

Package ‘ardlverse’

June 29, 2026

Type Package

Title Comprehensive ARDL: Panel, Bootstrap and Fourier Methods

Version 2.0.0

Date 2026-06-29

Description A unified framework for Autoregressive Distributed Lag (ARDL) modeling and cointegration analysis. Implements Panel ARDL with Pooled Mean Group (PMG), Mean Group (MG), and Dynamic Fixed Effects (DFE) estimators following Pesaran, Shin & Smith (1999) <[doi:10.1002/jae.616](https://doi.org/10.1002/jae.616)>. Provides bootstrap-based bounds testing per Pesaran, Shin & Smith (2001) <[doi:10.1002/jae.616](https://doi.org/10.1002/jae.616)>. Includes Quantile Nonlinear ARDL (QNARDL) combining distributional and asymmetric effects based on Shin, Yu & Greenwood-Nimmo (2014) <[doi:10.1007/978-1-4899-8008-3_9](https://doi.org/10.1007/978-1-4899-8008-3_9)>, and Fourier ARDL for modeling smooth structural breaks following Enders & Lee (2012) <[doi:10.1016/j.econlet.2012.05.019](https://doi.org/10.1016/j.econlet.2012.05.019)>. Features include Augmented ARDL (AARDL) with deferred t and F tests, Multiple-Threshold NARDL for complex asymmetries, Rolling/Recursive ARDL for time-varying relationships, and Panel NARDL for nonlinear panel cointegration. All methods include comprehensive diagnostics, publication-ready outputs, and visualization tools.

License GPL-3

URL <https://github.com/muhammedalkhalaf/ardlverse>

BugReports <https://github.com/muhammedalkhalaf/ardlverse/issues>

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0)

Imports stats, MASS, quantreg, lmtest, ggplot2, gridExtra, graphics, grDevices

Suggests testthat (>= 3.0.0), knitr, rmarkdown, covr

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Muhammad Abdullah Alkhalaf [aut, cre] (ORCID:
<https://orcid.org/0009-0002-2677-9246>),
Merwan Roudane [ctb] (Original Stata implementations),
Yeleazar Levchenko [ctb] (Bug fixes and PMG/MG/DFE corrections for
panel_ardl in v2.0.0 (KSE))
Maintainer Muhammad Abdullah Alkhalaf <muhammedalkhalaf@gmail.com>
Repository CRAN
Date/Publication 2026-06-29 08:30:07 UTC

Contents

ardlverse-package	2
aardl	4
ardl_diagnostics	6
asymmetry_test	8
boot_ardl	9
dynamic_multipliers	11
fourier_ardl	12
fourier_bounds_test	14
generate_oil_data	15
generate_panel_data	16
generate_ts_data	17
hausman_test	18
macro_data	19
macro_panel	19
mtnardl	20
oil_data	22
panel_ardl	23
pnardl	25
print.panel_ardl	27
pss_critical_values	28
qnardl	29
rardl	31
summary.panel_ardl	33
Index	34

ardlverse-package	<i>Comprehensive ARDL Modeling Framework</i>
-------------------	--

Description

A unified framework for Autoregressive Distributed Lag (ARDL) modeling and cointegration analysis. Implements Panel ARDL with Pooled Mean Group (PMG), Mean Group (MG), and Dynamic Fixed Effects (DFE) estimators. Provides bootstrap-based bounds testing for small samples, Quantile Nonlinear ARDL (QNARDL) combining distributional and asymmetric effects, and Fourier ARDL for modeling smooth structural breaks.

Details

The main functions are:

- `panel_ardl`: Panel ARDL with PMG, MG, DFE estimators
- `boot_ardl`: Bootstrap ARDL bounds test
- `qnardl`: Quantile Nonlinear ARDL
- `fourier_ardl`: Fourier ARDL for structural breaks
- `ardl_diagnostics`: Comprehensive model diagnostics

Supporting functions:

- `hausman_test`: PMG vs MG comparison
- `asymmetry_test`: Long-run asymmetry test
- `dynamic_multipliers`: Cumulative multipliers
- `pss_critical_values`: PSS (2001) critical values

Data generation for examples:

- `generate_panel_data`: Panel data
- `generate_ts_data`: Time series data
- `generate_oil_data`: Oil/gasoline price data

Author(s)

Muhammad Abdullah Alkhalaf <muhammedalkhalaf@gmail.com>

Maintainer: Muhammad Abdullah Alkhalaf <muhammedalkhalaf@gmail.com>

References

- Pesaran, M. H., Shin, Y., & Smith, R. P. (1999). Pooled mean group estimation of dynamic heterogeneous panels. *Journal of the American Statistical Association*.
- Pesaran, M. H., Shin, Y., & Smith, R. J. (2001). Bounds testing approaches to the analysis of level relationships. *Journal of Applied Econometrics*.
- Shin, Y., Yu, B., & Greenwood-Nimmo, M. (2014). Modelling asymmetric cointegration and dynamic multipliers in a nonlinear ARDL framework.
- Cho, J. S., Kim, T. H., & Shin, Y. (2015). Quantile cointegration in the autoregressive distributed-lag modeling framework. *Journal of Econometrics*.
- Banerjee, P., Arcabic, V., & Lee, H. (2017). Fourier ADL cointegration test. *Economic Modelling*.

Examples

```
# Panel ARDL
data <- generate_panel_data()
model <- panel_ardl(gdp ~ inflation + investment, data = data,
                    id = "country", time = "year", estimator = "pmg")
summary(model)

# Bootstrap bounds test
ts_data <- generate_ts_data()
boot <- boot_ardl(gdp ~ investment, data = ts_data, nboot = 1000)
summary(boot)

# QNARDL
oil <- generate_oil_data()
qmodel <- qnardl(gasoline ~ oil_price, data = oil, tau = c(0.25, 0.5, 0.75))
asymmetry_test(qmodel, var = "oil_price")
```

aardl

Augmented ARDL Bounds Test (AARDL)

Description

Implements the Augmented ARDL bounds testing approach with deferred t and F tests for cointegration analysis.

Usage

```
aardl(formula, data, p = 1, q = 1, case = 3,
      type = c("linear", "nardl", "fourier", "fnardl",
               "bootstrap", "bnardl", "fbootstrap", "fbnardl"),
      nboot = 2000, fourier_k = 1, threshold = 0, seed = NULL)

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

Arguments

formula	A formula specifying the model: <code>gdp ~ investment + trade + ...</code>
data	A data frame containing the time series data
p	Integer. Number of lags for dependent variable (default: 1)
q	Integer or vector. Number of lags for independent variables (default: 1)
case	Integer from 1-5 specifying deterministic components (default: 3)
type	Character. Model type: "linear", "nardl", "fourier", "fnardl", "bootstrap", "bnardl", "fbootstrap", "fbnardl"

nboot	Number of bootstrap replications (default: 2000)
fourier_k	Integer. Number of Fourier frequencies (default: 1, max: 3)
threshold	Numeric. Threshold value for NARDL decomposition (default: 0)
seed	Random seed for reproducibility
x, object	An object of class "aardl"
...	Additional arguments passed to methods

Details

The Augmented ARDL (AARDL) approach extends the standard ARDL bounds test by implementing additional diagnostic tests proposed by Sam, McNown & Goh (2019). This addresses potential weaknesses in the PSS bounds test by adding:

- Deferred t-test (t_{dep}): Tests significance of lagged dependent variable
- Deferred F-test (F_{ind}): Tests joint significance of independent variables
- Overall F-test with all deferred conditions

The function supports 8 sub-models including standard ARDL, NARDL, Fourier ARDL, and their bootstrap variants.

Value

An object of class "aardl" containing:

F_pss	PSS F-statistic for bounds test
t_dep	Deferred t-statistic for lagged dependent variable
F_ind	Deferred F-statistic for independent variables
conclusion	Cointegration decision based on all tests
model	The estimated ARDL model
long_run	Long-run coefficients
short_run	Short-run coefficients
diagnostics	Model diagnostic tests

References

Sam, C. Y., McNown, R., & Goh, S. K. (2019). An augmented autoregressive distributed lag bounds test for cointegration. *Economic Modelling*, 80, 130-141.

McNown, R., Sam, C. Y., & Goh, S. K. (2018). Bootstrapping the autoregressive distributed lag test for cointegration. *Applied Economics*, 50(13), 1509-1521.

See Also

[boot_ardl](#), [qnardl](#), [fourier_ardl](#)

Examples

```
# Generate example data
data <- generate_ts_data(n = 200)

# Standard Augmented ARDL
result <- aardl(gdp ~ investment + trade, data = data, p = 2, q = 2, case = 3)
summary(result)

# Augmented NARDL (nonlinear)
result_nardl <- aardl(gdp ~ investment + trade, data = data, type = "nardl")
summary(result_nardl)

# Fourier Augmented ARDL with bootstrap
result_fb <- aardl(gdp ~ investment, data = data, type = "bootstrap", nboot = 1000)
```

ardl_diagnostics	<i>ARDL Model Diagnostics</i>
------------------	-------------------------------

Description

Comprehensive diagnostic tests for ARDL models.

Usage

```
ardl_diagnostics(model, lags = 4, arch_lags = 4)

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

## S3 method for class 'ardl_diagnostics'
plot(x, which = 1:4, ...)
```

Arguments

<code>model</code>	An estimated model object (panel_ardl, boot_ardl, qnardl, or fourier_ardl)
<code>lags</code>	Integer. Number of lags for serial correlation tests (default: 4)
<code>arch_lags</code>	Integer. Number of lags for ARCH test (default: 4)
<code>object, x</code>	An object of class "ardl_diagnostics"
<code>which</code>	Integer vector. Which plots to display (1:4)
<code>...</code>	Additional arguments

Details

This function performs a battery of diagnostic tests:

- **Serial Correlation:** Breusch-Godfrey LM test and Ljung-Box test
- **Heteroskedasticity:** Breusch-Pagan and ARCH tests
- **Normality:** Jarque-Bera and Shapiro-Wilk tests
- **Functional Form:** RESET test
- **Stability:** CUSUM and CUSUMSQ tests

Value

An object of class "ardl_diagnostics" containing:

serial_corr	Breusch-Godfrey test results
ljung_box	Ljung-Box test results
hetero_bp	Breusch-Pagan test results
arch	ARCH-LM test results
normality	Jarque-Bera test results
shapiro	Shapiro-Wilk test results
reset	RESET test results
cusum	CUSUM test results
cusumsq	CUSUM of squares test results
residuals	Model residuals

See Also

[fourier_ardl](#), [boot_ardl](#)

Examples

```
# Estimate a model
data <- generate_ts_data(n = 100)
model <- fourier_ardl(gdp ~ investment, data = data)

# Run diagnostics
diag <- ardl_diagnostics(model)
summary(diag)
plot(diag)
```

asymmetry_test

Test Asymmetry in QNARDL Model

Description

Perform Wald test for long-run asymmetry in QNARDL models.

Usage

```
asymmetry_test(object, var)
```

Arguments

object	A qnardl object
var	Variable name to test asymmetry for

Details

Tests the null hypothesis that positive and negative long-run effects are equal ($H_0: \theta_{+} = \theta_{-}$) against the alternative of asymmetric effects ($H_1: \theta_{+} \neq \theta_{-}$).

The test is performed separately for each quantile.

Value

A data frame with test results by quantile:

tau	Quantile
theta_pos	Positive long-run coefficient
theta_neg	Negative long-run coefficient
diff	Difference ($\theta_{+} - \theta_{-}$)
wald_stat	Wald test statistic
p_value	P-value

See Also

[qnardl](#), [dynamic_multipliers](#)

Examples

```
data <- generate_oil_data(n = 200)

model <- qnardl(
  gasoline ~ oil_price,
  data = data,
  tau = c(0.25, 0.5, 0.75)
)
```



```
asymmetry_test(model, var = "oil_price")
```

boot_ardl

Bootstrap ARDL Bounds Test

Description

Perform bounds test for cointegration with bootstrap critical values.

Usage

```
boot_ardl(formula, data, p = 1, q = 1, case = 3,
           nboot = 2000, seed = NULL,
           parallel = FALSE, ncores = 2)
```

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

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

```
## S3 method for class 'boot_ardl'
plot(x, which = "both", ...)
```

Arguments

formula	A formula specifying the model: $\text{gdp} \sim \text{investment} + \text{trade} + \dots$
data	A data frame containing the time series data
p	Integer. Number of lags for dependent variable (default: 1)
q	Integer or vector. Number of lags for independent variables (default: 1)
case	Integer from 1-5 specifying deterministic components (default: 3)
nboot	Number of bootstrap replications (default: 2000)
seed	Random seed for reproducibility (default: NULL)
parallel	Logical. Use parallel processing (default: FALSE)
ncores	Number of cores for parallel processing (default: 2)
object, x	An object of class "boot_ardl"
which	Character. "F", "t", or "both" for plotting
...	Additional arguments

Details

This function implements bootstrap-based inference for the ARDL bounds test, which is particularly useful for small samples where asymptotic critical values may be unreliable.

Case specifications:

- 1: No intercept, no trend
- 2: Restricted intercept, no trend
- 3: Unrestricted intercept, no trend (default)
- 4: Unrestricted intercept, restricted trend
- 5: Unrestricted intercept, unrestricted trend

Value

An object of class "boot_ardl" containing:

F_stat	F-statistic for bounds test
t_stat	t-statistic for EC coefficient
boot_F	Bootstrap distribution of F-statistics
boot_t	Bootstrap distribution of t-statistics
cv_F	Critical values for F-test (90%, 95%, 99%)
cv_t	Critical values for t-test
p_value_F	Bootstrap p-value for F-test
p_value_t	Bootstrap p-value for t-test
conclusion	Test conclusion

References

McNown, R., Sam, C. Y., & Goh, S. K. (2018). Bootstrapping the autoregressive distributed lag test for cointegration. *Applied Economics*, 50(13), 1509-1521.

Pesaran, M. H., Shin, Y., & Smith, R. J. (2001). Bounds testing approaches to the analysis of level relationships. *Journal of Applied Econometrics*, 16(3), 289-326.

See Also

[pss_critical_values](#), [generate_ts_data](#)

Examples

```
# Generate example data
data <- generate_ts_data(n = 100)

# Bootstrap bounds test
boot_test <- boot_ardl(
  gdp ~ inflation + investment,
  data = data,
  p = 2, q = 2,
```

```
    nboot = 1000
  )
  summary(boot_test)
  plot(boot_test)
```

dynamic_multipliers	<i>Dynamic Multipliers for QNARDL</i>
---------------------	---------------------------------------

Description

Compute and plot cumulative dynamic multipliers for QNARDL models.

Usage

```
dynamic_multipliers(object, var, tau = 0.5, horizon = 20)
```

Arguments

object	A qnardl object
var	Variable name
tau	Quantile to compute multipliers for (default: 0.5)
horizon	Number of periods (default: 20)

Details

Computes the cumulative dynamic multipliers showing how the effect of a unit change in x accumulates over time towards the long-run equilibrium.

Separate multipliers are computed for positive and negative changes, allowing visualization of asymmetric adjustment paths.

Value

A data frame with:

horizon	Time horizon (0 to horizon)
multiplier	Cumulative multiplier value
type	Either "Positive" or "Negative"

See Also

[qnardl](#), [asymmetry_test](#)

Examples

```
data <- generate_oil_data(n = 200)

model <- qnardl(
  gasoline ~ oil_price,
  data = data,
  tau = c(0.25, 0.5, 0.75)
)

# Plot dynamic multipliers at median
dynamic_multipliers(model, var = "oil_price", tau = 0.5, horizon = 30)
```

fourier_ardl	<i>Fourier ARDL Model</i>
--------------	---------------------------

Description

Estimate ARDL models with Fourier terms for smooth structural breaks.

Usage

```
fourier_ardl(formula, data, p = 1, q = 1, k = 1, case = 3,
             selection = c("fixed", "aic", "bic"))

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

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

## S3 method for class 'fourier_ardl'
plot(x, which = "both", ...)
```

Arguments

formula	A formula specifying the model: <code>gdp ~ investment + trade + ...</code>
data	A data frame containing the time series data
p	Integer. Number of lags for dependent variable (default: 1)
q	Integer or vector. Number of lags for independent variables (default: 1)
k	Integer. Number of Fourier frequencies (default: 1, max: 3)
case	Integer from 1-5 specifying deterministic components (default: 3)
selection	Character. Method for selecting optimal k: "aic", "bic", or "fixed"
object, x	An object of class "fourier_ardl"
which	Character. "fourier", "fit", or "both" for plotting
...	Additional arguments

Details

This function implements the Fourier ARDL approach that captures smooth structural changes using trigonometric functions. The Fourier terms approximate unknown structural breaks without requiring prior specification of break dates.

The Fourier terms are defined as:

$$f_t = \sum_{k=1}^K [a_k \sin(2\pi kt/T) + b_k \cos(2\pi kt/T)]$$

where K is the number of frequencies and T is the sample size.

Value

An object of class "fourier_ardl" containing:

coefficients	Estimated coefficients
long_run	Long-run coefficients
fourier_coefs	Fourier term coefficients
ec_coef	Error correction coefficient
bounds_test	F and t statistics for bounds test
k	Number of Fourier frequencies used
aic	AIC values for k = 1, 2, 3
bic	BIC values for k = 1, 2, 3
structural_breaks	Detected structural break periods

References

- Banerjee, P., Arcabic, V., & Lee, H. (2017). Fourier ADL cointegration test to approximate smooth breaks with new evidence from crude oil market. *Economic Modelling*, 67, 114-124.
- Enders, W., & Lee, J. (2012). A unit root test using a Fourier series to approximate smooth breaks. *Oxford Bulletin of Economics and Statistics*, 74(4), 574-599.

See Also

[fourier_bounds_test](#), [generate_ts_data](#)

Examples

```
# Generate example data
data <- generate_ts_data(n = 100)

# Estimate Fourier ARDL with automatic selection
model <- fourier_ardl(
  gdp ~ investment + trade,
  data = data,
  selection = "aic"
```

```
)
summary(model)
plot(model)
```

fourier_bounds_test *Fourier ARDL Bounds Test*

Description

Perform bounds test with Fourier-adjusted critical values.

Usage

```
fourier_bounds_test(object)
```

Arguments

object A fourier_ardl object

Details

Performs the bounds test for cointegration using critical values adjusted for the presence of Fourier terms. The critical values account for the additional parameters estimated.

Value

A list containing:

F_stat	F-statistic
t_stat	t-statistic
cv_F	Critical values for F-test by significance level

See Also

[fourier_ardl](#), [pss_critical_values](#)

Examples

```
data <- generate_ts_data(n = 100)

model <- fourier_ardl(
  gdp ~ investment,
  data = data,
  k = 1
)

fourier_bounds_test(model)
```

generate_oil_data	<i>Generate Oil Price Data</i>
-------------------	--------------------------------

Description

Generate simulated oil and gasoline price data with asymmetric effects.

Usage

```
generate_oil_data(n = 200, seed = 789)
```

Arguments

n	Number of observations (default: 200)
seed	Random seed for reproducibility (default: 789)

Details

Generates a simulated dataset exhibiting the "rockets and feathers" phenomenon where gasoline prices respond more quickly to oil price increases than decreases. Suitable for demonstrating QNARDL methods and asymmetry testing.

Value

A data frame with columns:

week	Week number (1 to n)
gasoline	Retail gasoline price (cents/gallon)
oil_price	Crude oil price (\$/barrel)
exchange_rate	USD exchange rate index

See Also

[qnardl](#), [asymmetry_test](#)

Examples

```
# Generate oil price data
oil <- generate_oil_data(n = 200)
head(oil)

# Plot prices
plot(oil$week, oil$oil_price, type = "l", col = "blue",
     main = "Oil and Gasoline Prices")
lines(oil$week, oil$gasoline/4, col = "red") # Scaled for comparison
```

generate_panel_data	<i>Generate Example Panel Data</i>
---------------------	------------------------------------

Description

Generate simulated panel data for examples and testing.

Usage

```
generate_panel_data(n_groups = 10, n_time = 50, seed = 123)
```

Arguments

n_groups	Number of groups/countries (default: 10)
n_time	Number of time periods (default: 50)
seed	Random seed for reproducibility (default: 123)

Details

Generates a simulated panel dataset with cointegrated variables suitable for demonstrating panel ARDL methods. The data generating process includes country-specific fixed effects and error correction dynamics.

Value

A data frame with columns:

country	Country identifier (1 to n_groups)
year	Year (starting from 1970)
gdp	Log GDP per capita (I(1))
inflation	Inflation rate (I(0))
investment	Investment as % of GDP (I(1))
trade	Trade openness (I(1))

See Also

```
panel\_ardl, generate\_ts\_data
```

Examples

```
# Generate panel data for 5 countries, 30 years
panel <- generate_panel_data(n_groups = 5, n_time = 30)
head(panel)

# Check dimensions
table(panel$country)
```

generate_ts_data	<i>Generate Example Time Series Data</i>
------------------	--

Description

Generate simulated time series data for examples and testing.

Usage

```
generate_ts_data(n = 100, seed = 456)
```

Arguments

n	Number of observations (default: 100)
seed	Random seed for reproducibility (default: 456)

Details

Generates a simulated time series dataset with cointegrated variables and a smooth structural break (modeled via Fourier component). Suitable for demonstrating Fourier ARDL and bootstrap bounds testing methods.

Value

A data frame with columns:

quarter	Quarter number (1 to n)
gdp	Log real GDP (I(1) with structural break)
inflation	Inflation rate (I(0))
investment	Investment growth (I(1))
trade	Trade balance (I(1))

See Also

[fourier_ardl](#), [boot_ardl](#), [generate_panel_data](#)

Examples

```
# Generate 100 quarters of data
ts_data <- generate_ts_data(n = 100)
head(ts_data)

# Plot GDP series
plot(ts_data$quarter, ts_data$gdp, type = "l",
     main = "Simulated GDP with Structural Break")
```

hausman_test

*Hausman Test for Panel ARDL Estimators***Description**

Compare two panel_ardl estimators (typically MG vs PMG) to test for long-run slope homogeneity.

Usage

```
hausman_test(inefficient, efficient, sigmamore = TRUE)
```

Arguments

inefficient	A panel_ardl object - the always-consistent estimator (typically MG).
efficient	A panel_ardl object - the under-H0 efficient estimator (typically PMG).
sigmamore	Logical. If TRUE (default, matches Stata's sigmamore option), rescale the inefficient variance matrix to address non-positive-definite difference matrices.

Details

Under the null of long-run homogeneity, the PMG (efficient) estimator is consistent and efficient; the MG (inefficient) estimator is consistent but less efficient. A significant test statistic rejects PMG in favour of MG.

Argument order in v2.0.0 follows Stata's convention: pass the **inefficient** estimator first (e.g. MG), then the **efficient** one (e.g. PMG). This is a breaking change from earlier versions.

Value

An (invisible) list with statistic, df, p.value, and theta_diff.

References

Hausman, J. A. (1978). Specification tests in econometrics. *Econometrica*, 46(6), 1251-1271.

Examples

```
## Not run:
fit_mg <- panel_ardl(y ~ x, data = d, id = "i", time = "t", estimator = "mg")
fit_pmg <- panel_ardl(y ~ x, data = d, id = "i", time = "t", estimator = "pmg")
hausman_test(fit_mg, fit_pmg)

## End(Not run)
```

macro_data	<i>Time Series Macroeconomic Data</i>
------------	---------------------------------------

Description

A simulated time series dataset of macroeconomic variables for a single country over 100 quarters.

Usage

```
macro_data
```

Format

A data frame with 100 rows and 5 variables: quarter, gdp, inflation, investment, trade.

Source

Simulated data for demonstration purposes

Examples

```
data(macro_data)
head(macro_data)
```

macro_panel	<i>Simulated Macroeconomic Panel Data</i>
-------------	---

Description

A simulated panel dataset of macroeconomic variables for 10 countries over 50 years.

Usage

```
macro_panel
```

Format

A data frame with 500 rows and 6 variables: country, year, gdp, inflation, investment, trade.

Source

Simulated data for demonstration purposes

Examples

```
data(macro_panel)
head(macro_panel)

# Estimate Panel ARDL
model <- panel_ardl(
  gdp ~ inflation + investment,
  data = macro_panel,
  id = "country",
  time = "year",
  estimator = "pmg"
)
```

mtnardl

Multiple-Threshold Nonlinear ARDL (MT-NARDL)

Description

Extends NARDL to allow multiple threshold decomposition for capturing complex asymmetric relationships.

Usage

```
mtnardl(formula, data, thresholds = c(0), p = 1, q = 1, case = 3,
        auto_select = FALSE, n_thresholds = 2,
        bootstrap = FALSE, nboot = 2000, seed = NULL)

## S3 method for class 'mtnardl'
print(x, ...)
## S3 method for class 'mtnardl'
summary(object, ...)
## S3 method for class 'mtnardl'
plot(x, type = c("multipliers", "asymmetry"), ...)
```

Arguments

formula	A formula specifying the model: $\text{gdp} \sim \text{investment} + \text{trade} + \dots$
data	A data frame containing the time series data
thresholds	Numeric vector of threshold values (default: $c(0)$)
p	Integer. Number of lags for dependent variable (default: 1)
q	Integer or vector. Number of lags for independent variables (default: 1)
case	Integer from 1-5 specifying deterministic components (default: 3)
auto_select	Logical. Automatically select optimal thresholds (default: FALSE)
n_thresholds	Integer. Number of thresholds to select if <code>auto_select = TRUE</code>

bootstrap	Logical. Use bootstrap inference (default: FALSE)
nboot	Number of bootstrap replications (default: 2000)
seed	Random seed for reproducibility
x, object	An object of class "mtnardl"
type	Character. Plot type: "multipliers" or "asymmetry"
...	Additional arguments passed to methods

Details

The Multiple-Threshold NARDL (MT-NARDL) model extends the standard NARDL framework by allowing decomposition of variables into multiple regimes based on user-specified thresholds. This captures more nuanced asymmetric effects beyond simple positive/negative decomposition.

For example, with thresholds $c(-0.02, 0, 0.02)$, a variable is decomposed into:

- Large decreases ($< -2\%$)
- Small decreases (-2% to 0)
- Small increases (0 to 2%)
- Large increases ($> 2\%$)

Value

An object of class "mtnardl" containing:

model	The estimated MT-NARDL model
bounds_test	Bounds test results
long_run	Long-run coefficients for each regime
short_run	Short-run coefficients for each regime
thresholds	Threshold values used
asymmetry_tests	Wald tests for asymmetry between regimes
multipliers	Dynamic multipliers for each regime

References

Shin, Y., Yu, B., & Greenwood-Nimmo, M. (2014). Modelling asymmetric cointegration and dynamic multipliers in a nonlinear ARDL framework. In *Festschrift in Honor of Peter Schmidt* (pp. 281-314). Springer.

See Also

[qnardl](#), [aardl](#)

Examples

```
# Generate example data
data <- generate_oil_data(n = 300)

# Standard NARDL (single threshold at 0)
result1 <- mtnardl(gasoline ~ oil_price, data = data)

# Multiple thresholds for different shock sizes
result2 <- mtnardl(
  gasoline ~ oil_price,
  data = data,
  thresholds = c(-0.05, 0, 0.05)
)
summary(result2)
plot(result2, type = "multipliers")

# Auto-select optimal thresholds
result3 <- mtnardl(
  gasoline ~ oil_price,
  data = data,
  auto_select = TRUE,
  n_thresholds = 2
)
```

oil_data

*Oil Price and Gasoline Data***Description**

Weekly data on oil prices and retail gasoline prices, suitable for demonstrating asymmetric price transmission.

Usage

```
oil_data
```

Format

A data frame with 200 rows and 4 variables: week, gasoline, oil_price, exchange_rate.

Source

Simulated data for demonstration purposes

Examples

```
data(oil_data)

# Test for asymmetric price transmission
model <- qnardl(
  gasoline ~ oil_price + exchange_rate,
  data = oil_data,
  tau = c(0.25, 0.5, 0.75)
)
asymmetry_test(model, var = "oil_price")
```

panel_ardl

*Panel ARDL Estimation (PMG / MG / DFE)***Description**

Estimate Panel ARDL models with Pooled Mean Group (PMG), Mean Group (MG), and Dynamic Fixed Effects (DFE) estimators following Pesaran, Shin & Smith (1999). This implementation replicates Stata's xtpmg command (Blackburne & Frank 2007).

Usage

```
panel_ardl(
  formula,
  data,
  id,
  time,
  p = 1,
  q = 1,
  estimator = c("pmg", "mg", "dfe"),
  ec = TRUE,
  trend = FALSE,
  start_time = NULL,
  cluster = FALSE,
  maxiter = 100,
  tol = 1e-06
)
```

Arguments

formula	A formula $y \sim x_1 + x_2 + \dots$
data	A data frame containing panel data.
id	Character. Name of the panel/group identifier column.
time	Character. Name of the time variable column.

p	Integer. Lags for dependent variable (default 1).
q	Integer or numeric vector. Lags for independent variables (default 1).
estimator	Character. One of "pmg", "mg", "dfe".
ec	Logical. If TRUE, display error-correction form (default TRUE).
trend	Logical. Include a deterministic time trend (default FALSE).
start_time	Optional. Restrict estimation to observations at or after this time.
cluster	Logical. If TRUE, use cluster-robust standard errors where applicable (default FALSE).
maxiter	Integer. Maximum iterations for PMG optimization (default 100).
tol	Numeric. Convergence tolerance (default 1e-6).

Details

The model is specified as:

$$\Delta y_{it} = \phi_i(y_{i,t-1} - \theta'_i x_{it}) + \sum_{j=1}^{p-1} \lambda_{ij} \Delta y_{i,t-j} + \sum_{j=0}^{q-1} \delta'_{ij} \Delta x_{i,t-j} + \mu_i + \varepsilon_{it}$$

Three estimators are available:

- **PMG**: Constrains long-run coefficients across groups while allowing short-run coefficients to vary. Efficient under long-run homogeneity.
- **MG**: Estimates separate regressions per group and averages. Consistent but less efficient than PMG.
- **DFE**: Traditional dynamic fixed effects with all coefficients pooled. Consistent only under slope homogeneity.

Value

An object of class "panel_ardl" with components:

- long_run: long-run coefficients (theta).
- short_run: short-run coefficient summaries.
- ec_coef: error-correction coefficient (phi).
- vcov: variance-covariance matrix of long-run coefficients.
- sigma: residual standard error.
- estimator: the estimator used.

References

- Pesaran, M. H., Shin, Y., & Smith, R. P. (1999). Pooled mean group estimation of dynamic heterogeneous panels. *Journal of the American Statistical Association*, 94(446), 621-634.
- Blackburne, E. F., & Frank, M. W. (2007). Estimation of nonstationary heterogeneous panels. *Stata Journal*, 7(2), 197-208.

Examples

```
## Not run:
fit <- panel_ardl(y ~ x1 + x2, data = my_panel,
                 id = "country", time = "year",
                 p = 1, q = 1, estimator = "pmg")

summary(fit)

## End(Not run)
```

pnardl	<i>Panel Nonlinear ARDL (Panel NARDL)</i>
--------	---

Description

Panel data estimation with asymmetric/nonlinear ARDL specification using PMG, MG, or DFE estimators.

Usage

```
pnardl(formula, data, id, time, p = 1, q = 1,
       estimator = c("pmg", "mg", "dfe"),
       threshold = 0,
       effect = c("individual", "time", "twoways"),
       bootstrap = FALSE, nboot = 500)

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

Arguments

formula	A formula specifying the model: <code>gdp ~ investment + trade + ...</code>
data	A data frame in long panel format
id	Character. Name of the group/panel identifier variable
time	Character. Name of the time variable
p	Integer. Number of lags for dependent variable (default: 1)
q	Integer or vector. Number of lags for independent variables (default: 1)
estimator	Character. One of "pmg", "mg", or "dfe" (default: "pmg")
threshold	Numeric. Threshold for asymmetric decomposition (default: 0)
effect	Character. Type of effects: "individual", "time", "twoways" (default: "individual")
bootstrap	Logical. Use bootstrap for inference (default: FALSE)
nboot	Number of bootstrap replications (default: 500)
x, object	An object of class "pnardl"
...	Additional arguments passed to methods

Details

The Panel NARDL model extends the Shin et al. (2014) NARDL framework to panel data settings. It allows for asymmetric short-run and long-run effects while controlling for cross-sectional heterogeneity.

Three estimators are available:

- **PMG**: Pooled Mean Group - constrains long-run coefficients to be homogeneous while allowing short-run heterogeneity
- **MG**: Mean Group - estimates separate models for each unit and averages coefficients
- **DFE**: Dynamic Fixed Effects - assumes all coefficients are homogeneous except fixed effects

Value

An object of class "pnardl" containing:

coefficients	Estimated coefficients
long_run_pos	Long-run coefficients for positive changes
long_run_neg	Long-run coefficients for negative changes
short_run	Short-run coefficients
ec_coef	Error correction coefficient
asymmetry_test	Wald test for long-run asymmetry
unit_results	Individual unit estimation results (for MG)
hausman	Hausman test comparing PMG vs MG

References

Shin, Y., Yu, B., & Greenwood-Nimmo, M. (2014). Modelling asymmetric cointegration and dynamic multipliers in a nonlinear ARDL framework.

Pesaran, M. H., Shin, Y., & Smith, R. P. (1999). Pooled mean group estimation of dynamic heterogeneous panels. Journal of the American Statistical Association.

See Also

[panel_ardl](#), [qnardl](#), [mtnardl](#)

Examples

```
# Generate panel data
data <- generate_panel_data(n_groups = 20, n_time = 50)

# Panel NARDL with PMG estimator
result <- pnardl(
  gdp ~ investment + trade,
  data = data,
  id = "country",
  time = "year",
  estimator = "pmg"
```

```
)
summary(result)

# Panel NARDL with MG estimator
result_mg <- pnardl(
  gdp ~ investment,
  data = data,
  id = "country",
  time = "year",
  estimator = "mg"
)

# Panel NARDL with bootstrap standard errors
result_boot <- pnardl(
  gdp ~ investment,
  data = data,
  id = "country",
  time = "year",
  bootstrap = TRUE,
  nboot = 200
)
```

`print.panel_ardl`*Print Method for panel_ardl Objects*

Description

Concise display of a fitted panel_ardl model.

Usage

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

Arguments

<code>x</code>	A panel_ardl object.
<code>...</code>	Additional arguments (currently ignored).

Value

Invisibly returns `x`.

pss_critical_values *PSS Asymptotic Critical Values*

Description

Get Pesaran, Shin & Smith (2001) asymptotic critical values for bounds test.

Usage

```
pss_critical_values(k, case = 3, level = "5%")
```

Arguments

k	Number of regressors (excluding the lagged dependent variable)
case	Deterministic specification (1-5, default: 3)
level	Significance level: "10%", "5%", or "1%"

Details

Returns the I(0) and I(1) critical value bounds from the original Pesaran, Shin & Smith (2001) tables.

Case specifications:

- 1: No intercept, no trend
- 2: Restricted intercept, no trend
- 3: Unrestricted intercept, no trend (default)
- 4: Unrestricted intercept, restricted trend
- 5: Unrestricted intercept, unrestricted trend

Value

A list with:

F_bounds	List with I0 and I1 bounds for F-test
t_bounds	List with I0 and I1 bounds for t-test
k	Number of regressors
case	Case specification
level	Significance level

References

Pesaran, M. H., Shin, Y., & Smith, R. J. (2001). Bounds testing approaches to the analysis of level relationships. *Journal of Applied Econometrics*, 16(3), 289-326.

See Also

[boot_ardl](#), [fourier_bounds_test](#)

Examples

```
# Get 5% critical values for 3 regressors, Case III
cv <- pss_critical_values(k = 3, case = 3, level = "5%")
cv$F_bounds # I(0) and I(1) bounds for F-test
cv$t_bounds # I(0) and I(1) bounds for t-test
```

qnardl	<i>Quantile Nonlinear ARDL (QNARDL)</i>
--------	---

Description

Estimate Quantile Nonlinear ARDL models combining distributional and asymmetric effects.

Usage

```
qnardl(formula, data, tau = c(0.25, 0.5, 0.75),
       p = 1, q = 1, decompose = NULL, trend = FALSE)
```

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

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

```
## S3 method for class 'qnardl'
plot(x, var = NULL, type = "long_run", ...)
```

Arguments

formula	A formula specifying the model: <code>gdp ~ investment + trade + ...</code>
data	A data frame containing the time series data
tau	Numeric vector of quantiles to estimate (default: <code>c(0.25, 0.5, 0.75)</code>)
p	Integer. Number of lags for dependent variable (default: 1)
q	Integer or vector. Number of lags for independent variables (default: 1)
decompose	Character vector. Variables to decompose into +/- components. Default is all x variables.
trend	Logical. Include time trend (default: FALSE)
object, x	An object of class "qnardl"
var	Variable to plot (default: first decomposed variable)
type	"long_run" or "asymmetry"
...	Additional arguments

Details

This function implements the QNARDL model which extends the NARDL framework of Shin, Yu & Greenwood-Nimmo (2014) to a quantile regression setting.

The model allows for:

- **Asymmetric effects:** Positive and negative changes in X can have different impacts on Y
- **Distributional heterogeneity:** Effects can vary across quantiles of the conditional distribution

The partial sum decomposition separates each regressor into positive and negative components.

Value

An object of class "qnardl" containing:

coefficients	Estimated coefficients for each quantile
long_run_pos	Positive long-run coefficients by quantile
long_run_neg	Negative long-run coefficients by quantile
short_run	Short-run coefficients by quantile
ec_coef	Error correction coefficients by quantile
asymmetry_test	Wald test for long-run asymmetry
tau	Quantiles estimated

References

Cho, J. S., Kim, T. H., & Shin, Y. (2015). Quantile cointegration in the autoregressive distributed-lag modeling framework. *Journal of Econometrics*, 188(1), 281-300.

Shin, Y., Yu, B., & Greenwood-Nimmo, M. (2014). Modelling asymmetric cointegration and dynamic multipliers in a nonlinear ARDL framework. In *Festschrift in Honor of Peter Schmidt* (pp. 281-314). Springer.

See Also

[asymmetry_test](#), [dynamic_multipliers](#), [generate_oil_data](#)

Examples

```
# Generate oil price data
data <- generate_oil_data(n = 200)

# Estimate QNARDL
model <- qnardl(
  gasoline ~ oil_price + exchange_rate,
  data = data,
  tau = c(0.1, 0.25, 0.5, 0.75, 0.9)
)
summary(model)
plot(model, var = "oil_price")
```

rardl	<i>Rolling and Recursive ARDL (R-ARDL)</i>
-------	--

Description

Time-varying ARDL bounds test using rolling or recursive windows to detect structural changes in cointegration relationships.

Usage

```
rardl(formula, data, method = c("rolling", "recursive"),
      window = 50, min_obs = 40, p = 1, q = 1, case = 3,
      parallel = FALSE, ncores = 2)
```

```
## S3 method for class 'rardl'
print(x, ...)
## S3 method for class 'rardl'
summary(object, ...)
## S3 method for class 'rardl'
plot(x, type = c("F", "ec", "lr", "all"), ...)
```

Arguments

formula	A formula specifying the model: $\text{gdp} \sim \text{investment} + \text{trade} + \dots$
data	A data frame containing the time series data
method	Character. Either "rolling" or "recursive" (default: "rolling")
window	Integer. Window size for rolling method (default: 50)
min_obs	Integer. Minimum observations for recursive method (default: 40)
p	Integer. Number of lags for dependent variable (default: 1)
q	Integer or vector. Number of lags for independent variables (default: 1)
case	Integer from 1-5 specifying deterministic components (default: 3)
parallel	Logical. Use parallel processing (default: FALSE)
ncores	Integer. Number of cores for parallel processing
x, object	An object of class "rardl"
type	Character. Plot type: "F", "ec", "lr", or "all"
...	Additional arguments passed to methods

Details

The Rolling ARDL (R-ARDL) approach implements time-varying bounds testing to detect structural changes in cointegration relationships. Two methods are available:

- **Rolling:** Fixed-size window that moves through the sample

- **Recursive:** Expanding window starting from initial observations

This is useful for detecting:

- Time-varying cointegration relationships
- Structural breaks in long-run parameters
- Changes in error correction speed

Value

An object of class "rardl" containing:

F_stats	Time series of F-statistics
t_stats	Time series of t-statistics (EC coefficient)
cointegration	Time series of cointegration decisions
ec_coefs	Time-varying error correction coefficients
lr_coefs	Time-varying long-run coefficients
dates	Index/dates for each window
breaks	Detected structural break points

References

Pesaran, M. H., Shin, Y., & Smith, R. J. (2001). Bounds testing approaches to the analysis of level relationships. *Journal of Applied Econometrics*, 16(3), 289-326.

See Also

[boot_ardl](#), [aardl](#)

Examples

```
# Generate example data
data <- generate_ts_data(n = 300)

# Rolling ARDL with 60-observation window
roll_result <- rardl(gdp ~ investment + trade, data = data, method = "rolling", window = 60)
plot(roll_result, type = "all")
summary(roll_result)

# Recursive ARDL
rec_result <- rardl(gdp ~ investment + trade, data = data, method = "recursive", min_obs = 50)
plot(rec_result, type = "F")
```

summary.panel_ardl	<i>Summary Method for panel_ardl Objects</i>
--------------------	--

Description

Detailed summary of a fitted panel_ardl model with long-run, short-run, and error-correction coefficients.

Usage

```
## S3 method for class 'panel_ardl'  
summary(object, digits = 6, ...)
```

Arguments

object	A panel_ardl object.
digits	Integer. Decimal digits to display (default 6).
...	Additional arguments (currently ignored).

Value

Invisibly returns the summary as a list.

Index

- * **asymmetry**
 - mtnardl, 20
- * **cointegration**
 - aardl, 4
- * **datagen**
 - generate_oil_data, 15
 - generate_panel_data, 16
 - generate_ts_data, 17
- * **datasets**
 - macro_data, 19
 - macro_panel, 19
 - oil_data, 22
- * **htest**
 - asymmetry_test, 8
 - fourier_bounds_test, 14
 - pss_critical_values, 28
- * **models**
 - aardl, 4
 - mtnardl, 20
 - pnardl, 25
 - rardl, 31
- * **nonlinear**
 - qnardl, 29
- * **package**
 - ardlverse-package, 2
- * **panel**
 - pnardl, 25
- * **regression**
 - ardl_diagnostics, 6
 - boot_ardl, 9
 - dynamic_multipliers, 11
 - fourier_ardl, 12
 - qnardl, 29
- * **structural breaks**
 - rardl, 31
- * **ts**
 - ardl_diagnostics, 6
 - boot_ardl, 9
 - fourier_ardl, 12
 - aardl, 4, 21, 32
 - ardl_diagnostics, 3, 6
 - ardlverse (ardlverse-package), 2
 - ardlverse-package, 2
 - asymmetry_test, 3, 8, 11, 15, 30
 - boot_ardl, 3, 5, 7, 9, 17, 29, 32
 - dynamic_multipliers, 3, 8, 11, 30
 - fourier_ardl, 3, 5, 7, 12, 14, 17
 - fourier_bounds_test, 13, 14, 29
 - generate_oil_data, 3, 15, 30
 - generate_panel_data, 3, 16, 17
 - generate_ts_data, 3, 10, 13, 16, 17
 - hausman_test, 3, 18
 - macro_data, 19
 - macro_panel, 19
 - mtnardl, 20, 26
 - oil_data, 22
 - panel_ardl, 3, 16, 23, 26
 - plot.ardl_diagnostics
 - (ardl_diagnostics), 6
 - plot.boot_ardl (boot_ardl), 9
 - plot.fourier_ardl (fourier_ardl), 12
 - plot.mtnardl (mtnardl), 20
 - plot.qnardl (qnardl), 29
 - plot.rardl (rardl), 31
 - pnardl, 25
 - print.aardl (aardl), 4
 - print.boot_ardl (boot_ardl), 9
 - print.fourier_ardl (fourier_ardl), 12
 - print.mtnardl (mtnardl), 20
 - print.panel_ardl, 27
 - print.pnardl (pnardl), 25
 - print.qnardl (qnardl), 29

`print.rardl (rardl)`, 31
`pss_critical_values`, 3, 10, 14, 28
`qnardl`, 3, 5, 8, 11, 15, 21, 26, 29
`rardl`, 31
`summary.aardl (aardl)`, 4
`summary.ardl_diagnostics`
 (`ardl_diagnostics`), 6
`summary.boot_ardl (boot_ardl)`, 9
`summary.fourier_ardl (fourier_ardl)`, 12
`summary.mtnardl (mtnardl)`, 20
`summary.panel_ardl`, 33
`summary.pnardl (pnardl)`, 25
`summary.qnardl (qnardl)`, 29
`summary.rardl (rardl)`, 31