Package: Westerlund (via r-universe)

November 6, 2026

Type Package

Title Panel Cointegration Tests Based on Westerlund (2007)

Version 0.1.4

Description Implements a functional approximation of the four panel cointegration tests developed by Westerlund (2007) <doi:10.1111/j.1468-0084.2007.00477.x>. The tests are based on structural rather than residual dynamics and allow for heterogeneity in both the long-run cointegrating relationship and the short-run dynamics. The package includes logic for automated lag and lead selection via AIC/BIC, Bartlett kernel long-run variance estimation, and a bootstrap procedure to handle cross-sectional dependence. It also includes a bootstrapping distribution visualization function for diagnostic purposes.

License MIT + file LICENSE

URL https://github.com/bosco-hung/WesterlundTest

BugReports https://github.com/bosco-hung/WesterlundTest/issues

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices, utils, scales, dplyr, ggplot2, tidyr

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev

Repository https://bosco-hung.r-universe.dev

Date/Publication 2026-10-05 19:23:39 UTC

RemoteUrl https://github.com/bosco-hung/westerlundtest

RemoteRef HEAD

RemoteSha 7f54f21bb5b1a6c946c0222984663e8e22637e72

RemoteSubdir R

Contents

calc_lrvar_bartlett . . . 2
DisplayWesterlund . . . 4
get_diff . . . 8
get_lag . . . 10
get_ts_val . . . 12
plot.westerlund_test . . . 13
print.summary.westerlund_test . . . 15
print.westerlund_test . . . 15
shiftNA . . . 16
summary.westerlund_test . . . 17
westerlund_test . . . 18
westerlund_test_mg . . . 21
westerlund_test_reg . . . 23
WesterlundBootstrap . . . 24
WesterlundPlain . . . 26

Index **30**

`calc_lrvar_bartlett` *Long-Run Variance Estimation with Bartlett Kernel*

Description

Computes a Bartlett-kernel (Newey–West style) long-run variance estimate for a univariate series using Stata-like conventions: missing values are removed (`na.omit`), autocovariances are scaled by $1/n$ (not $1/(n-j)$), and optional centering is controlled by `nodemean`.

Usage

```
calc_lrvar_bartlett(x, maxlag, nodemean = FALSE)
```

Arguments

`x`	A numeric vector. Missing values are removed prior to computation.
`maxlag`	Non-negative integer. Maximum lag order m used in the Bartlett kernel. If `maxlag=0`, the function returns the sample variance estimate γ_0.
`nodemean`	Logical. If `FALSE` (default), the series is centered by subtracting its mean before computing autocovariances. If `TRUE`, the function assumes `x` is already centered and does not de-mean it.

Details

Let x_t be the input series after removing missing values. If `nodemean=FALSE`, the function replaces x_t with $x_t - \bar{x}$.

Define the lag-j autocovariance using the Stata-style scaling:

$$\gamma_j = \frac{1}{n} \sum_{t=j+1}^{n} x_t x_{t-j}, \qquad j = 0, 1, \ldots, m,$$

where n is the length of the cleaned series and $m =$ `maxlag`. Note that the scaling uses $1/n$ for all j (rather than $1/(n-j)$).

The Bartlett kernel weight at lag j is:

$$w_j = 1 - \frac{j}{m+1}.$$

The long-run variance estimate is then:

$$\widehat{\Omega} = \gamma_0 + 2 \sum_{j=1}^{m} w_j \gamma_j.$$

If the cleaned series has zero length, the function returns `NA`.

Value

A single numeric value:

- The Bartlett-kernel long-run variance estimate $\widehat{\Omega}$ (scalar),
- or `NA` if `x` contains no non-missing values.

Technical Notes

This section illustrates how `calc_lrvar_bartlett()` computes a Bartlett-kernel long-run variance (LRV) estimate and how its options map to common time-series preprocessing choices.

Stata-like conventions: This helper function is designed to match Stata-style calculations:

- **Missing values are dropped**: `na.omit(x)` is applied first.
- **Scaling by** $1/n$: autocovariances at all lags divide by n, not by $n - j$.
- **Optional de-meaning**: controlled by `nodemean`.

Centering: By default (`nodemean=FALSE`), the series is centered: $x_t \leftarrow x_t - \bar{x}$. Set `nodemean=TRUE` when you have already centered the series elsewhere.

Choosing maxlag: `maxlag` sets the truncation point m. Larger m captures more serial correlation but increases estimation noise.

See Also

`westerlund_test`

Examples

```
## Example 1: Basic usage
x <- rnorm(200)
calc_lrvar_bartlett(x, maxlag = 4)

## Example 2: maxlag = 0 returns gamma_0
calc_lrvar_bartlett(x, maxlag = 0)

## Example 3: Handle missing values (they are removed)
x_na <- x
x_na[c(5, 10, 50)] <- NA
calc_lrvar_bartlett(x_na, maxlag = 4)

## Example 4: Compare centering choices
## Default: de-mean internally
lr1 <- calc_lrvar_bartlett(x, maxlag = 4, nodemean = FALSE)

## Pre-center and skip de-meaning
x_centered <- x - mean(x)
lr2 <- calc_lrvar_bartlett(x_centered, maxlag = 4, nodemean = TRUE)
```

DisplayWesterlund *Standardize and Display Westerlund ECM Panel Cointegration Test Results*

Description

Formats, standardizes, and prints the Westerlund (2007) ECM-based panel cointegration test results for the null hypothesis of no cointegration. Given raw statistics G_t, G_a, P_t, and P_a, the function computes standardized Z-statistics using tabulated (or hard-coded) asymptotic moments and reports left-tail asymptotic p-values. If a bootstrap distribution is supplied, it also computes bootstrap (robust) p-values for the raw statistics.

Usage

```
DisplayWesterlund(
  stats,
  bootstats = NULL,
  nobs,
  nox,
  constant = FALSE,
  trend = FALSE,
  meanlag = -1,
  meanlead = -1,
  realmeanlag = -1,
  realmeanlead = -1,
  auto = 0,
  westerlund = FALSE,
```

```
  aic = TRUE,
  verbose = FALSE
)
```

Arguments

stats	A numeric vector of length 4 or a 1x4 numeric matrix containing the raw test statistics in the order c(Gt, Ga, Pt, Pa).
bootstats	Optional. A numeric matrix with bootstrap replications of the raw statistics, with 4 columns in the order [Gt, Ga, Pt, Pa]. If provided, robust (bootstrap) p-values are computed and displayed.
nobs	Integer. Number of valid cross-sectional units (series) used in the test.
nox	Integer. Number of covariates in the long-run relationship (length of xvars in upstream calls). Used to select asymptotic moments for standardization in the non-westerlund case.
constant	Logical. Indicates whether a constant was included in the cointegrating relationship. Affects the deterministic-case row used for asymptotic moment lookup.
trend	Logical. Indicates whether a trend was included. If TRUE, constant is assumed to be TRUE and the deterministic-case row for moment lookup changes accordingly.
meanlag	Integer. Mean selected lag length (rounded down to integer) reported in the header when auto is non-zero.
meanlead	Integer. Mean selected lead length (rounded down to integer) reported in the header when auto is non-zero.
realmeanlag	Numeric. Unrounded mean selected lag length (for display when auto is non-zero).
realmeanlead	Numeric. Unrounded mean selected lead length (for display when auto is non-zero).
auto	Logical/integer. If non-zero, the header prints the average AIC-selected lag and lead lengths based on realmeanlag and realmeanlead.
westerlund	Logical. If TRUE, uses the hard-coded Westerlund-specific moment constants for standardization (separate constants for trend vs no-trend). If FALSE, uses lookup tables indexed by deterministic case and number of covariates.
aic	Logical. If TRUE, AIC was used for lag/lead selection; If FALSE, BIC was used.
verbose	Logical. If TRUE, prints additional output.

Details

What this function does. DisplayWesterlund() takes raw Westerlund ECM test statistics and produces:

- standardized Z-statistics for G_t, G_a, P_t, P_a,
- left-tail asymptotic p-values using pnorm(),
- optionally, bootstrap (robust) p-values when bootstats is provided,

- a console table summarizing the results.

Standardization and deterministic cases. The function selects a deterministic-case index:

- Row 1: no constant, no trend,
- Row 2: constant, no trend,
- Row 3: constant and trend,

implemented as `row_idx = (as.integer(constant) + as.integer(trend)) + 1`. In the non-`westerlund` case, asymptotic means and variances are taken from hard-coded lookup matrices indexed by `row_idx` and `nox`. Z-statistics are computed using:

$$Z = \frac{\sqrt{N}S - \sqrt{N}\mu}{\sqrt{\sigma^2}}$$

for mean-group statistics and analogous formulas for pooled statistics as implemented in the code, where $N =$ `nobs` and S is the raw statistic.

In the `westerlund=TRUE` case, the function uses a separate set of hard-coded mean/variance constants, with different values depending on whether `trend` is included.

Asymptotic p-values. All asymptotic p-values are computed as left-tail probabilities: `pnorm(Z)`.

Bootstrap p-values (robust p-values). If `bootstats` is supplied, bootstrap p-values are computed for each raw statistic using a left-tail empirical rule:

$$\hat{p} = \frac{r+1}{B+1}, \qquad r = \sum_{b=1}^{B} \mathbb{I}(S_b \leq S_{\text{obs}}),$$

after dropping non-finite bootstrap draws. This finite-sample correction is an intentional feature of this R implementation.

Return values. In addition to printing, the function returns a list containing the raw statistics, Z-statistics, asymptotic p-values, and (if applicable) bootstrap p-values.

Value

A list containing at minimum:

- `gt`, `ga`, `pt`, `pa`: raw test statistics,
- `gt_z`, `ga_z`, `pt_z`, `pa_z`: standardized Z-statistics,
- `gt_pval`, `ga_pval`, `pt_pval`, `pa_pval`: left-tail asymptotic p-values.

If `bootstats` is provided, the list also includes:

- `gt_pvalboot`, `ga_pvalboot`, `pt_pvalboot`, `pa_pvalboot`: bootstrap (robust) p-values for the raw statistics.

Vignette

This section explains how to use `DisplayWesterlund()` and how its output connects to the broader testing workflow.

Where does `DisplayWesterlund()` fit?: Typically, the workflow is:

1. Compute observed raw statistics via `WesterlundPlain`.
2. Optionally compute a bootstrap distribution via `WesterlundBootstrap`.
3. Call `DisplayWesterlund()` to standardize, print, and return p-values.

The user-facing `westerlund_test` wraps these steps and collects the returned scalars.

Input format for `stats`: `stats` can be either a numeric vector `c(Gt, Ga, Pt, Pa)` or a `1x4` matrix with `[1,]` corresponding to `Gt, Ga, Pt, Pa`.

Asymptotic standardization and left-tail p-values: The function converts raw statistics to Z-statistics using hard-coded asymptotic means and variances. P-values are computed as `pnorm(Z)`, i.e., a left-tail test.

Bootstrap (robust) p-values: If `bootstats` is provided, robust p-values are computed by comparing the observed raw statistic to its bootstrap distribution, using a finite-sample correction: $(r + 1)/(B + 1)$ where r is the number of bootstrap draws less than or equal to the observed statistic.

Examples:

```
## Example 1: Asymptotic-only display (no bootstrap)
stats <- c(Gt = -2.1, Ga = -9.5, Pt = -1.8, Pa = -6.2)

res1 <- DisplayWesterlund(
  stats      = stats,
  bootstats = NULL,
  nobs       = 18,
  nox        = 2,
  constant   = TRUE,
  trend      = FALSE,
  auto       = 0,
  westerlund = FALSE
)

## Example 2: With bootstrap distribution (robust p-values)
set.seed(123)
bootstats <- cbind(
  rnorm(399, mean = -1.8, sd = 0.9),
  rnorm(399, mean = -8.0, sd = 3.0),
  rnorm(399, mean = -1.5, sd = 1.0),
  rnorm(399, mean = -5.0, sd = 3.5)
)

res2 <- DisplayWesterlund(
  stats      = stats,
```

```
  bootstats = bootstats,
  nobs       = 18,
  nox        = 2,
  constant   = TRUE,
  trend      = FALSE,
  auto       = 1,
  realmeanlag  = 1.35,
  realmeanlead = 0.40,
  westerlund = FALSE
)
```

References

Westerlund, J. (2007). Testing for error correction in panel data. *Oxford Bulletin of Economics and Statistics*, 69(6), 709–748.

See Also

`westerlund_test`, `WesterlundPlain`, `WesterlundBootstrap`

`get_diff` *First Difference with Strict Time Indexing*

Description

Computes the first difference of a time-indexed series using strict time-based lagging. The function respects gaps in the time index and returns `NA` when the previous time period does not exist, mirroring Stata's `D.` operator.

Usage

```
get_diff(vec, tvec)
```

Arguments

`vec`	A numeric (or atomic) vector of observations.
`tvec`	A vector of time indices corresponding one-to-one with `vec`. Each value must uniquely identify a time period within the series.

Details

This helper function computes first differences as:

$$\Delta x_t = x_t - x_{t-1},$$

where the lagged value x_{t-1} is obtained using `get_lag`, which performs strict time-based lookup.

Internally, the function calls:

```
val_t_minus_1 <- get_lag(vec, tvec, 1)
```

and then subtracts this lagged vector from `vec`. If the time index contains gaps, or if the previous time period does not exist for a given observation, the lagged value is `NA` and the corresponding difference is also `NA`.

No interpolation or implicit shifting is performed; missing time periods propagate as missing differences.

Value

A vector of the same length as `vec`, containing the first differences aligned by the time index. Elements are `NA` when the previous time period does not exist.

Time Indexing Logic

This section explains how `get_diff()` computes first differences and why strict time indexing matters in the presence of gaps.

Relation to Stata's D. operator: The function replicates the behaviour of Stata's first-difference operator `D.x`. When time periods are missing, Stata returns missing values rather than differencing across gaps. Because `get_diff()` relies on `get_lag`, it follows the same rule.

Why not use diff()?: The base R function `diff()` computes differences based on vector positions. This implicitly assumes a complete and regularly spaced time index. When time periods are missing, `diff()` can produce misleading results by differencing across gaps. `get_diff()` avoids this by differencing only when the previous time period exists.

See Also

`get_lag`, `get_ts_val`, `westerlund_test`

Examples

```
## Example 1: Regular time series
t <- 1:5
x <- c(10, 20, 30, 40, 50)

get_diff(x, t)
# [1] NA 10 10 10 10

## Example 2: Time series with a gap
t_gap <- c(1, 2, 4, 5)
x_gap <- c(10, 20, 40, 50)

get_diff(x_gap, t_gap)
# [1] NA 10 NA 10

## Explanation:
## At t = 4, the previous period t-1 = 3 does not exist, so the difference is NA.

## Example 3: Comparison with diff()
```

```
diff(x_gap)
# [1] 10 20 10
```

---

get_lag *Time-Indexed Lag Extraction with Explicit Gaps*

Description

Extracts lagged (or led) values from a time-indexed vector using a strict time-based lookup. The function respects gaps in the time index and returns NA when the requested lagged time does not exist, mirroring the behaviour of Stata's time-series lag/lead operators.

Usage

```
get_lag(vec, tvec, k)
```

Arguments

vec	A numeric (or atomic) vector of observations.
tvec	A vector of time indices corresponding one-to-one with vec. Each value must uniquely identify a time period within the series.
k	Integer. The lag order. Positive values correspond to lags (e.g., k = 1 for $t-1$), while negative values correspond to leads (e.g., k = -1 for $t+1$).

Details

This helper function performs strict time-based lagging rather than position-based shifting. Internally, it constructs a mapping from time indices to observed values:

```
val_map <- setNames(vec, tvec)
```

For each observation at time t, the function retrieves the value associated with time $t-k$. If that time value is not present in tvec, the result is NA.

This behaviour is particularly important when working with irregular time series or panel data with gaps. In such cases, simple vector shifting can incorrectly carry values across missing time periods, while get_lag() preserves the correct alignment.

No interpolation, padding, or reordering is performed; missing time periods propagate as NA in the lagged (or led) series.

Value

A vector of the same length as vec, containing the lagged (or led) values aligned by the time index. Elements are NA when the requested time period does not exist.

Technical Background

This section illustrates the logic of `get_lag()` and explains how it differs from standard position-based lagging.

Why not simple shifting?: In many applications, lags are computed by shifting vectors by positions. This implicitly assumes a complete and regular time index. When time periods are missing, such shifting produces incorrect lag values. `get_lag()` avoids this by explicitly matching on time values.

Relation to Stata operators: The function mirrors the behaviour of Stata's time-series operators:

- `L.x`: lagged value at $t - 1$,
- `L2.x`: lagged value at $t - 2$,
- `F.x`: lead value at $t + 1$.

As in Stata, gaps in the time index result in missing values in the lagged series.

See Also

`get_ts_val`, `westerlund_test`, `calc_lrvar_bartlett`

Examples

```
## Example 1: Regular time series
t <- 1:5
x <- c(10, 20, 30, 40, 50)

get_lag(x, t, k = 1)
# [1] NA 10 20 30 40

get_lag(x, t, k = -1)
# [1] 20 30 40 50 NA

## Example 2: Time series with a gap
t_gap <- c(1, 2, 4, 5)
x_gap <- c(10, 20, 40, 50)

get_lag(x_gap, t_gap, k = 1)
# [1] NA 10 NA 40

## Explanation:
## At t = 4, the requested time t-1 = 3 does not exist, so NA is returned.

## Example 3: Higher-order lags
get_lag(x_gap, t_gap, k = 2)
# [1] NA NA NA 20
```

get_ts_val *Strict Time-Series Mapping with Explicit Gaps*

Description

Retrieves lagged (or led) values from a time-indexed vector using a strict time-based mapping.

Usage

```
get_ts_val(vec, tvec, lag)
```

Arguments

vec	A numeric (or atomic) vector of observations.
tvec	A vector of time indices corresponding one-to-one with vec. Values must uniquely identify time periods within the series.
lag	Integer. The lag order. Positive values correspond to lags (e.g., lag = 1 for $t-1$), while negative values correspond to leads (e.g., lag = -1 for $t+1$).

Details

This helper function implements a strict time-based lookup rather than position-based indexing. Internally, it constructs a named mapping from time indices to observed values:

```
val_map <- setNames(vec, tvec)
```

For each observation at time t, the function computes the target time $t - \text{lag}$ and retrieves the corresponding value from the map.

If the target time does not exist in tvec, the function returns NA for that observation. This behaviour exactly mirrors Stata's lag/lead operators in the presence of gaps.

Importantly, the function does **not** interpolate or shift values when time periods are missing. A gap in the time index propagates as NA in the lagged (or led) series.

Value

A vector of the same length as vec, containing the lagged (or led) values aligned by time index. Elements are NA when the requested time period does not exist.

Implementation Details

This section explains why get_ts_val() is useful and how it differs from standard lagging approaches based on vector positions.

Why strict time-based lagging matters: In panel and time-series econometrics, lagged variables should be defined with respect to the time index, not the row position. When data contain gaps in time, simple shifting (e.g., c(NA, x[-length(x)])) can produce incorrect values. get_ts_val() avoids this by explicitly matching on time values.

Relation to Stata operators: This function replicates the behaviour of Stata's time-series operators:

- `L.x`: lagged value at $t-1$,
- `L2.x`: lagged value at $t-2$,
- `F.x`: lead value at $t+1$.

When time periods are missing, Stata returns missing values rather than shifting across gaps.

See Also

`westerlund_test`, `calc_lrvar_bartlett`

Examples

```
## Example 1: Regular time series
t <- 1:5
x <- c(10, 20, 30, 40, 50)

## Lag by one period
get_ts_val(x, t, lag = 1)
# [1] NA 10 20 30 40

## Lead by one period
get_ts_val(x, t, lag = -1)
# [1] 20 30 40 50 NA

## Example 2: Time series with a gap
t_gap <- c(1, 2, 4, 5)
x_gap <- c(10, 20, 40, 50)

## Lag by one period: note the NA at time 4
get_ts_val(x_gap, t_gap, lag = 1)
# [1] NA 10 NA 40

## Explanation:
## - At t = 4, t - 1 = 3 does not exist in t_gap, so NA is returned.

## Example 3: Higher-order lags
get_ts_val(x_gap, t_gap, lag = 2)
# [1] NA NA NA 20
```

---

plot.westerlund_test *Plot Bootstrap Distributions for Westerlund ECM Panel Cointegration Tests*

Description

Creates a 2-by-2 bootstrap density display for `Gt`, `Ga`, `Pt`, and `Pa`, including the observed statistic, a configurable lower-tail bootstrap critical value, and optional robust p-value annotations.

Usage

```
## S3 method for class 'westerlund_test'
plot(
  x,
  title = "Westerlund Test: Bootstrap Distributions",
  conf_level = 0.05,
  save_path = NULL,
  dpi = 300,
  figsize = c(12, 10),
 colors = list(obs = "#D55E00", crit = "#0072B2", fill = "grey80", density = "grey30"),
  lwd = list(obs = 1, crit = 0.8, density = 0.5),
  alpha = 0.5,
  show_grid = TRUE,
  show_robust_p = TRUE,
  ...
)
```

Arguments

x	A westerlund_test object containing bootstrap distributions.
title	Main plot title.
conf_level	Lower-tail bootstrap critical probability in (0,1).
save_path	Optional output path. If non-NULL, the plot is saved with ggsave.
dpi	Saved image resolution.
figsize	Numeric width and height in inches.
colors	Named colors for observed line, critical line, density fill, and density outline.
lwd	Named line widths for observed, critical, and density lines.
alpha	Density fill transparency.
show_grid	Logical; display major grid lines.
show_robust_p	Logical; annotate bootstrap p-values in the facets.
...	Reserved for S3 compatibility.

Details

The observed statistic is shown as a solid line and the empirical conf_level quantile as a dashed line. Since the Westerlund tests are lower-tail tests, rejection occurs when the observed statistic lies to the left of the critical value.

Value

A ggplot object. If save_path is supplied, the same plot is also saved to disk.

See Also

westerlund_test

print.summary.westerlund_test *Print Summary for Westerlund Test*

Description

Print method for summary objects of class `summary.westerlund_test`.

Usage

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

Arguments

`x`	An object of class `summary.westerlund_test`.
`...`	Additional arguments passed to or from other methods.

Value

Invisibly returns the input object `x`.

See Also

`summary.westerlund_test`, `westerlund_test`

print.westerlund_test *Print Method for Westerlund ECM Panel Cointegration Tests*

Description

Prints raw statistics, standardized Z-scores, asymptotic p-values, and bootstrap p-values when available.

Usage

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

Arguments

`x`	A `westerlund_test` object.
`...`	Unused.

Value

Returns x invisibly.

See Also

westerlund_test, summary.westerlund_test

---

shiftNA *NA-Padded Lag and Lead Operator*

Description

Shifts a vector forward or backward by a specified number of positions and fills out-of-range values with NA. This helper is designed for position-based transformations where missing values should be explicitly propagated as NA rather than assumed to be zero.

Usage

```
shiftNA(v, k)
```

Arguments

v	A vector (numeric, character, etc.).
k	Integer. Shift order. Positive values correspond to lags (shifting the series downward, i.e.\ $t - k$), while negative values correspond to leads (shifting the series upward, i.e.\ $t + k$).

Details

This function performs a position-based shift of the input vector v. Unlike strict time-indexed helpers (such as get_lag), shiftNA() does not rely on an explicit time index and does not propagate gaps based on timestamps. Instead, it performs a simple index shift, padding the resulting empty slots with NA.

Let n denote the length of v. The behaviour is:

- k = 0: return v unchanged,
- k > 0: lag by k positions; the first k elements are NA,
- k < 0: lead by $|k|$ positions; the last $|k|$ elements are NA.

Formally, for k > 0,

$$\text{out}_t = \begin{cases} \text{NA}, & t \le k, \\ v_{t-k}, & t > k, \end{cases}$$

and for k < 0,

$$\text{out}_t = \begin{cases} v_{t+|k|}, & t \le n - |k|, \\ \text{NA}, & t > n - |k|. \end{cases}$$

Value

A vector of the same length as v, containing the shifted values with NA inserted where the shift would otherwise go out of range.

Usage and Propagation

This section explains the implications of using NA padding in recursive or difference-based calculations.

Why NA-padding?: Using NA is the standard behavior in R for out-of-bounds operations. It ensures that subsequent calculations (like v - shiftNA(v, 1)) correctly result in NA for boundary cases, preventing the accidental use of arbitrary values (like zero) in statistical estimations unless explicitly intended.

Difference from get_lag(): Unlike get_lag, shiftNA() is position-based and ignores time indices. Use shiftNA() when you need a fast, simple shift on vectors that are already correctly ordered.

See Also

WesterlundBootstrap, get_lag, get_diff

Examples

```
v <- c(1, 2, 3, 4, 5)

## Lag by one (k = 1)
shiftNA(v, 1)
# [1] NA 1 2 3 4

## Lead by one (k = -1)
shiftNA(v, -1)
# [1] 2 3 4 5 NA

## Larger shifts
shiftNA(v, 2)
# [1] NA NA 1 2 3

shiftNA(v, -2)
# [1] 3 4 5 NA NA
```

---

summary.westerlund_test

Summary for Westerlund Test

Description

Summary method for objects of class westerlund_test.

Usage

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

Arguments

object	An object of class westerlund_test.
...	Additional arguments passed to or from other methods.

Value

An object of class summary.westerlund_test.

---

westerlund_test *Westerlund ECM-Based Panel Cointegration Test*

Description

Computes the Westerlund (2007) ECM-based panel cointegration statistics G_t, G_a, P_t, and P_a, with optional automatic lag/lead selection, bootstrap inference, unit-level ECM output, and mean-group summaries.

Usage

```
westerlund_test(
  data,
  yvar,
  xvars,
  idvar,
  timevar,
  constant = FALSE,
  trend = FALSE,
  lags = 1,
  leads = NULL,
  westerlund = FALSE,
  aic = TRUE,
  bootstrap = -1,
  indiv.ecm = FALSE,
  lrwindow = 2,
  seed = NULL,
  verbose = FALSE
)
```

Arguments

data	A data.frame containing the panel data.
yvar	String naming the dependent variable.
xvars	Character vector naming the long-run regressors. At most six are supported; with westerlund=TRUE, at most one is allowed.
idvar	String naming the panel identifier.
timevar	String naming the time variable.
constant	Logical. Include a constant.
trend	Logical. Include a trend. Requires constant=TRUE.
lags	Integer or length-2 integer vector giving a fixed lag or selection range. Defaults to 1.
leads	Integer or length-2 integer vector, or NULL. Defaults to 0.
westerlund	Logical. Use the Westerlund-specific selection/variance conventions.
aic	Logical. Use AIC for automatic selection; if FALSE, use BIC.
bootstrap	Integer number of bootstrap replications; values $<= 0$ disable bootstrapping.
indiv.ecm	Logical. Store individual ECM coefficient tables.
lrwindow	Non-negative integer Bartlett long-run variance window.
seed	Optional integer bootstrap seed. If NULL, the current R RNG state is used without resetting it.
verbose	Logical. Print detailed output.

Details

The function permits unbalanced panels, but the usable time series within each unit must be continuous after removing missing model observations. Automatic lag/lead selection is performed unit by unit when ranges are supplied.

Bootstrap p-values use the finite-sample correction $(r + 1)/(B + 1)$, where r is the number of finite bootstrap statistics less than or equal to the observed statistic. This convention is intentionally retained in both the R and Python implementations.

Value

An object of class westerlund_test, a list containing:

- test_stats: raw Gt, Ga, Pt, and Pa.
- z_scores: standardized Z-scores.
- p_values: asymptotic left-tail p-values.
- boot_pvals: finite-sample-corrected bootstrap p-values when requested.
- bootstrap_distributions: bootstrap draws of the four raw statistics.
- unit_data: unit-level alpha estimates, standard errors, beta vectors, selected lags/leads, normalization terms, long-run variance components, and observation counts. Cross-language aliases are included for compatibility.

- `indiv_data`: detailed per-unit intermediate results.
- `mean_group`: mean-group alpha/beta estimates and their standard errors.
- `settings`: model and selection settings, including the selection method, bandwidth, bootstrap count, seed, and average panel length.
- `metadata`: compact metadata aligned with the Python implementation.
- `mg_results`: mean-group reporting tables.
- `mg_tables`: alias of the mean-group reporting tables for cross-language consistency.
- `indiv_reg`: stored individual ECM tables when `indiv.ecm=TRUE`.

References

Westerlund, J. (2007). Testing for error correction in panel data. *Oxford Bulletin of Economics and Statistics*, 69(6), 709–748.

See Also

`WesterlundPlain`, `WesterlundBootstrap`, `DisplayWesterlund`, `plot.westerlund_test`

Examples

```
# 1. Generate a small synthetic panel dataset
set.seed(123)
N <- 10
T <- 30

df <- data.frame(
  id = rep(1:N, each = T),
  time = rep(1:T, N),
  y = rnorm(N * T),
  x1 = rnorm(N * T)
)

# 2. Run Asymptotic (Plain) Test
res_plain <- westerlund_test(
  data = df,
  yvar = "y",
  xvars = "x1",
  idvar = "id",
  timevar = "time",
  constant = TRUE,
  lags = 1,
  leads = 0,
  verbose = FALSE
)

print(res_plain$test_stats)

# 3. Run Bootstrap Test with automatic lag selection
# Note: bootstrap replications are kept low for example purposes
res_boot <- westerlund_test(
```

```
  data = df,
  yvar = "y",
  xvars = "x1",
  idvar = "id",
  timevar = "time",
  constant = TRUE,
  lags = c(0, 1),
  bootstrap = 50,
  verbose = TRUE
)

summary(res_boot)
```

`westerlund_test_mg` *Print Mean-Group ECM Output and Long-Run Relationship Tables*

Description

Internal reporting helper that prints two Stata-style coefficient tables: (1) the mean-group error-correction model (short-run ECM output) and (2) the estimated long-run relationship with short-run adjustment. This function delegates all table computation and formatting to `westerlund_test_reg`.

Usage

```
westerlund_test_mg(b, V, b2, V2, auto, verbose = FALSE)
```

Arguments

`b`	A numeric vector of coefficients for the long-run relationship and short-run adjustment terms (as reported in the second table).
`V`	A numeric variance-covariance matrix corresponding to `b`.
`b2`	A numeric vector of coefficients for the mean-group error-correction model (as reported in the first table).
`V2`	A numeric variance-covariance matrix corresponding to `b2`.
`auto`	Logical/integer. If non-zero, prints a note indicating that short-run coefficients (apart from the error-correction term) are omitted because selected lag/lead lengths may differ across units.
`verbose`	Logical. If `TRUE`, prints additional output.

Details

What it prints. The function prints two sections:

1. **Mean-group error-correction model:** This corresponds to reporting the mean-group ECM coefficients (often denoted `b2` with variance `V2`). If `auto` is non-zero, a note is printed to remind that short-run coefficients may be omitted due to unit-specific lag/lead selection.

2. **Estimated long-run relationship and short run adjustment:** This corresponds to reporting the long-run relationship and the adjustment dynamics (b with variance V).

In each section, westerlund_test_reg is called to compute standard errors, z-statistics, two-sided p-values, and 95% confidence intervals, and to print the formatted coefficient table using stats::printCoefmat().

Intended use. This is an internal display helper used in Stata-aligned reporting flows. It is not required for computing the Westerlund test statistics themselves.

Value

A named list containing two matrices:

mg_model	A numeric matrix containing the Mean-group ECM results (coefficients, SE, z, p-values, and CI).
long_run	A numeric matrix containing the long-run relationship and adjustment results.

Vignette

This section shows the expected calling pattern for westerlund_test_mg() and explains the role of the auto flag.

Two-table reporting: In mean-group implementations, it is common to show:

- a short-run ECM table (possibly with omitted coefficients when lag/lead lengths differ by unit),
- a long-run relationship table with the short-run adjustment term.

westerlund_test_mg() prints these tables sequentially.

Examples:

```
## Long-run relationship table inputs
b <- c(alpha = -0.20, beta = 1.05)
V <- diag(c(0.03^2, 0.10^2))
rownames(V) <- colnames(V) <- names(b)

## Mean-group ECM table inputs
b2 <- c(ec = -0.18)
V2 <- matrix(0.04^2, nrow = 1, ncol = 1)
rownames(V2) <- colnames(V2) <- names(b2)

## Print both sections (auto = TRUE prints the omission note)
westerlund_test_mg(b, V, b2, V2, auto = TRUE, verbose = FALSE)
```

See Also

westerlund_test_reg, westerlund_test

`westerlund_test_reg` *Formatted Coefficient Table for Westerlund Test Reporting*

Description

Builds a Stata-style coefficient table using either Z inference or t inference. Mean-group reporting uses Z inference, while individual OLS ECM output uses t inference with residual degrees of freedom.

Usage

```
westerlund_test_reg(
  b,
  V,
  verbose = FALSE,
  distribution = c("z", "t"),
  df = NULL
)
```

Arguments

`b`	Named numeric coefficient vector.
`V`	Variance-covariance matrix corresponding to `b`.
`verbose`	Logical. Print the coefficient table.
`distribution`	Either `"z"` or `"t"`.
`df`	Positive residual degrees of freedom, required for `distribution="t"`.

Details

Standard errors are the square roots of `diag(V)`. The reported statistic is coefficient divided by standard error. Two-sided p-values and 95 percent confidence intervals are computed from either the standard normal or Student t distribution according to `distribution`.

Value

A numeric matrix containing coefficients, standard errors, the selected test statistic, two-sided p-values, and 95 percent confidence intervals.

See Also

`westerlund_test_mg`, `westerlund_test`

Examples

```
b <- c(alpha = -0.2, beta = 1.05)
V <- diag(c(0.03^2, 0.10^2))
names(b) <- rownames(V) <- colnames(V) <- c("alpha", "beta")
westerlund_test_reg(b, V)
westerlund_test_reg(b, V, distribution = "t", df = 25)
```

WesterlundBootstrap *Bootstrap Routine for Westerlund ECM Panel Cointegration Tests*

Description

Generates the null bootstrap distribution of the Westerlund ECM panel cointegration statistics using Stata-aligned restricted regressions, time-cluster resampling, and bootstrap reconstruction of the panel.

Usage

```
WesterlundBootstrap(
  data,
  touse,
  idvar,
  timevar,
  yvar,
  xvars,
  constant = FALSE,
  trend = FALSE,
  lags = 1,
  leads = NULL,
  westerlund = FALSE,
  aic = TRUE,
  bootstrap = 100,
  indiv.ecm = FALSE,
  lrwindow = 2,
  seed = NULL,
  verbose = FALSE
)
```

Arguments

`data`	Original panel `data.frame`.
`touse`	Logical vector identifying usable rows.
`idvar`	Panel identifier column name.
`timevar`	Time column name.
`yvar`	Dependent variable name.
`xvars`	Regressor names.

constant	Include a constant in the restricted bootstrap regression.
trend	Include a trend.
lags	Fixed lag or length-2 lag range. Defaults to 1.
leads	Fixed lead or length-2 lead range; NULL means 0.
westerlund	Use the Westerlund-specific restricted-model information criterion.
aic	Use AIC rather than BIC outside Westerlund-specific mode.
bootstrap	Number of bootstrap replications.
indiv.ecm	Retained for interface compatibility.
lrwindow	Bartlett long-run variance window passed to re-estimation.
seed	Optional integer seed. If NULL, the current R RNG state is used.
verbose	Print progress information.

Details

For each unit the routine estimates the restricted short-run model under the null of no error correction. Residuals are demeaned within unit and Δx is centered over all usable observations. The bootstrap then reproduces the Stata-style expandcl 2, actual-time cluster(t) sampling, and newttt/tussent/newtt temporal alignment while retaining unit-specific panel lengths.

When ranges are supplied, lag/lead selection is performed in the restricted model. In westerlund=TRUE mode the criterion is

$$\log(RSS/(T_i - p - q - 1)) + 2(p + q + c + r + 1)/(T_i - p_{max} - q_{max}).$$

For literal compatibility with the 2010 xtwest source, automatic lead reconstruction uses the final processed unit's selected currlead as the global lead-loop upper bound.

The routine sets the RNG kind to Mersenne-Twister. Supplying seed restarts that generator reproducibly; with seed=NULL, the existing RNG state is used.

Value

A list with BOOTSTATS, a bootstrap by 4 numeric matrix containing Gt, Ga, Pt, and Pa draws.

References

Westerlund, J. (2007). Testing for error correction in panel data. *Oxford Bulletin of Economics and Statistics*, 69(6), 709–748.

See Also

westerlund_test, WesterlundPlain

`WesterlundPlain` *Compute Raw Westerlund ECM Panel Cointegration Statistics (Plain Routine)*

Description

Internal plain (non-bootstrap) routine for computing the four Westerlund (2007) ECM-based panel cointegration test statistics G_t, G_a, P_t, and P_a. The function estimates unit-specific ECM regressions to form the mean-group statistics and then constructs pooled (panel) statistics using cross-unit aggregation and partialling-out steps. Time indexing is handled strictly via gap-aware lag/difference helpers.

Usage

```
WesterlundPlain(
  data,
  touse,
  idvar,
  timevar,
  yvar,
  xvars,
  constant = FALSE,
  trend = FALSE,
  lags,
  leads = NULL,
  lrwindow = 2,
  westerlund = FALSE,
  aic = TRUE,
  bootno = FALSE,
  indiv.ecm = FALSE,
  verbose = FALSE
)
```

Arguments

`data`	A `data.frame` containing panel data.
`touse`	Logical vector of length `nrow(data)` indicating rows eligible for estimation. Rows are further filtered to remove missing `yvar` and `xvars`.
`idvar`	String. Column identifying cross-sectional units.
`timevar`	String. Column identifying time.
`yvar`	String. Name of the dependent variable (levels).
`xvars`	Character vector. Names of regressors in the long-run relationship (levels).
`constant`	Logical. If `TRUE`, includes a constant term in the ECM design matrix.
`trend`	Logical. If `TRUE`, includes a linear time trend in the ECM design matrix.

`lags`	Integer or length-2 integer vector. Fixed lag order or range `c(min,max)` for short-run dynamics. If a range is supplied, the routine performs an information-criterion search over candidate lag/lead combinations.
`leads`	Integer or length-2 integer vector, or `NULL`. Fixed lead order or range `c(min,max)`. If `NULL`, defaults to 0.
`lrwindow`	Integer. Bartlett kernel window (maximum lag) used in long-run variance calculations via `calc_lrvar_bartlett`.
`westerlund`	Logical. If `TRUE`, uses a Westerlund-specific information criterion and trimming logic for variance estimation.
`aic`	Logical. If `TRUE`, uses AIC for lag/lead selection when ranges. If `FALSE`, uses BIC.
`bootno`	Logical. If `TRUE`, prints a short header and progress dots (intended for higher-level routines).
`indiv.ecm`	Logical. If `TRUE`, gets output of individual ECM regressions.
`verbose`	Logical. If `TRUE`, prints additional output.

Details

Purpose and status. `WesterlundPlain()` is typically called internally by `westerlund_test`. It returns the four *raw* test statistics and lag/lead diagnostics needed for printing and standardization.

Workflow overview. The routine proceeds in two main stages:

1. **Unit-specific ECM regressions (Loop 1):** For each cross-sectional unit, it constructs an ECM with Δy_t as the dependent variable and includes deterministic terms (optional), y_{t-1}, x_{t-1}, lagged Δy_t, and leads/lags of Δx_t. Lags and leads are computed using strict time-indexed helpers (`get_lag`, `get_diff`), which respect gaps in the time index. If `lags` and/or `leads` are provided as ranges, an information-criterion search selects the lag/lead orders for each unit. The routine stores the unit-level error-correction estimate $\hat{\alpha}_i$ and its standard error.
2. **Pooled (panel) aggregation (Loop 2):** Using the mean of selected lag/lead orders across units, the routine constructs pooled quantities needed for P_t and P_a via partialling-out regressions and cross-unit aggregation of residual products.

Long-run variance calculations. Long-run variances are computed using `calc_lrvar_bartlett` with `maxlag = lrwindow`. In `westerlund=TRUE` mode, the routine applies Stata-like trimming at the start/end of the differenced series based on selected lags/leads prior to long-run variance estimation.

Returned statistics. Let $\hat{\alpha}_i$ denote the unit-specific error-correction coefficient on y_{t-1} (as constructed in the ECM), with standard error $\widehat{\mathrm{se}}(\hat{\alpha}_i)$. The routine computes:

- G_t: the mean of the individual t-ratios $\hat{\alpha}_i / \widehat{\mathrm{se}}(\hat{\alpha}_i)$,
- G_a: a scaled mean-group statistic using a unit-specific normalization factor derived from long-run variances,
- P_t: a pooled t-type statistic based on a pooled $\hat{\alpha}$ and its pooled standard error,
- P_a: a pooled scaled statistic using an average effective time dimension.

Value

A nested list containing:

- `stats`: A list of the four raw Westerlund test statistics:
  - `Gt`: Mean-group tau statistic.
  - `Ga`: Mean-group alpha statistic.
  - `Pt`: Pooled tau statistic.
  - `Pa`: Pooled alpha statistic.
- `indiv_data`: A named list where each element corresponds to a cross-sectional unit (ID), containing:
  - `ai`: The estimated speed of adjustment (alpha).
  - `seai`: The standard error of alpha (adjusted for degrees of freedom).
  - `betai`: Vector of long-run coefficients ($\beta = -\lambda/\alpha$).
  - `blag, blead`: The lags and leads selected for that specific unit.
  - `ti`: Raw observation count for the unit.
  - `tnorm`: Degrees of freedom used for normalization.
  - `reg_coef`: If `indiv.ecm = TRUE`, the full coefficient matrix from `westerlund_test_reg`.
- `results_df`: A summary `data.frame` containing all unit-level results in vectorized format.
- `settings`: A list of routine metadata:
  - `constant`: Logical indicating if a constant was included.
  - `trend`: Logical indicating if a trend was included.
  - `meanlag`, `meanlead`: Integer averages of the selected unit lags/leads.
  - `realmeanlag`, `realmeanlead`: Numeric averages of the selected unit lags/leads.
  - `auto`: Logical; `TRUE` if automatic selection (ranges) was used.

Internal Logic

Two-stage structure: Loop 1 (mean-group) estimates unit-specific ECMs. Each unit produces an estimated error-correction coefficient on y_{t-1} and an associated standard error. These are aggregated into G_t and G_a.
Loop 2 (pooled) fixes a common short-run structure based on the average selected lag/lead orders and constructs pooled residual products to obtain P_t and P_a.

Strict time indexing and gaps: All lags and differences are computed using strict time-based helpers (`get_lag`, `get_diff`). This ensures that gaps in the time index propagate as missing values rather than shifting across gaps.

References

Westerlund, J. (2007). Testing for error correction in panel data. *Oxford Bulletin of Economics and Statistics*, 69(6), 709–748.

See Also

`westerlund_test`, `WesterlundBootstrap`, `get_lag`, `get_diff`, `calc_lrvar_bartlett`

Examples

```
set.seed(123)
N <- 5
T <- 20
df <- data.frame(
  id = rep(1:N, each = T),
  t  = rep(1:T, N),
  y  = rnorm(N * T),
  x1 = rnorm(N * T),
  x2 = rnorm(N * T)
)

touse <- rep(TRUE, nrow(df))

plain_res <- WesterlundPlain(
  data       = df,
  touse      = touse,
  idvar      = "id",
  timevar    = "t",
  yvar       = "y",
  xvars      = c("x1","x2"),
  lags       = 1,
  leads      = 0
)

# Accessing results from the nested structure:
stats <- plain_res$stats
print(c(Gt = stats$Gt, Ga = stats$Ga, Pt = stats$Pt, Pa = stats$Pa))

# Checking unit-specific coefficients for ID '101'
unit_101 <- plain_res$indiv_data[["101"]]
print(unit_101$ai)
```

Index

calc_lrvar_bartlett, 2, *11*, *13*, *27, 28*

DisplayWesterlund, 4, *20*

get_diff, 8, *17*, *27, 28*
get_lag, *8, 9*, 10, *16, 17*, *27, 28*
get_ts_val, *9*, *11*, 12

plot.westerlund_test, 13, *20*
print.summary.westerlund_test, 15
print.westerlund_test, 15

shiftNA, 16
summary.westerlund_test, *15, 16*, 17

westerlund_test, *3*, *7–9*, *11*, *13–16*, 18, *22, 23*, *25*, *27, 28*
westerlund_test_mg, 21, *23*
westerlund_test_reg, *21, 22*, 23
WesterlundBootstrap, *7, 8*, *17*, *20*, 24, *28*
WesterlundPlain, *7, 8*, *20*, *25*, 26