

Package ‘mvardlurt’

March 16, 2026

Type Package

Title Multivariate ARDL Unit Root Test

Version 1.0.2

Description Implements the multivariate autoregressive distributed lag (ARDL) unit root test proposed by Sam, McNown, Goh, and Goh (2024) <[doi:10.1080/03796205.2024.2439101](https://doi.org/10.1080/03796205.2024.2439101)>. The test augments the standard ADF regression with lagged levels of a covariate to improve power when cointegration exists. Bootstrap critical values ensure correct size regardless of nuisance parameters. Provides automatic lag selection via AIC/BIC, diagnostic tests, and comprehensive inference tables following the four-case framework.

License GPL-3

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

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

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.0.0)

Imports grDevices, graphics, stats, utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

NeedsCompilation no

Author Muhammad Alkhalaf [aut, cre, cph] (ORCID:
<<https://orcid.org/0009-0002-2677-9246>>)

Maintainer Muhammad Alkhalaf <muhammedalkhalaf@gmail.com>

Repository CRAN

Date/Publication 2026-03-16 15:50:10 UTC

Contents

mvardlurt-package	2
autoplot.mvardlurt	3
mvardlurt	4
plot.mvardlurt	6
print.mvardlurt	8

Index	10
--------------	-----------

mvardlurt-package	<i>Multivariate ARDL Unit Root Test</i>
-------------------	-----------------------------------------

Description

Implements the multivariate autoregressive distributed lag (ARDL) unit root test proposed by Sam, McNown, Goh, and Goh (2024). The test augments the standard ADF regression with lagged levels of a covariate to improve power when cointegration exists. Bootstrap critical values ensure correct size regardless of nuisance parameters.

Details

The main function is `mvardlurt`, which performs the multivariate ARDL unit root test. The package provides:

- Automatic lag selection via AIC or BIC
- Bootstrap critical values for correct test size
- Three deterministic specifications (none, intercept, intercept + trend)
- Four-case decision framework for inference
- Comprehensive print and summary output
- Diagnostic plots

The test produces two statistics:

- **t-statistic:** Tests $H_0 : \pi = 0$ (unit root)
- **F-statistic:** Tests $H_0 : \delta = 0$ (no cointegration)

The four-case framework interprets results as:

- Case I: Both reject \rightarrow Cointegration
- Case II: Reject t, Accept F \rightarrow y may be I(0)
- Case III: Accept t, Reject F \rightarrow Spurious
- Case IV: Both accept \rightarrow No cointegration

Author(s)

Muhammad Alkhalaf

References

Sam, C. Y., McNown, R., Goh, S. K., & Goh, K. L. (2024). A multivariate autoregressive distributed lag unit root test. *Studies in Economics and Econometrics*, 1-17. doi:10.1080/03796205.2024.2439101

See Also

[mvardlurt](#)

Examples

```
# Generate cointegrated data
set.seed(123)
n <- 200
x <- cumsum(rnorm(n))
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run the test
result <- mvardlurt(y, x, case = 3, reps = 200)
print(result)
```

autoplot.mvardlurt *Create a Combined Diagnostic Plot for mvardlurt Objects*

Description

Creates a 2x2 panel of diagnostic plots for the multivariate ARDL unit root test including residuals vs fitted, Q-Q plot, residuals over time, and ACF.

Usage

```
## S3 method for class 'mvardlurt'
autoplot(x, ...)
```

Arguments

x An object of class "mvardlurt".
... Additional arguments passed to plotting functions.

Value

Invisibly returns x.

Author(s)

Muhammad Alkhalaf

See Also

[mvardlurt](#), [plot.mvardlurt](#)

Examples

```
set.seed(123)
n <- 100
x <- cumsum(rnorm(n))
y <- 0.5 * x + rnorm(n, sd = 0.5)

result <- mvardlurt(y, x, reps = 100)
autoplot.mvardlurt(result)
```

mvardlurt

Multivariate ARDL Unit Root Test

Description

Implements the multivariate autoregressive distributed lag (ARDL) unit root test proposed by Sam, McNown, Goh, and Goh (2024). The test augments the standard ADF regression with lagged levels of a covariate (independent variable) to improve power, especially when cointegration exists. Bootstrap critical values ensure correct size regardless of nuisance parameters.

Usage

```
mvardlurt(y, x, case = 3L, maxlag = 10L, ic = "aic",
          fixlag = NULL, reps = 1000L, level = 0.95,
          seed = 12345L, boot = TRUE)
```

Arguments

y	A numeric vector or time series. The dependent variable.
x	A numeric vector or time series. The independent variable (covariate).
case	Integer. Deterministic specification: <ul style="list-style-type: none"> • 1: No deterministic terms • 3: Intercept only (default) • 5: Intercept and linear trend
maxlag	Integer. Maximum lag order for AIC/BIC selection. Default is 10. Must be between 0 and 10.
ic	Character. Information criterion for lag selection: "aic" (default) or "bic".
fixlag	Optional numeric vector of length 2, specifying fixed lag orders $c(p, q)$ for Δy and Δx respectively. If provided, overrides automatic lag selection.
reps	Integer. Number of bootstrap replications. Default is 1000. Minimum is 100.
level	Numeric. Confidence level for inference (0 to 1). Default is 0.95.
seed	Integer. Random seed for reproducibility. Default is 12345.
boot	Logical. Whether to compute bootstrap critical values. Default is TRUE.

Details

The test estimates the following ARDL regression:

$$\Delta y_t = \pi y_{t-1} + \delta x_{t-1} + \sum_{j=1}^p \gamma_j \Delta y_{t-j} + \sum_{j=1}^q \theta_j \Delta x_{t-j} + \text{deterministics} + \varepsilon_t$$

The test produces two statistics:

- **t-statistic:** Tests $H_0 : \pi = 0$ (unit root in y)
- **F-statistic:** Tests $H_0 : \delta = 0$ (no cointegration)

Based on the four-case framework (Sam et al., 2024):

- **Case I:** Reject both \rightarrow Cointegration
- **Case II:** Reject t, Accept F \rightarrow Degenerate case 1 (y may be I(0))
- **Case III:** Accept t, Reject F \rightarrow Degenerate case 2 (spurious)
- **Case IV:** Accept both \rightarrow No cointegration

Value

An object of class "mvardlurt" containing:

tstat	t-statistic for the unit root test (on π)
fstat	F-statistic for the cointegration test (on δ)
fstat_p	Asymptotic p-value for the F-statistic
pi_coef	Coefficient estimate of π (lagged y)
pi_se	Standard error of π
delta_coef	Coefficient estimate of δ (lagged x)
delta_se	Standard error of δ
lr_mult	Long-run multiplier $-\delta/\pi$ (if $\pi \neq 0$)
opt_p	Selected lag order for Δy
opt_q	Selected lag order for Δx
case	Deterministic case used
casename	Description of the deterministic case
reps	Number of bootstrap replications
nobs	Number of observations used
aic	AIC value of the selected model
bic	BIC value of the selected model
r_squared	R-squared of the regression
t_cv	Bootstrap critical values for t-statistic (10%, 5%, 2.5%, 1%)
f_cv	Bootstrap critical values for F-statistic (10%, 5%, 2.5%, 1%)
ic_table	Matrix of IC values for all (p, q) combinations

decision	List containing test decisions and significance levels
model	The fitted lm object
y	Original dependent variable
x	Original independent variable
residuals	Residuals from the fitted model

Author(s)

Muhammad Alkhalaf

References

Sam, C. Y., McNown, R., Goh, S. K., & Goh, K. L. (2024). A multivariate autoregressive distributed lag unit root test. *Studies in Economics and Econometrics*, 1-17. doi:10.1080/03796205.2024.2439101

Examples

```
# Generate example data with cointegration
set.seed(123)
n <- 200
x <- cumsum(rnorm(n))
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run the test (reduced reps for speed)
result <- mvardlurt(y, x, case = 3, reps = 200)
print(result)

# With fixed lags
result2 <- mvardlurt(y, x, fixlag = c(2, 2), reps = 200)
```

plot.mvardlurt

Plot Method for mvardlurt Objects

Description

Creates diagnostic plots for the multivariate ARDL unit root test.

Usage

```
## S3 method for class 'mvardlurt'
plot(x, which = c(1, 2, 3, 4),
      ask = (length(which) > 1 && dev.interactive()), ...)
```

Arguments

<code>x</code>	An object of class "mvardlurt".
<code>which</code>	Integer vector indicating which plots to produce: <ul style="list-style-type: none">• 1: Residuals vs Fitted• 2: Q-Q plot of residuals• 3: Residuals over time• 4: ACF of residuals• 5: Time series of y and x• 6: Information criterion surface Default is <code>c(1, 2, 3, 4)</code> .
<code>ask</code>	Logical. If TRUE, prompt before each plot. Default is TRUE when multiple plots are requested in an interactive session.
<code>...</code>	Additional arguments passed to plotting functions.

Value

Invisibly returns `x`.

Author(s)

Muhammad Alkhalaf

See Also

[mvardlurt](#), [autoplot.mvardlurt](#)

Examples

```
set.seed(123)
n <- 100
x <- cumsum(rnorm(n))
y <- 0.5 * x + rnorm(n, sd = 0.5)

result <- mvardlurt(y, x, reps = 100)

# Default diagnostic plots
plot(result, ask = FALSE)

# IC surface plot
plot(result, which = 6)
```

print.mvdlurt *Methods for mvdlurt Objects*

Description

Print, summary, and accessor methods for objects of class "mvdlurt".

Usage

```
## S3 method for class 'mvdlurt'  
print(x, ...)  
  
## S3 method for class 'mvdlurt'  
summary(object, ...)  
  
## S3 method for class 'mvdlurt'  
coef(object, ...)  
  
## S3 method for class 'mvdlurt'  
residuals(object, ...)  
  
## S3 method for class 'mvdlurt'  
fitted(object, ...)
```

Arguments

x	An object of class "mvdlurt".
object	An object of class "mvdlurt".
...	Additional arguments (ignored).

Value

print and summary invisibly return x/object.
coef returns a named numeric vector with pi, delta, and lr_mult.
residuals returns the numeric vector of residuals from the fitted model.
fitted returns the numeric vector of fitted values.

Author(s)

Muhammad Alkhalaf

See Also

[mvdlurt](#)

Examples

```
set.seed(123)
n <- 100
x <- cumsum(rnorm(n))
y <- 0.5 * x + rnorm(n, sd = 0.5)

result <- mvardlurt(y, x, reps = 100)

# Print method
print(result)

# Summary method
summary(result)

# Extract coefficients
coef(result)

# Extract residuals
head(residuals(result))

# Extract fitted values
head(fitted(result))
```

Index

* package

mvardlurt-package, 2

autoplot.mvardlurt, 3, 7

coef.mvardlurt (print.mvardlurt), 8

fitted.mvardlurt (print.mvardlurt), 8

mvardlurt, 2, 3, 4, 7, 8

mvardlurt-package, 2

plot.mvardlurt, 3, 6

print.mvardlurt, 8

residuals.mvardlurt (print.mvardlurt), 8

summary.mvardlurt (print.mvardlurt), 8