

bootCT: An R Package for Bootstrap Cointegration Tests in ARDL Models

by Gianmarco Vacca, Maria Zoia, Stefano Bertelli

Abstract The Autoregressive Distributed Lag approach to cointegration or bound testing, proposed by Pesaran in 2001, has become prominent in empirical research. Although this approach has many advantages over the classical cointegration tests, it is not exempt from drawbacks, such as possible inconclusive inference and distortion in size. Recently, Bertelli and coauthors developed a bootstrap approach to the bound tests to overcome these drawbacks. This paper introduces the R package bootCT, which implements this method by deriving the bootstrap versions of the bound tests and of the asymptotic F-test on the independent variables proposed by Sam and coauthors in 2019. As a spinoff, a general method for generating random multivariate time series following a given VECM/ARDL structure is provided in the package. Empirical applications showcase the main functionality of the package.

1 Introduction

Cointegration and error correction are fundamental concepts in the analysis of economic data, insofar as they provide an appropriate framework for testing economic hypotheses about growth and fluctuation. Several approaches have been proposed in the literature to determine whether two or more non-stationary time series are cointegrated, meaning they share a common long-run relationship. There are two basic types of tests for cointegration: single equation tests and VAR-based tests. The former check the presence of unit roots in cointegration residuals (see, e.g., Engle and Granger, 1987; Engle and Yoo, 1987; Mackinnon, 1991; Gabriel et al., 2002; Cook, 2006) or test the significance of the error-correction (EC) term coefficient (Kremers et al., 1992; Maddala and Kim, 1998; Arranz and Escribano, 2000; Ericsson and MacKinnon, 2002). The latter, such as the Johansen (1991) approach, tackle the problem of detecting cointegrating relationships in a VAR model. This latter approach, albeit having the advantage of avoiding the issue of normalization, as well as allowing the detection of multiple cointegrating vectors, is far from being perfect. In the VAR system all variables are treated symmetrically, as opposed to the standard univariate models that usually have a clear interpretation in terms of exogenous and endogenous variables. Furthermore, in a VAR system all the variables are estimated at the same time, which is problematic if the relation between some variables is flawed, that is affected by some source of error. In this case a simultaneous estimation process tends to propagate the error affecting one equation to the others. Furthermore, a multidimensional VAR models employs plenty of degrees of freedom.

The recent cointegration approach, known as Autoregressive Distributed Lag (ARDL) approach to cointegration or bound testing, proposed by Pesaran et al. (2001) (PSS), falls in the former strand of literature. It has become prominent in empirical research because it shows several advantages with respect to traditional methods for testing cointegration. First, it is applicable also in cases of mixed order integrated variables, albeit with integration not exceeding the first order. Thus, it evades the necessity of pre-testing the variables and, accordingly, avoids some common practices that may prevent finding cointegrating relationships, such as dropping variables or transforming them into stationary form (see McNown et al., 2018). Second, cointegration bound tests are performed in an ARDL model that allows different lag orders for each variable, thus providing a more flexible framework than other commonly employed approaches. Finally, unlike other cointegration techniques, which are sensitive to the sample size, the ARDL approach provides robust and consistent results for small sample sizes.

Notably, the ARDL bound testing methodology has quickly spread in economics and econometrics to study the cointegrating relationships between macroeconomic and financial variables, to evaluate the long-run impact of energy variables, or to assess recent environmental policies and their impact on the economy. Among the many applications, see for instance Haseeb et al. (2019); Reda and Nourhan (2020); Menegaki (2019); Yilanci et al. (2020); Hussain et al. (2019); Abbasi et al. (2021).

The original bound tests proposed by Pesaran et al. (2001) are an F -test for the significance of the coefficients of all lagged level variables entering the error correction term (F_{ov}), and a t -test for the coefficient of the lagged dependent variable. When either the dependent or the independent variables do not appear in the long-run relationship, a degenerate case arises. The bound t -test provides answers on the occurrence of a degenerate case of second type, while the occurrence of a degeneracy case of first type can be assessed by testing whether the dependent variable is of integration order $I(1)$. This type of check violates the spirit and motivation of the bound tests, which are supposed to be applicable in situations of unknown order of integration for the variables.

Recently, McNown et al. (2018) pointed out how, due to the low power problem of unit root tests, investigating the presence of a first type degeneracy by testing the integration order of the dependent variable may lead to incorrect conclusions. Therefore, they suggested checking for its occurrence by testing the significance of the lagged levels of the independent variables via an extra F -test (F_{ind}), which was also worked out in its asymptotic version (SMK; Sam et al., 2019).

Besides problems in testing the occurrence of degenerate cases, in general, the main drawback of the bound tests is the occurrence of potentially inconclusive results, if the test statistic lies between the bounds of the test distribution under the null. Furthermore, the asymptotic distributions of the statistics may provide a poor approximation of the true distributions in small samples. Finite sample critical values, even if only for a subset of all possible model specifications, have been worked out in the literature (see Mills and Pentecost, 2001; Narayan and Smyth, 2004; Kanioura and Turner, 2005; Narayan, 2005), while Kripfganz and Schneider (2020) provided the quantiles of the asymptotic distributions of the tests as functions of the sample size, the lag order and the number of long-run forcing variables. However, this relevant improvement does not eliminate the uncertainty related to the inconclusive regions, or the existence of other critical issues related to the underlying assumptions of the bound test framework, such as the (weak) exogeneity of the independent variables or the non-stationarity of the dependent variable.

To overcome the mentioned bound test drawbacks, Bertelli et al. (2022) proposed bootstrapping the ARDL cointegration test. Inference can always be pursued with ARDL bootstrap tests, unlike what happens with both the PSS tests and the SMK test on the independent variables. Bootstrap ARDL tests were first put forward by McNown et al. (2018) in an unconditional ARDL model, which omits the instantaneous differences of the exogenous variables in the ARDL equation, rather than a conditional one, as originally proposed by Pesaran et al. (2001). The unconditional model is often used, for reason of practical convenience, in empirical research. Simulation results in Bertelli et al. (2022) have highlighted the importance of employing the appropriate specification, especially under degenerate cases. In fact, it has been pointed out that a correct detection of these cases requires the comparison of the test outcomes in both the conditional and unconditional settings. Erroneous conclusions, based exclusively on one model specification, can thus be avoided.

In this paper, bootstrap bound tests, thereby including the bootstrap versions of the F_{ov} , t and F_{ind} bound tests, are carried out in a conditional ARDL model setting. This approach allows to overcome the problem of inconclusive regions of the standard bound tests. A comparison with the outcomes engendered by the unconditional ARDL bootstrap tests is nevertheless provided for the F_{ind} test, to avoid erroneous inference in presence of degenerate cases.

The paper is organized as follows. Section 2.2 introduces the theoretical results of the ARDL cointegration bound tests. Section 2.3 details the steps carried out by the bootstrap procedure, which allows the construction of the (bootstrap) distribution - under the null - for the F_{ov} , t , conditional F_{ind} and unconditional F_{ind} tests. Section 2.4 introduces the R package **bootCT** (Vacca and Bertelli, 2023) and its functionalities: a method for the generation of random multivariate time series that follow a user-specified VECM/ARDL structure, with some examples, and the main function that carries out the aforementioned bootstrap tests, while also computing the PSS and SMK bound tests. The trade-off between accuracy and computational time of the bootstrap procedure is also investigated, under several scenarios in terms of sample size and number of replications. Notably, a function that performs the PSS bound tests is already available in the **dynamac** package (Jordan and Philips, 2020), while no R routine has so far been implemented for the SMK test, to the best of our knowledge. Section 2.5 gives some empirical applications that employ the core function of the package and its possible outputs. Section 2.6 concludes. Appendix 2.7 briefly delves into technical details of the conditional ARDL model and its possible specifications¹.

2 Cointegration bound tests in ARDL models

The starting point of the approach proposed by Pesaran et al. (2001) is a $(K + 1)$ VAR(p) model

$$\mathbf{A}(L)(\mathbf{z}_t - \boldsymbol{\mu} - \boldsymbol{\eta}t) = \boldsymbol{\varepsilon}_t \quad \boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}), \quad \mathbf{A}(L) = \left(\mathbf{I}_{K+1} - \sum_{j=1}^p \mathbf{A}_j L^j \right) \quad t = 1, 2, \dots, T. \quad (1)$$

¹The R packages, either used in the creation of **bootCT** or employed in the analyses presented in this paper, are **magrittr** (Bache and Wickham, 2022), **gtools** (Bolker et al., 2022), **pracma** (Borchers, 2022), **Rcpp** (Eddelbuettel, 2013), **RcppArmadillo** (Eddelbuettel et al., 2023), **Rmisc** (Hope, 2022), **dynamac** (Jordan and Philips, 2020), **ARDL** (Natsopoulos and Tzeremes, 2021), **aod** (Lesnoff et al., 2012), **vars** and **urca** (Pfaff, 2008a,b), **aTSA** (Qiu, 2015), **tseries** (Trapletti and Hornik, 2023), **reshape2**, **ggplot2** and **stringr** (Wickham, 2007, 2016, 2022), **tidyverse** and **dplyr** (Wickham et al., 2019, 2023).

Here, \mathbf{A}_j are square $(K+1)$ matrices, \mathbf{z}_t a vector of $(K+1)$ variables, $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ are $(K+1)$ vectors representing the drift and the trend respectively, and $\det(\mathbf{A}(z)) = 0$ for $|z| \geq 1$. If the matrix $\mathbf{A}(1) = \mathbf{I}_{K+1} - \sum_{j=1}^p \mathbf{A}_j$ is singular, the components of \mathbf{z}_t turn out to be integrated and possibly cointegrated.

The VECM representation of (1) is given by (see Appendix 2.7.1 for details)

$$\Delta \mathbf{z}_t = \boldsymbol{\alpha}_0 + \boldsymbol{\alpha}_1 t - \mathbf{A}(1) \mathbf{z}_{t-1} + \sum_{j=1}^{p-1} \boldsymbol{\Gamma}_j \Delta \mathbf{z}_{t-j} + \boldsymbol{\varepsilon}_t. \quad (2)$$

Now, to study the adjustment to the equilibrium of a single variable y_t , given the other \mathbf{x}_t variables, the vectors \mathbf{z}_t and $\boldsymbol{\varepsilon}_t$ are partitioned

$$\mathbf{z}_t = \begin{bmatrix} y_t \\ (1,1) \\ \mathbf{x}_t \\ (K,1) \end{bmatrix}, \quad \boldsymbol{\varepsilon}_t = \begin{bmatrix} \varepsilon_{yt} \\ (1,1) \\ \boldsymbol{\varepsilon}_{xt} \\ (K,1) \end{bmatrix}. \quad (3)$$

The matrix $\mathbf{A}(1)$, which is assumed to be singular to allow cointegration, is partitioned conformably to \mathbf{z}_t as ²

$$\mathbf{A}(1) = \begin{bmatrix} a_{yy} & \mathbf{a}'_{yx} \\ (1,1) & (1,K) \\ \mathbf{a}_{xy} & \mathbf{A}_{xx} \\ (K,1) & (K,K) \end{bmatrix}. \quad (4)$$

Under the assumption

$$\boldsymbol{\varepsilon}_t \sim N\left(\mathbf{0}, \begin{bmatrix} \sigma_{yy} & \sigma'_{yx} \\ (1,1) & (1,K) \\ \boldsymbol{\sigma}_{xy} & \boldsymbol{\Sigma}_{xx} \\ (K,1) & (K,K) \end{bmatrix}\right), \quad (5)$$

the following holds

$$\varepsilon_{yt} = \boldsymbol{\omega}' \boldsymbol{\varepsilon}_{xt} + v_{yt} \sim N(0, \sigma_{y.x}), \quad (6)$$

where $\sigma_{y.x} = \sigma_{yy} - \boldsymbol{\omega}' \boldsymbol{\sigma}_{xy}$ with $\boldsymbol{\omega}' = \boldsymbol{\sigma}'_{yx} \boldsymbol{\Sigma}_{xx}^{-1}$, and v_{yt} is independent of $\boldsymbol{\varepsilon}_{xt}$.

Substituting (6) into (2) and assuming that the \mathbf{x}_t variables are exogenous towards the ARDL parameters (that is, setting $\mathbf{a}_{xy} = \mathbf{0}$ in $\mathbf{A}(1)$) yields the system (see Appendix 2.7.1 for details)

$$\Delta y_t = \alpha_{0,y} + \alpha_{1,y} t - a_{yy} EC_{t-1} + \sum_{j=1}^{p-1} \gamma'_{y,x,j} \Delta \mathbf{z}_{t-j} + \boldsymbol{\omega}' \Delta \mathbf{x}_t + v_{yt} \quad (7)$$

$$\Delta \mathbf{x}_t = \boldsymbol{\alpha}_{0x} + \boldsymbol{\alpha}_{1x} t + \mathbf{A}_{(x)} \mathbf{z}_{t-1} + \boldsymbol{\Gamma}_{(x)}(L) \Delta \mathbf{z}_t + \boldsymbol{\varepsilon}_{xt}, \quad (8)$$

where

$$\gamma'_{y,x,j} = \gamma'_{y,j} - \boldsymbol{\omega}' \boldsymbol{\Gamma}_{(x),j} \quad (9)$$

$$\alpha_{0,y} = \alpha_{0y} - \boldsymbol{\omega}' \boldsymbol{\alpha}_{0x}, \quad \alpha_{1,y} = \alpha_{1y} - \boldsymbol{\omega}' \boldsymbol{\alpha}_{1x}, \quad (10)$$

and where the error correction term, EC_{t-1} , expressing the long-run equilibrium relationship between y_t and \mathbf{x}_t , is given by

$$EC_{t-1} = y_{t-1} - \theta_0 - \theta_1 t - \boldsymbol{\theta}' \mathbf{x}_{t-1}, \quad (11)$$

with

$$\theta_0 = \mu_y - \boldsymbol{\theta}' \boldsymbol{\mu}_x, \quad \theta_1 = \eta_y - \boldsymbol{\theta}' \boldsymbol{\eta}_x, \quad \boldsymbol{\theta}' = -\frac{\tilde{\mathbf{a}}'_{y,x}}{a_{yy}} = -\frac{\mathbf{a}'_{yx} - \boldsymbol{\omega}' \mathbf{A}_{xx}}{a_{yy}}. \quad (12)$$

Thus, no cointegration occurs when $\tilde{\mathbf{a}}_{y,x} = \mathbf{0}$ or $a_{yy} = 0$. These two circumstances are referred to as degenerate case of second and first type, respectively. Degenerate cases imply no cointegration between y_t and \mathbf{x}_t .

To test the hypothesis of cointegration between y_t and \mathbf{x}_t , Pesaran et al. (2001) proposed an F -test, F_{ov} hereafter, based on the hypothesis system

$$H_0 : a_{yy} = 0 \cap \tilde{\mathbf{a}}_{y,x} = \mathbf{0} \quad (13)$$

$$H_1 : a_{yy} \neq 0 \cup \tilde{\mathbf{a}}_{y,x} \neq \mathbf{0}. \quad (14)$$

²If the explanatory variables are stationary \mathbf{A}_{xx} is non-singular ($rk(\mathbf{A}_{xx}) = K$), while when they are integrated but without cointegrating relationship \mathbf{A}_{xx} is a null matrix.

Note that H_1 covers also the degenerate cases

$$H_1^{y,x} : a_{yy} = 0, \tilde{\mathbf{a}}_{y,x} \neq \mathbf{0} \quad (15)$$

$$H_1^{yy} : a_{yy} \neq 0, \tilde{\mathbf{a}}_{y,x} = \mathbf{0}. \quad (16)$$

The exact distribution of the F statistic under the null is unknown, but it is limited from above and below by two asymptotic distributions: one corresponding to the case of stationary regressors, and another corresponding to the case of first-order integrated regressors. As a consequence, the test is called bound test and has an inconclusive area.³

Pesaran et al. (2001) worked out two sets of (asymptotic) critical values: one, $\{\tau_{L,F}\}$, for the case when $\mathbf{x}_t \sim I(0)$ and another, $\{\tau_{UL,F}\}$, for the case when $\mathbf{x}_t \sim I(1)$. These values vary in accordance with the number of regressors in the ARDL equation, the sample size and the assumptions made about the deterministic components (intercept and trend) of the data generating process.

In this regard, Pesaran et al. (2001) introduced five different specifications for the ARDL model, depending on its deterministic components, which are (see Appendix 2.7.2 for details)

I. *No intercept and no trend*

$$\Delta y_t = -a_{yy}EC_{t-1} + \sum_{j=1}^{p-1} \gamma'_{y,x,j} \Delta \mathbf{z}_{t-j} + \omega' \Delta \mathbf{x}_t + v_{yt}, \quad (17)$$

where $EC_{t-1} = y_{t-1} - \theta' \mathbf{x}_{t-1}$,

II. *Restricted intercept and no trend*

$$\Delta y_t = -a_{yy}EC_{t-1} + \sum_{j=1}^{p-1} \gamma'_{y,x,j} \Delta \mathbf{z}_{t-j} + \omega' \Delta \mathbf{x}_t + v_{yt}, \quad (18)$$

where $EC_{t-1} = y_{t-1} - \theta_0 - \theta' \mathbf{x}_{t-1}$. The intercept extracted from the EC term is $\alpha_{0,y}^{EC} = a_{yy}\theta_0$.

III. *Unrestricted intercept and no trend*

$$\Delta y_t = \alpha_{0,y} - a_{yy}EC_{t-1} + \sum_{j=1}^{p-1} \gamma'_{y,x,j} \Delta \mathbf{z}_{t-j} + \omega' \Delta \mathbf{x}_t + v_{yt}, \quad (19)$$

where $EC_{t-1} = y_{t-1} - \theta' \mathbf{x}_{t-1}$.

IV. *Unrestricted intercept, restricted trend*

$$\Delta y_t = \alpha_{0,y} - a_{yy}EC_{t-1} + \sum_{j=1}^{p-1} \gamma'_{y,x,j} \Delta \mathbf{z}_{t-j} + \omega' \Delta \mathbf{x}_t + v_{yt}, \quad (20)$$

where $EC_{t-1} = y_{t-1} - \theta_1 t - \theta' \mathbf{x}_{t-1}$. The trend extracted from the EC term is $\alpha_{1,y}^{EC} = a_{yy}\theta_1$.

V. *Unrestricted intercept, unrestricted trend*

$$\Delta y_t = \alpha_{0,y} + \alpha_{1,y}t - a_{yy}EC_{t-1} + \sum_{j=1}^{p-1} \gamma'_{y,x,j} \Delta \mathbf{z}_{t-j} + \omega' \Delta \mathbf{x}_t + v_{yt}, \quad (21)$$

where $EC_{t-1} = y_{t-1} - \theta' \mathbf{x}_{t-1}$.

The model in (7) proposed by Pesaran et al. (2001) represents the correct framework in which to carry out bound tests. However, bound test are often performed in an unconditional ARDL model setting, specified as

$$\Delta y_t = \alpha_{0,y} + \alpha_{1,y}t - a_{yy}EC_{t-1} + \sum_{j=1}^{p-1} \gamma'_j \Delta \mathbf{z}_{t-j} + \varepsilon_{yt}, \quad (22)$$

which omits the term $\omega' \Delta \mathbf{x}_t$.

Bertelli et al. (2022) have highlighted that bootstrap tests performed in these two ARDL specifications can lead to contrasting results. To explain this divergence, note that the conditional model makes use of the following vector in the EC term

$$\tilde{\mathbf{a}}'_{y,x} = \mathbf{a}'_{yx} - \omega' \mathbf{A}_{xx} \quad (23)$$

³The knowledge of the rank of the cointegrating matrix is necessary to overcome this impasse.

(divided by a_{yy} , see (12)) to carry out bound tests, while the unconditional one only uses the vector \mathbf{a}'_{yx} , (divided by a_{yy}), since it neglects the term $\omega' \mathbf{A}_{xx}$.⁴ This can lead to contrasting inference in two instances. The first happens when a degeneracy of first type occurs in the conditional model, that is

$$\tilde{\mathbf{a}}'_{y.x} = \mathbf{0}, \quad (24)$$

because

$$\mathbf{a}'_{yx} = \omega' \mathbf{A}_{xx}. \quad (25)$$

In this case, the conditional model rejects cointegration, while the unconditional one concludes the opposite. The other case happens when a degeneracy of first type occurs in the unconditional model, that is

$$\mathbf{a}'_{yx} = \mathbf{0}, \quad (26)$$

but

$$\tilde{\mathbf{a}}'_{y.x} = \omega' \mathbf{A}_{xx} \neq \mathbf{0}. \quad (27)$$

In this case, the unconditional model rejects cointegration, while the conditional one concludes for the existence of cointegrating relationships, which are however spurious. Only a comparison of the outcomes of the F_{ind} test performed in both the conditional and unconditional ARDL equation can help to disentangle this problem.⁵

In the following, bootstrap tests are carried out in the conditional ARDL model (7). However, when a degeneracy of first type occurs in the unconditional model, the outcomes of the F_{ind} bootstrap test performed in both the conditional and unconditional settings are provided. This, as previously outlined, is performed to avoid the acceptance of spurious long-run relationships among the dependent variable and the independent variables.

3 The new bootstrap procedure

The bootstrap procedure here proposed focuses on a ARDL model specified as in (17)-(21), depending on the assumptions on the deterministic components.

The bootstrap procedure consists of the following steps:

1. The ARDL model is estimated via OLS and the related test statistics F_{ov} , t or F_{ind} are computed.
2. In order to construct the distribution of each test statistic under the corresponding null, the same model is re-estimated imposing the appropriate restrictions on the coefficients according to the test under consideration.
3. Following McNown et al. (2018), the ARDL restricted residuals are then computed. For example, under Case III, the residuals are

$$\hat{v}_{yt}^{F_{ov}} = \Delta y_t - \hat{\alpha}_{0,y} - \sum_{j=1}^{p-1} \hat{\gamma}'_{y.x,j} \Delta \mathbf{z}_{t-j} - \hat{\omega}' \Delta \mathbf{x}_t \quad (28)$$

$$\hat{v}_{yt}^t = \Delta y_t - \hat{\alpha}_{0,y} + \tilde{\mathbf{a}}'_{y.x} \mathbf{x}_{t-1} - \sum_{j=1}^{p-1} \hat{\gamma}'_{y.x,j} \Delta \mathbf{z}_{t-j} - \hat{\omega}' \Delta \mathbf{x}_t \quad (29)$$

$$\hat{v}_{yt}^{F_{ind}} = \Delta y_t - \hat{\alpha}_{0,y} + \hat{a}_{yy} y_{t-1} - \sum_{j=1}^{p-1} \hat{\gamma}'_{y.x,j} \Delta \mathbf{z}_{t-j} - \hat{\omega}' \Delta \mathbf{x}_t. \quad (30)$$

Here, the apex " $\hat{\cdot}$ " denotes the estimated parameters. The other cases can be dealt with in a similar manner.

4. The VECM model

$$\Delta \mathbf{z}_t = \boldsymbol{\alpha}_0 - \mathbf{A} \mathbf{z}_{t-1} + \sum_{j=1}^{p-1} \boldsymbol{\Gamma}_j \Delta \mathbf{z}_{t-j} + \boldsymbol{\varepsilon}_t \quad (31)$$

is estimated as well (imposing weak exogeneity), and the residuals

$$\hat{\varepsilon}_{xt} = \Delta \mathbf{x}_t - \hat{\alpha}_{0x} + \hat{\mathbf{A}}_{xx} \mathbf{x}_{t-1} - \sum_{j=1}^{p-1} \hat{\boldsymbol{\Gamma}}_{(x)j} \Delta \mathbf{z}_{t-j} \quad (32)$$

⁴The latter is introduced in the ARDL equation by the operation of conditioning y_t on the other variables \mathbf{x}_t of the model

⁵In fact, as $\omega' \mathbf{A}_{xx} \mathbf{x}_t \approx I(0)$, the conclusion that $y_t \approx I(0)$ must hold. This in turn entails that no cointegration occurs between y_t and \mathbf{x}_t .

are computed. This approach guarantees that the residuals $\hat{\varepsilon}_{xt}$, associated to the variables \mathbf{x}_t explained by the marginal model (8), are uncorrelated with the ARDL residuals \hat{v}_{yt} .

5. A large set of B bootstrap replicates are sampled from the residuals calculated as in (28), (29), (30) and (32). In each replication, the following operations are carried out:

- (a) Each set of $(T - p)$ resampled residuals (with replacement) $\hat{v}_{zt}^{(b)} = (\hat{v}_{yt}^{(b)}, \hat{\varepsilon}_{xt}^{(b)})$ is re-centered (see Davidson and MacKinnon, 2005)

$$\hat{v}_{yt}^{(b)} = \hat{v}_{yt}^{(b)} - \frac{1}{T - p} \sum_{t=p+1}^T \hat{v}_{yt}^{(b)} \quad (33)$$

$$\hat{\varepsilon}_{x_{it}}^{(b)} = \hat{\varepsilon}_{x_{it}}^{(b)} - \frac{1}{T - p} \sum_{t=p+1}^T \hat{\varepsilon}_{x_{it}}^{(b)} \quad i = 1, \dots, K. \quad (34)$$

- (b) A sequential set of $(T - p)$ bootstrap observations, $y_t^*, \mathbf{x}_t^* \quad t = p + 1, \dots, T$, is generated as follows

$$y_t^* = y_{t-1}^* + \Delta y_t^*, \quad \mathbf{x}_t^* = \mathbf{x}_{t-1}^* + \Delta \mathbf{x}_t^*, \quad (35)$$

where $\Delta \mathbf{x}_t^*$ are obtained from (32) and Δy_t^* from either (28), (29) or (30) after replacing in each of these equations the original residuals with the bootstrap ones.

The initial conditions, that is the observations before $t = p + 1$, are obtained by drawing randomly p observations in block from the original data, so as to preserve the data dependence structure.

- (c) An unrestricted ARDL model is estimated via OLS using the bootstrap observations, and the statistics $F_{ov}^{(b), H_0}, t^{(b), H_0}, F_{ind}^{(b), H_0}$ are computed.

6. The bootstrap distributions of $\{F_{ov}^{(b), H_0}\}_{b=1}^B$, $\{F_{ind}^{(b), H_0}\}_{b=1}^B$ and $\{t^{(b), H_0}\}_{b=1}^B$ under the null are then employed to determine the critical values of the tests. By denoting with M_b^* the ordered bootstrap test statistic, and with α the nominal significance level, the bootstrap critical values are determined as follows

$$c_{\alpha, M}^* = \min \left\{ c : \sum_{b=1}^B \mathbf{1}_{\{M_b^* > c\}} \leq \alpha \right\} \quad M \in \{F_{ov}, F_{ind}\} \quad (36)$$

for the F tests and

$$c_{\alpha, t}^* = \max \left\{ c : \sum_{b=1}^B \mathbf{1}_{\{t_b^* < c\}} \leq \alpha \right\} \quad (37)$$

for the t test.

Here, $\mathbf{1}_{\{x \in A\}}$ is the indicator function, which is equal to one if the condition in subscript is satisfied and zero otherwise.

The null hypothesis is rejected if the F statistic computed at step 1, F_{ov} or F_{ind} , is greater than the respective $c_{\alpha, M}^*$, or if the t statistic computed at the same step is lower than $c_{\alpha, t}^*$.

4 Illustration of the bootCT package

This section describes the main functionalities of the **bootCT** package. The functions included in the package are essentially of two types. The function `sim_vecm_ardl` generates data according to a given data generating process (DGP), assuming either the presence or the absence of cointegrating relationships between variables, or degenerate cases. The function `boot_ardl` tests the presence of cointegrating relationships employing the Pesaran ARDL bound tests (F_{ov} and t), the SMK bound test on lagged independent variables (F_{ind}), and the novel ARDL bootstrap testing procedure.

4.1 Generating a multivariate time series: the `sim_vecm_ardl` function

The function `sim_vecm_ardl` allows to simulate a multivariate time series from a given conditional ARDL specification for a dependent variable y_t and a VAR/VECM specification for the remaining independent variables \mathbf{x}_t . In this regard, it represents an interesting addition to extant data generating procedures for VAR/VECM models. The arguments of this function can be divided into two subgroups. A group of parameters pertains the VECM model (7) and (8), with \mathbf{A}_{xx} identifying the matrix of the long-run relationships among the \mathbf{x}_t variables, and Γ_j 's, $j = 1, \dots, p - 1$ the short-run matrices of the system variables. Additionally, the parameter a_{yy} weighs the EC term for y_t , while a'_{yx} is the parameter

vector weighting the variables x_t in the ARDL equation. The vector a'_{yx} , after conditioning y_t on the other variables (x_t , see model 7) becomes $\tilde{a}'_{y,x} = a'_{yx} - \omega' A_{xx}$.

The second group of parameters concerns the model intercept and trend of the VAR specification, μ and η , which in the VECM representation become $\alpha_0 = A\mu + (I_{K+1} - \sum_{i=1}^{p-1} \Gamma_i - A)\eta$ and $\alpha_1 = A\eta$ and in the conditional ARDL become $\alpha_{0,y}^{EC} = a_{yy}(\mu_y - \tilde{a}'_{y,x}\mu_x) + \gamma'_{y,x}(1)\eta$ and $\alpha_{1,y}^{EC} = a_{yy}(\eta_y - \tilde{a}'_{y,x}\eta_x)$. As explained in Appendix 2.7.2, intercept and trend appear in the error correction (EC) term of the ARDL equation only when restricted. Accordingly, they both do not appear in the EC in the case I, the intercept does not appear in the EC term in cases III, IV and V (it is freely set to $\alpha_{0,y}$) while the trend appears in the EC term only in the case IV (it is freely set to $\alpha_{1,y}$ for case V). Accordingly, when these terms are not restricted, they need to be supplied by the user.

The approach used to specify the function inputs offers great control to the user, in terms of generating specific (conditional) ARDL-based cointegration structures.

The function `sim_vecm_ardl` takes the following arguments:

- `nobs`: number of observations to generate;
- `case`: indicates the conditional ARDL specification in terms of deterministic component (intercept and trend) among the five specifications proposed by Pesaran et al. (2001), given in (17)-(21).
- `sigma.in`: covariance matrix, Σ , of the error term ε_t ;
- `gamma.in`: list of short-run parameter matrices Γ_j ;
- `axx.in`: cointegrating relationships, A_{xx} , pertaining the independent variables in the marginal VECM model;
- `ayx.uc.in`: vector of parameters, as in a_{yx} ;
- `ayy.in`: the a_{yy} term, weighting the EC term in the ARDL equation;
- `mu.in`: mean vector, μ , in the starting VAR specification, used to define the VECM intercept for CASE II;
- `eta.in`: trend vector, η , in the starting VAR specification, used to define the VECM trend for case IV;
- `azero.in`: unrestricted intercept of the VECM specification (valid only for cases III, IV and V), when the intercept is not involved in the EC term;
- `aone.in`: unrestricted coefficient of the trend in the VECM specification (valid only for case V), when the trend is not involved in the EC term;
- `burn.in`: additional observations burn-in observations to be generated. A total of `burn.in` + `nobs` observations are generated, but only the last `nobs` are kept in the data;
- `seed.in`: seed number for the generation of $\varepsilon_t \sim N(0, \Sigma)$.

If parameter values for `mu.in`, `eta.in`, `azero.in`, or `aone.in` and case number turn out to be in contradiction, an error message is displayed.

As output, the function gives out a list containing the data, both in level and first difference, along with all the parameter values given as input. Additionally, all intermediate transformation of parameters via VECM transformation or as a by-product of conditioning y_t on x_t are included in the output.

Figure 1 depicts three-time series, `dep_1_0`, `ind_1_0` and `ind_2_0`, generated using this function and affected by a cointegrating relationship, one panel for each case, from I to V. The variable `dep_1_0` represents the dependent variable y_t of the ARDL equation, while `ind_1_0` and `ind_2_0` the independent ones, x_{1t} and x_{2t} .

The code used to generate the data for case I is the following:

```
corrm = matrix(c( 0,      0, 0,
                 0.25,    0, 0,
                 0.4, -0.25, 0), nrow = 3, ncol = 3, byrow = T)

Corrm = (corrm + t(corrm)) + diag(3)

sds = diag(c(1.3, 1.2, 1))

sigma.in = (sds %%% Corrm %%% t(sds))

gamma1 = matrix(c(0.6,    0, 0.2,
                 0.1, -0.3, 0,
                 0, -0.3, 0.2), nrow = 3, ncol = 3, byrow=T)

gamma2= gamma1 * 0.3
```

```

omegat = sigma.in[1, -1] %*% solve(sigma.in[-1, -1])
axx.in = matrix(c( 0.3, 0.5,
                  -0.4, 0.3), nrow = 2, ncol = 2, byrow = T)
ayx.uc.in = c(0.4, 0.4)
ayy.in = 0.6

data.vecm.ardl_1 =
sim_vecm_ardl(nobs = 200,
              case = 1,
              sigma.in = sigma.in,
              gamma.in = list(gamma1, gamma2),
              axx.in = axx.in,
              ayx.uc.in = ayx.uc.in,
              ayy.in = ayy.in,
              mu.in = rep(0, 3),
              eta.in = rep(0, 3),
              azero.in = rep(0, 3),
              aone.in = rep(0, 3),
              burn.in = 100,
              seed.in = 999)

```

Additionally, Figure 2 displays other three time series, dep_1_0 (y_t), ind_1_0 (x_{1t}) and ind_2_0 (x_{2t}), when a degeneracy of second type occurs ($a_{yy} = 0$) in the long-run relationship in the ARDL equation of dep_1_0 on ind_1_0 , ind_2_0 . The five panels represents the behavior of these series in the Cases from I to V. It is worth noting the different scenario implied by these cases: case III depicts a trend for the y_t variable, case IV highlights the inclusion of a trend in the cointegrating relationship, and case V exhibits a quadratic trend in the y_t variable.

Finally, the flowchart in Figure 3 details the internal steps of the function `sim_vecm_ardl` and the data generation workflow. There, it is specified how the parameters of the VAR, VECM and ARDL equation are introduced. Attention is paid on whether the error correction mechanism involves either intercept or trend (or both) via the internal computation of the parameters θ_0 and θ_1 (and thus $\alpha_{0,y}^{EC}$ and $\alpha_{1,y}^{EC}$). When the EC term does not involve intercept and/or trend, α_0 and α_1 are supplied by the user, depending on the case under study.

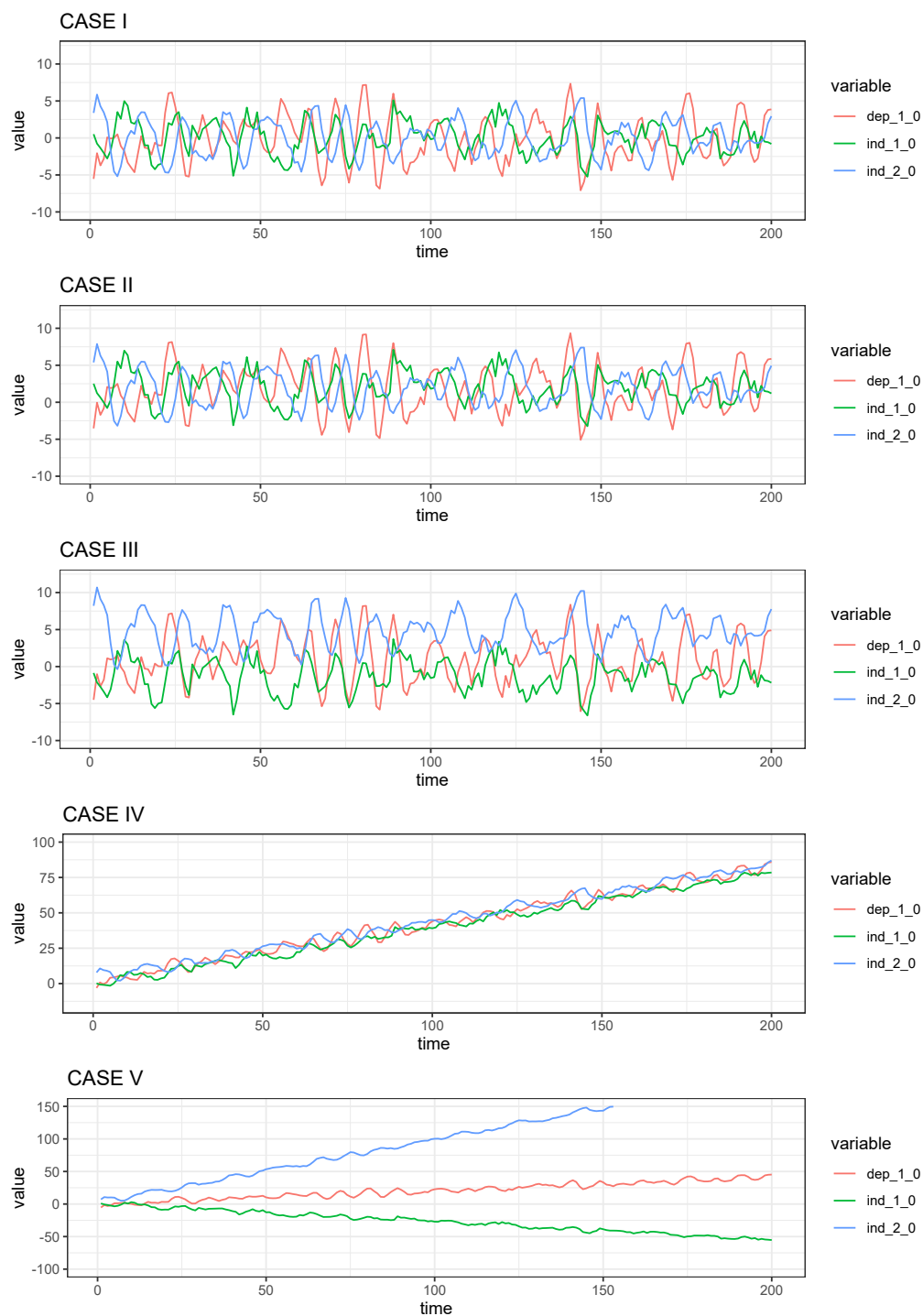


Figure 1: Simulated data from the VECM / conditional ARDL specifications, for every case. Made with [ggplot](#) (Wickham, 2016).

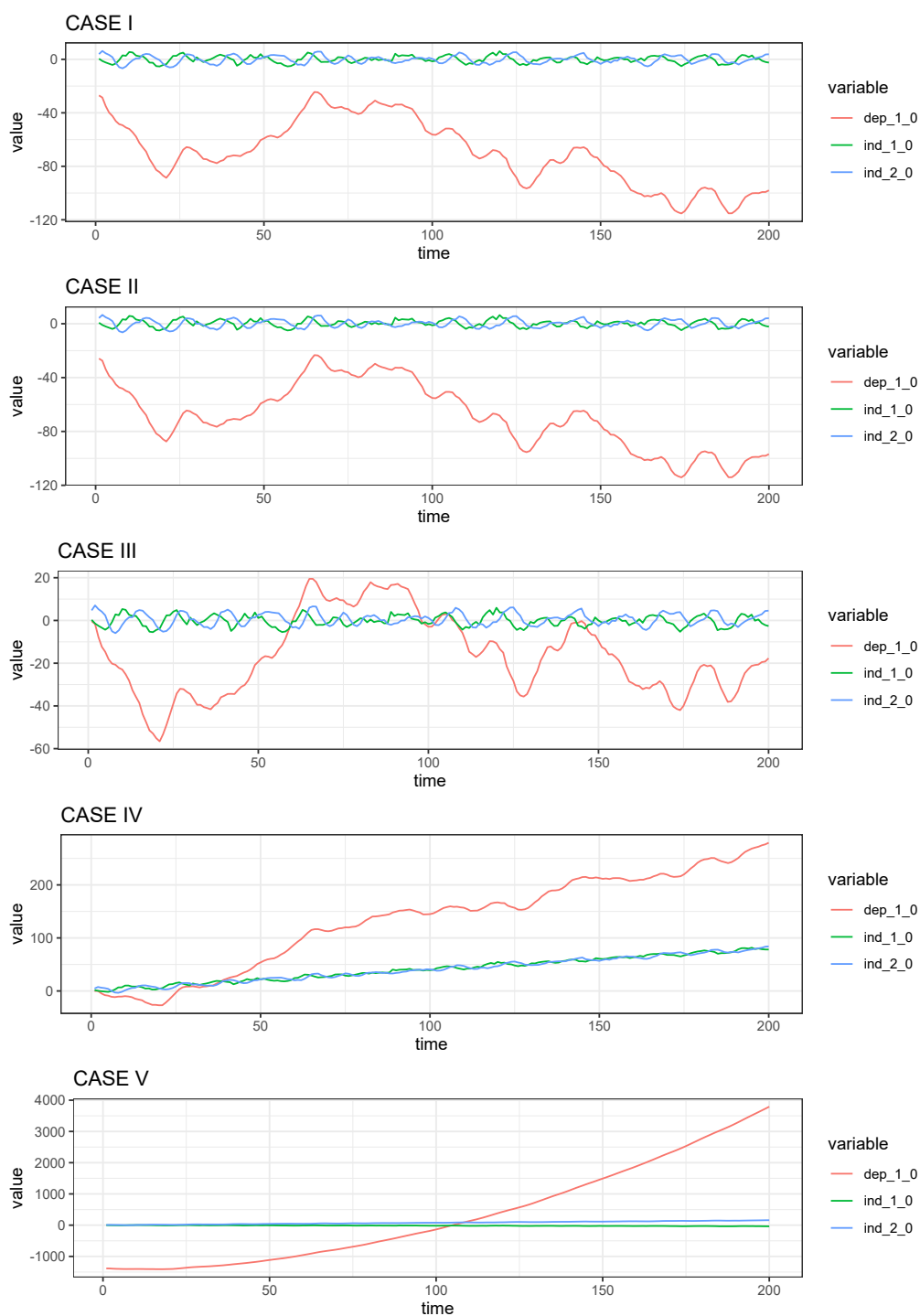


Figure 2: Simulated data from the VECM / conditional ARDL specifications (degenerate case of type 2, $a_{yy} = 0$), for every case. Made with `ggplot`.

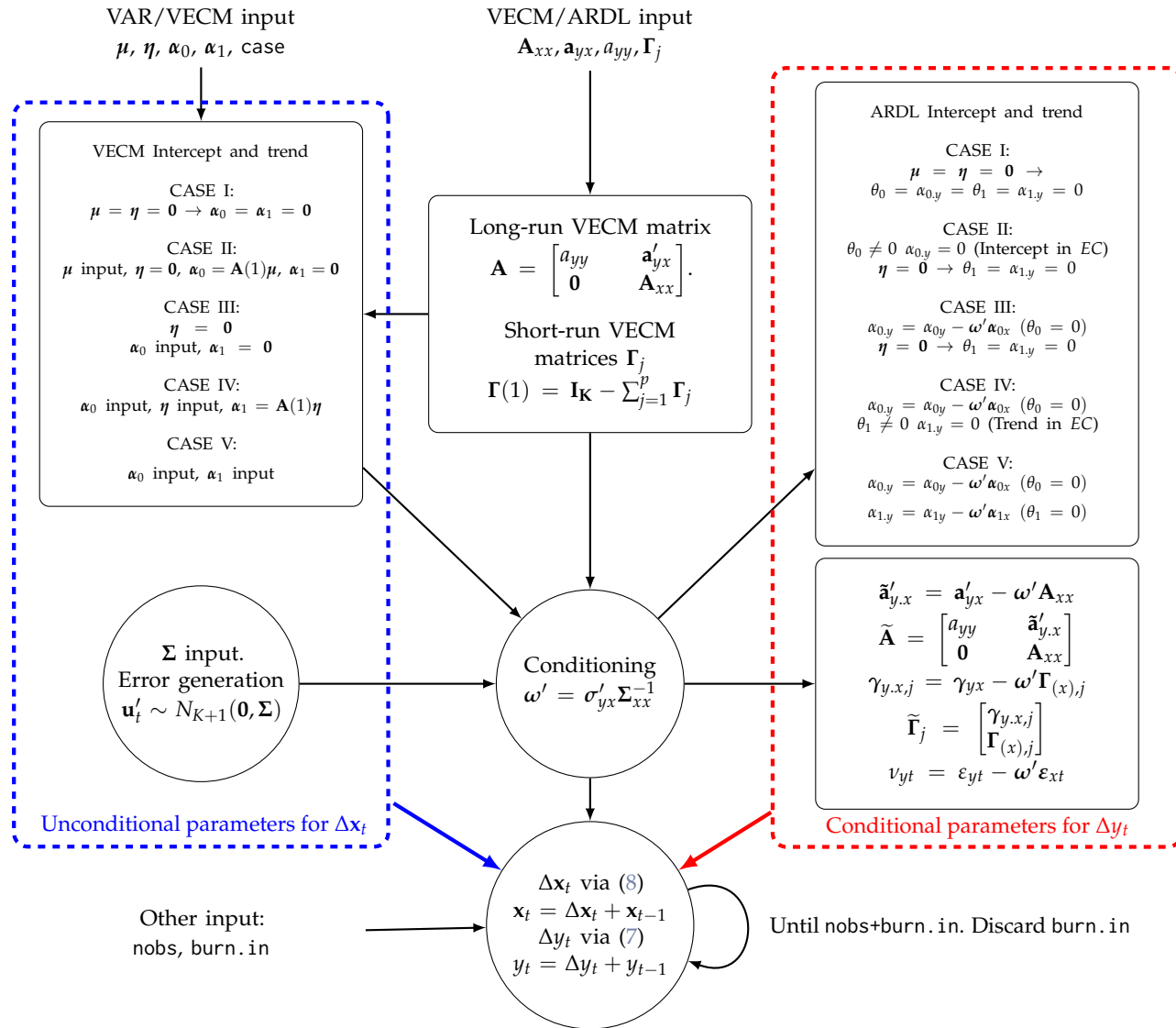


Figure 3: Flowchart of the `sim_vecm_ardl` function inner steps. When applying (7) and (8), $y_{t_j} = 0$, $\Delta y_{t_j} = 0$, $\mathbf{x}_{t_j} = \mathbf{0}$, $\Delta \mathbf{x}_{t_j} = \mathbf{0}$ for any $t_j < 1$. Boxes denote parameter definitions and transformations. Circles denote crucial actions, Empty nodes denote function inputs.

4.2 Bootstrapping the ARDL bound tests: the `boot_ardl` function

This function develops the bootstrap procedure detailed previously. As an option in the initial estimation phase, it offers the possibility of automatically choosing the best order for the lagged differences of all the variables in the ARDL and VECM models. This is done by using several criteria. In particular, AIC, BIC, AICc, R^2 and R^2_{adj} are used as lag selection criteria for the ARDL model, while the overall minimum between AIC, HQIC, SC and FPE is used for the lag selection for the VECM. In particular, the `auto_ardl` function in the package **ARDL** (Natsiopoulos and Tzeremes, 2021) selects the best ARDL order in terms of the short-run parameter vectors $\gamma_{y,x,j}$, while the `VARselect` function in the package **vars** (Pfaff, 2008a) selects the best VECM order in terms of the short-run parameter matrices $\Gamma_{(x),j}$. Furthermore, the user can input a significance threshold for the retention of single parameters in the Γ_j and in the $\gamma_{y,x,j}$ vectors.

The function `boot_ardl` takes the following arguments:

- `data`: input dataset. Must contain a dependent variable and a set of independent variables;
- `yvar`: name of the dependent variable enclosed in quotation marks. If unspecified, the first variable in the dataset is used;
- `xvar`: vector of names of the independent variables, each enclosed in quotation marks. If unspecified, all variables in the dataset except the first are used;
- `fix.ardl`: vector (j_1, \dots, j_K) , containing the maximum orders of the lagged differences (i.e., $\Delta y_{t-j_1}, \Delta x_{1,t-j_2}, \dots, \Delta x_{1,t-j_K}$) for the short term part of the ARDL equation, chosen in advance;
- `info.ardl`: (alternatively to `fix.ardl`) the information criterion used to choose the best lag order for the short term part of the ARDL equation. It must be one between AIC (default), AICc, BIC, R^2 , R^2_{adj} ;
- `fix.vecm`: scalar m containing the maximum order of the lagged differences (i.e., Δz_{t-m}) for the short term part of the VECM equation, chosen in advance;
- `info.vecm`: (alternatively to `fix.vecm`) the information criterion used to choose the best lag order for the short term part of the VECM equation. Must be one among AIC (default), HQIC, SC, FPE;
- `maxlag`: (in conjunction with `info.ardl` / `info.vecm`) maximum number of lags for the `auto_ardl` function in the package **ARDL**, and for the `VARselect` function in the package **vars**;
- `a.ardl`: significance threshold for the short-term ARDL coefficients ($\gamma_{y,x,j}$) in the ARDL model estimation;
- `a.vecm`: significance threshold for the short-term VECM coefficients (in Γ_j) in the VECM model estimation;
- `nboot`: number of bootstrap replications;
- `case`: type of the specification for the conditional ARDL in terms of deterministic components (intercept and trend) among the five proposed by Pesaran et al. (2001), given in (17)-(21);
- `a.boot.H0`: probability/ies α by which the critical quantiles of the bootstrap distribution(s) $c^*_{\alpha, F_{ov}}$, $c^*_{\alpha, t}$ and $c^*_{\alpha, F_{ind}}$ must be calculated;
- `print`: if set to TRUE, shows the progress bar.

`boot_ardl` makes use of the `lag_mts` function which produces lagged versions of a given matrix of time series, each column with a separate order. `lag_mts` takes as parameters the data included in a matrix X and the lag orders in a vector k , with the addition of a boolean parameter `last.only`, which allows to specify whether only the k -th order lags have to be retained, or all the lag orders from the first to the k -th.

`boot_ardl` also acts as a wrapper for the most common methodologies detecting cointegration, offering a comprehensive view on the testing procedures involved in the analysis. The resulting object, of class `bootCT`, contains all the information about

- The conditional ARDL model estimates, and the unconditional VECM model estimates;
- the bootstrap tests performed in the conditional ARDL model;
- the Pesaran, Shin and Smith bound testing procedure (F_{ov} and t -test, when applicable);
- the Sam, McNown and Goh bound testing procedure for F_{ind} , when applicable;
- the Johansen rank and trace cointegration tests on the independent variables.

Internally, the bootstrap data generation under the null is executed via a Rcpp function, employing the **Rcpp** and **RcppArmadillo** packages (Eddelbuettel, 2013), so as to greatly speed up computational times. As explained in the previous section, cointegration tests in the unconditional ARDL model are performed in order to uncover the presence of spurious cointegrating relationships.

To this end, the function provides

- the bootstrap critical values of the F_{ov} , t and F_{ind} tests in the conditional model, at level α , boot.H0 , along with the same statistics computed in the conditional model.
- a flag, called `fakecoint`, that indicates divergence between the outcomes of the F_{ind} test performed in both the conditional and unconditional model. In this circumstance, as explained before, there is no cointegration (see Bertelli et al., 2022).

A summary method has been implemented to present the results in a visually clear manner. It accepts the additional argument "out" that lets the user choose which output(s) to visualize: `ARDL` prints the conditional ARDL model summary, `VECM` prints the VECM model summary, `cointARDL` prints the summary of the bound tests and the bootstrap tests, `cointVECM` prints the summary of the Johansen test on the independent variables.

A detailed flowchart showing the function's workflow is displayed in Figure 4. There, the expressions "C ARDL" and "UC ARDL" stand for conditional and unconditional ARDL model, respectively.

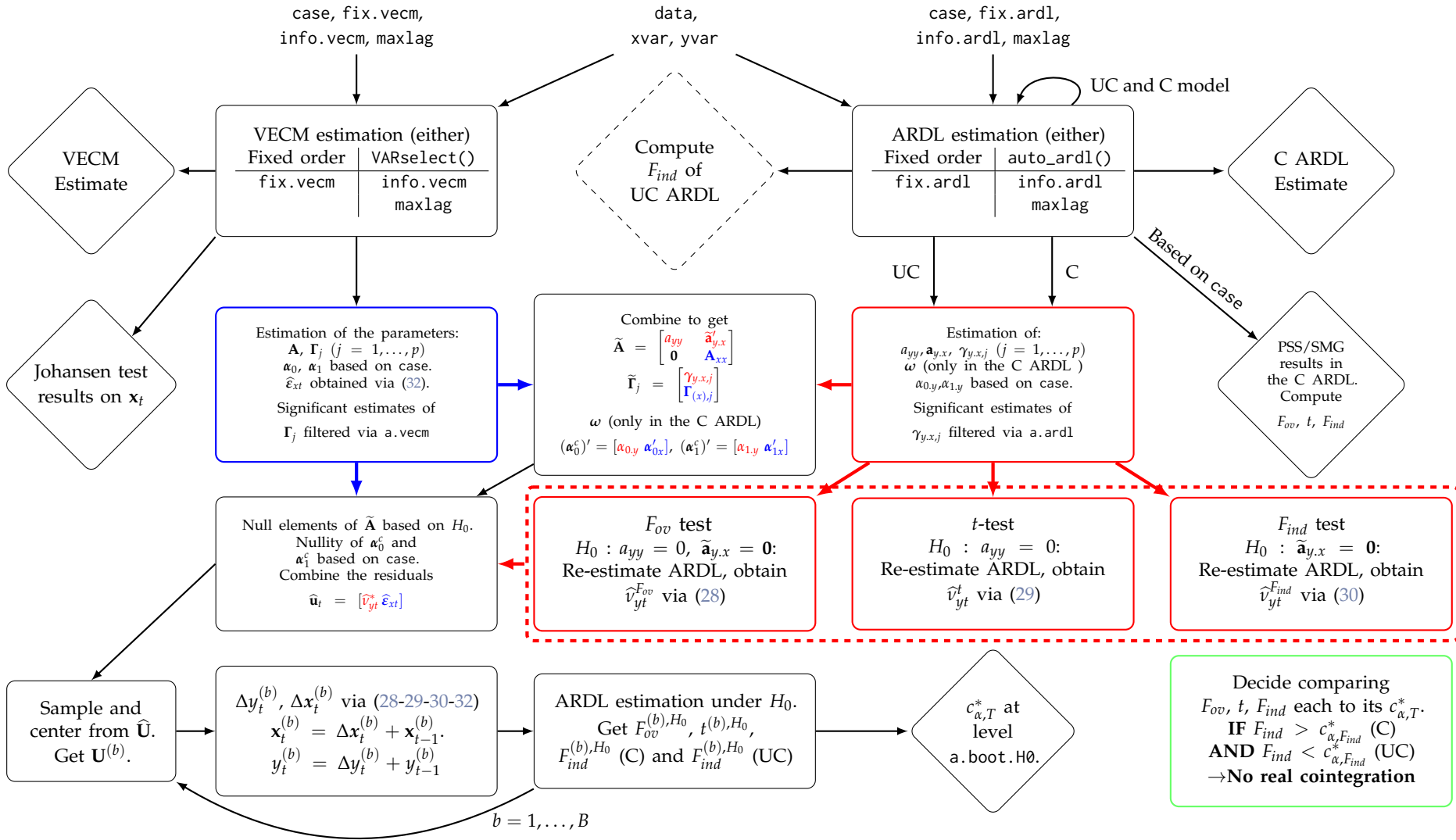


Figure 4: Flowchart of the `boot_ardl` function inner steps. Boxes denote parameter definitions and transformations. Diamonds denote function outputs. Dashed diamonds denote intermediate output (not shown after function call). Empty nodes denote function inputs. The first $p + 1$ rows of $\mathbf{z}_t^{(b)}$ are set equal to the first $p + 1$ rows of the original data. The best lag order for each difference variable in the ARDL model is determined via `auto_ardl()`. It is reported as a unique value p in $\gamma_{y,x,j}$ for brevity in the flowchart.

4.3 Execution time and technical remarks

In order to investigate the sensitivity of the procedure to different sample sizes and number of bootstrap replicates, an experiment has been run using a three-dimensional time series of length $T = \{50, 80, 100, 200, 500\}$, generating 100 datasets for each sample size with the `sim_vecm_ardl` function (Case II, with cointegrated variables, and 2 lags in the short-run section of the model). Then, the `boot_ardl` function has been called

```
boot_ardl(data = df_sim,
          nboot = bootr,
          case = 2,
          fix.ardl = rep(2, 3),
          fix.vecm = 2)
```

In the code above, `bootr` has been set equal to $B = \{200, 500, 1000, 2000\}$, the number of lags has been assumed known (`fix.ardl` and `fix.vecm`), while default values have been used for every other argument (such as `a.ardl`, `a.vecm` and `a.boot.H0`).

Table 1 shows the average running time per replication together with the coefficient of variation (%) of the bootstrap critical values of the F_{ov} test, for each value of T and B , across 100 replications for each scenario.

Naturally, the running time increases as both sample size and bootstrap replicates increase. However, it can be noticed how the coefficients of variation tend to stabilize for $B \geq 1000$, especially for $T > 80$, at the 5% significance level. Therefore, it is recommended a number of bootstrap replicates of at least $B = 1000$ for higher sample size, or at least $B = 2000$ for smaller samples. The analysis has been carried out using an Intel(R) Core(TM) i7-1165G7 CPU @ 2.80GHz processor, 16GB of RAM.

T	B	Exec. Time (sec)	$cv^{(F_{ov})}(5\%)$	$cv^{(F_{ov})}(2.5\%)$	$cv^{(F_{ov})}(1\%)$
50	200	23.38	8.648	10.925	13.392
50	500	48.37	6.312	6.952	8.640
50	1000	96.65	4.806	5.613	6.288
50	2000	231.15	4.255	4.226	4.946
80	200	23.46	7.251	8.936	11.263
80	500	50.19	4.998	6.220	7.946
80	1000	143.00	3.882	4.453	5.305
80	2000	255.64	2.912	3.623	4.518
100	200	37.89	7.707	8.583	10.955
100	500	52.86	4.691	5.304	7.557
100	1000	184.51	3.512	4.567	5.695
100	2000	212.65	3.519	3.674	4.185
200	200	35.46	6.644	7.173	10.365
200	500	76.78	4.734	5.355	6.225
200	1000	148.25	3.124	4.177	5.034
200	2000	484.51	2.811	3.361	3.907
500	200	54.47	6.641	8.694	10.414
500	500	133.17	5.137	5.816	6.408
500	1000	271.87	3.905	4.585	5.283
500	2000	561.71	3.221	3.490	4.145

Table 1: Average execution times (in seconds) of the `boot_ardl` function, for different combinations of sample size T and bootstrap replicates B . Coefficients of variation (cv) reported for the F_{ov} bootstrap critical values at level 5%, 2.5% and 1%.

5 Empirical applications

This section provides two illustrative application which highlight the performance of the bootstrap ARDL tests.

5.1 An application to the German macroeconomic dataset

In the first example, the occurrence of a long-run relationship between consumption [C], income [INC], and investment [INV] of Germany has been investigated via a set of ARDL models, where each variable takes in turn the role of dependent one, while the remaining are employed as independent. The models have been estimated by employing the dataset of [Lütkepohl \(2005\)](#) which includes quarterly data of the series over the years 1960 to 1982. The data have been employed in logarithmic form. Figure 5 displays these series over the sample period.

Before applying the bootstrap procedure, the order of integration of each series has been analyzed. Table 2 shows the results of ADF test performed on both the series and their first-differences ($k = 3$ maximum lags). The results confirm the applicability of the ARDL framework as no series is integrated of order higher than one.

The following ARDL equations have been estimated:

I First ARDL equation (C | INC, INV):

$$\begin{aligned} \Delta \log C_t = & \alpha_{0,y} - a_{yy} \log C_{t-1} - a_{y,x_1} \log INC_{t-1} - a_{y,x_2} \log INV_{t-1} + \\ & \sum_{j=1}^{p-1} \gamma_{y,j} \Delta \log C_{t-j} + \sum_{j=1}^{s-1} \gamma_{x_1,j} \Delta \log INC_{t-j} + \sum_{j=1}^{r-1} \gamma_{x_2,j} \Delta \log INV_{t-j} + \\ & \omega_1 \Delta \log INC_t + \omega_2 \Delta \log INV_t + v_t. \end{aligned} \quad (38)$$

II Second ARDL equation (INC | C, INV):

$$\begin{aligned} \Delta \log INC_t = & \alpha_{0,y} - a_{yy} \log INC_{t-1} - a_{y,x_1} \log C_{t-1} - a_{y,x_2} \log INV_{t-1} + \\ & \sum_{j=1}^{p-1} \gamma_{y,j} \Delta \log INC_{t-j} + \sum_{j=1}^{s-1} \gamma_{x_1,j} \Delta \log C_{t-j} + \sum_{j=1}^{r-1} \gamma_{x_2,j} \Delta \log INV_{t-j} + \\ & \omega_1 \Delta \log C_t + \omega_2 \Delta \log INV_t + v_t. \end{aligned} \quad (39)$$

III Third ARDL equation (INV | C, INC):

$$\begin{aligned} \Delta \log INV_t = & \alpha_{0,y} - a_{yy} \log INV_{t-1} - a_{y,x_1} \log C_{t-1} - a_{y,x_2} \log INC_{t-1} + \\ & \sum_{j=1}^{p-1} \gamma_{y,j} \Delta \log INV_{t-j} + \sum_{j=1}^{s-1} \gamma_{x_1,j} \Delta \log C_{t-j} + \sum_{j=1}^{r-1} \gamma_{x_2,j} \Delta \log INC_{t-j} + \\ & \omega_1 \Delta \log C_t + \omega_2 \Delta \log INC_t + v_t. \end{aligned} \quad (40)$$

Table 3 shows the estimation results for each ARDL and VECM model. It is worth noting that the instantaneous difference of the independent variables are highly significant in each conditional ARDL model. Thus, neglecting these variables in the ARDL equation, as happens in the unconditional version of the model, may potentially lead to biased estimates and incorrect inference. For the sake of completeness, also the results of the marginal VECM estimation are reported for each model.

The code to prepare the data, available in the package as the `ger_macro` dataset, is:

```
data("ger_macro")
LNDATA = apply(ger_macro[, -1], 2, log)
col_ln = paste0("LN", colnames(ger_macro)[-1])
LNDATA = as.data.frame(LNDATA)
colnames(LNDATA) = col_ln
```

Then, the `boot_ardl` function is called, to perform the bootstrap tests. In the code chunk below, Model I is considered.

```
set.seed(999)
BCT_res_CONS = 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)
```

to which follows the call to the summary function

```
summary(BCT_res_CONS, out = "ARDL")
summary(BCT_res_CONS, out = "VECM")
summary(BCT_res_CONS, out = "cointVECM")
summary(BCT_res_CONS, out = "cointARDL")
```

The first summary line displays the output in the ARDL column of Table 3 and the second column of Table 4, Model I. The second line corresponds to the VECM columns of Table 3, Model I - only for the independent variables. The information on the rank of the A_{xx} in Table 3 is inferred from the third line. Finally, the fourth summary line corresponds to the test results in Table 4, Model I. A textual indication of the presence of spurious cointegration is displayed at the bottom of the "cointARDL" summary, if detected.

In this example, the bootstrap and bound testing procedures are in agreement only for model I, indicating the existence of a cointegrating relationship. Additionally, no spurious cointegration is detected for this model. As for models II and III, the null hypothesis is not rejected by the bootstrap tests, while the PSS and SMG bound tests fail to give a conclusive answer in the F_{ind} test.

The running time of the entire analysis is of roughly 11 minutes, using an Intel(R) Core(TM) i7-1165G7 CPU @ 2.80GHz processor, 16GB of RAM.

Series	lag	level variable		first difference	
		ADF	p.value	ADF	p-value
$\log C_t$	0	-1.690	0.450	-9.750	< 0.01
	1	-1.860	0.385	-5.190	< 0.01
	2	-1.420	0.549	-3.130	0.030
	3	-1.010	0.691	-2.720	0.080
$\log INC_t$	0	-2.290	0.217	-11.140	< 0.01
	1	-1.960	0.345	-7.510	< 0.01
	2	-1.490	0.524	-5.120	< 0.01
	3	-1.310	0.587	-3.290	0.020
$\log INV_t$	0	-1.200	0.625	-8.390	< 0.01
	1	-1.370	0.565	-5.570	< 0.01
	2	-1.360	0.570	-3.300	0.020
	3	-1.220	0.619	-3.100	0.032

Table 2: ADF preliminary test (null hypothesis: random walk with drift).

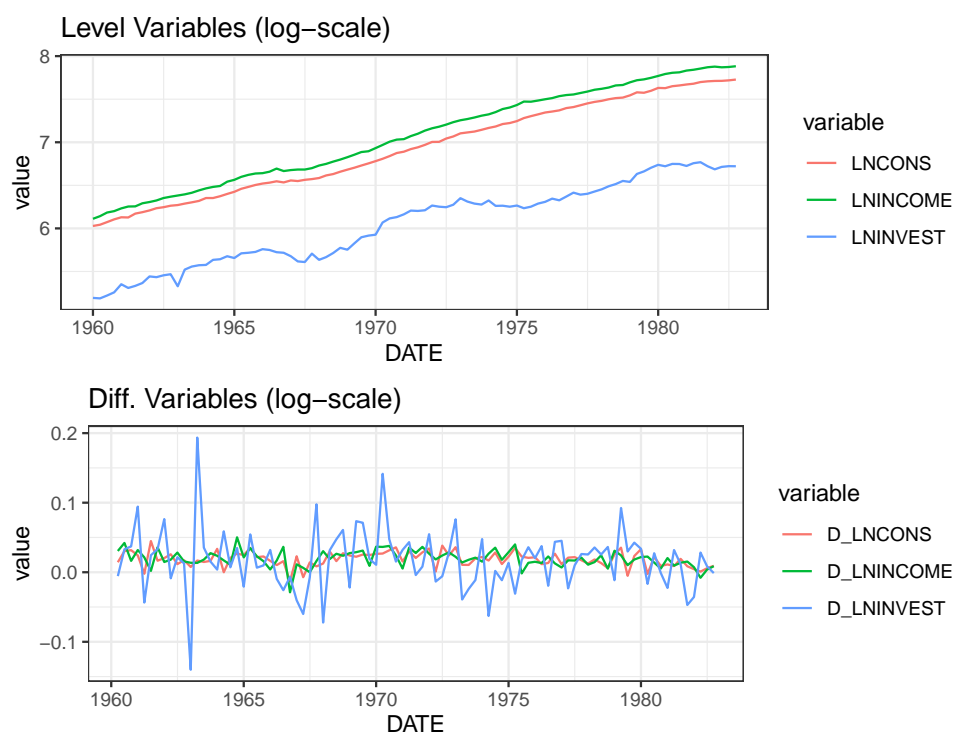


Figure 5: log-consumption/investment/income graphs (level variables and first differences). Made with [ggplot](#).

	Model I			Model II			Model III		
	ARDL $\Delta \log C_t$	VECM $\Delta \log INV_t$ $\Delta \log INC_t$		ARDL $\Delta \log INC_t$	$\Delta \log C_t$	VECM $\Delta \log INV_t$	ARDL $\Delta \log INV_t$	VECM $\Delta \log C_t$ $\Delta \log INC_t$	
$\log C_{t-1}$	-0.307 *** (0.055)			0.168 * (0.081)	-0.0011 (0.0126)	0.1286* (0.0540)	0.611 . (0.339)	-0.2727*** (0.0704)	-0.0508 (0.0796)
$\log INC_{t-1}$	0.297 *** (0.055)	0.124 * (0.054)	-0.017 (0.014)	-0.183* (0.079)			-0.491 (0.340)	0.2619*** (0.0681)	0.0464 (0.0772)
$\log INV_{t-1}$	-0.001 (0.011)	-0.152 * (0.063)	0.016 (0.017)	0.0209 (0.0135)	-0.00107 (0.0142)	-0.1531* (0.0607)	-0.1212* (0.060)		
$\Delta \log C_{t-1}$	-0.248 ** (0.079)	0.899 * (0.442)	0.211 . (0.113)	0.375*** (0.1086)		0.9288* (0.442)	1.113 * (0.441)		0.2072 . (0.1142)
$\Delta \log C_{t-2}$		0.744 (0.431)				0.8049 . (0.4345)			
$\Delta \log INC_{t-1}$				-0.1404 (0.1095)					
$\Delta \log INC_{t-2}$					0.2675** (0.0958)			0.1522. (0.0912)	
$\Delta \log INV_{t-1}$		-0.18 (0.111)	0.035 (0.029)			-0.189 . (0.1097)	-0.175 (0.1075)		0.0479 . (0.0282)
$\Delta \log INV_{t-2}$			0.049 . (0.027)		0.0591* (0.0245)			0.0578* (0.0223)	0.0562* (0.0266)
$\Delta \log C_t$				0.7070*** (0.1093)			1.8540*** (0.5425)		
$\Delta \log INC_t$	0.471*** (0.074)						-0.445*** (0.4726)		
$\Delta \log INV_t$	0.065** (0.019)			-0.0230 (0.025)					
const.	0.048 *** (0.013)	0.036 (0.066)	0.033 * (0.017)	0.002 (0.018)	0.0266 . (0.0155)	0.023 (0.0666)	-0.056 (0.072)	0.0517** (0.0157)	0.0378* (0.0177)
J-test	$rk(\mathbf{A}_{xx}) = 2$			$rk(\mathbf{A}_{xx}) = 2$			$rk(\mathbf{A}_{xx}) = 2$		

Table 3: Conditional ARDL and VECM results for the consumption/income/investment dataset, along with rank of the \mathbf{A}_{xx} matrix via the Johansen (J) test. Significance codes: (***) 1%; (**) 5%; (.) 10%.

Model	Lags	Test	Boot. Critical Values	PSS / SMG Threshold		Statistic	Outcome	
				I(0) 5%	I(1) 5%		Boot	Bound
I	(1,0,0)	F_{ov}	3.79	3.79	4.85	10.75	Y	Y
		t	-2.88	-2.86	-3.53	-5.608		
		F_{ind}	4.92	3.01	5.42	15.636		
II	(1,1,0)	F_{ov}	5.79	3.79	4.85	2.867	N	U
		t	-3.69	-2.86	-3.53	-2.315		
		F_{ind}	7.38	3.01	5.42	3.308		
III	(1,1,0)	F_{ov}	5.50	3.79	4.85	3.013	N	U
		t	-3.32	-2.86	-3.53	-2.020		
		F_{ind}	6.63	3.01	5.42	4.189		

Table 4: Cointegration analysis for the three ARDL equations in the German macroeconomic data. The optimal number of ARDL lags in the short-run - in the form (y, x_1, x_2) , matching the model definition - bootstrap critical values, bound test thresholds and test statistics for each test are shown (case III). The outcome columns draw conclusions on each type of model (bootstrap or bound): Y = cointegrated, N = not cointegrated, D1 = degenerate of type 1, D2 = degenerate of type 2, U = inconclusive inference.

5.2 An application on Italian Macroeconomic Data

Following Bertelli et al. (2022), the relationship between foreign direct investment [FDI], exports [EXP], and gross domestic product [GDP] in Italy is investigated. The data of these three yearly variables have been retrieved from the World Bank Database and cover the period from 1970 to 2020. In the analysis, the log of the variables has been used and [EXP] and [FDI] have been adjusted using the GDP deflator. Figure 6 displays these series over the sample period.

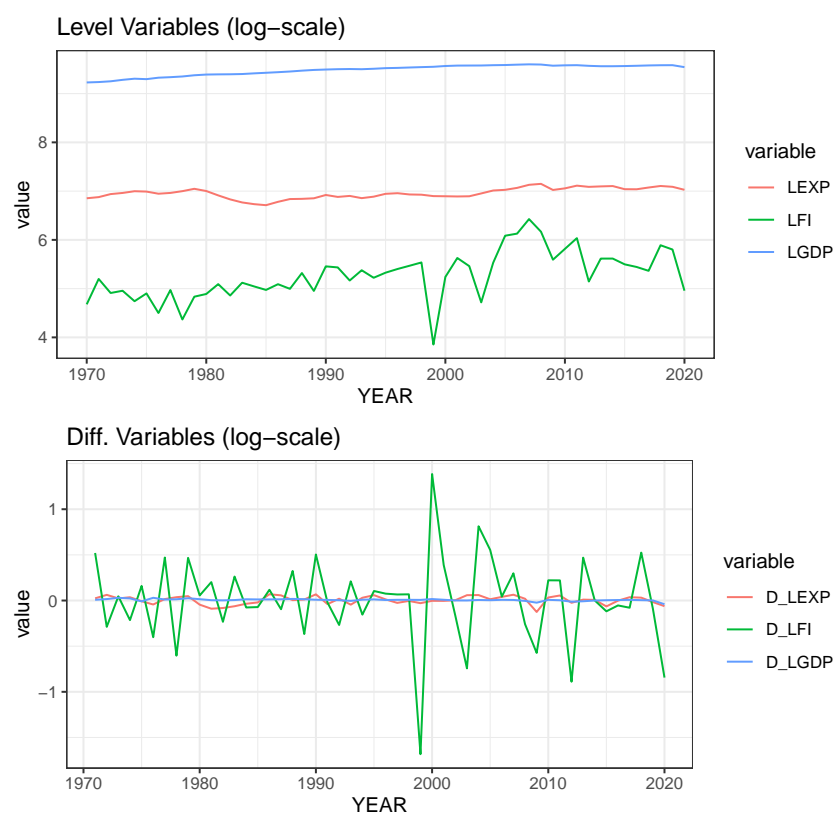


Figure 6: log-GDP/export/investment graphs (level variables and first differences). Made with ggplot.

Table 5 shows the outcomes of the ADF test performed on each variable, which ensures that the integration order is not higher than one for all variables. Table 6 shows the results of bound and bootstrap tests performed in ARDL model by taking each variable, in turn, as the dependent one. The following ARDL equations have been estimated:

I First ARDL equation (GDP | EXP, FDI):

$$\begin{aligned} \Delta \log \text{GDP}_t = & \alpha_{0,y} - a_{yy} \log \text{GDP}_{t-1} - a_{y,x_1} \log \text{EXP}_{t-1} - a_{y,x_2} \log \text{FDI}_{t-1} + \\ & \sum_{j=1}^{p-1} \gamma_{y,j} \Delta \log \text{GDP}_{t-j} + \sum_{j=1}^{s-1} \gamma_{x_1,j} \Delta \log \text{EXP}_{t-j} + \sum_{j=1}^{r-1} \gamma_{x_2,j} \Delta \log \text{FDI}_{t-j} + \\ & \omega_1 \Delta \log \text{EXP}_t + \omega_2 \Delta \log \text{FDI}_t + v_t \end{aligned} \quad (41)$$

. For this model, a degenerate case of the first type can be observed, while the simpler bound testing procedure does not signal cointegration.

II Second ARDL equation (EXP | GDP, FDI):

$$\begin{aligned} \Delta \log \text{EXP}_t = & \alpha_{0,y} - a_{yy} \log \text{EXP}_{t-1} - a_{y,x_1} \log \text{GDP}_{t-1} - a_{y,x_2} \log \text{FDI}_{t-1} + \\ & \sum_{j=1}^{p-1} \gamma_{y,j} \Delta \log \text{EXP}_{t-j} + \sum_{j=1}^{s-1} \gamma_{x_1,j} \Delta \log \text{GDP}_{t-j} + \sum_{j=1}^{r-1} \gamma_{x_2,j} \Delta \log \text{FDI}_{t-j} + \\ & \omega_1 \Delta \log \text{GDP}_t + \omega_2 \Delta \log \text{FDI}_t + v_t. \end{aligned} \quad (42)$$

For this model, the ARDL bootstrap test indicates absence of cointegration, while the bound testing approach is inconclusive for the F_{ind} test.

III Third ARDL equation (FDI | GDP, EXP):

$$\begin{aligned} \Delta \log \text{FDI}_t = & \alpha_{0,y} - a_{yy} \log \text{FDI}_{t-1} - a_{y,x_1} \log \text{GDP}_{t-1} - a_{y,x_2} \log \text{EXP}_{t-1} + \\ & \sum_{j=1}^{p-1} \gamma_{y,j} \Delta \log \text{FDI}_{t-j} + \sum_{j=1}^{s-1} \gamma_{x_1,j} \Delta \log \text{GDP}_{t-j} + \sum_{j=1}^{r-1} \gamma_{x_2,j} \Delta \log \text{EXP}_{t-j} + \\ & \omega_1 \Delta \log \text{GDP}_t + \omega_2 \Delta \log \text{EXP}_t + v_t. \end{aligned} \quad (43)$$

For this model, the long-run cointegrating relationship is confirmed using both bootstrap and bound testing. No spurious cointegration is detected.

The code to load the data and perform the analysis (e.g. for Model I) is:

```
data("ita_macro")
BCT_res_GDP = boot_ardl(data = ita_macro,
  yvar = "LGDP",
  xvar = c("LEXP", "LFI"),
  maxlag = 5,
  a.ardl = 0.1,
  a.vecm = 0.1,
  nboot = 2000,
  case = 3,
  a.boot.H0 = c(0.05),
  print = T)
```

For the sake of simplicity, the conditional ARDL and VECM marginal models outputs included in each cointegrating analysis is omitted. The summary for the cointegration tests for Model I is called via

```
summary(BCT_res_GDP, out = "ARDL") # extract lags
summary(BCT_res_GDP, out = "cointARDL") # ARDL cointegration
```

This empirical application further highlights the importance of dealing with inconclusive inference via the bootstrap procedure, while naturally including the effect of conditioning in the ARDL model, as highlighted in Bertelli et al. (2022).

6 Conclusion

The **bootCT** package allows the user to perform bootstrap cointegration tests in ARDL models by overcoming the problem of inconclusive inference which is a well-known drawback of standard bound tests. The package makes use of different functions. The function `boot_ardl` performs the bootstrap

Variable	No Drift, No Trend				Drift, No Trend				Drift and Trend			
	Lag = 0	Lag = 1	Lag = 2	Lag = 3	Lag = 0	Lag = 1	Lag = 2	Lag = 3	Lag = 0	Lag = 1	Lag = 2	Lag = 3
$\log GDP_t$	0.99	0.974	0.941	0.796	< 0.01	< 0.01	< 0.01	0.084	0.99	0.99	0.99	0.99
$\log FDI_t$	0.572	0.599	0.675	0.725	< 0.01	0.0759	0.3199	0.5174	< 0.01	0.013	0.151	0.46
$\log EXP_t$	0.787	0.71	0.698	0.684	0.479	0.288	0.467	0.433	0.629	0.35	0.463	0.379
$\Delta \log GDP_t$	< 0.01	< 0.0164	0.0429	0.0402	< 0.01	0.0861	0.3989	0.4267	< 0.01	< 0.01	0.0166	0.017
$\Delta \log FDI_t$	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01
$\Delta \log EXP_t$	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.0336	0.0315

Table 5: ADF preliminary test for the second example.

Model	Lags	Test	Boot. Critical Values	PSS / SMG Threshold		Statistic	Outcome	
				I(0) 5%	I(1) 5%		Boot	Bound
I	(1,1,0)	F_{ov}	3.730	4.070	5.190	9.758	D1	N
		t	-2.020	-2.860	-3.530	-2.338		
		F_{ind}	3.710	3.220	5.620	2.273		
II	(1,0,0)	F_{ov}	5.400	4.070	5.190	2.649	N	U
		t	-3.380	-2.860	-3.530	-1.889		
		F_{ind}	5.630	3.220	5.620	3.481		
III	(1,0,0)	F_{ov}	5.360	4.070	5.190	6.716	Y	Y
		t	-3.550	-2.860	-3.530	-4.202		
		F_{ind}	6.500	3.220	5.620	7.017		

Table 6: Cointegration analysis for the three ARDL equations in the Italian macroeconomic data. The optimal number of ARDL lags in the short-run - in the form (y, x_1, x_2) , matching the model definition - bootstrap critical values, bound test thresholds and test statistics for each test are shown (case III). The outcome columns draw conclusions on each type of model (bootstrap or bound): Y = cointegrated, N = not cointegrated, D1 = degenerate of type 1, D2 = degenerate of type 2, U = inconclusive inference.

tests, and it acts as a wrapper of both the bootstrap and the standard bound tests, including also the Johansen test on the independent variables of the model. Finally, it also performs the bound F -test on the lagged independent variables, so far not available in other extant R packages. The function `sim_vecm_ardl`, which allows the simulation of multivariate time series data following a user-defined DGP, enriches the available procedures for multivariate data generation, while the function `lag_mts` provides a supporting tool in building datasets of lagged variables for any practical purpose. Finally, the use of Rcpp functions gives a technical advantage in terms of computational speed, performing the bootstrap analysis within an acceptable time frame.

7 Appendix

7.1 Section A - the methodological framework of (conditional) VECM and ARDL models

Expanding the matrix polynomial $\mathbf{A}(z)$ about $z = 1$, yields

$$\mathbf{A}(z) = \mathbf{A}(1)z + (1 - z)\mathbf{\Gamma}(z), \quad (44)$$

where

$$\mathbf{A}(1) = \mathbf{I}_{K+1} - \sum_{j=1}^p \mathbf{A}_j \quad (45)$$

$$\mathbf{\Gamma}(z) = \mathbf{I}_{K+1} - \sum_{i=1}^{p-1} \mathbf{\Gamma}_i z^i, \quad \mathbf{\Gamma}_i = - \sum_{j=i+1}^p \mathbf{A}_j. \quad (46)$$

The VECM model (2) follows accordingly, and

$$\alpha_0 = \mathbf{A}(1)\mu + (\mathbf{\Gamma}(1) - \mathbf{A}(1))\eta, \quad \alpha_1 = \mathbf{A}(1)\eta. \quad (47)$$

Assuming that $\mathbf{A}(1)$ is singular and that the variables \mathbf{x}_t are cointegrated. This entails the following

$$\begin{aligned} \mathbf{A}(1) &= \begin{bmatrix} a_{yy} & \mathbf{a}'_{yx} \\ (1,1) & (1,K) \\ \mathbf{a}_{xy} & \mathbf{A}_{xx} \\ (K,1) & (K,K) \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{C}' \\ (K+1,r+1) & (r+1,K+1) \end{bmatrix} = \begin{bmatrix} b_{yy} & \mathbf{b}'_{yx} \\ \mathbf{b}_{xy} & \mathbf{B}_{xx} \end{bmatrix} \begin{bmatrix} c_{yy} & \mathbf{c}'_{yx} \\ \mathbf{c}_{xy} & \mathbf{C}_{xx} \end{bmatrix} = \\ &= \begin{bmatrix} b_{yy}c_{yy} + \mathbf{b}'_{yx}\mathbf{c}_{xy} & b_{yy}\mathbf{c}'_{yx} + \mathbf{b}'_{yx}\mathbf{C}_{xx} \\ \mathbf{b}_{xy}c_{yy} + \mathbf{B}_{xx}\mathbf{c}_{xy} & \mathbf{b}_{xy}\mathbf{c}'_{yx} + \mathbf{B}_{xx}\mathbf{C}_{xx} \end{bmatrix}, \quad rk(\mathbf{A}(1)) = rk(\mathbf{B}) = rk(\mathbf{C}), \end{aligned} \quad (48)$$

where \mathbf{B} and \mathbf{C} are full column rank matrices arising from the rank-factorization of $\mathbf{A}(1) = \mathbf{B}\mathbf{C}'$ with \mathbf{C} matrix of the long-run relationships of the process and \mathbf{B}_{xx} , \mathbf{C}_{xx} arising from the rank factorization of $\mathbf{A}_{xx} = \mathbf{B}_{xx}\mathbf{C}'_{xx}$, with $rk(\mathbf{A}_{xx}) = rk(\mathbf{B}_{xx}) = rk(\mathbf{C}_{xx}) = r$ ⁶.

By partitioning the vectors α_0 , α_1 , the matrix $\mathbf{A}(1)$ and the polynomial matrix $\mathbf{\Gamma}(L)$ conformably to \mathbf{z}_t , as follows

$$\alpha_0 = \begin{bmatrix} \alpha_{0y} \\ (1,1) \\ \alpha_{0x} \\ (K,1) \end{bmatrix}, \quad \alpha_1 = \begin{bmatrix} \alpha_{1y} \\ (1,1) \\ \alpha_{1x} \\ (K,1) \end{bmatrix} \quad (49)$$

$$\mathbf{A}(1) = \begin{bmatrix} \mathbf{a}'_{(y)} \\ (1,K+1) \\ \mathbf{A}_{(x)} \\ (K,K+1) \end{bmatrix} = \begin{bmatrix} a_{yy} & \mathbf{a}'_{yx} \\ (1,1) & (1,K) \\ \mathbf{a}_{xy} & \mathbf{A}_{xx} \\ (K,1) & (K,K) \end{bmatrix}, \quad \mathbf{\Gamma}(L) = \begin{bmatrix} \gamma'_y(L) \\ (1,K+1) \\ \mathbf{\Gamma}_{(x)}(L) \\ (K,K+1) \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(L) & \gamma'_{yx}(L) \\ (1,1) & (1,K) \\ \gamma_{xy}(L) & \mathbf{\Gamma}_{xx}(L) \\ (K,1) & (K,K) \end{bmatrix} \quad (50)$$

, and substituting (6) into (2) yields

$$\Delta \mathbf{z}_t = \begin{bmatrix} \Delta y_t \\ \Delta \mathbf{x}_t \end{bmatrix} = \begin{bmatrix} \alpha_{0,y} \\ \alpha_{0,x} \end{bmatrix} + \begin{bmatrix} \alpha_{1,y} \\ \alpha_{1,x} \end{bmatrix} t - \begin{bmatrix} \mathbf{a}'_{(y).x} \\ \mathbf{A}_{(x)} \end{bmatrix} \begin{bmatrix} y_{t-1} \\ \mathbf{x}_{t-1} \end{bmatrix} + \begin{bmatrix} \gamma'_{y.x}(L) \\ \mathbf{\Gamma}_{(x)}(L) \end{bmatrix} \Delta \mathbf{z}_t + \begin{bmatrix} \omega' \Delta \mathbf{x}_t \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} v_{yt} \\ \varepsilon_{xt} \end{bmatrix} \quad (51)$$

, where

$$\alpha_{0,y} = \alpha_{0y} - \omega' \alpha_{0x}, \quad \alpha_{1,y} = \alpha_{1y} - \omega' \alpha_{1x} \quad (52)$$

$$\mathbf{a}'_{(y).x} = \mathbf{a}'_{(y)} - \omega' \mathbf{A}_{(x)}, \quad \gamma'_{y.x}(L) = \gamma'_y(L) - \omega' \mathbf{\Gamma}_{(x)}(L). \quad (53)$$

According to (51), the long-run relationships of the VECM turn out to be now included in the matrix

$$\begin{bmatrix} \mathbf{a}'_{(y).x} \\ \mathbf{A}_{(x)} \end{bmatrix} = \begin{bmatrix} a_{yy} - \omega' \mathbf{a}_{xy} & \mathbf{a}'_{yx} - \omega' \mathbf{A}_{xx} \\ \mathbf{a}_{xy} & \mathbf{A}_{xx} \end{bmatrix}. \quad (54)$$

To rule out the presence of long-run relationships between y_t and \mathbf{x}_t in the marginal model, the \mathbf{x}_t variables are assumed to be exogenous with respect to the ARDL parameters, that is \mathbf{a}_{xy} is assumed to be a null vector. Accordingly, the long-run matrix in (54) becomes

$$\tilde{\mathbf{A}} = \begin{bmatrix} a_{yy} & \mathbf{a}'_{yx} - \omega' \mathbf{A}_{xx} \\ \mathbf{0} & \mathbf{A}_{xx} \end{bmatrix} = \begin{bmatrix} a_{yy} & \tilde{\mathbf{a}}'_{y.x} \\ \mathbf{0} & \mathbf{A}_{xx} \end{bmatrix} = \begin{bmatrix} b_{yy}c_{yy} & b_{yy}\mathbf{c}'_{yx} + (\mathbf{b}'_{yx} - \omega' \mathbf{B}_{xx})\mathbf{C}'_{xx} \\ \mathbf{0} & \mathbf{B}_{xx}\mathbf{C}'_{xx} \end{bmatrix}. \quad (55)$$

⁶If the explanatory variables are stationary \mathbf{A}_{xx} is non-singular ($rk(\mathbf{A}_{xx}) = K$), while when they are integrated but without cointegrating relationship \mathbf{A}_{xx} is a null matrix

After these algebraic transformations, the ARDL equation for Δy_t can be rewritten as in (7).

In light of the factorization (48) of the matrix $\mathbf{A}(1)$, the long-run equilibrium vector θ can be expressed as

$$\theta' = -\frac{1}{a_{yy}} \begin{bmatrix} b_{yy} & (\mathbf{b}_{yx} - \omega' \mathbf{B}_{xx}) \end{bmatrix} \begin{bmatrix} \mathbf{c}'_{yx} \\ \mathbf{c}'_{xx} \end{bmatrix}, \quad (56)$$

(1,r+1) (r+1,K)

where $\tilde{\mathbf{a}}_{y,x} = \mathbf{a}_{yx} - \omega' \mathbf{A}_{xx}$.

Bearing in mind that \mathbf{C}'_{xx} is the cointegrating matrix for the variables \mathbf{x}_t , the equation (56) leads to the following conclusion

$$rk \begin{bmatrix} \mathbf{c}'_{yx} \\ \mathbf{c}'_{xx} \end{bmatrix} = \begin{cases} r \rightarrow y_t \sim I(0) \\ r+1 \rightarrow y_t \sim I(1) \end{cases}, \quad (57)$$

where $r = rk(\mathbf{A}_{xx})$ and $0 \leq r \leq K$.

7.2 Section B - Intercept and trend specifications

Pesaran et al. (2001) introduced five different specifications for the ARDL model, which depend on the deterministic components that can be absent or restricted to the values they assume in the parent VAR model. In this connection, note that, in light of (47), the drift and the trend coefficient in the conditional VECM (51) are defined as

$$\alpha_0^c = \tilde{\mathbf{A}}(1)(\mu - \eta) + \tilde{\mathbf{\Gamma}}(1)\eta, \quad \alpha_1^c = \tilde{\mathbf{A}}(1)\eta, \quad (58)$$

where $\tilde{\mathbf{A}}(1)$ is as in (55) and $\tilde{\mathbf{\Gamma}}(1) = \begin{bmatrix} \gamma'_{y,x}(1) \\ \mathbf{\Gamma}'_{(x)}(1) \end{bmatrix}$.

Accordingly, after partitioning the mean and the drift vectors as

$$\begin{matrix} \mu' \\ (1,K+1) \end{matrix} = \begin{matrix} [\mu_y, \mu_x'] \\ (1,1) (1,K) \end{matrix}, \quad \begin{matrix} \eta' \\ (1,K+1) \end{matrix} = \begin{matrix} [\eta_y, \eta_x'] \\ (1,1) (1,K) \end{matrix}, \quad (59)$$

the intercept and the coefficient of the trend of the ARDL equation (7) are defined as

$$\alpha_{0,y}^{EC} = \mathbf{e}_1' \alpha_0^c = a_{yy} \mu_y - \tilde{\mathbf{a}}'_{y,x} \mu_x + \gamma'_{y,x}(1) \eta = a_{yy}(\mu_y - \theta' \mu_x) + \gamma'_{y,x}(1) \eta, \quad \theta' = -\frac{\tilde{\mathbf{a}}'_{y,x}}{a_{yy}} \quad (60)$$

$$\alpha_{1,y}^{EC} = \mathbf{e}_1' \alpha_1^c = a_{yy} \eta_y - \tilde{\mathbf{a}}'_{y,x} \eta_x = a_{yy}(\eta_y - \theta' \eta_x), \quad (61)$$

where \mathbf{e}_1 is the $K+1$ first elementary vector.

In the error correction term

$$EC_{t-1} = y_{t-1} - \theta_0 - \theta_1 t - \theta' \mathbf{x}_{t-1} \quad (62)$$

the parameters that partake in the calculation of intercept and trend are

$$\theta_0 = \mu_y - \theta' \mu_x, \quad \theta_1 = \eta_y - \theta' \eta_x. \quad (63)$$

In particular, these latter are not null only when they are assumed to be restricted in the model specification.

The five specifications proposed by Pesaran et al. (2001) are

I *No intercept and no trend:*

$$\mu = \eta = \mathbf{0}. \quad (64)$$

It follows that

$$\theta_0 = \theta_1 = \alpha_{0,y} = \alpha_{1,y} = 0. \quad (65)$$

Accordingly, the model is as in (17).

II *Restricted intercept and no trend:*

$$\alpha_0^c = \tilde{\mathbf{A}}(1)\mu, \quad \eta = \mathbf{0}, \quad (66)$$

which entails

$$\theta_0 \neq 0 \quad \alpha_{0,y}^{EC} = a_{yy} \theta_0, \quad \alpha_{0,y} = \theta_1 = \alpha_{1,y} = 0. \quad (67)$$

Therefore, the intercept stems from the EC term of the ARDL equation. The model is specified as in (18)

III *Unrestricted intercept and no trend:*

$$\alpha_0^c \neq \tilde{\mathbf{A}}(1)\mu, \quad \eta = \mathbf{0}. \quad (68)$$

Thus,

$$\alpha_{0.y} \neq 0, \quad \theta_0 = \theta_1 = \alpha_{1.y} = 0. \quad (69)$$

Accordingly, the model is as in (19).

IV *Unrestricted intercept, restricted trend:*

$$\alpha_0^c \neq \tilde{\mathbf{A}}(1)(\mu - \eta) + \tilde{\Gamma}(1)\eta \quad \alpha_1^c = \tilde{\mathbf{A}}(1)\eta, \quad (70)$$

which entails

$$\alpha_{0.y} \neq 0, \quad \theta_0 = 0 \quad \theta_1 \neq 0 \quad \alpha_{1.y}^{EC} = a_{yy}\theta_1 \quad \alpha_{1.y} = 0. \quad (71)$$

Accordingly, the trend stems from the EC term of the ARDL equation. The model is as in (20).

V *Unrestricted intercept, unrestricted trend:*

$$\alpha_0^c \neq \tilde{\mathbf{A}}(1)(\mu - \eta) + \tilde{\Gamma}(1)\eta \quad \alpha_1^c \neq \tilde{\mathbf{A}}(1)\eta. \quad (72)$$

Accordingly,

$$\alpha_{0.y} \neq 0 \quad \alpha_{1.y} \neq 0, \quad \theta_0 = \theta_1 = 0. \quad (73)$$

The model is as in (21).

References

- K. R. Abbasi, M. Shahbaz, Z. Jiao, and M. Tufail. How energy consumption, industrial growth, urbanization, and co2 emissions affect economic growth in pakistan? a novel dynamic ardl simulations approach. *Energy*, 221:119793, 2021. doi: 10.1016/j.energy.2021.119793. [p39]
- M. A. Arranz and A. Escribano. Cointegration testing under structural breaks: A robust extended error correction model. *Oxford Bulletin of Economics and Statistics*, 62(1):23–52, 2000. doi: 10.1111/1468-0084.00158. [p39]
- S. M. Bache and H. Wickham. *magrittr: A Forward-Pipe Operator for R*, 2022. URL <https://CRAN.R-project.org/package=magrittr>. R package version 2.0.3. [p40]
- S. Bertelli, G. Vacca, and M. Zoia. Bootstrap cointegration tests in ardl models. *Economic Modelling*, 116:105987, 2022. doi: 10.1016/j.econmod.2022.105987. [p40, 42, 51, 58, 59]
- B. Bolker, G. R. Warnes, and T. Lumley. *gtools: Various R Programming Tools*, 2022. URL <https://CRAN.R-project.org/package=gtools>. R package version 3.9.4. [p40]
- H. W. Borchers. *pracma: Practical Numerical Math Functions*, 2022. URL <https://CRAN.R-project.org/package=pracma>. R package version 2.4.2. [p40]
- S. Cook. The power of single equation tests for cointegration. *Applied Economics Letters*, 13(5):265–267, 2006. doi: 10.1080/13504850500398534. [p39]
- R. Davidson and J. G. MacKinnon. The case against jive. *Journal of Applied Econometrics*, 21(6):827–833, 2005. doi: 10.1002/jae.873. [p44]
- D. Eddelbuettel. *Seamless R and C++ Integration with Rcpp*. Springer, New York, 2013. doi: 10.1007/978-1-4614-6868-4. ISBN 978-1-4614-6867-7. [p40, 50]
- D. Eddelbuettel, R. Francois, D. Bates, B. Ni, and C. Sanderson. *RcppArmadillo: ‘Rcpp’ Integration for the ‘Armadillo’ Templated Linear Algebra Library*, 2023. URL <https://CRAN.R-project.org/package=RcppArmadillo>. R package version 0.12.4.0.0. [p40]
- R. F. Engle and C. W. Granger. Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276, 1987. doi: 10.2307/1913236. [p39]
- R. F. Engle and B. S. Yoo. Forecasting and testing in co-integrated systems. *Journal of Econometrics*, 35(1):143–159, 1987. doi: 10.1016/0304-4076(87)90085-6. [p39]
- N. R. Ericsson and J. G. MacKinnon. Distributions of error correction tests for cointegration. *The Econometrics Journal*, 5(2):285–318, 2002. doi: 10.1111/1368-423X.00085. [p39]
- V. J. Gabriel, Z. Psaradakis, and M. Sola. A simple method of testing for cointegration subject to multiple regime changes. *Economics Letters*, 76(2):213–221, 2002. [p39]
- M. Haseeb, I. S. Z. Abidin, Q. M. A. Hye, and N. H. Hartani. The impact of renewable energy on economic well-being of malaysia: Fresh evidence from auto regressive distributed lag bound testing approach. *International Journal of Energy Economics and Policy*, 9(1):269, 2019. doi: 10.32479/ijeep.7229. [p39]
- R. M. Hope. *Rmisc: Ryan Miscellaneous*, 2022. URL <https://CRAN.R-project.org/package=Rmisc>. R package version 1.5.1. [p40]
- H. I. Hussain, M. A. Salem, A. Z. A. Rashid, and F. Kamarudin. Environmental impact of sectoral energy consumption on economic growth in malaysia: Evidence from ardl bound testing approach. *Ekoloji Dergisi*, (107), 2019. [p39]
- S. Johansen. Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models. *Econometrica: journal of the Econometric Society*, pages 1551–1580, 1991. doi: 10.2307/2938278. [p39]
- S. Jordan and A. Q. Philips. *dynamac: Dynamic Simulation and Testing for Single-Equation ARDL Models*, 2020. URL <https://CRAN.R-project.org/package=dynamac>. R package version 0.1.11. [p40]
- A. Kanioura and P. Turner. Critical values for an f-test for cointegration in a multivariate model. *Applied Economics*, 37(3):265–270, 2005. doi: 10.1080/00036840412331315051. [p40]

- J. J. Kremers, N. R. Ericsson, and J. J. Dolado. The power of cointegration tests. *Oxford bulletin of economics and statistics*, 54(3):325–348, 1992. doi: 10.1111/j.1468-0084.1992.tb00005.x. [p39]
- S. Kripfganz and D. C. Schneider. Response surface regressions for critical value bounds and approximate p-values in equilibrium correction models 1. *Oxford Bulletin of Economics and Statistics*, 82(6): 1456–1481, 2020. doi: 10.1111/obes.12377. [p40]
- Lesnoff, M., Lancelot, and R. aod: *Analysis of Overdispersed Data*, 2012. URL <https://cran.r-project.org/package=aod>. R package version 1.3.2. [p40]
- H. Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005. doi: 10.1007/978-3-540-27752-1. [p54]
- J. G. Mackinnon. Critical values for cointegration tests. In *Eds.), Long-Run Economic Relationship: Readings in Cointegration*. Oxford Press, 1991. [p39]
- G. S. Maddala and I.-M. Kim. Unit roots, cointegration, and structural change. 1998. doi: 10.1017/CBO9780511751974. [p39]
- R. McNown, C. Y. Sam, and S. K. Goh. Bootstrapping the autoregressive distributed lag test for cointegration. *Applied Economics*, 50(13):1509–1521, 2018. doi: 10.1080/00036846.2017.1366643. [p39, 40, 43]
- A. N. Menegaki. The ardl method in the energy-growth nexus field; best implementation strategies. *Economies*, 7(4):105, 2019. doi: 10.3390/economies7040105. [p39]
- T. C. Mills and E. J. Pentecost. The real exchange rate and the output response in four eu accession countries. *Emerging Markets Review*, 2(4):418–430, 2001. doi: 10.1016/S1566-0141(01)00027-9. [p40]
- P. K. Narayan. The saving and investment nexus for china: evidence from cointegration tests. *Applied economics*, 37(17):1979–1990, 2005. doi: 10.1080/00036840500278103. [p40]
- P. K. Narayan and R. Smyth. Crime rates, male youth unemployment and real income in australia: evidence from granger causality tests. *Applied Economics*, 36(18):2079–2095, 2004. doi: 10.1080/0003684042000261842. [p40]
- K. Natsiopoulou and N. Tzeremes. *ARDL: ARDL, ECM and Bounds-Test for Cointegration*, 2021. URL <https://CRAN.R-project.org/package=ARDL>. R package version 0.1.1. [p40, 50]
- M. H. Pesaran, Y. Shin, and R. J. Smith. Bounds testing approaches to the analysis of level relationships. *Journal of applied econometrics*, 16(3):289–326, 2001. doi: 10.1002/jae.616. [p39, 40, 41, 42, 45, 50, 62]
- B. Pfaff. Var, svar and svec models: Implementation within R package vars. *Journal of Statistical Software*, 27(4), 2008a. URL <https://www.jstatsoft.org/v27/i04/>. [p40, 50]
- B. Pfaff. *Analysis of Integrated and Cointegrated Time Series with R*. Springer, New York, second edition, 2008b. URL <https://www.pfaffikus.de>. ISBN 0-387-27960-1. [p40]
- D. Qiu. *aTSA: Alternative Time Series Analysis*, 2015. URL <https://CRAN.R-project.org/package=aTSA>. R package version 3.1.2. [p40]
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022. URL <https://www.R-project.org/>. [p]
- A. M. Reda and E. Nourhan. Using the ardl bound testing approach to study the inflation rate in egypt. *Economic consultant*, (3 (31)):24–41, 2020. doi: 10.46224/ecoc.2020.3.2. [p39]
- C. Y. Sam, R. McNown, and S. K. Goh. An augmented autoregressive distributed lag bounds test for cointegration. *Economic Modelling*, 80:130–141, 2019. doi: 10.1016/j.econmod.2018.11.001. [p40]
- A. Trapletti and K. Hornik. *tseries: Time Series Analysis and Computational Finance*, 2023. URL <https://CRAN.R-project.org/package=tseries>. R package version 0.10-54. [p40]
- G. Vacca and S. Bertelli. *bootCT: Bootstrapping the ARDL Tests for Cointegration*, 2023. R package version 2.0.0. [p40]
- H. Wickham. Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12):1–20, 2007. URL <http://www.jstatsoft.org/v21/i12/>. [p40]
- H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL <https://ggplot2.tidyverse.org>. [p40, 47]

- H. Wickham. *stringr: Simple, Consistent Wrappers for Common String Operations*, 2022. URL <https://CRAN.R-project.org/package=stringr>. R package version 1.5.0. [p40]
- H. Wickham, M. Averick, J. Bryan, W. Chang, L. D. McGowan, R. François, G. Golemund, A. Hayes, L. Henry, J. Hester, M. Kuhn, T. L. Pedersen, E. Miller, S. M. Bache, K. Müller, J. Ooms, D. Robinson, D. P. Seidel, V. Spinu, K. Takahashi, D. Vaughan, C. Wilke, K. Woo, and H. Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686. [p40]
- H. Wickham, R. François, L. Henry, K. Müller, and D. Vaughan. *dplyr: A Grammar of Data Manipulation*, 2023. URL <https://CRAN.R-project.org/package=dplyr>. R package version 1.1.2. [p40]
- V. Yilanci, S. Bozoklu, and M. S. Gorus. Are brics countries pollution havens? evidence from a bootstrap ardl bounds testing approach with a fourier function. *Sustainable Cities and Society*, 55: 102035, 2020. doi: 10.1016/j.scs.2020.102035. [p39]

Gianmarco Vacca
Department of Economic Policy. Università Cattolica del Sacro Cuore
Largo Gemelli, 1, Milan.
Italy
(0000-0002-8996-5524)
gianmarco.vacca@unicatt.it

Maria Zoia
Department of Economic Policy. Università Cattolica del Sacro Cuore
Largo Gemelli, 1, Milan.
Italy
(0000-0002-8169-781X)
maria.zoia@unicatt.it

Stefano Bertelli
CRO Area, Internal Validation and Controls Department, Operational Risk and ICAAP Internal Systems,
Intesa Sanpaolo, Milan
Viale Stelvio, 55/57, Milan.
Italy
stefano.bertelli@intesasanpaolo.com