Package ‘boundedur’

March 16, 2026

Type Package

Title Unit Root Tests for Bounded Time Series

Version 1.0.1

Description Implements unit root tests for bounded time series following Cavaliere and Xu (2014) <doi:10.1016/j.jeconom.2013.08.012>. Standard unit root tests (ADF, Phillips-Perron) have non-standard limiting distributions when the time series is bounded. This package provides modified ADF and M-type tests (MZ-alpha, MZ-t, MSB) with p-values computed via Monte Carlo simulation of bounded Brownian motion. Supports one-sided (lower bound only) and two-sided bounds, with automatic lag selection using the MAIC criterion of Ng and Perron (2001) <doi:10.1111/1468-0262.00256>.

License GPL-3

URL https://github.com/muhammedalkhalaf/boundedur

BugReports https://github.com/muhammedalkhalaf/boundedur/issues

Encoding UTF-8

Depends R (>= 3.5.0)

Imports stats

Suggests testthat (>= 3.0.0)

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Muhammad Alkhalaf [aut, cre, cph] (ORCID: <https://orcid.org/0009-0002-2677-9246>),
Giuseppe Cavaliere [ctb] (Original methodology),
Fang Xu [ctb] (Original methodology)

Maintainer Muhammad Alkhalaf <muhammedalkhalaf@gmail.com>

Repository CRAN

Date/Publication 2026-03-16 15:50:22 UTC

Contents

`boundedur` *Unit Root Tests for Bounded Time Series*

Description

Performs unit root tests for time series constrained within known bounds, following the methodology of Cavaliere and Xu (2014). Provides modified ADF and M-type test statistics with p-values computed via Monte Carlo simulation of bounded Brownian motion.

Usage

```
boundedur(
  y,
  lbound,
  ubound = Inf,
  test = c("all", "adf", "adf_alpha", "adf_t", "mz_alpha", "mz_t", "msb"),
  lags = NULL,
  maxlag = NULL,
  detrend = c("constant", "none"),
  nsim = 499,
  nstep = NULL,
  seed = NULL
)
```

Arguments

`y` Numeric vector. The time series to test.

`lbound` Numeric. Lower bound for the series.

`ubound` Numeric or `Inf`. Upper bound for the series. Use `Inf` for one-sided (lower bound only) testing.

`test` Character. Which test(s) to perform. One of:

- `"all"`: All available tests (default)
- `"adf"`: Both ADF-alpha and ADF-t
- `"adf_alpha"`: ADF normalized bias test only
- `"adf_t"`: ADF t-statistic test only
- `"mz_alpha"`: MZ-alpha test only
- `"mz_t"`: MZ-t test only
- `"msb"`: MSB test only

lags	Integer or NULL. Number of lagged differences to include. If NULL (default), selected automatically using MAIC.
maxlag	Integer or NULL. Maximum lag for automatic selection. If NULL, uses floor(12 * (T/100)^0.25).
detrend	Character. Detrending method: • "constant": Demean the series (default) • "none": No detrending
nsim	Integer. Number of Monte Carlo replications for p-value computation. Default is 499.
nstep	Integer or NULL. Number of discretization steps for Brownian motion simulation. If NULL, uses sample size T.
seed	Integer or NULL. Random seed for reproducibility.

Details

Standard unit root tests assume the series is unbounded, leading to non-standard limiting distributions when bounds are present. This function implements the bounded unit root tests of Cavaliere and Xu (2014), which account for the effect of bounds on the limiting distribution.

The null hypothesis is that the series has a unit root while respecting the bounds. The alternative is stationarity.

Value

An object of class "boundedur" containing:

statistics	Named vector of test statistics
p_values	Named vector of p-values
results	Data frame with statistics, p-values, and decisions
n	Sample size
lags	Number of lags used
lbound	Lower bound
ubound	Upper bound
c_lower	Standardized lower bound parameter
c_upper	Standardized upper bound parameter
sigma2_lr	Long-run variance estimate
detrend	Detrending method used
nsim	Number of Monte Carlo replications
call	The matched call

Test Statistics

ADF-alpha Augmented Dickey-Fuller normalized bias: $T(\hat{\rho} - 1)$

ADF-t Augmented Dickey-Fuller t-statistic for ρ

MZ-alpha Modified Phillips-Perron normalized bias

MZ-t Modified Phillips-Perron t-statistic

MSB Modified Sargan-Bhargava statistic

P-value Computation

P-values are computed by Monte Carlo simulation of bounded Brownian motion. The number of replications (`nsim`) controls accuracy; larger values give more precise p-values but increase computation time.

References

Cavaliere, G., & Xu, F. (2014). Testing for unit roots in bounded time series. *Journal of Econometrics*, 178(2), 259-272. doi:10.1016/j.jeconom.2013.08.012

Ng, S., & Perron, P. (2001). Lag length selection and the construction of unit root tests with good size and power. *Econometrica*, 69(6), 1519-1554. doi:10.1111/14680262.00256

Examples

```
# Generate bounded random walk (interest rate between 0 and 10)
set.seed(123)
n <- 200
y <- numeric(n)
y[1] <- 5
for (i in 2:n) {
  y[i] <- y[i-1] + rnorm(1, 0, 0.5)
  y[i] <- max(0, min(10, y[i]))  # Reflect at bounds
}

# Test for unit root with known bounds
result <- boundedur(y, lbound = 0, ubound = 10, nsim = 199)
print(result)
summary(result)

# One-sided bound (e.g., price level, lower bound = 0)
result_lower <- boundedur(y, lbound = 0, ubound = Inf, nsim = 199)
```

`select_lag_maic` *Select Optimal Lag using MAIC Criterion*

Description

Selects the optimal number of lags for the ADF regression using the Modified Akaike Information Criterion (MAIC) of Ng and Perron (2001).

Usage

```
select_lag_maic(y, maxlag = NULL, detrend = "constant")
```

Arguments

`y`	Numeric vector. Time series data.
`maxlag`	Integer or `NULL`. Maximum lag to consider. If `NULL`, uses the rule `floor(12 * (T/100)^0.25)`.
`detrend`	Character. Detrending method: "constant" (demean) or "none". Default is "constant".

Details

The MAIC criterion is defined as:

$$MAIC(k) = \ln(\hat{\sigma}_k^2) + 2(k+1)/T$$

where $\hat{\sigma}_k^2$ is the residual variance from the ADF regression with k lags.

This criterion provides better size properties than standard AIC for unit root testing.

Value

A list with class `"lag_selection"` containing:

`selected_lag`	Optimal lag selected by MAIC
`maic`	MAIC value at optimal lag
`all_maic`	Vector of MAIC values for all lags
`maxlag`	Maximum lag considered

References

Ng, S., & Perron, P. (2001). Lag length selection and the construction of unit root tests with good size and power. *Econometrica*, 69(6), 1519-1554. doi:10.1111/14680262.00256

Examples

```
# Generate random walk
set.seed(123)
y <- cumsum(rnorm(200))

# Select lag
lag_sel <- select_lag_maic(y)
print(lag_sel)
```

simulate_bounded_bm *Simulate Bounded Brownian Motion*

Description

Simulates a discretized reflected Brownian motion constrained between bounds, following the methodology of Cavaliere and Xu (2014).

Usage

```
simulate_bounded_bm(n, c_lower, c_upper = Inf)
```

Arguments

n	Integer. Number of time steps for discretization.
c_lower	Numeric. Standardized lower bound parameter.
c_upper	Numeric or Inf. Standardized upper bound parameter. Use Inf for one-sided (lower) bound only.

Details

The function simulates a standard Brownian motion and applies reflection at the boundaries. For two-sided bounds, both upper and lower reflections are applied. For one-sided bounds (c_upper = Inf), only lower reflection is used.

The standardized bound parameters c_lower and c_upper are computed from the original bounds as:

$$c = (b - X_0)/(\sigma\sqrt{T})$$

where b is the bound, X_0 is the initial value, σ is the long-run standard deviation, and T is the sample size.

Value

A numeric vector of length n + 1 containing the simulated bounded Brownian motion path, starting at 0.

References

Cavaliere, G., & Xu, F. (2014). Testing for unit roots in bounded time series. *Journal of Econometrics*, 178(2), 259-272. doi:10.1016/j.jeconom.2013.08.012

Examples

```
# Simulate bounded Brownian motion with two-sided bounds
set.seed(123)
bm <- simulate_bounded_bm(n = 1000, c_lower = -2, c_upper = 2)
plot(bm, type = "l", main = "Bounded Brownian Motion")
abline(h = c(-2, 2), col = "red", lty = 2)

# One-sided bound (lower only)
bm_lower <- simulate_bounded_bm(n = 1000, c_lower = -1, c_upper = Inf)
```

Index