Package: bootCT (via r-universe)

October 26, 2026

Type Package

Title Bootstrapping the ARDL Tests for Cointegration

Version 2.1.0

Author Gianmarco Vacca, Stefano Bertelli

Maintainer Gianmarco Vacca <gianmarco.vacca@unicatt.it>

Description The bootstrap ARDL tests for cointegration is the main functionality of this package. It also acts as a wrapper of the most commond ARDL testing procedures for cointegration: the bound tests of Pesaran, Shin and Smith (PSS; 2001 - <doi:10.1002/jae.616>) and the asymptotic test on the independent variables of Sam, McNown and Goh (SMG: 2019 - <doi:10.1016/j.econmod.2018.11.001>). Bootstrap and bound tests are performed under both the conditional and unconditional ARDL models.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp, pracma, aod, ARDL, dynamac, vars, gtools, dplyr, stringr, urca, magrittr, usethis

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 3.5.0)

RoxygenNote 7.2.3

LazyData true

Suggests knitr, rmarkdown, reshape2, ggplot2

VignetteBuilder knitr

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2024-01-16 02:40:50 UTC

RemoteUrl https://github.com/cran/bootCT

RemoteRef HEAD

RemoteSha 674b4983e2bed76f04ec07645fa50ab7a327cc58

Contents

boot_ardl 2
ger_macro 4
ita_macro 5
lag_mts 5
sim_vecm_ardl 6
smk_crit 8
summary.bootCT 9

Index **10**

`boot_ardl` *Bootstrap ARDL*

Description

This is the main function of the package. It performs the bootstrap version of the ARDL bound test for cointegration.

Usage

```
boot_ardl(
  data,
  yvar = NULL,
  xvar = NULL,
  fix.ardl = NULL,
  info.ardl = "AIC",
  fix.vecm = NULL,
  info.vecm = "AIC",
  maxlag = 5,
  a.ardl = 0.05,
  a.vecm = 0.05,
  nboot = 2000,
  case = 3,
  a.boot.H0 = c(0.05, 0.025, 0.01),
  print = T
)
```

Arguments

`data`	Input dataset. Must contain a dependent and a set of independent variables.
`yvar`	Name of the dependent variable, enclosed in quotation marks. If NULL, the first variable will be used.
`xvar`	Vector of names of the independent variables, each enclosed in quotation marks. If NULL, all variables except the first will be used.
`fix.ardl`	Fixed lagged differences for the short term part of the ARDL equation.

info.ardl	Selection criterion for the auto_ardl function. Options are "AIC", "AICc", BIC, "R2", "adjR2", if fix.ardl is null. Defaults to AIC.
fix.vecm	Fixed lagged differences for the short term part of the VECM equation.
info.vecm	Selection criterion for the VARselect function. Options are "AIC", "HQ", "SC", "FPE", if fix.vecm is null. Defaults to AIC.
maxlag	Max number of lags for the auto_ardl and VARselect procedures, if fix.ardl or fix.vecm are null, respectively.
a.ardl	Threshold significance for the short-term ARDL coefficients significance.
a.vecm	Threshold significance for the short-term VECM coefficients significance.
nboot	Number of bootstrap replications.
case	Model case, pertaining to the treatment of intercept and trend. Must be integer from 1 to 5. Defaults to 3.
a.boot.H0	Probability/ies by which the critical quantiles of the bootstrap distribution(s) must be calculated.
print	Show the progress bar.

Value

List of several elements including

- data: the data used to perform estimation and testing
- ARDL: the estimated ARDL conditional model
- VECM: the estimated VECM unconditional model
- jo.testX: Johansen cointegration test on the independent variables
- pssbounds: the PSS bound test output
- smgbounds: the SMG bound test critical values
- fov.stat: the test statistics on the conditional Fov tests
- t.stat: the test statistics on the conditional t test
- find.stat: the test statistics on the conditional Find tests
- quantfov: the bootstrap conditional F Overall test critical value(s)
- quantt: the bootstrap conditional t-test critical value(s)
- quantfind: the bootstrap conditional F Independent test critical value(s)
- fakecoint: indication of the situation in which $a_{y.x} \neq 0$ but $a_{y.x}^{UC} = 0$, signaling absence of cointegration.

Examples

```
## Not run:
#LOAD DATA
data("ger_macro")
# DATA PREPARATION
LNDATA = apply(ger_macro[,-1], 2, log)
col_ln = paste0("LN", colnames(ger_macro)[-1])
```

```
LNDATA = as.data.frame(LNDATA)
colnames(LNDATA) = col_ln
LNDATA$DATE = ger_macro$DATE
#ARDL BOOT
BCT_res = boot_ardl(data = LNDATA,
                    yvar = "LNCONS",
                    xvar = c("LNINCOME","LNINVEST"),
                    maxlag = 5,
                    a.ardl = 0.1,
                    a.vecm = 0.1,
                    nboot = 2000,
                    case = 3,
                    a.boot.H0 = c(0.05),
                    print = T)
summary(boot_res)

## End(Not run)
```

`ger_macro` *Investment, Income and Consumption dataset.*

Description

The data set contains quarterly, seasonally adjusted time series for West German fixed investment, disposable income, and consumption expenditures in billions of DM from 1960Q1 to 1982Q4. It was produced from file E1 of the data sets associated with Lutkepohl (2007). Originally obtained from Deutsche Bundesbank.

Usage

```
ger_macro
```

Format

A data frame with 92 rows and 4 variables:

DATE Quarter

INVEST Fixed investment (DM Billions)

INCOME Disposable income (DM Billions)

CONS Consumption expenditures (DM Billions)

Source

http://www.jmulti.de/download/datasets/e1.dat

ita_macro *Export, Foreign Investment and GDP dataset.*

Description

The data set contains yearly, time series for Italian log GDP, log foreign direct investment, and log exports, from 1970 to 2020. Obtained from the World Bank Database.

Usage

```
ita_macro
```

Format

A data frame with 51 rows and 4 variables:

YEAR From 1970 to 2020

LEXP log Exports (Adjusted for DGP deflator)

LFI log Foreign Direct Investment (Adjusted for DGP deflator)

LGDP log GDP

Source

World Bank Database

lag_mts *Create matrix of lagged variables*

Description

This function lags a set of variables in a matrix, each with a separate index. It is also possible to retain only the last lag order.

Usage

```
lag_mts(X, k, last.only = F)
```

Arguments

X	numeric matrix whose columns are subject to lagging
k	vector of lag orders
last.only	If TRUE only the k-th order lag will be computed, otherwise all lags from 1 to k

Value

a matrix whose columns are the original variables and the k-th order lagged variables. Column name suffix ".lx".

Examples

```
data(ger_macro)

lag_mts(X = ger_macro, k = 3, last.only = FALSE)
```

`sim_vecm_ardl` *Generate data from a VECM/ARDL equation*

Description

Generate data from a VECM/ARDL equation

Usage

```
sim_vecm_ardl(
  nobs,
  case = 1,
  sigma.in,
  gamma.in,
  axx.in,
  ayx.uc.in,
  ayy.in,
  mu.in = NULL,
  eta.in = NULL,
  azero.in = NULL,
  aone.in = NULL,
  burn.in = nobs * 0.5,
  seed.in = NULL
)
```

Arguments

`nobs`	number of observations.
`case`	case related to intercept and trend
`sigma.in`	error covariance matrix $\mathbf{\Sigma}$
`gamma.in`	list of VECM short-run parameter matrices $\mathbf{\Gamma}_j$
`axx.in`	long-run relationships between the independent variables $\mathbf{A}_{xx}$
`ayx.uc.in`	long-run unconditional relationship between dependent and independent variables, $\mathbf{a}_{yx}$. The second component ayxC, derived from conditioning, is calculated as $\mathbf{a}_{yx}^{(C)} = -\boldsymbol{\omega}'\mathbf{A}_{xx}$

ayy.in	long-run relationship for the dependent variable a_{yy}
mu.in	VAR intercept vector $\boldsymbol{\mu}$ (CASE II)
eta.in	VAR trend vector $\boldsymbol{\eta}$ (CASE IV)
azero.in	VECM intercept $\boldsymbol{\alpha}_0$ (CASE III-IV-V)
aone.in	VECM trend $\boldsymbol{\alpha}_1$ (CASE V)
burn.in	burn-in number of observations
seed.in	optional seed number for random error generation.

Value

A list that includes

- dims: a vector with the dataset dimension
- case: the case given as input
- data: the generated data
- diffdata: the data first difference
- ut: the generated random error matrix.
- sigma: the error covariance matrix $\boldsymbol{\Sigma}$.
- omega: the $\boldsymbol{\omega}$ vector of parameters generated via conditioning
- at: the conditional long-run parameter matrix $\widetilde{\mathbf{A}}$
- ayy: the coefficient weighting the EC term, a_{yy}
- ayx.uc: the unconditional subvector of the ARDL equation $\mathbf{a}_{yx}$
- ayx2: the conditioning effect $\omega' A_{xx}$
- ayx.c: the conditional subvector of the ARDL equation $\widetilde{a}_{y.x} = a_{yx} - \omega' A_{xx}$
- gammalist: the list of unconditional $\boldsymbol{\Gamma}_j$ parameter matrices
- psilist: the list of conditional $\boldsymbol{\gamma}_{y.x,j}$ parameter matrices
- vmu: the VAR intercept $\boldsymbol{\mu}$
- veta: the VAR trend $\boldsymbol{\eta}$
- azero: the unconditional VECM intercept $\boldsymbol{\alpha}_0$
- aone: the unconditional VECM trend $\boldsymbol{\alpha}_1$
- azero.c: the conditional VECM intercept $\boldsymbol{\alpha}_0^c$
- aone.c: the conditional VECM trend $\boldsymbol{\alpha}_1^c$
- interc.ardl: the conditional ARDL intercept $\alpha_{0.y}$ (case > 2)
- trend.ardl: the conditional ARDL trend $\alpha_{1.y}$ (case = 5)
- theta0: the θ_0 coefficient in the EC term (case = 2)
- theta1: the θ_1 coefficient in the EC term (case = 4)
- interc.ec: the conditional ARDL intercept derived from the EC tem $\alpha_{0.y}^{EC}$ (case = 2)
- trend.ec: the conditional ARDL trend derived from the EC tem $\alpha_{1.y}^{EC}$ (case = 4)

Examples

```
#PARAMETERS

#Sigma
corrm = matrix(0, ncol = 3, nrow = 3)
corrm[2,1] = 0.25
corrm[3,1] = 0.4
corrm[3,2] = -0.25
corrs = (corrm + t(corrm)) + diag(3)
sds = diag(c(1.3, 1.2, 1))
sigma = (sds %*% corrs %*% t(sds))

#Gamma
gammax = list()
gammax[[1]] = matrix(c(0.6, 0, 0.2, 0.1, -0.3, 0, 0, -0.3, 0.2), nrow = 3, ncol = 3, byrow = TRUE)
gammax[[2]] = matrix(c(0.2, 0, 0.1, 0.05, -0.15, 0, 0, 0, 0.1), nrow = 3, ncol = 3, byrow = TRUE)

#DATA GENERATION
data_sim = sim_vecm_ardl(nobs = 200,
                         case = 3,
                         sigma.in = sigma,
                         gamma.in = gammax,
                         axx.in = matrix(c(0.3, 0.5, 0.4, 0.3), nrow = 2, ncol = 2),
                         ayx.uc.in = c(0.5,0.6),
                         ayy.in = 0.7,
                         mu.in = rep(0, 3),
                         eta.in = rep(0, 3),
                         azero.in = rep(0.4, 3),
                         aone.in = rep(0, 3),
                         burn.in = 50,
                         seed.in = 10)
```

---

smk_crit *Critical values of the F-test on the independent variables in the conditional ARDL model.*

Description

This internal data contains critical values of the Find test of Sam et al (2018), for several sample sizes and lag orders. Applicable only for cases I, III, and V. Critical values at significance levels 1%, 2.5%, 5% and 10%.

Usage

```
smk_crit
```

Format

A data frame with 144 rows and 17 variables

case case related to intercept and trend specification. Only I, III or V

prob significance levels. 0.01, 0.025, 0.05 or 0.10

num sample size, 30 to 80 and beyond for asymptotic critical values

I0_x right threshold for accepting the null of zero coefficients in the Find test. x: 1 to 7

I1_x left threshold for rejecting the null of zero coefficients in the Find test. x: 1 to

---

summary.bootCT *Summary method*

Description

This function summarizes the ARDL bootstrap test and all the other asymptotic procedures all together.

Usage

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

Arguments

object	an object of class "bootCT"
...	additional arguments, e.g. out: subset of output to print. Options (can be multiple) are: "all", "ARDL", "VECM", "cointVECM", "cointARDL". Defaults to "all".

Value

the function returns a list of summary statistics, already present in the function boot_ardl, and displays them in an appropriate manner. Depending on the out argument, ARDL/VECM estimation outputs and/or ARDL/VECM cointegration tests can be displayed.

Index

∗ **datasets**
ger_macro, 4
ita_macro, 5
smk_crit, 8

boot_ardl, 2

ger_macro, 4

ita_macro, 5

lag_mts, 5

sim_vecm_ardl, 6
smk_crit, 8
summary.bootCT, 9