

# Package: tsqr (via r-universe)

June 22, 2026

**Title** Sequential Threshold-Spatial-Quantile Panel Estimation

**Version** 0.1.2

**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](https://doi.org/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](https://doi.org/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](https://doi.org/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** <https://cran.r-universe.dev>

**Date/Publication** 2026-06-22 18:33:23 UTC

**RemoteUrl** <https://github.com/cran/tsqr>

**RemoteRef** HEAD

**RemoteSha** d03b700fce54499c14a31b3efeffc9a851bd3904

## Contents

canay_quantile . . . . .	2
maharashtra_panel . . . . .	4
ptr_hansen . . . . .	5
sdm_impact . . . . .	7
tsq_consistency_check . . . . .	8
tsq_panel . . . . .	9
tsq_plot . . . . .	11
<b>Index</b>	<b>12</b>

---

canay_quantile	<i>Canay Two-Step Panel Quantile Regression</i>
----------------	---

---

## Description

Implements the Canay (2011) two-step panel quantile estimator. In the first step, unit and time fixed effects are estimated from a standard two-way fixed effects model and subtracted from the outcome. In the second step, standard quantile regression is applied to the corrected outcome 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 $y \sim x + \text{controls}$ .
data	A data frame.
index	A character vector of length 2 giving unit and time identifiers.
taus	Numeric vector of quantile levels. Default is $c(0.10, 0.25, 0.50, 0.75, 0.90)$ .
n_boot	Integer. Number of bootstrap replications for standard errors. Default is 500.
seed	Integer. Random seed for reproducibility. Default is 42.
ptr_result	Optional object of class <code>ptr_hansen</code> . If supplied, the function also estimates the quantile regression on the high-regime subsample defined by <code>threshold_var</code> exceeding the estimated threshold.
threshold_var	Character string naming the threshold variable used to define the high-regime subsample. Required if <code>ptr_result</code> is supplied.
verbose	Logical. Default is TRUE.

**Details**

The returned list contains `coefs` (a data frame of quantile-specific coefficients for all variables and all `taus`), `se` (a matching data frame of bootstrap standard errors), `slope_tests` (a data frame of pairwise slope equality z-tests for the key variable), `gradient` (a named numeric vector giving the coefficient on the key variable at each quantile), `gradient_highstress` (as `gradient` for the high-regime subsample, or NULL if `ptr_result` was not supplied), `fe_corrected` (the fixed-effect-corrected outcome vector), and `call` (the matched function call).

**Value**

A list of class `canay_quantile` containing the quantile coefficients, bootstrap standard errors, pairwise slope equality tests, and (optionally) the high-stress subsample results. See Details.

**References**

Canay IA (2011). "A Simple Approach to Quantile Regression for Panel Data." *Econometrics Journal*, 14(3), 368-386.

**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)
```

---

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). This dataset is the empirical illustration used throughout this package.

## 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 prices, in INR crore.

**ln\_GDVA** Natural log of GDVA\_agri\_crore.

**FarmIncome\_Rs000** Farm household income proxy, INR thousands.

**HYV\_area\_000ha** Area under high-yielding crop varieties, thousand hectares.

**ln\_HYV** Natural log of HYV\_area\_000ha.

**Rainfall\_mm** Annual rainfall, millimetres.

**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.

**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 under irrigation.

**VarCost\_Rs** Variable input cost per hectare, INR.

**KVK\_intensity** Agricultural training intensity.

**KVK\_expd** Agricultural training expenditure, INR lakhs.

**ExtWorker\_per1000** Extension worker density 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 operational 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 secondary government and research institution sources for the state of Maharashtra, India.

### 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 $y \sim x + \text{controls}$ (do not include the threshold interaction terms; these are added internally).
data	A data frame containing all variables in formula and the threshold variable.
threshold_var	Character string naming the threshold variable (the moderating variable hypothesised to govern the structural break).
index	A character vector of length 2 giving the unit and time identifiers, e.g. <code>c("district", "year")</code> .
trim	A numeric vector of length 2 giving the lower and upper percentile trimming bounds for the threshold grid. Default is <code>c(0.15, 0.85)</code> .
n_boot	Integer. Number of bootstrap replications for the likelihood ratio test. Default is 300.
seed	Integer. Random seed for reproducibility. Default is 42.
verbose	Logical. If TRUE, prints progress messages. Default is TRUE.

## Details

The returned list contains the following elements: `threshold` (the optimal threshold estimate), `ci` (a named numeric vector with lower and upper bounds of the 95 percent confidence interval), `LR_stat` (the likelihood ratio statistic), `p_value` (the bootstrap p-value for the null of no threshold), `cv` (a named vector of bootstrap critical values at the 10, 5, and 1 percent levels), `coef_regime1` and `coef_regime2` (the coefficient on the key variable in each regime), `regime_shift` (the difference between the two regime coefficients), `ssr_grid` (a data frame of threshold candidates and their sum of squared residuals), `LR_boot` (the vector of bootstrap likelihood ratio statistics), `fit_regime1` and `fit_regime2` (the fitted regime models), and `call` (the matched function call).

## Value

A list of class `ptr_hansen` containing the threshold estimate, confidence interval, likelihood ratio statistic, bootstrap p-value, regime-specific coefficients, and the fitted regime models. See Details.

## References

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

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

## 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)

```

---

sdm\_impact

*Spatial Durbin Model with Impact Decomposition*


---

## Description

Constructs a k-nearest neighbour spatial weights matrix, runs pre-tests for spatial dependence, estimates a panel Spatial Durbin Model (SDM), 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 $y \sim x + \text{controls}$ .
data	A data frame.
index	A character vector of length 2 giving unit and time identifiers.
coords	A matrix or data frame with two columns giving the longitude and latitude (or x and y) coordinates of each unique cross-sectional unit, in the same order as the sorted unique values of <code>data[[index[1]]]</code> .
k	Integer. Number of nearest neighbours for the spatial weights matrix. Default is 4.
verbose	Logical. Default is TRUE.

## Details

The returned list contains *W* (the row-standardised spatial weights matrix), *impacts* (a data frame with Direct, Indirect, and Total columns for each variable), *spillover\_ratio* (the Indirect over Total ratio for the key variable), *rho* (the spatial autoregressive coefficient), *moran\_y* and *moran\_x* (Moran's I test results on the cross-sectional means of the outcome and key variable), *fit* (the fitted spatial model object), and *call* (the matched function call).

## Value

A list of class `sdm_impact` containing the spatial weights matrix, the impact decomposition table, the spillover ratio, the spatial autoregressive coefficient, Moran's I test results, and the fitted model object. See Details.

## References

LeSage J, Pace RK (2009). *Introduction to Spatial Econometrics*. CRC Press, Boca Raton.

## Examples

```
## Not run:
# sdm_impact() requires coordinate columns not included in the
# bundled maharashtra_panel dataset. Supply your own coordinates:
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)

## End(Not run)
```

---

tsq\_consistency\_check *Cross-Layer Consistency Check*

---

## Description

Evaluates whether the regime shift identified by [ptr\\_hansen\(\)](#), the spillover ratio identified by [sdm\\_impact\(\)](#), and the distributional gradient identified by [canay\\_quantile\(\)](#) are mutually consistent with a common underlying structural relationship.

## Usage

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

## Arguments

<code>ptr_result</code>	Object of class <code>ptr_hansen</code> .
<code>sdm_result</code>	Object of class <code>sdm_impact</code> .
<code>qr_result</code>	Object of class <code>canay_quantile</code> .
<code>verbose</code>	Logical. Default is <code>TRUE</code> .

## Details

Criterion 1 checks whether the regime shift and the spillover ratio have the same sign. Criterion 2 checks whether the distributional gradient is steeper in the high-stress subsample than in the full sample. Criterion 3 checks whether the full-sample gradient is monotonically declining from the lowest to the highest quantile.

**Value**

A list of class `tsq_consistency` with logical elements `criterion1`, `criterion2`, `criterion3`, and `all_pass`, together with a `summary_table` data frame. See Details.

**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"
)
check <- tsq_consistency_check(ptr_mock, sdm_mock, qr_mock, verbose = FALSE)
print(check)
```

---

 tsq\_panel

---

*Sequential Threshold-Spatial-Quantile Panel Estimation*


---

**Description**

Runs the full four-step sequential panel estimation 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 $y \sim x + \text{controls}$ . The first variable on the right-hand side is treated as the key variable of interest.
data	A data frame.
index	A character vector of length 2: <code>c("unit_id", "time_id")</code> .
threshold_var	Character string naming the threshold variable for <code>ptr_hansen()</code> .
coords	A matrix or data frame with coordinate columns, one row per unique cross-sectional unit in sorted order. Required for the spatial step.
k	Integer. Nearest neighbours for the spatial weights matrix. Default is 4.
taus	Numeric vector of quantile levels. Default is <code>c(0.10, 0.25, 0.50, 0.75, 0.90)</code> .
trim	Numeric vector of length 2 for threshold grid trimming. Default is <code>c(0.15, 0.85)</code> .
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. Random seed. Default is 42.
verbose	Logical. Default is TRUE.

**Value**

A list of class `tsq_panel` containing `step1_baseline`, `step2_ptr`, `step3_sdm`, `step4_qr`, `consistency`, and `call`.

**Examples**

```
## Not run:
# This example requires coordinate data not included in the bundled
# maharashtra_panel dataset; see ?sdm_impact for a worked example.
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)

## End(Not run)
```

---

tsq_plot	<i>Plot TSQ Protocol Results</i>
----------	----------------------------------

---

**Description**

Produces diagnostic plots for TSQ protocol results: a likelihood ratio profile for threshold identification, a spillover decomposition bar chart, or a distributional gradient plot across quantiles.

**Usage**

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

**Arguments**

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

**Value**

A ggplot object, or a named list of ggplot objects when `type = "all"` and `x` is of class `tsq_panel`.

**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")
```

# Index

canay\_quantile, [2](#)  
canay\_quantile(), [8](#), [9](#)

maharashtra\_panel, [4](#)

ptr\_hansen, [5](#)  
ptr\_hansen(), [8–10](#)

sdm\_impact, [7](#)  
sdm\_impact(), [8](#), [9](#)

tsq\_consistency\_check, [8](#)  
tsq\_consistency\_check(), [9](#)  
tsq\_panel, [9](#)  
tsq\_plot, [11](#)