linearFeedbackModel", unload = TRUE)
install.packages("../linearFeedbackModel", repos = NULL, type = "source")
library(linearFeedbackModel)

# Create some data - follow example in Blundell, Griffith and Windmeijer (2002):

nT          <- 8     # Number of time periods
N           <- 500   # Number of individuals
gamma       <- 0.5   # Coefficient on lagged dependent variable
beta        <- 0.5   # Coefficient on x
rho         <- 0.5   # For the starting values of x
tau         <- 0.1   # For effect of individual effects on x
var_eta     <- 0.5   # Variance of distribution of individual fixed effects
var_epsilon <- 0.5   # Variance of distribution of x

# Parameters to be estimated:
theta <- c(gamma, beta)

# Draw individual effects:
eta <- rnorm(N, mean = 0, sd = sqrt(var_eta))

# Draws for starting values of x:
xi <- rnorm(N, mean = 0, sd = sqrt(var_epsilon / (1 - rho^2)))

# Draw initial conditions for x and y:
```

```
x0 <- (tau / (1 - rho)) * eta + xi
y0 <- sapply(1:N, function(i) rpois(1, lambda = exp(x0 * beta + eta[i])))

# Dataset to be created: List for each time period
df <- vector(”list”, length = nT)

# Create first time period:
x <- rho * x0 + tau * eta + rnorm(N, mean = 0, sd = sqrt(var_epsilon))
y <- sapply(1:N, function(i)
            rpois(1, lambda = gamma * y0 + exp(beta * x0[i] + eta[i])))
df[[1]] <- data.frame(i = 1:N, t = 1, x = x, y = y)

# Create all following time periods:
for (t in 2:nT) {
  x <- rho * df[[t - 1]]$x + tau * eta +
       rnorm(N, mean = 0, sd = sqrt(var_epsilon))
  y <- sapply(1:N, function(i) rpois(1, lambda = gamma * df[[t - 1]]$y[i] +
                                     exp(beta * df[[t - 1]]$x[i] + eta[i])))
  df[[t]] <- data.frame(i = 1:N, t = t, x = x, y = y)
}
# Combine into one dataset:
df <- do.call(rbind, df)

# Convert to pdata.frame:
data <- pdata.frame(df, index = c(”i”, ”t”))

test <- lfm(y ~ lag(y, k = 1) + x | lag(y, k = 2:4) + lag(x, k = 1:4),
            data = data, effect = ”individual”, model = ”onestep”)
print(summary(test))
```

Index