Package: xtpqardl (via r-universe)

October 31, 2026

Type Package

Title Panel Quantile Autoregressive Distributed Lag Model

Version 1.0.5

Date 2026-10-30

Description Estimation of Panel Quantile Autoregressive Distributed Lag (PQARDL) models that combine panel ARDL methodology with quantile regression. Supports Pooled Mean Group (PMG), Mean Group (MG), and Dynamic Fixed Effects (DFE) estimators across multiple quantiles. Computes long-run cointegrating parameters, error correction term speed of adjustment, half-life of adjustment, and performs Wald tests for parameter equality across quantiles. Based on the econometric frameworks of Pesaran, Shin, and Smith (1999) <doi:10.1080/01621459.1999.10474156>, Cho, Kim, and Shin (2015) <doi:10.1016/j.jeconom.2015.05.003>, and Bildirici and Kayikci (2022).

License GPL-3

URL https://github.com/muhammedalkhalaf/xtpqardl

BugReports https://github.com/muhammedalkhalaf/xtpqardl/issues

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports stats, utils, quantreg

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

RoxygenNote 7.3.1

Repository https://muhammedalkhalaf.r-universe.dev

Date/Publication 2026-10-31 05:48:21 UTC

RemoteUrl https://github.com/muhammedalkhalaf/xtpqardl

RemoteRef HEAD

RemoteSha 6d30336185b6670cc4aad765ffb21571db619d70

Contents

coef.xtpqardl 2
compute_irf 3
pqardl_sample 4
print.irf.xtpqardl 5
print.summary.xtpqardl 6
print.wald_test.xtpqardl 6
print.xtpqardl 7
summary.xtpqardl 7
vcov.xtpqardl 8
wald_test 9
xtpqardl 11

Index **17**

`coef.xtpqardl` *Coefficients Method for xtpqardl Objects*

Description

Extract estimated coefficients from a Panel Quantile ARDL model.

Usage

```
## S3 method for class 'xtpqardl'
coef(object, type = c("beta", "rho", "all"), ...)
```

Arguments

`object`	An object of class `"xtpqardl"`.
`type`	Character string specifying which coefficients to extract: `"beta"` for long-run coefficients (default), `"rho"` for ECT speed of adjustment, or `"all"` for both.
`...`	Additional arguments (currently unused).

Value

A named numeric vector or list of coefficients.

Examples

```
data(pqardl_sample)
fit <- xtpqardl(
  formula = d_y ~ d_x1 + d_x2,
  data = pqardl_sample,
  id = "country",
  time = "year",
  lr = c("L_y", "x1", "x2"),
  tau = c(0.25, 0.50, 0.75)
)
coef(fit)
coef(fit, type = "rho")
```

`compute_irf` *Compute Impulse Response Function*

Description

Computes the impulse response function (IRF) for Panel Quantile ARDL models, showing the response to a one-unit shock via the error correction mechanism.

Usage

```
compute_irf(object, horizon = 20)
```

Arguments

`object`	An object of class `"xtpqardl"`.
`horizon`	Integer specifying the number of periods for the IRF. Default is 20.

Details

The IRF for the error correction model is computed as:

$$IRF_t(\tau) = (1 + \rho(\tau))^t$$

which shows the decay of a unit shock over time through the error correction mechanism. Values approach zero as t grows when $-2 < \rho(\tau) < 0$; for $-2 < \rho(\tau) \leq -1$ the response oscillates in sign (the half-life of `xtpqardl` is defined for $-1 < \rho < 0$ only). Outside $(-2, 0)$ the entries are `NA` (printed as "div.").

Value

A matrix with rows representing time periods and columns representing quantiles. Each entry shows the response at that period for that quantile.

Examples

```
data(pqardl_sample)
fit <- xtpqardl(
  formula = d_y ~ d_x1 + d_x2,
  data = pqardl_sample,
  id = "country",
  time = "year",
  lr = c("L_y", "x1", "x2"),
  tau = c(0.25, 0.50, 0.75)
)
irf <- compute_irf(fit, horizon = 15)
print(irf)
```

---

pqardl_sample *Simulated Panel Data for PQARDL Estimation*

Description

A simulated panel dataset for demonstrating Panel Quantile ARDL estimation. Contains 10 countries observed over 30 years with variables suitable for error correction modeling.

Usage

```
pqardl_sample
```

Format

A data frame with 300 rows and 9 variables:

country Factor indicating the panel unit (10 countries)

year Integer year variable (1990-2019)

y Dependent variable in levels (e.g., GDP per capita)

x1 First explanatory variable in levels (e.g., investment)

x2 Second explanatory variable in levels (e.g., trade openness)

L_y Lagged dependent variable (y at t-1)

d_y First difference of y

d_x1 First difference of x1

d_x2 First difference of x2

Details

The data are simulated from a panel error correction model with heterogeneous adjustment speeds across countries. The true long-run relationship is:

$$y_{it} = \beta_1 x_{1,it} + \beta_2 x_{2,it} + \mu_i + \varepsilon_{it}$$

with error correction dynamics:

$$\Delta y_{it} = \rho_i (y_{i,t-1} - \beta_1 x_{1,i,t-1} - \beta_2 x_{2,i,t-1}) + \gamma_1 \Delta x_{1,it} + \gamma_2 \Delta x_{2,it} + u_{it}$$

where $\rho_i \sim U(-0.6, -0.2)$ varies by panel.

Source

Simulated data for package demonstration.

Examples

```
data(pqardl_sample)
head(pqardl_sample)

# Check panel structure
table(pqardl_sample$country)
```

---

print.irf.xtpqardl *Print Method for irf.xtpqardl Objects*

Description

Print Method for irf.xtpqardl Objects

Usage

```
## S3 method for class 'irf.xtpqardl'
print(x, ...)
```

Arguments

x	An object of class "irf.xtpqardl".
...	Additional arguments (currently unused).

Value

Invisibly returns the input object.

`print.summary.xtpqardl` *Print Method for summary.xtpqardl Objects*

Description

Print Method for summary.xtpqardl Objects

Usage

```
## S3 method for class 'summary.xtpqardl'
print(x, digits = 4, ...)
```

Arguments

`x`	An object of class `"summary.xtpqardl"`.
`digits`	Number of significant digits to display. Default is 4.
`...`	Additional arguments (currently unused).

Value

Invisibly returns the input object.

`print.wald_test.xtpqardl` *Print Method for wald_test.xtpqardl Objects*

Description

Print Method for wald_test.xtpqardl Objects

Usage

```
## S3 method for class 'wald_test.xtpqardl'
print(x, ...)
```

Arguments

`x`	An object of class `"wald_test.xtpqardl"`.
`...`	Additional arguments (currently unused).

Value

Invisibly returns the input object.

print.xtpqardl *Print Method for xtpqardl Objects*

Description

Print Method for xtpqardl Objects

Usage

```
## S3 method for class 'xtpqardl'
print(x, ...)
```

Arguments

x	An object of class "xtpqardl".
...	Additional arguments (currently unused).

Value

Invisibly returns the input object.

summary.xtpqardl *Summary Method for xtpqardl Objects*

Description

Produces a detailed summary of Panel Quantile ARDL estimation results, including long-run coefficients, ECT speed of adjustment, half-life of adjustment, and short-run parameters by quantile.

Usage

```
## S3 method for class 'xtpqardl'
summary(object, ...)
```

Arguments

object	An object of class "xtpqardl".
...	Additional arguments (currently unused).

Value

An object of class "summary.xtpqardl" containing formatted tables of results.

Examples

```
data(pqardl_sample)
fit <- xtpqardl(
  formula = d_y ~ d_x1 + d_x2,
  data = pqardl_sample,
  id = "country",
  time = "year",
  lr = c("L_y", "x1", "x2"),
  tau = c(0.25, 0.50, 0.75)
)
summary(fit)
```

`vcov.xtpqardl` *Variance-Covariance Matrix for xtpqardl Objects*

Description

Extract the variance-covariance matrix of the estimated parameters.

Usage

```
## S3 method for class 'xtpqardl'
vcov(object, type = c("beta", "rho"), ...)
```

Arguments

`object`	An object of class `"xtpqardl"`.
`type`	Character string specifying which covariance matrix to extract: `"beta"` for long-run coefficients (default), or `"rho"` for ECT coefficients.
`...`	Additional arguments (currently unused).

Value

A variance-covariance matrix.

wald_test *Wald Test for Parameter Equality Across Quantiles*

Description

Performs Wald tests for the null hypothesis that parameters are equal across different quantiles. Tests both long-run coefficients (beta) and ECT speed of adjustment (rho).

Usage

```
wald_test(object, joint = TRUE, all = FALSE)

## S3 method for class 'xtpqardl'
wald_test(object, joint = TRUE, all = FALSE)
```

Arguments

object	An object of class "xtpqardl".
joint	Logical. If TRUE (default), performs joint test for all coefficients. If FALSE, performs individual tests for each variable.
all	Logical, for model = "dfe" only. If FALSE (default) the Koenker-Bassett tests in kb_test restrict only the economic slopes of the pooled regression (speed of adjustment, long-run levels, short run). If TRUE they restrict every non-intercept coefficient, including the $N-1$ panel-intercept contrasts, which is what quantreg::anova(..., joint = TRUE) does on the equivalent regression y ~ slopes + factor(id).

Details

The Wald test statistic is computed as:

$$W = (R\theta)'[RVR']^{-1}(R\theta)$$

where θ is the vector of coefficients, V is the variance-covariance matrix, and R is a restriction matrix testing equality of coefficients across quantiles (each quantile against the first). Under the null hypothesis of equal coefficients, W follows a chi-squared distribution with degrees of freedom equal to the number of restrictions. For MG the covariance is the Pesaran-Smith between-panel covariance of the stacked panel estimates; for DFE it is the joint covariance described in xtpqardl (one estimator for every block, named in dfe_cov); for PMG the cross-quantile blocks are not estimated and beta_test is not available.

Koenker-Bassett tests (DFE). kb_test applies the same restriction scheme to the pooled quantile-regression coefficients themselves (rho, theta, short run) with the joint covariance $W = [(\min(\tau_a,\tau_b) - \tau_a\tau_b)H_a^{-1}X'XH_b^{-1}]_{ab}$ of xtpqardl (Koenker and Bassett 1982, Theorem 3.1, with the estimator chosen by dfe_cov). As in quantreg::anova the statistic is reported as T_n/ndf against an F distribution with ndf restrictions and $ddf = nm - ndf$ denominator degrees of freedom (n observations, m quantiles); with dfe_cov = "nid" and all = TRUE the joint and individual rows reproduce

`anova(rq_1, ..., rq_m, se = "nid")` on the equivalent regression with `factor(id)`. With the default `all = FALSE` the joint test has $ndf = k(m-1)$ where k is the number of economic slopes, so the hypothesis is the one stated in the printout, and the $N-1$ contrasts (whose equality across quantiles is of no economic interest and dilutes the test) are not restricted.

Rank of the theta covariance. Cho, Kim and Shin (2015, Theorem 1(ii)) show that the asymptotic covariance of the long-run level coefficients $\gamma(\tau)$ (theta) has rank one under their normalisation. In their words, "has a rank equal to unity by Theorem 1(ii). This implies that, if R has a rank greater than unity, the corresponding Wald test has a degenerate asymptotic null distribution." (the page numbers of that passage are not given here; the library copy has no page markers). The individual rows of `kb_test` for each theta coefficient restrict one coefficient across quantiles; the joint test with more than one long-run regressor restricts several theta coefficients at once and relies on the finite-sample full rank of their sandwich covariance.

The audit's Monte Carlo is consistent with this concern, but does not isolate it. In the `kb_joint_all` column of its size grid (nid estimator, all non-intercept coefficients restricted, iid N(0,1) errors, N = 5, T = 20, R = 500) the rejection rate at nominal 5 percent was 16.2 percent with one long-run regressor and 23.2 percent with two. The two designs also differ in the number of restrictions and the nid density bias documented in the audit is a confounder, so the increase should be read as consistent with, not as evidence for, the rank argument. The long-run beta = -theta/rho is tested in `beta_test`, with the delta method.

Value

An object of class `"wald_test.xtpqardl"` containing:

beta_test Wald test result for long-run coefficients

rho_test Wald test result for ECT coefficients

individual_beta Individual tests for each long-run variable (if joint = FALSE)

kb_test For `model = "dfe"`: Koenker and Bassett (1982) equality tests across quantiles on the pooled quantile-regression coefficients, computed from the joint covariance `dfe_coef_V` (estimator `dfe_cov` of the fit): `joint` (one row: F statistic, numerator and denominator degrees of freedom, p-value) and `individual` (one row per restricted coefficient), the names of the restricted coefficients in `restricted`, the estimator in `se` and whether the panel contrasts were included in `all`. `NULL` for MG and PMG.

tau Quantiles tested

References

Cho JS, Kim TH, Shin Y (2015). "Quantile Cointegration in the Autoregressive Distributed-Lag Modeling Framework." *Journal of Econometrics*, 188(1), 281-300. doi:10.1016/j.jeconom.2015.05.003

Koenker R, Bassett G (1982). "Robust Tests for Heteroscedasticity Based on Regression Quantiles." *Econometrica*, 50(1), 43-61. doi:10.2307/1912528

Examples

```
data(pqardl_sample)
fit <- xtpqardl(
  formula = d_y ~ d_x1 + d_x2,
  data = pqardl_sample,
```

```
  id = "country",
  time = "year",
  lr = c("L_y", "x1", "x2"),
  tau = c(0.25, 0.50, 0.75)
)
wald_test(fit)
```

---

xtpqardl *Panel Quantile Autoregressive Distributed Lag Model*

Description

Estimate Panel Quantile ARDL (PQARDL) models that combine panel ARDL methodology with quantile regression. Supports Pooled Mean Group (PMG), Mean Group (MG), and Dynamic Fixed Effects (DFE) estimators.

Usage

```
xtpqardl(
  formula,
  data,
  id,
  time,
  lr,
  tau = c(0.25, 0.5, 0.75),
  p = 1,
  q = 1,
  model = c("pmg", "mg", "dfe"),
  lagsel = NULL,
  pmax = 4,
  qmax = 4,
  constant = TRUE,
  pmg_weights = c("inverse_variance", "equal"),
  dfe_cov = c("iid", "nid", "ker"),
  allow_gaps = FALSE,
  hausman = c("strict", "ginv")
)
```

Arguments

`formula`	A formula specifying the model. The response variable should be in first differences (e.g., `d.y ~ d.x1 + d.x2`), where short-run dynamics are estimated. Transformations such as `log(x)`, `I(x^2)` and interactions `a:b` are expanded with `model.matrix` before the lags are built, so the lag orders in `q` refer to the columns of the expanded design (in the order of `colnames(model.matrix(formula, data))` without the intercept).

`data` A data frame containing panel data with variables specified in the formula and `lr` argument.

`id` Character string specifying the panel (cross-section) identifier variable name.

`time` Character string specifying the time variable name. It must be numeric (for example the year as an integer). Lags are built by matching `time - j` within each panel, never by row position; see Details.

`lr` Character vector of long-run level variable names. The first element should be the lagged dependent variable level (for the error correction term), and remaining elements are the long-run explanatory variables. The variables enter the regression exactly as supplied (no further lag is applied); supply lagged regressors, for example `L_x1`, if the error correction term should use x_{t-1}. The names must be columns of `data` (no transformations).

`tau` Numeric vector of quantiles to estimate, each in (0,1). Default is `c(0.25, 0.50, 0.75)`.

`p` Integer specifying the autoregressive lag order for the dependent variable. Default is 1.

`q` Integer or integer vector specifying the distributed lag order(s) for explanatory variables. If a single integer, the same lag order is applied to all variables. Default is 1.

`model` Character string specifying the estimation method: `"pmg"` for Pooled Mean Group (default), `"mg"` for Mean Group, or `"dfe"` for Dynamic Fixed Effects.

`lagsel` Character string for automatic lag selection. If `"bic"` or `"aic"`, the lag orders are selected by least squares on the panel regressions, see Details. Default is `NULL` (no automatic selection).

`pmax` Maximum p to consider in lag selection. Default is 4.

`qmax` Maximum q to consider in lag selection. Default is 4.

`constant` Logical. Include a constant term? Default is `TRUE`.

`pmg_weights` Character, for `model = "pmg"` only. `"inverse_variance"` (default) weights each panel's long-run estimate by the inverse of its delta-method covariance, as described in Details; it requires that covariance to be estimable for at least two panels per quantile. `"equal"` pools with equal weights (the pooled long run is then the simple average of the panel estimates and its covariance the mean-group covariance); use it when the panels are too short for the covariances to be estimated.

`dfe_cov` Character, for `model = "dfe"` only: the estimator of the joint (all quantiles) covariance matrix of the pooled quantile regression coefficients. `"iid"` (default) is the Koenker and Bassett (1982, Theorem 3.1) form with the scalar sparsity estimate of `summary.rq(se = "iid")`; `"nid"` uses the Hendricks-Koenker local densities of `summary.rq(se = "nid")`; `"ker"` the Powell kernel densities of `summary.rq(se = "ker")`. One estimator is used for every block of the matrix; see Details for the choice.

`allow_gaps` Logical. If `FALSE` (default) the function stops when a panel has a gap in `time`. If `TRUE` the panels with gaps are kept: lags are matched by the time index, and a row whose lagged period is missing is dropped from that panel's regression.

hausman — Character, for model = "pmg" only: "strict" (default) returns NA with a warning when the difference $V(MG) - V(PMG)$ is not positive definite; "ginv" inverts it with a Moore-Penrose inverse over the eigenvalues larger than 1e-8 times the largest, with degrees of freedom equal to that rank, and warns.

Details

The PQARDL model extends the standard panel ARDL framework to allow for heterogeneous effects across the conditional distribution of the response variable. The error correction representation is:

$$\Delta y_{it} = \rho_i(\tau) \cdot ECT_{i,t-1} + \sum_{j=1}^{p-1} \phi_{ij} \Delta y_{i,t-j} + \sum_{m=0}^{q-1} \theta_{im} \Delta x_{i,t-m} + \varepsilon_{it}(\tau)$$

where $ECT = y_{i,t-1} - \beta(\tau)'X$ is the error correction term built from the lr variables as supplied, $\rho(\tau)$ is the speed of adjustment (negative for convergence), and $\beta(\tau)$ are the long-run parameters, $\beta = -\theta/\rho$ (the plug-in estimator of Cho, Kim and Shin 2015, equation (3)).

Lags, gaps and missing values. Within each panel the lag j of a variable at time t is the value at time $t - j$, where the unit step is the smallest positive time difference in the data. Duplicated (id, time) pairs are an error. A gap in a panel's time index is an error unless allow_gaps = TRUE, in which case the rows whose lagged period is missing are dropped. Rows with a missing value in the response, a regressor or a lag are dropped panel by panel (the leading rows lost to the user's own differencing and lagging are the usual case); the number of dropped rows is returned in n_dropped. Exactly collinear regressors stop with an error naming the columns.

model = "mg" averages the panel estimates (Pesaran and Smith covariance $\sum_i (b_i - \bar{b})(b_i - \bar{b})'/(N(N-1))$). With fewer than three panels this covariance is unreliable (with N = 2 the audit's Monte Carlo rejected 26 to 31 percent at nominal 5 percent) and the function warns. model = "pmg" pools the long run by minimum distance, weighting each panel by the inverse of its delta-method covariance, then re-estimates every panel with the pooled long run imposed; the speed of adjustment and the short run are the mean group of that second stage. A Hausman test of long-run homogeneity (MG against PMG) is computed quantile by quantile; the statistic is $d'(V_{MG} - V_{PMG})^{-1}d$ with d the difference of the long-run estimates at that quantile. The quadratic form is defined only when the difference matrix is positive definite (Hausman 1978 gives non-negative definiteness asymptotically); see hausman.

DFE covariance. model = "dfe" pools all panels in one quantile regression per quantile with an intercept and $N - 1$ panel-intercept contrasts. All quantiles share the design X, so the joint covariance of the coefficient vectors $b(\tau_a)$, $b(\tau_b)$ is estimated by one formula for every block, with the estimator named in dfe_cov:

$$Cov(b(\tau_a), b(\tau_b)) = (\min(\tau_a, \tau_b) - \tau_a \tau_b) H_a^{-1} (X'X) H_b^{-1}.$$

With "iid" (default) $H_a^{-1} = s(\tau_a)(X'X)^{-1}$, where $s(\tau) = 1/f(F^{-1}(\tau))$ is the sparsity estimated as in summary.rq(se = "iid"); this is Theorem 3.1 of Koenker and Bassett (1982), whose covariance has typical element $[\min(u_i, u_j) - u_i u_j]/(f(Q(u_i))f(Q(u_j)))$ times $D^{-1} = \lim(X'X/n)^{-1}$, under i.i.d. errors. With "nid" and "ker", $H_\tau = X' diag(f_i(\tau)) X$ with the local density estimates of summary.rq(se = "nid") (Hendricks-Koenker) or summary.rq(se = "ker") (Powell kernel), including quantreg's bandwidth rule (the bandwidth is halved while $\tau \pm h$ leaves (0, 1)). The diagonal blocks equal summary.rq(se = dfe_cov)$cov. The default is "iid" because in the audit of

version 1.0.4 (panel designs with N = 5 to 20, T = 20 to 40, i.i.d. errors) the Hendricks-Koenker densities under-stated the variance of the speed of adjustment by 23 to 55 percent in the presence of the panel intercepts (Monte Carlo variance 0.00161 against a mean "nid" estimate 0.00072 at N = 20, T = 40), so that the "nid" equality tests rejected 9 to 23 percent at nominal 5 percent for N at most 10 and 47 to 76 percent at N = 20, whereas the "iid" form rejected 4.3, 5.0 and 7.0 percent in the three designs checked. For the default in the N = 10, T = 40, k = 1 design (iid errors, nominal 5 percent) `rho_test` rejected between 6 and 9 percent in two independent Monte Carlo batches (R = 200 and R = 500); a third batch (R = 300) gave 5.7 percent. The test is therefore mildly over-sized in that cell, against 1.2 percent with the 1.0.4 mixed matrix. `"nid"` and `"ker"` remain available for users who need them; they are not recommended for the cross-quantile tests of panels with many intercepts. The same joint matrix is the basis of the Koenker-Bassett tests of `wald_test`; if the estimator fails for a quantile (for example an extreme quantile in a small sample), the affected blocks are `NA` and a warning names the quantile and the reason. The standard errors of ρ and β come from the diagonal blocks and the delta method.

Rank of the covariance of theta. Cho, Kim and Shin (2015, Theorem 1(ii)) show that under their normalisation the asymptotic covariance of the long-run level coefficients $\gamma(\tau)$ (`theta` here, the coefficients of `lr[-1]`) has rank one, and they restrict their Wald tests on γ to restriction matrices of rank one, because a higher rank gives a degenerate null distribution. The finite-sample sandwich blocks for θ used here are of full rank, so a joint test that restricts several θ coefficients at once (as the Koenker-Bassett joint test of `wald_test` does when $k_x > 1$) relies on finite-sample full rank of a matrix whose limit is singular under their assumptions. See `wald_test` for the size numbers. Tests on a single θ_j across quantiles (the individual rows) are of rank $m-1$ in the quantile direction.

Lag selection. `lagsel` fits every (p, q) combination with $p \le$ `pmax`, $q \le$ `qmax` (the same q for all regressors) by least squares (`lm.fit`) panel by panel, sums the residual sums of squares and the sample sizes over panels, and penalises the total number of estimated coefficients (the sum over the panels of the columns of their regressions): $BIC = n\log(RSS/n) + K\log n$, $AIC = n\log(RSS/n) + 2K$. The criterion is least squares, not the quantile check loss, and applies to the mean regression.

Short panels. Each panel regression has $1+k_x+(p-1)+\sum q+1$ coefficients. When T is close to that number the covariance of the panel estimates cannot be estimated by `summary.rq(se = "iid")` (its sparsity estimate needs the quantiles at $\tau \pm h$); the function then tries the other `summary.rq` estimators and finally an iid formula with a kernel density estimate of the sparsity, and warns once naming how many panel-quantile pairs switched and to which estimator (details in `panel_cov_status`). If no covariance is available for at least two panels, PMG stops with a message listing the status of every panel; `pmg_weights = "equal"` or a smaller model are the alternatives. Panels dropped from the PMG pooling because their covariance is not invertible are named in a warning.

Value

An object of class `"xtpqardl"` containing:

beta_mg Matrix of mean group long-run coefficients across quantiles

rho_mg Vector of mean group ECT speed of adjustment by quantile

halflife_mg Half-life $\ln(0.5)/\ln(1+\rho)$ of the mean group speed of adjustment, by quantile

sr_mg Matrix of mean group short-run coefficients

phi_mg Matrix of mean group AR coefficients (if p > 1)

hausman For `model = "pmg"`, a data frame with one row per quantile: the Hausman test of long-run homogeneity (MG against PMG) at that quantile, with `statistic`, `df`, `p.value`, the number of non-positive eigenvalues of the difference matrix and the method used

beta_V Variance-covariance matrix for beta_mg. For MG and DFE the full matrix is estimated (for DFE every block comes from the estimator named in `dfe_cov`, see Details); for PMG the cross-quantile blocks are `NA`

rho_V Variance-covariance matrix for rho_mg

dfe_cov For `model = "dfe"`, the covariance estimator used (`"iid"`, `"nid"` or `"ker"`)

dfe_coef For `model = "dfe"`, the matrix of pooled quantile-regression coefficients (rows: coefficients, columns: tau)

dfe_coef_V For `model = "dfe"`, the joint covariance matrix of `dfe_coef` stacked by quantile, used by `wald_test` for the Koenker-Bassett tests

dfe_slopes For `model = "dfe"`, the names of the economic slopes (speed of adjustment, long-run levels, short run), as opposed to the intercept and the panel-intercept contrasts

rq_fits For `model = "dfe"`, the list of `rq` fits (one per quantile) of the pooled regression

panel_cov_status A data frame with one row per panel and quantile (MG, PMG) or per quantile (DFE) reporting whether the covariance could be estimated, which estimator was used and why not

beta_all Matrix of per-panel long-run coefficients

rho_all Matrix of per-panel ECT coefficients

halflife_all Matrix of per-panel half-life values

tau Vector of estimated quantiles

p AR lag order used

q Distributed lag order(s) used

model Estimation method used

n_obs Total number of observations

n_dropped Number of rows dropped because a regressor, the response or a lag was missing

n_panels Number of panels

valid_panels Number of successfully estimated panels

depvar Dependent variable name

lrvars Long-run variable names

call The matched call

References

Pesaran MH, Shin Y, Smith RP (1999). "Pooled Mean Group Estimation of Dynamic Heterogeneous Panels." *Journal of the American Statistical Association*, 94(446), 621-634. doi:10.1080/01621459.1999.10474156

Cho JS, Kim TH, Shin Y (2015). "Quantile Cointegration in the Autoregressive Distributed-Lag Modeling Framework." *Journal of Econometrics*, 188(1), 281-300. doi:10.1016/j.jeconom.2015.05.003

Bildirici M, Kayikci F (2022). "Uncertainty, Renewable Energy, and CO2 Emissions in Top Renewable Energy Countries: A Panel Quantile Regression Approach." *Energy*, 247, 124303.

Hausman JA (1978). "Specification Tests in Econometrics." *Econometrica*, 46(6), 1251-1271. doi:10.2307/1913827

Koenker R, Bassett G (1978). "Regression Quantiles." *Econometrica*, 46(1), 33-50. doi:10.2307/1913643

Koenker R, Bassett G (1982). "Robust Tests for Heteroscedasticity Based on Regression Quantiles." *Econometrica*, 50(1), 43-61. doi:10.2307/1912528

Examples

```
# Load example panel data
data(pqardl_sample)

# Estimate PQARDL model at 25th, 50th, and 75th quantiles
fit <- xtpqardl(
  formula = d_y ~ d_x1 + d_x2,
  data = pqardl_sample,
  id = "country",
  time = "year",
  lr = c("L_y", "x1", "x2"),
  tau = c(0.25, 0.50, 0.75),
  model = "pmg"
)

# View results
summary(fit)

# Wald test for parameter equality across quantiles
wald_test(fit)
```

Index

∗ **datasets**
pqardl_sample, 4

coef.xtpqardl, 2
compute_irf, 3

pqardl_sample, 4
print.irf.xtpqardl, 5
print.summary.xtpqardl, 6
print.wald_test.xtpqardl, 6
print.xtpqardl, 7

summary.xtpqardl, 7

vcov.xtpqardl, 8

wald_test, 9, *14, 15*

xtpqardl, *3*, *9*, 11