create_fframe

Add empty rows with time stamps to each cross-sectional unit in the panel

Description

Creates a forecast frame as required by the predict.pmpp() method. To each cross-sectional unit in the data, a specified number of rows are added that contain only this unit’s ID and the selected time ID.

Usage

create_fframe(indata, timestamps, panel_ind = colnames(indata[, 1:2]), overwrite = FALSE)

Arguments

indata data.frame with a panel structure
timestamps vector of time IDs for the added time periods
panel_ind vector of length 2 indicating names of variables indexing units and time periods respectively
overwrite logical; if TRUE, existing rows in the data are overwritten with empty rows if their time ID is in timestamps

Value

A data.frame with empty rows added.

Author(s)

Michal Oleszak
**get_kernel**

Obtain 2D kernel density estimates given sufficient statistics for lambdas and the initial data $Y_0$

**Description**

Obtain 2D kernel density estimates given sufficient statistics for lambdas and the initial data $Y_0$

**Usage**

```r
get_kernel(lambdas, sigma2, dens_grid, N, T, Y0)
```

**Arguments**

- `lambdas`: sufficient statistics for the intercept term
- `sigma2`: variance of the shocks
- `dens_grid`: grid over which the density is to be computed
- `N`: cross-sectional dimension of the data
- `T`: time dimension of the data
- `Y0`: initial observations of the dependent variable

---

**get_lambda0**

Produce sufficient statistics ($\lambda_0$) given the common coefficients ($\rho_0$)

**Description**

Produce sufficient statistics ($\lambda_0$) given the common coefficients ($\rho_0$)

**Usage**

```r
get_lambda0(rho, alpha = rep(0, n_alpha), N, T, n_alpha, Y_mat, X_mat, W, Z_mat)
```
Arguments

rho         lagged dependent variable coefficients
alpha       external variables coefficients
N           cross-sectional dimension of the data
T           time dimension of the data
n_alpha     number of external variables
Y_mat       dependent variable matrix
X_mat       lagged dependent variable matrix
W           cross-sectionally invariant variables - not used now
Z_mat       external variable matrix

get_sigma2  Produce variance of the shocks estimated using GMM residues (sigma2_0) given the common coefficients (rho0)

Description

Produce variance of the shocks estimated using GMM residues (sigma2_0) given the common coefficients (rho0)

Usage

get_sigma2(rho, alpha = 0, common_par_method, X_star, Y_star, Z_star,
           X_mat, Y_mat, Z_mat, n_alpha)

Arguments

rho         lagged dependent variable coefficients
alpha       external variables coefficients
common_par_method
            method for estimating common parameters
X_star      auxiliary matrix for OFD transformation
Y_star      auxiliary matrix for OFD transformation
Z_star      auxiliary matrix for OFD transformation
X_mat       lagged dependent variable matrix
Y_mat       dependent variable matrix
Z_mat       external variable matrix
n_alpha     number of external variables
GMM_parametric

Produce posterior means of lambda's for the parametric GMM implementation given autoregressive coefficient (rho)

Description

Produce posterior means of lambda's for the parametric GMM implementation given autoregressive coefficient (rho)

Usage

GMM_parametric(rho, alpha = 0, optim_method, init, n_lambda, n_alpha,
X_mat, Y_mat, Z_mat, W, T, N, aux_Y0, common_par_method, X_star, Y_star,
Z_star)

Arguments

- rho: lagged dependent variable coefficients
- alpha: external variables coefficients
- optim_method: optimization method
- init: initial values for the optimization routine
- n_lambda: number of columns in W; currently always set to 1
- n_alpha: number of external variables
- X_mat: lagged dependent variable matrix
- Y_mat: dependent variable matrix
- Z_mat: external variable matrix
- W: cross-sectionally invariant variables - not used now
- T: time dimension of the data
- N: cross-sectional dimension of the data
- aux_Y0: auxiliary matrix with initial observations of the dependent variable
- common_par_method: method for estimating common parameters
- X_star: auxiliary matrix for OFD transformation
- Y_star: auxiliary matrix for OFD transformation
- Z_star: auxiliary matrix for OFD transformation
**Description**

State-of-the-art gaussian kernel density estimator for one-dimensional data. The estimator does not use the commonly employed 'gaussian rule of thumb'. As a result, it outperforms many plug-in methods on multimodal densities with widely separated modes. This function is the cleaned-up version of the code written and published by Z. I. Botev at: [http://web.maths.unsw.edu.au/~zdravkobotev/](http://web.maths.unsw.edu.au/~zdravkobotev/)

**Usage**

```r
d = kde(data, n, MIN, MAX)
```

**Arguments**

- `data`: a vector of data from which the density estimate is constructed;
- `n`: the number of mesh points used in the uniform discretization of the interval [MIN, MAX]; n has to be a power of two; if n is not a power of two, then n is rounded up to the next power of two; the default value of n is n=2^12;
- `MIN`: minimum of the interval [MIN, MAX] on which the density estimate is constructed; default value: MIN = min(data) - Range / 10
- `MAX`: maximum of the interval [MIN, MAX] on which the density estimate is constructed; default value: MAX = max(data) + Range / 10

**Value**

A matrix with two rows of length n, where the second row contains the density values on the mesh in the first row.

**References**


**Examples**

```r
set.seed(1)
data <- c(rnorm(10^3), rnorm(10^3) * 2 + 30)
d <- kde(data)
plot(d[1,], d[2,], type = 'l', xlab = 'x', ylab = 'density f(x)')
```
**kde2D**

Compute a two-dimensional kernel density estimate

---

**Description**

The kernel is assumed to be Gaussian. Bandwidth matrix is diagonal. The two bandwidth parameters are chosen optimally without ever using/assuming any parametric model for the data or any "rules of thumb". Unlike many other procedures, this one is immune to accuracy failures in the estimation of multimodal densities with widely separated modes. This function in meant to be the R implementation of the MATLAB kde2d() function written and published by Z. I. Botev at: [http://web.maths.unsw.edu.au/~zdravkobotev/](http://web.maths.unsw.edu.au/~zdravkobotev/)

**Usage**

```r
kde2D(data, n = 2^8, limits = NULL)
```

**Arguments**

- `data` : N by 2 matrix with the two variables as columns
- `n` : size of the n by n grid over which the density is computed
- `limits` : limits of the bounding box over which the density is computed; format: `c(lower_Xlim, upper_Xlim, lower_Ylim, upper_Ylim)`

**Value**

A list with bandwidth, density and grids for the two dimensions.

**Author(s)**

Michal Oleszak

**References**

loglikelihood_GMM  
Produce negative log-likelihood in the GMM case

Description
Produce negative log-likelihood in the GMM case

Usage
loglikelihood_GMM(theta, rho_GMMpar, alpha_GMMpar, sigma2_GMMpar, n_alpha, 
X_mat, Y_mat, Z_mat, W, T, N, aux_Y0)

Arguments
theta  
vector of homogeneous parameters
rho_GMMpar  
lagged dependent variables coefficient estimates from the GMM
alpha_GMMpar  
external variables coefficient estimates from the GMM
sigma2_GMMpar  
variance of the shocks estimated using GMM residuals
n_alpha  
number of external variables
X_mat  
lagged dependent variable matrix
Y_mat  
dependent variable matrix
Z_mat  
external variable matrix
W  
cross-sectionally invariant variables - not used now
T  
time dimension of the data
N  
cross-sectional dimension of the data
aux_Y0  
auxiliary matrix with initial observations of the dependent variable

loglikelihood_QMLE  
Produce (negative) log marginal likelihood for QMLE with correlated random coefficients

Description
Produce (negative) log marginal likelihood for QMLE with correlated random coefficients

Usage
loglikelihood_QMLE(param, n_alpha, X_mat, Y_mat, Z_mat, W, T, N, aux_Y0)
**Arguments**

- `param`: vectors of parameters to optimize over
- `n_alpha`: number of external variables
- `X_mat`: lagged dependent variable matrix
- `Y_mat`: dependent variable matrix
- `Z_mat`: external variable matrix
- `W`: cross-sectionally invariant variables - not used now
- `T`: time dimension of the data
- `N`: cross-sectional dimension of the data
- `aux_Y0`: auxiliary matrix with initial observations of the dependent variable

---

**plot.pmpp** *(Plot method for objects of class pmpp.)*

**Description**

Plot method for objects of class pmpp.

**Usage**

```r
## S3 method for class 'pmpp'
plot(x, ...)
```

**Arguments**

- `x`: object of class pmpp, as returned by pmpp()
- `...`: other arguments passed to the method

**Value**

No object is returned. Displays a ggplot of density of the estimated individual-specific effects.

**Examples**

```r
data(EmplUK, package = "plm")
pmpp_model <- pmpp(dep_var = "emp", data = EmplUK)
plot(pmpp_model)
```
**Posterior Mean Panel Predictor for dynamic panel modelling**

**Description**

This function estimates parameters of the Posterior Mean Panel Predictor (PMPP) model based on an empirical-Bayes approach to obtain unit-specific fixed effects.

**Usage**

```r
pmpp(dep_var, data, panel_ind = colnames(data[, 1:2]), exp_var = NULL, 
    csi_var = NULL, post_mean_method = "gaussian", 
    common_par_method = "QMLE", optim_method = "quadratic", 
    dens_grid = 2^10, gmm_model = "twosteps", gmm_inst = 99, 
    pure_data = FALSE)
```

**Arguments**

- `dep_var`: character string indicating name of dependent variable
- `data`: data.frame or matrix with input data
- `panel_ind`: vector of length 2 indicating names of variables indexing units and time periods respectively
- `exp_var`: vector of character strings indicating names of exogeneous explanatory variables
- `csi_var`: vector of character strings indicating names of cross-sectionally invariant explanatory variables; feature not supported yet
- `post_mean_method`: method for estimating the heterogeneous intercept parameters, one of "gaussian", "kernel"
- `common_par_method`: method for estimating the common parameters, one of "QMLE", "GMM_ABond", "GMM_BBond", GMM_ABover", "GMM_SSYS"
- `optim_method`: which optimisation routine to use, one of "gradient", "quadratic", "annealing"
- `dens_grid`: size of the grid over which data is interpolated for kernel density estimation; larger value may yield higher accuracy, but increases computation time
- `gmm_model`: number of steps for computing optimal GMM matrix, one of "onestep", "twosteps", "threesteps"; "threesteps" can be used for "GMM_SSYS" only
- `gmm_inst`: number of lagged values of the dependent variable to be used as GMM instruments in Arellano-Bond/Blundell-Bond setting
- `pure_data`: if TRUE, removes indexing/subsetting from model’s call on data, facilitating use in a loop
The PMPP model is a two-step procedure. First, the homogeneous parameters are estimated using one of the QMLE or GMM-based methods:

- Arellano-Bond estimator (Difference GMM),
- Arellano-Bover estimator (Level GMM),
- Blundell-Bond estimator (System GMM),
- Sub-optimal System GMM estimator,
- Quasi-Maximum Likelihood estimator.

Parameter `common_par_method` can be used to select the method for common parameters estimation. All the above methods only provide estimates of the homogeneous parameters, i.e. the ones measuring impact of lagged response and external variables. The intercept is removed in the estimation process. In the second step of the PMPP modelling, the individual-specific intercept is calculated based on the formula for posterior mean (Tweedie's Formula). It involves approximating certain density function, which can be done in two ways:

- Parametrically, assuming Gaussian distribution,
- Using a 2D kernel density estimator.

Parameter `post_mean_method` can be used to select the method used for intercept estimation. For technical details on the methods, see the references.

An object of class `pmpp`; a list with parameter estimates, fitted values, residuals, in-sample error measures and information on the data and function call.

Michal Oleszak


data(EmplUK, package = "plm")
pmpp_model <- pmpp(dep_var = "emp", data = EmplUK)
pmpp_data

Transform a single variable in the matrix format into the long panel format

Description

This function transforms a matrix of data with cross-sectional and time dimensions in rows and columns or columns and rows into a panel-structured, 3-column data frame

Usage

pmpp_data(indata, t_dim = “cols”, var_name = “Y”)

Arguments

- indata: matrix with a single variable
- t_dim: character string, one of: ‘cols’, ’rows’; whether time dimension in indata is across columns or rows
- var_name: character string; name of the variable in indata

Value

A data.frame with 3 columns: unit, time and variable’s values.

Author(s)

Michal Oleszak

Examples

set.seed(1)
matrix_var <- matrix(rnorm(100), nrow = 20)
panel_var <- pmpp_data(matrix_var)

pmpp_predinterval

Random-Window Block Bootstrap for prediction intervals for PMPP model

Description

Produces prediction intervals for Posterior Mean Panel Predictor model by means of resampling with replacement from model’s residuals. Block Bootstrap method takes into account heteroskedasticity of the error terms, both across units and over time. Block window is chosen randomly.
Usage

pmpp_predinterval(model, fframe, boot_reps = 1000, block_size = NULL,
                    confidence = 0.95, iter = NULL)

Arguments

model PMPP model, as returned by pmpp()
fframe data.frame with the same columns as input data to model, but with empty rows
        added to each cross-sectional unit, as created by create_fframe()
boot_reps integer; number of bootstrap replications
block_size integer; width of the re-sampled block of residuals
confidence numeric in (0,1); confidence level of the interval
iter iterating constant, to be used in a loop when extraction from call is needed

Value

A data.frame with panel indices, lower and upper bounds and midpoint.

Author(s)

Michal Oleszak

References

        from consumer goods sector.", Erasmus University Thesis Repository

Examples

## Not run: data(EmplUK, package = "plm")
pmpp_model <- pmpp(dep_var = "emp", data = EmplUK)
my_fframe <- create_fframe(EmplUK, 1983:1985)
intervals <- pmpp_predinterval(pmpp_model, my_fframe, boot_reps = 10)
## End(Not run)

post_mean_lambda_par

Provide posterior means of lambda_i's based on the Parametric Post-
erior Mean estimator with correlated random coefficients

Description

Provide posterior means of lambda_i’s based on the Parametric Posterior Mean estimator with cor-
related random coefficients
predict.pmpp

Compute forecasts with a PMPP model

Usage
predict.pmpp(object, fframe = NULL, iter = NULL, ...)

Arguments
- object: an object of class pmpp()
- fframe: data.frame with the same columns as input data to model, but with empty rows added to each cross-sectional unit, as created by create_fframe()
- iter: iterating constant, to be used in a loop when extraction from call is needed
- ...: other arguments passed to the method

Value
A data.frame with predicted and true values.

Author(s)
Michal Oleszak

Examples
```r
data(EmplUK, package = "plm")
pmpp_model <- pmpp(dep_var = "emp", data = EmplUK)
my_fframe <- create_fframe(EmplUK, 1983:1985)
prediction <- predict(pmpp_model, my_fframe)
```
### Description

Computes an enhanced version of the Blundell-Bond (System-GMM) estimator for panel data by means of replacing the standard GMM-weighting matrix by its sub-optimal version, thus increasing estimator's efficiency.

### Usage

```r
ssys_gmm(Y, model = c("onestep", "twosteps", "threesteps"))
```

### Arguments

- **Y**: matrix of size (T x N) with the dependent variable
- **model**: one of: onestep, twosteps, threesteps; more steps should increase efficiency, but might be computationally infeasible (a singular matrix needs to be inverted); if this is the case, generalised inverse is used

### Value

The estimated value of the auto-regressive parameter.

### Author(s)

Michal Oleszak

### References


### Summary.pmpp

**Summary method for objects of class pmpp.**

### Description

Summary method for objects of class pmpp.

### Usage

```r
## S3 method for class 'pmpp'
summary(object, file = "", ...)```
Arguments

- object: object of class pmpp, as returned by pmpp()
- file: a connection, or a character string naming the file to print to
- ... other parameters passed further

Value

A summary object for class pmpp.
Index

create_fframe, 2
get_kernel, 3
get_lambda0, 3
get_sigma2, 4
GMM_parametric, 5
kde, 6
kde2D, 7
loglikelihood_GMM, 8
loglikelihood_QMLE, 8
plot.pmpp, 9
pmpp, 10
pmpp_data, 12
pmpp_predinterval, 12
post_mean_lambda_par, 13
predict.pmpp, 14
ssys_gmm, 15
summary.pmpp, 15