Package 'tsqr'

July 6, 2026

Title Sequential Threshold-Spatial-Quantile Panel Estimation

Version 0.2.0

Description Implements a sequential panel estimation protocol for regional economic panels that combines three estimation layers in a fixed order. The first layer applies a two-way fixed effects baseline. The second layer applies the panel threshold regression method of Hansen (1999) <doi:10.1016/S0304-4076(99)00025-1> to identify structural breaks at an unknown threshold of a moderating variable, with bootstrap inference following Hansen (2000) <doi:10.1111/1468-0262.00124>. The third layer applies a spatial Durbin model with an impact decomposition following LeSage and Pace (2009, ISBN:978-1-4200-6424-7) to quantify direct and indirect spillover effects. The fourth layer applies the two-step panel quantile estimator of Canay (2011) <doi:10.1111/j.1368-423X.2011.00349.x> to document distributional heterogeneity in the outcome. The threshold identified in the second layer defines a subsample used as structured input to the fourth layer, and a consistency check evaluates whether the three sets of results are jointly compatible with a common underlying structural relationship. An illustrative panel of 33 districts of the state of Maharashtra, India, observed over 10 agricultural years, is included with the package.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports plm (>= 2.6-0), spdep (>= 1.2-0), spatialreg (>= 1.2-0), quantreg (>= 5.90), ggplot2 (>= 3.4.0), stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Prakash Vhankade [aut, cre] (ORCID: <https://orcid.org/0009-0005-2436-0603>)

Maintainer Prakash Vhankade <prakash.vhankade@gipe.ac.in>

Repository CRAN

Date/Publication 2026-07-06 13:50:02 UTC

Contents

`canay_quantile` *Canay Two-Step Panel Quantile Regression*

Description

Implements the Canay (2011) two-step panel quantile estimator. In the first step, unit and time fixed effects are estimated from a two-way fixed effects model and subtracted from the outcome. In the second step, standard quantile regression is applied at each requested quantile.

Usage

```
canay_quantile(
  formula,
  data,
  index,
  taus = c(0.1, 0.25, 0.5, 0.75, 0.9),
  n_boot = 500,
  seed = 42,
  ptr_result = NULL,
  threshold_var = NULL,
  verbose = TRUE
)
```

Arguments

`formula`	A formula of the form `y ~ x + controls`.
`data`	A data frame.
`index`	A character vector of length 2: `c("unit_id", "time_id")`.
`taus`	Numeric vector of quantile levels. Default is `c(0.10, 0.25, 0.50, 0.75, 0.90)`.
`n_boot`	Integer. Bootstrap replications for standard errors. Default is `500`.

seed	Integer or NULL. Random seed for reproducibility. Set to NULL to use the current random state without fixing the seed. Default is 42.
ptr_result	Optional object of class ptr_hansen. If supplied, the function also estimates the quantile regression on the high-regime subsample defined by threshold_var exceeding the estimated threshold.
threshold_var	Character string naming the threshold variable. Required if ptr_result is supplied.
verbose	Logical. Default is TRUE.

Value

A list of class canay_quantile with elements coefs, se, slope_tests, gradient, gradient_highstress, fe_corrected, taus, key_var, and call.

References

Canay IA (2011). "A Simple Approach to Quantile Regression for Panel Data." *Econometrics Journal*, 14(3), 368-386. doi:10.1111/j.1368423X.2011.00349.x

Examples

```
data(maharashtra_panel)
result <- canay_quantile(
  formula = ln_GDVA ~ ln_HYV + MCDS_days + Irrigation_pct,
  data    = maharashtra_panel,
  index   = c("district", "year"),
  taus    = c(0.25, 0.50, 0.75),
  n_boot  = 50,
  seed    = 42,
  verbose = FALSE
)
print(result)
coef(result)
```

---

maharashtra_panel *Maharashtra Agricultural District Panel Dataset*

Description

A balanced panel dataset covering 33 districts of the state of Maharashtra, India, over 10 agricultural years from 2014-15 to 2023-24 (330 observations, 30 variables).

Usage

```
data(maharashtra_panel)
```

Format

A data frame with 330 rows and 30 variables:

district District name.

dist_id District numeric identifier, 1 to 33.

year Calendar year, 2014 to 2023.

agri_year Agricultural year label, e.g. "2014-15".

GDVA_agri_crore Gross District Value Added from Agriculture and Allied Sectors, constant 2011-12 prices, INR crore.

ln_GDVA Natural log of GDVA_agri_crore. Dependent variable.

FarmIncome_Rs000 Farm household income, INR thousands.

HYV_area_000ha Area under high-yielding varieties, 000 ha.

ln_HYV Natural log of HYV_area_000ha. Key regressor.

Rainfall_mm Annual rainfall, mm.

ln_Rainfall Natural log of Rainfall_mm.

Temp_C Mean annual temperature, degrees Celsius.

TempAnomaly Standardised within-district temperature anomaly.

MCDS_days Maximum Consecutive Dry Spell, days. Primary threshold variable.

RainAnomaly Standardised within-district rainfall anomaly.

Pesticide_MT Pesticide consumption, metric tonnes.

NPK_MT NPK fertiliser consumption, metric tonnes.

Irrigation_pct Share of gross cropped area irrigated.

VarCost_Rs Variable input cost per hectare, INR.

KVK_intensity Agricultural training intensity.

KVK_expd Agricultural training expenditure, INR lakhs.

ExtWorker_per1000 Extension workers per 1000 farm households.

PACS_count Number of primary agricultural credit societies.

APMC_yards Number of agricultural market yards.

Road_density Rural road density.

AvgHolding_ha Average holding size, hectares.

SmallFarmer_share Share of holdings below 2 hectares.

CropIntensity Cropping intensity index.

Herfindahl Crop concentration index, 0 to 1.

Rainfed_share Share of gross cropped area that is rainfed.

Source

Compiled from public government and research institution sources for Maharashtra, India. Deposited at Mendeley Data: doi:10.17632/sr2gbfxk2d.2.

Examples

```
data(maharashtra_panel)
head(maharashtra_panel)
```

`ptr_hansen` *Hansen Panel Threshold Regression*

Description

Estimates a two-regime panel threshold regression model following Hansen (1999, 2000), with two-way fixed effects, grid search for the optimal threshold, bootstrap likelihood ratio test, and confidence interval construction by likelihood ratio inversion.

Usage

```
ptr_hansen(
  formula,
  data,
  threshold_var,
  index,
  trim = c(0.15, 0.85),
  n_boot = 300,
  seed = 42,
  verbose = TRUE
)
```

Arguments

`formula`	A formula of the form `y ~ x + controls`. The first right-hand-side variable is treated as the key variable of interest.
`data`	A data frame.
`threshold_var`	Character string naming the threshold variable.
`index`	A character vector of length 2: `c("unit_id", "time_id")`.
`trim`	Numeric vector of length 2 giving the lower and upper percentile trimming bounds for the threshold grid. Default is `c(0.15, 0.85)`.
`n_boot`	Integer. Bootstrap replications for the likelihood ratio test. Default is `300`.
`seed`	Integer or NULL. Random seed for reproducibility. Set to `NULL` to use the current random state without fixing the seed. Default is `42`.
`verbose`	Logical. If `TRUE`, prints progress messages. Default is `TRUE`.

Value

A list of class `ptr_hansen` with elements `threshold`, `ci`, `LR_stat`, `p_value`, `cv`, `coef_regime1`, `coef_regime2`, `regime_shift`, `ssr_grid`, `LR_boot`, `fit_regime1`, `fit_regime2`, and `call`.

References

Hansen BE (1999). "Threshold Effects in Non-Dynamic Panels: Estimation, Testing, and Inference." *Journal of Econometrics*, 93(2), 345-368. doi:10.1016/S03044076(99)000251

Hansen BE (2000). "Sample Splitting and Threshold Estimation." *Econometrica*, 68(3), 575-603. doi:10.1111/14680262.00124

Examples

```
data(maharashtra_panel)
result <- ptr_hansen(
  formula       = ln_GDVA ~ ln_HYV + MCDS_days + Irrigation_pct,
  data          = maharashtra_panel,
  threshold_var = "MCDS_days",
  index         = c("district", "year"),
  n_boot        = 50,
  seed          = 42,
  verbose       = FALSE
)
print(result)
summary(result)
coef(result)
confint(result)
```

`sdm_impact` *Spatial Durbin Model with Impact Decomposition*

Description

Constructs a k-nearest neighbour spatial weights matrix, runs Moran's I pre-tests for spatial dependence, estimates a panel Spatial Durbin Model, and computes the direct, indirect, and total impact decomposition following LeSage and Pace (2009).

Usage

```
sdm_impact(formula, data, index, coords, k = 4, verbose = TRUE)
```

Arguments

`formula`	A formula of the form `y ~ x + controls`.
`data`	A data frame.
`index`	A character vector of length 2: `c("unit_id", "time_id")`.
`coords`	A matrix or data frame with two columns giving the longitude and latitude coordinates of each unique cross-sectional unit in the same order as `sort(unique(data[[index[1]]]))`.
`k`	Integer. Number of nearest neighbours. Default is `4`.
`verbose`	Logical. Default is `TRUE`.

Value

A list of class `sdm_impact` with elements `W`, `impacts`, `spillover_ratio`, `rho`, `moran_y`, `moran_x`, `fit`, and `call`.

References

LeSage J, Pace RK (2009). *Introduction to Spatial Econometrics*. CRC Press, Boca Raton. ISBN 978-1-4200-6424-7.

Examples

```
# Requires coordinate columns not in the bundled maharashtra_panel.
# Replace with real district centroids in your application.
data(maharashtra_panel)
units  <- sort(unique(maharashtra_panel$district))
coords <- data.frame(
  longitude = stats::runif(length(units), 73, 80),
  latitude  = stats::runif(length(units), 16, 22)
)
result <- sdm_impact(
  formula = ln_GDVA ~ ln_HYV + MCDS_days + Irrigation_pct,
  data    = maharashtra_panel,
  index   = c("district", "year"),
  coords  = coords,
  k       = 4
)
print(result)
coef(result)
```

---

tsq_consistency_check *Cross-Layer Consistency Check*

Description

Evaluates whether the regime shift from `ptr_hansen()`, the spillover ratio from `sdm_impact()`, and the distributional gradient from `canay_quantile()` are mutually consistent with a common structural relationship.

Usage

```
tsq_consistency_check(ptr_result, sdm_result, qr_result, verbose = TRUE)
```

Arguments

`ptr_result`	Object of class `ptr_hansen`.
`sdm_result`	Object of class `sdm_impact`.
`qr_result`	Object of class `canay_quantile`.
`verbose`	Logical. Default is `TRUE`.

Value

A list of class `tsq_consistency` with elements `criterion1`, `criterion2`, `criterion3`, `all_pass`, and `summary_table`.

Examples

```
ptr_mock <- structure(
  list(threshold = 23, regime_shift = 0.30,
       coef_regime1 = 0.346, coef_regime2 = 0.646),
  class = "ptr_hansen"
)
sdm_mock <- structure(
  list(spillover_ratio = 0.467, rho = 0.31),
  class = "sdm_impact"
)
qr_mock <- structure(
  list(gradient = c(Q10 = 0.57, Q50 = 0.55, Q90 = 0.52),
       gradient_highstress = c(Q10 = 0.62, Q50 = 0.56, Q90 = 0.50),
       taus = c(0.10, 0.50, 0.90), key_var = "ln_HYV"),
  class = "canay_quantile"
)
result <- tsq_consistency_check(ptr_mock, sdm_mock, qr_mock)
print(result)
```

---

tsq_panel *Sequential Threshold-Spatial-Quantile Panel Estimation*

Description

Wrapper that runs the full four-step TSQ protocol: a two-way fixed effects baseline, `ptr_hansen()`, `sdm_impact()`, and `canay_quantile()`, followed by `tsq_consistency_check()`.

Usage

```
tsq_panel(
  formula,
  data,
  index,
  threshold_var,
  coords,
  k = 4,
  taus = c(0.1, 0.25, 0.5, 0.75, 0.9),
  trim = c(0.15, 0.85),
  n_boot_ptr = 300,
  n_boot_qr = 500,
  seed = 42,
  verbose = TRUE
)
```

Arguments

formula	A formula of the form y ~ x + controls.
data	A data frame.
index	A character vector of length 2: c("unit_id", "time_id").
threshold_var	Character string naming the threshold variable.
coords	A matrix or data frame with coordinate columns, one row per unique cross-sectional unit in sorted order.
k	Integer. Nearest neighbours for the spatial weights matrix. Default is 4.
taus	Numeric vector of quantile levels. Default is c(0.10, 0.25, 0.50, 0.75, 0.90).
trim	Numeric vector of length 2 for threshold grid trimming. Default is c(0.15, 0.85).
n_boot_ptr	Integer. Bootstrap replications for the threshold test. Default is 300.
n_boot_qr	Integer. Bootstrap replications for quantile standard errors. Default is 500.
seed	Integer or NULL. Random seed. Set to NULL to use the current random state. Default is 42.
verbose	Logical. Default is TRUE.

Value

A list of class tsq_panel with elements step1_baseline, step2_ptr, step3_sdm, step4_qr, consistency, and call.

Examples

```
# Requires coordinate data; replace with real district centroids.
data(maharashtra_panel)
units  <- sort(unique(maharashtra_panel$district))
coords <- data.frame(
  longitude = stats::runif(length(units), 73, 80),
  latitude  = stats::runif(length(units), 16, 22)
)
result <- tsq_panel(
  formula       = ln_GDVA ~ ln_HYV + MCDS_days + Irrigation_pct,
  data          = maharashtra_panel,
  index         = c("district", "year"),
  threshold_var = "MCDS_days",
  coords        = coords,
  n_boot_ptr    = 50,
  n_boot_qr     = 50
)
print(result)
coef(result)
```

`tsq_plot` *Plot TSQ Protocol Results*

Description

Produces diagnostic plots for TSQ protocol results.

Usage

```
tsq_plot(x, type = "all", ...)
```

Arguments

`x`	An object of class `tsq_panel`, `ptr_hansen`, `sdm_impact`, or `canay_quantile`.
`type`	Character. One of `"threshold"`, `"spillover"`, `"gradient"`, or `"all"`. Default is `"all"`.
`...`	Currently unused.

Value

A `ggplot` object or named list of `ggplot` objects.

Examples

```
data(maharashtra_panel)
ptr <- ptr_hansen(
  formula       = ln_GDVA ~ ln_HYV + MCDS_days + Irrigation_pct,
  data          = maharashtra_panel,
  threshold_var = "MCDS_days",
  index         = c("district", "year"),
  n_boot        = 50,
  verbose       = FALSE
)
p <- tsq_plot(ptr, type = "threshold")
print(p)
```

Index