

# Package ‘OrthoPanels’

May 7, 2026

**Title** Dynamic Panel Models with Orthogonal Reparameterization of Fixed Effects

**Version** 1.2-4

**Description** Implements the orthogonal reparameterization approach recommended by Lancaster (2002) to estimate dynamic panel models with fixed effects (and optionally: panel specific intercepts). The approach uses a likelihood-based estimator and produces estimates that are asymptotically unbiased as N goes to infinity, with a T as low as 2.

**Depends** R (>= 3.1.0)

**Imports** MASS

**Suggests** testthat, knitr, rmarkdown

**License** GPL (>= 3)

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Maintainer** Mark Pickup <mark.pickup@sfu.ca>

**Author** Davor Cubranic [aut],  
Mark Pickup [aut, cre],  
Paul Gustafson [aut],  
Geoffrey Evans [aut],  
Jonoska Stojkova Biljana [ctb]

**Repository** CRAN

**Date/Publication** 2022-06-09 05:20:01 UTC

## Contents

OrthoPanels-package . . . . .	2
abond_panel . . . . .	3
BES_panel . . . . .	4

caterplot . . . . .	4
caterplot_longRun . . . . .	5
confint.opm . . . . .	6
DIC . . . . .	7
hist.opm . . . . .	8
longRunEffects . . . . .	9
opm . . . . .	9
plot.opm . . . . .	11
p_rho . . . . .	12
quantile.opm . . . . .	13
<b>Index</b>	<b>14</b>

---

OrthoPanels-package     *OrthoPanels: Orthogonalized Panel Model*

---

## Description

This package includes the function `opm()`, which implements the orthogonal reparameterization approach recommended by Lancaster (2002) to estimate dynamic panel models with fixed effects (and optionally: panel specific intercepts). The OLS estimator for such models is biased with a fixed (small)  $N$  (Nickell 1981). Equivalently, a maximum likelihood estimation leads to an incidental parameters problem (Neyman and Scott 1948; Lancaster 2000). The approach by Lancaster (2002) uses an orthogonal reparameterization of the fixed effects to produce a likelihood-based estimator of the remaining parameters that is exact and consistent as  $N$  approaches infinity for  $T$  greater than or equal to 2.

## Details

Orthopanels can accomodate unbalanced panel data, in that some respondents may drop out early (attrition) and some respondents may enter the panel late (refreshment). It is assumed that once respondents enter the panel, they will have observations up until they dropout and then NAs in subsequent waves. The estimation is conducted under the assumption that the data is missing at random

## References

- Lancaster, T. (2000) The incidental parameter problem since 1948. *Journal of Econometrics*, **95**, 391–413.
- Lancaster, T. (2002) Orthogonal parameters and panel data. *Review of Economic Studies*, **69**, 647–666.
- Neyman, J. and Scott, E. L. (1948) Consistent estimation from partially consistent observations. *Econometrica*, **16**, 1–32.
- Nickell, S. (1981) Biases in dynamic models with fixed effects. *Econometrica*, **49**, 1417–1426.

---

abond\_panel

*UK Company Data Panel*

---

### Description

The dynamics of labour demand of firm *id* in the United Kingdom in year *year* as a function of real product wages, gross capital stock and industry output. This is done using the data used by Arellano and Bond (1991).

### Format

A data frame with 813 rows and 16 variables

### Details

A survey of 1845 respondents using 3 waves of panel survey data from the 2010 British Election Study. The variables are as follows:

- id case number
- year time wave
- n log of employment in firm id at time year
- w natural log of the real product wage
- k natural log of gross capital stock
- ys natural log of industry output
- l\_w lag of w
- l\_k lag of k
- l2\_k two-step lag of k
- l\_ys lag of ys
- l2\_ys two-step lag of ys
- yr1980..yr1984 time dummies

### References

Arellano M., and Bond S. (1991) Some Tests of Specification for Panel Data: Monte Carlo Evidence and an Application to Employment Equations. *Review of Economic Studies*, **58(2)**, 277–297.

---

 BES\_panel

*Responses from the 2010 British Election Study*


---

### Description

A survey of 1845 respondents using 3 waves of panel survey data from the 2010 British Election Study. The variables are as follows:

### Format

A data frame with 5535 rows and 11 variables

### Details

- n case number
- t time wave
- Econ Assessment of change in the national economic situation over the past 12 months (1-5, 1='got a lot worse', 5='got a lot better')
- Clegg Evaluation of Liberal Party leader Nick Clegg (0-10, 0='strongly dislike' and 10='strongly like')
- Brown Evaluation of Labour Party leader Gordon Brown
- Cameron Evaluation of Conservative Party leader David Cameron
- Approve Approval of the government, as expressed by feeling about the ruling Labour Party (0-10, 0='strongly dislike', 10='strongly like')
- NHS Assesment of the current government's handling of the National Health Service (1-5, 1='very badly', 5='very well')
- Terror Assesment of the current government's handling of terrorism (1-5, 1='very badly', 5='very well')
- PID Personal identification with the Labour Party (0/1, 0='no', 1='yes')
- Tax Preference for policy on taxes and health and social spending (0-10, 0='cut taxes a lot and spend much less', 10='increase taxes a lot and spend much more')

---

 caterplot

*Caterpillar Plots of opm Model Parameters*


---

### Description

Creates side-by-side plots of equal-tailed credible intervals of opm model parameters. The intervals are displayed as horizontal lines, with 90% interval using a thicker line width and 95% interval a thinner one. The posterior median is indicated with a dot.

**Usage**

```
caterplot(
  x,
  parm,
  main = paste("Caterpillar plot of", xname),
  xlab = "Range of parameter samples",
  labels = colnames(ranges)
)
```

**Arguments**

x	an instance of class opm
parm	a specification of which parameters are to be plotted, either a vector of names ("rho", "sig2" and "beta" are the only legal values) or a vector of positional indices. If missing, all parameters are considered.
main, xlab	useful defaults for the plot title and X-axis label
labels	labels for each parameter's interval: see <a href="#">axis</a>

**Value**

A matrix of 2.5%, 5%, 50%, 95%, and 97.5% quantiles for each of the desired parameters, with parameters arranged in columns.

**Examples**

```
## Not run:
caterplot(o, main = NULL, labels = expression(alpha, beta, sigma^2))

## End(Not run)
```

---

caterplot_longRun	<i>Caterpillar Plots of long run effects based on opm Model Parameters</i>
-------------------	--

---

**Description**

Creates side-by-side plots of equal-tailed credible intervals of opm the long run effects parameters. The intervals are displayed as horizontal lines, with 90% interval using a thicker line width and 95% interval a thinner one. The posterior median is indicated with a dot.

**Usage**

```
caterplot_longRun(
  x,
  parm = NULL,
  main = "Caterpillar plot of long run effects",
```

```

    xlab = "Range of parameter samples",
    probs = c(0.025, 0.05, 0.5, 0.95, 0.975),
    labels = colnames(ranges)
  )

```

### Arguments

x	an instance of class opm
parm	a specification of which parameters are to be plotted, a vector of names are the only legal values. If missing, all parameters are considered.
main, xlab	useful defaults for the plot title and X-axis label
probs	a vector specifying the quantiles, the defaults is 2.5%, 5%, 50%, 95%, and 97.5% quantiles
labels	labels for each parameter's interval: see <a href="#">axis</a>

### Value

A matrix of 2.5%, 5%, 50%, 95%, and 97.5% quantiles for each of the desired parameters, with model parameters arranged in columns.

### Examples

```

## Not run:
caterplot_longRun(o, main = NULL)

## End(Not run)

```

---

confint.opm

*Credible Intervals for Model Parameters*

---

### Description

Computes equal-tailed credible intervals for one or more parameters in a fitted opm model. The method used is the quantile interval of the posterior sample.

### Usage

```

## S3 method for class 'opm'
confint(object, parm, level = 0.95, ...)

```

### Arguments

object	an instance of class opm whose credible intervals are wanted
parm	a specification of which parameters are to be given credible intervals, either a vector of names ("rho", "sig2", and "beta" are the only legal values) or a vector of positional indices. If missing, all parameters are considered.
level	the size of the interval (e.g., 0.95 for 95% C.I.)
...	additional argument(s) for methods

**Value**

A matrix with columns giving lower and upper limits of the credible interval for each parameter. These will be labeled as (1 - level/2) and 1 - (1 - level)/2 in % (by default, "2.5%" and "97.5%").

**See Also**

[confint](#)

---

DIC	<i>Deviance Information Criterion (DIC)</i>
-----	---

---

**Description**

Computes the Deviance Information Criterion (DIC), which is a generalization of the Akaike Information Criterion. Models with smaller DIC are considered to fit better than models with larger DIC.

**Usage**

```
DIC(object, ...)
```

**Arguments**

object	an instance of class opm whose DIC is wanted.
...	further arguments passed to other methods.

**Details**

DIC is defined as  $DIC = 2\bar{D} - D_\theta$  where:  $\bar{D} = -2\text{mean}(\log\text{-likelihood at parameter samples})$   
 $D_\theta = -2 * \log(\text{likelihood expected value of parameters})$

DIC is calculated as:  $2 * (-2 * \text{mean}(\log\text{-likelihood at each element of parameter samples}))$   
 $- (-2 * \log(\text{likelihood at mean parameter sample value}))$

**Value**

a numeric value with the corresponding DIC

**Note**

Note the speed of computation of the DIC is proportional to the number of sampled values of the parameters in the opm object.

---

 hist.opm

*Histogram of an opm Object*


---

### Description

Method for `hist` applied to `opm` objects. Each parameter will be plotted in a separate figure.

### Usage

```
## S3 method for class 'opm'
hist(
  x,
  parm,
  ask = dev.interactive(),
  plot = TRUE,
  main = NULL,
  xlab = NULL,
  ...
)
```

### Arguments

<code>x</code>	an instance of class <code>opm</code>
<code>parm</code>	a specification of which parameters are to be plotted, either a vector of names ("rho", "sig2" and "beta" are the only legal values) or a vector of positional indices. If missing, all parameters are considered.
<code>ask</code>	if "TRUE", and the R session is interactive, the user is asked to press a key before a new figure (i.e., histogram of the next model parameter) is drawn.
<code>plot</code>	if "TRUE" (default), the resulting object of class "histogram" is plotted by <code>plot.hist</code> .
<code>main, xlab</code>	(optional) vector of titles and X-axis labels for <i>each</i> figure.
<code>...</code>	further arguments passed to the <code>hist</code> function operating on the individual parameter's samples

### Value

A list of objects of class "histogram", one for each requested model parameter. The elements are named after the parameter.

---

longRunEffects	<i>Long run effects based on the opm Model Parameters</i>
----------------	---

---

**Description**

Computes long run effects and confidence intervals of opm Model Parameters

**Usage**

```
longRunEffects(opm_obj, parm = NULL, probs = c(0.025, 0.5, 0.975))
```

**Arguments**

opm_obj	an instance of class opm
parm	a specification of which parameters are to be plotted, a vector of names are the only legal values. If missing, all parameters are considered.
probs	a vector of specified quantiles, by default, the c(0.025,0.5,0.975) are ("probs")

**Value**

A matrix with quantiles on the rows, with number of rows specified as length of the probs vector for the specified quantiles, with covariates on the columns

**Examples**

```
## Not run:
longRunEffects(opm_obj)
longRunEffects(opm_obj, probs=c(0.975, 0.16, 0.5, 0.84, 0.025))

## End(Not run)
```

---

opm	<i>Fitting orthogonal panel models</i>
-----	--

---

**Description**

opm is used to fit orthogonal panel models.

**Usage**

```
opm(x, ...)
```

## Default S3 method:

```
opm(x, y, n.samp, add.time.indicators = FALSE, ...)
```

## S3 method for class 'formula'

```
opm(x, data = environment(x), subset = NULL, index = 1:2, n.samp, ...)
```

**Arguments**

<code>x</code>	a formula (see description of parameter formula below) or an array of dimension time x variable x case of terms.
<code>...</code>	further arguments passed to other methods.
<code>y</code>	a matrix of dimensions time x case of responses.
<code>n.samp</code>	number of samples to use to estimate the parameters.
<code>add.time.indicators</code>	(logical) if TRUE, adds dummy variables for time.
<code>data</code>	an optional data frame, list, or environment containing the variables in the model. If not found in data, the variables are taken from <code>environment(x)</code> , typically the environment from which <code>opm</code> is called.
<code>subset</code>	an optional vector specifying a subset of observations to be used in the fitting process.
<code>index</code>	a two-element vector containing the index of the case and time variables, respectively. Variable indices can be specified by name or position. This argument is ignored if the model is not specified by the formula, because the index is implicit in the organization of the terms and response arrays.

**Details**

The model can be either specified symbolically with the formula `response ~ term1 + term2 ...` or with the terms and response given as a pair of 3- and 2-dimensional arrays, `x` and `y` respectively. The arrays have to be in the format time x variable x case for terms and time x case for the response.

The lagged dependent variable does not need to be included in the formula or data, as it is included automatically.

**Value**

An object of class `opm` with the following elements:

`samples` parameter samples used to estimate the model, as a list with following elements:

`rho` a vector of `n.samp` samples of  $\rho$ .

`v` a vector of `n.samp` samples of  $\frac{1}{\sigma^2}$ .

`beta` an `n.samp` x `variable` matrix of samples of  $\beta$ .

`call` the matched call

`index` the index variables, when using the formula interface

`time.indicators` TRUE if dummy time variables are used (see Notes), FALSE otherwise

`terms` the terms object used

The function `summary` (i.e., `summary.opm`) can be used to obtain or print a summary of the results. The generic accessor functions `coefficients`, `fitted.values`, `residuals`, `logLik`, and `df.residual` can be used to extract various useful features of the value returned by `opm`.

**Note**

Dummy time variables exist as an additional column for each wave of data, excluding the first and second wave (i.e., at  $t = 0$  and  $t = 1$  using the terminology from Lancaster (2000)). The new variables are named `tind.t`, where  $t = 2, \dots$ , and appear as such as elements of the estimated beta coefficient.

**Examples**

```
set.seed(123)
N <- 5
T <- 2
beta <- .5
rho <- .5
v <- 1

f <- runif(N, -2, 2)
K <- length(beta)
beta <- matrix(beta, K, 1)

## $x_i = 0.75 f + N(0, 1)$:
x <- array(.75*f, dim=c(N, K, (T+1))) + rnorm(N*K*(T+1))

## $y_{i,t} = \rho y_{i,t-1} + \beta x_{i,t} + f_i + N(0,1)$:
y <- matrix(0, N, T+1)
for (t in seq_len(T+1)) {
  yy <- if (t>1) y[,t-1] else 0
  y[,t] <- rho * yy + f + x[, ,t] %% beta + rnorm(N, sd = sqrt(1/v))
}

d <- data.frame(i = rep(seq(N), T+1),
               t = rep(seq(T+1), each = N),
               as.data.frame(matrix(aperm(x, c(1, 3, 2)), N*(T+1), K,
                                   dimnames = list(NULL, paste0('x', seq(K)))))),
               y = c(y))
opm(y~x1, d, n.samp = 10)
```

---

plot.opm

*Plot Method for an opm Object*


---

**Description**

Method for `plot` applied to `opm` objects. Each parameter will be plotted as a density plot in a separate figure.

**Usage**

```
## S3 method for class 'opm'
plot(x, parm, ask = dev.interactive(), main = NULL, xlab = NULL, ...)
```

**Arguments**

x	an instance of class opm
parm	a specification of which parameters are to be plotted, either a vector of names ("rho", "sig2" and "beta" are the only legal values) or a vector of positional indices. If missing, all parameters are considered.
ask	if "TRUE", and the R session is interactive, the user is asked to press a key before a new figure (i.e., histogram of the next model parameter) is drawn.
main, xlab	(optional) vector of titles and X-axis labels for <i>each</i> figure.
...	further arguments passed to the <code>plot</code> function operating on the individual parameter's samples

**Value**

A list of objects of class "density", one for each requested model parameter. The elements are named after the parameter.

---

p_rho	<i>Returns the posterior density <math>p(\rho \Theta)</math></i>
-------	--

---

**Description**

Returns the posterior density  $p(\rho|\Theta)$

**Usage**

```
p_rho(x, y, rho, log.p = FALSE)
```

**Arguments**

x	an array of dimension time x variable x case of terms.
y	a matrix of dimension time x case of responses.
rho	vector of quantiles.
log.p	if TRUE, probabilities are given as log(p).

---

`quantile.opm`*Posterior Sample Quantiles*

---

**Description**

Produces quantiles of the posterior samples corresponding to the given probabilities. In other words, it is equivalent to computing `quantile(x, ...)`, where `x` is the original Monte Carlo sample of the parameter `"parm"`, as produced by `opm`.

**Usage**

```
## S3 method for class 'opm'  
quantile(x, parm, ...)
```

**Arguments**

<code>x</code>	an instance of class <code>opm</code> whose sample quantiles are wanted
<code>parm</code>	a specification of which parameters are to be given quantiles, either a vector of names ( <code>"rho"</code> , <code>"sig2"</code> , and <code>"beta"</code> are the only legal values) or a vector of positional indices. If missing, all parameters are considered.
<code>...</code>	further arguments passed to the <code>quantile</code> function operating on the individual parameter's samples

**Value**

A matrix of quantiles for each of the desired parameters, with parameters arranged in columns. If arguments include `"names = FALSE"`, the quantile labels won't be included (i.e., the rownames of the matrix will be `NULL`).

**See Also**

[quantile](#)

# Index

abond\_panel, 3  
axis, 5, 6

BES\_panel, 4

caterplot, 4  
caterplot\_longRun, 5  
confint, 7  
confint.opm, 6

DIC, 7

hist, 8  
hist.opm, 8

longRunEffects, 9

opm, 8, 9, 11, 13  
OrthoPanels-package, 2

p\_rho, 12  
plot, 11, 12  
plot.opm, 11

quantile, 13  
quantile.opm, 13