Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

FCVAR: An R Package for the Fractionally Cointegrated Vector Autoregressive Model

Lealand Morin
University of Central Florida

Morten Ørregaard Nielsen
Queen's University and CREATES

Michał Ksawery Popiel
Analysis Group

Abstract

This article illustrates how to estimate the fractionally cointegrated vector autoregression model (FCVAR) in R, based on a companion package in MATLAB. This model is used to detect equilibrium relationships between variables observed over time. The cointegrated vector autoregression model (CVAR) can detect an equilibrium relationship between variables that are integrated, i.e. exhibit unit root behavior, where deviations from this relationship are not integrated. The fractionally cointegrated VAR model can detect relationships between variables that are cointegrated of a fractional order, with deviations that can be fractionally integrated but of a lower order than the variables themselves. This allows for the detection of relationships with deviations that correct more slowly. The **FCVAR** package ties together the features of these models in a maximum likelihood approach to allow for a comprehensive set of testing options.

Keywords: cofractional process, cointegration rank, fractional autoregressive model, fractional cointegration, fractional unit root, MATLAB, R, VAR model.

1. Introduction: Cointegration and fractional integration in R

Many time series exhibit features of fractional integration. The detection of equlibrium relationships between such variables is problematic if this feature is not built into the model. The fractionally cointegrated vector autoregression model (FCVAR) is designed to detect equilibrium relationships between fractionally integrated variables. A restricted version, the cointegrated vector autoregression model (CVAR) can detect an equilibrium relationship between variables that are integrated, i.e. exhibit unit root behavior, where deviations from this relationship are not integrated. In contrast, the fractionally cointegrated VAR model can detect relationships between variables that are cointegrated of a fractional order, with deviations that can also be fractionally integrated but of a lower order than the variables themselves. This allows for the detection of relationships with deviations that correct more slowly than with the CVAR.

The packages currently available concentrate more heavily on the features of one of two types of models: those with cointegrated series and those with fractionally integrated series. Many packages focus on the CVAR alone, which is restricted to integer orders of integration. Some focus on specific features of the variables, such as estimating the integration order, identifying whether a cointegrating relationship exists, or estimating a single-equation model. Those that do consider fractionally cointegrated systems follow approaches developed earlier in the literature. The software in **FCVAR** treats the time series as a system and estimates all parameters together in a maximum likelihood framework. This provides a flexible set of options for conducting inference on many features of the cointegrating relationship.

The most common tests impose linear restrictions on the parameters defining the cointegrating space and the process of error correction. In **FCVAR**, the user can also test equality and inequality restrictions on the fractional differencing parameters. In addition, the user can test any linear restriction on the short-run autoregressive parameters. Since the parameters are all jointly estimated in one parametric maximum likelihood framework, it is possible to jointly test any combination of these restrictions with a likelihood ratio test.

A comprehensive listing of the R packages (R Core Team 2017) available for time series analysis was compiled by Hyndman (2020). Of the packages that estimate integer-order cointegration, such as in the CVAR, several packages perform Engle-Granger tests, following Engle and Granger (1987). One such package is **aTSA** for *Alternative Time Series Analysis* (Qiu 2015), another is the **egcm** package in (Clegg 2017). Other packages have implemeted the cointegration tests in Phillips and Ouliaris (1990). This amounts to running a regression of the response variable on a set of regressors and testing the residuals for a unit root, following Phillips and Perron (1988). The **tseries** package (Trapletti, Hornik, and LeBaron 2019) implements this test, as well as the **urca** package (Pfaff, Zivot, and Stigler 2016). There are also a variety of choices among modified ordinary least squares (OLS) approaches, including the **cointReg** package in Aschersleben and Wagner (2016), implementing the methods of Phillips and Hansen (1990), Phillips and Loretan (1991), Saikkonen (1991), and Stock and Watson (1993). Following a maximum likelihood approach, the **tsDyn** package and the **urca** package implement the cointegrated VAR model, as in Johansen (1995). Of the packages designed for the CVAR model, **urca** is perhaps the closest available to the **FCVAR** package, in terms of the testing opportunities available.

The methods above allow for only a discrete form of cointegration between the series: models restricted to integer orders of integration. There are several packages designed for series with fractional integration or long memory. A number of these pacakages are focused on estimation of $ARFIMA$ models, or autoregressive fractionally integrated moving average models. The **fracdiff**[1] package by (Maechler, Fraley, Leisch, Reisen, Lemonte, and Hyndman

[1] There appears to be some duplication between this package and **FCVAR**: the `diffseries()` function in **fracdiff** is based on the same algorithm in Jensen and Nielsen (2014) as `FracDiff()` in **FCVAR**, except that `diffseries()` demeans the data first. Specifically, `fracdiff::diffseries(x, d) - FCVAR::FracDiff(x - mean(x), d)` is numerically very small. The demeaning step is not required to estimate the FCVAR model, as the parameters in the conditional mean are estimated jointly with the others while optimizing the likelihood

2020) includes one such example. Other options include the **arfima** package by Veenstra and McLeod (2018) and **nsarfima** package by (Groebe 2019). The latter package uses a few types of optimization methods built on the $ARFIMA$ model, including both maximum likelihood (as in Beran (1995)) and minimum distance (as in Mayoral (2007)) estimators. The **arfima** package follows a careful optimization procedure that displays several sets of estimates corresponding to multiple local optima—a common occurrence in these models—and this feature is accounted for in **FCVAR** as well.

The package **LongMemoryTS** is in a class of its own, in that it uses a wide variety of methods to investigate both fractional integration and cointegrating relationships. The authors implement the early semiparametric approaches for estimating the cointegrating vector, including that of Robinson (1994) and later Robinson and Marinucci (2003), Marmol and Velasco (2004), Christensen and Nielsen (2006), and Robinson (2008). They also implement more recent procedures of Nielsen (2010) and Wang, Wang, and Chan (2015), as well as the frequency-domain test for fractional cointegration in Souza, Reisen, Franco, and Bondon (2018). One feature that is missing, however, is the maximum likelihood method of Johansen (1995), which was extended to fractional processes in Johansen (2008). This framework allows for joint testing of a comprehensive set of restrictions and provides the foundation on which **FCVAR** is built. The **FCVAR** package in R complements a MATLAB package **FCVARmodel.m**, which is documented in Nielsen and Popiel (2016) and estimates the FCVAR model using similar syntax.

The next section describes the FCVAR model and the restricted models that can be estimated with this program. Section 3 describes an example of a modeling session, which is a replication of one of the tables of results in Jones, Nielsen, and Popiel (2014). Section 4 describes other examples, which demonstrate some additional functionality of the software.

2. The fractionally cointegrated VAR model

The fractionally cointegrated vector autoregressive (FCVAR) model was proposed in Johansen (2008) and analyzed by, e.g., Johansen and Nielsen (2010, 2012). For a time series X_t of dimension p, the fractionally cointegrated VAR model is given in error correction form as

$$\Delta^d X_t = \alpha\beta'\Delta^{d-b}L_b X_t + \sum_{i=1}^{k}\Gamma_i\Delta^d\ L_b^i X_t + \varepsilon_t, \qquad (1)$$

where ε_t is p-dimensional $i.i.d.(0,\Omega)$, Δ^d is the fractional difference operator, and $L_b = 1-\Delta^b$ is the fractional lag operator.[2] Johansen and Nielsen (2012) imposed two restrictions on the parameter space, $d \geq b$ and $d-b < 1/2$, in their asymptotic analysis. However, these restrictions were relaxed in Johansen and Nielsen (2018a,b).

Model (1) includes the Johansen (1995) CVAR model as the special case $d = b = 1$; see Johansen and Nielsen (2018b). Some of the parameters are well-known from the CVAR model and these have the usual interpretations in the FCVAR model. The most important of these are the long-run parameters α and β, which are $p \times r$ matrices with $0 \leq r \leq p$.

function.

[2]Both the fractional difference and fractional lag operators are defined in terms of their binomial expansion in the lag operator, L. Note that the expansion of L_b has no term in L^0 and thus only lagged disequilibrium errors appear in (1).

The rank r is termed the cointegration, or cofractional, rank. The columns of β constitute the r cointegration (cofractional) vectors such that $\beta' X_t$ are the cointegrating combinations of the variables in the system, i.e. the long-run equilibrium relations. The parameters in α are the adjustment or loading coefficients which represent the speed of adjustment towards equilibrium for each of the variables. The short-run dynamics of the variables are governed by the parameters $\Gamma = (\Gamma_1, \ldots, \Gamma_k)$ in the autoregressive augmentation.

The FCVAR model has two additional parameters compared with the CVAR model, namely the fractional parameters d and b. Here, d denotes the fractional integration order of the observable time series and b determines the degree of fractional cointegration, i.e. the reduction in fractional integration order of $\beta' X_t$ compared to X_t itself. These parameters are estimated jointly with the remaining parameters. This model thus has the same main structure as in the standard CVAR model in that it allows for modeling of both cointegration and adjustment towards equilibrium, but is more general since it accommodates fractional integration and cointegration.

In the next four subsections we briefly describe the accommodation of deterministic terms as well as estimation and testing in the FCVAR model.

2.1. Deterministic terms

There are several ways to accommodate deterministic terms in the FCVAR model (1). The inclusion of the so-called restricted constant was considered in Johansen and Nielsen (2012), and the so-called unrestricted constant term was considered in Dolatabadi, Nielsen, and Xu (2016). A general formulation that encompasses both models is[3]

$$\Delta^d X_t = \alpha \Delta^{d-b} L_b (\beta' X_t + \rho') + \sum_{i=1}^{k} \Gamma_i \Delta^d \, L_b^i X_t + \xi + \varepsilon_t. \tag{2}$$

The parameter ρ is the so-called restricted constant term (since the constant term in the model is restricted to be of the form $\alpha\rho'$), which is interpreted as the mean level of the long-run equilibria when these are stationary, i.e. $E\beta' X_t + \rho' = 0$. The parameter ξ is the unrestricted constant term, which gives rise to a deterministic trend in the levels of the variables. When $d = 1$ this trend is linear. Thus, the model (2) contains both a restricted constant and an unrestricted constant. In the usual CVAR model, i.e. with $d = b = 1$, the former would be absorbed in the latter, but in the fractional model they can both be present and are interpreted differently. For the representation theory related to (2), and in particular for additional interpretation of the two types of constant terms, see Dolatabadi *et al.* (2016).

An alternative formulation of deterministic terms was suggested by Johansen and Nielsen (2016), albeit in a simpler model, with the aim of reducing the impact of pre-sample observations of the process. This model is

$$\Delta^d (X_t - \mu) = \alpha \beta' \Delta^{d-b} L_b (X_t - \mu) + \sum_{i=1}^{k} \Gamma_i \Delta^d \, L_b^i (X_t - \mu) + \varepsilon_t, \tag{3}$$

[3] In Dolatabadi *et al.* (2016) the constants are included as $\rho' \pi_t(1)$ and $\xi \pi_t(1)$, where $\pi_t(u)$ denotes coefficients in the binomial expansion of $(1-z)^{-u}$. This is mathematically convenient, but makes no difference in terms of the practical implementation.

which can be derived easily from the unobserved components formulation

$$X_t = \mu + X_t^0, \quad \Delta^d X_t^0 = L_b \alpha \beta' X_t^0 + \sum_{i=1}^{k} \Gamma_i \Delta^d \, L_b^i X_t^0 + \varepsilon_t. \tag{4}$$

The formulation (3), or equivalently (4), includes the restricted constant, which may be obtained as $\rho' = \beta'\mu$. More generally, the level parameter μ is meant to accommodate a non-zero starting point for the first observation on the process, i.e., for X_1. It has the added advantage of reducing the bias arising due to pre-sample behavior of X_t, at least in simple models, even when conditioning on no initial values (see below). For details, see Johansen and Nielsen (2016).

2.2. Maximum likelihood estimation

It is assumed that a sample of length $T+N$ is available on X_t, where N denotes the number of observations used for conditioning, for details see Johansen and Nielsen (2016). The models (1), (2), and (3) are estimated by conditional maximum likelihood, conditional on N initial values, by maximizing the function

$$\log L_T(\lambda) = -\frac{Tp}{2}(\log(2\pi) + 1) - \frac{T}{2} \log\det \left\{ T^{-1} \sum_{t=N+1}^{T+N} \varepsilon_t(\lambda)\varepsilon_t(\lambda)' \right\}, \tag{5}$$

where the residuals are defined as

$$\varepsilon_t(\lambda) = \Delta^d X_t - \alpha \Delta^{d-b} L_b(\beta' X_t + \rho') - \sum_{i=1}^{k} \Gamma_i \Delta^d \, L_b^i X_t - \xi, \quad \lambda = (d, b, \alpha, \beta, \Gamma, \rho, \xi), \tag{6}$$

for model (2), and hence also for submodels of model (2), such as (1), with the appropriate restrictions imposed on ρ and ξ. For model (3) the residuals are

$$\varepsilon_t(\lambda) = \Delta^d (X_t - \mu) - \alpha \beta' \Delta^{d-b} L_b(X_t - \mu) - \sum_{i=1}^{k} \Gamma_i \Delta^d \, L_b^i (X_t - \mu), \quad \lambda = (d, b, \alpha, \beta, \Gamma, \mu). \tag{7}$$

It is shown in Johansen and Nielsen (2012) and Dolatabadi *et al.* (2016) how, for fixed (d, b), the estimation of model (2) reduces to regression and reduced rank regression as in Johansen (1995). In this way, the parameters $(\alpha, \beta, \Gamma, \rho, \xi)$ can be concentrated out of the likelihood function, and numerical optimization is only needed to optimize the profile likelihood function over the two fractional parameters, d and b. In model (3) we can similarly concentrate the parameters (α, β, Γ) out of the likelihood function resulting in numerical optimization over (d, b, μ), making the estimation of model (3) slightly more involved numerically than that of model (2).

For model (2) with $\xi = 0$, Johansen and Nielsen (2012) shows that asymptotic theory is standard when $b < 0.5$, and for the case $b > 0.5$ asymptotic theory is non-standard and involves fractional Brownian motion of type II. Specifically, when $b > 0.5$, Johansen and Nielsen (2012) shows that under i.i.d. errors with suitable moment conditions, the conditional maximum likelihood parameter estimates $(\hat{d}, \hat{b}, \hat{\alpha}, \hat{\Gamma}_1, \ldots, \hat{\Gamma}_k)$ are asymptotically Gaussian, while $(\hat{\beta}, \hat{\rho})$ are locally asymptotically mixed normal. These results allow asymptotically standard (chi-squared) inference on all parameters of the model, including the cointegrating relations and orders of fractionality, using quasi-likelihood ratio tests. As in the CVAR model, see Johansen (1995), the same results hold for the same parameters in the full models (2) and (3), whereas the asymptotic distribution theory for the remaining parameters, ξ and μ, is currently unknown.

2.3. Cointegration rank tests

Letting $\Pi = \alpha\beta'$, the likelihood ratio (LR) test statistic of the hypothesis $\mathcal{H}_r : \text{rank}(\Pi) = r$ against $\mathcal{H}_p : \text{rank}(\Pi) = p$ is of particular interest because it deals with an important empirical question. This statistic is often denoted the "trace" statistic. Let $\theta = (d, b)$ for model (2) and $\theta = (d, b, \mu)$ for model (3) denote the parameters for which the likelihood is numerically maximized. Then let $L(\theta, r)$ be the profile likelihood function given rank r, where (α, β, Γ), and possibly (ρ, ξ) if appropriate, have been concentrated out by regression and reduced rank regression; see Johansen and Nielsen (2012) and Dolatabadi *et al.* (2016) for details.

The profile likelihood function is maximized both under the hypothesis $\mathcal{H}_r$ and under $\mathcal{H}_p$ and the LR test statistic is then $\text{LR}_T(q) = 2\log(L(\hat{\theta}_p, p)/L(\hat{\theta}_r, r))$, where

$$L(\hat{\theta}_p, p) = \max_{\theta} L(\theta, p), \quad L(\hat{\theta}_r, r) = \max_{\theta} L(\theta, r),$$

and $q = p - r$. This problem is qualitatively different from that in Johansen (1995) since the asymptotic distribution of $\text{LR}_T(q)$ depends qualitatively (and quantitatively) on the parameter b. In the case with $0 < b < 1/2$ (sometimes known as "weak cointegration"), $\text{LR}_T(q)$ has a standard asymptotic distribution, see Johansen and Nielsen (2012, Theorem 11(ii)), namely

$$\text{LR}_T(q) \overset{D}{\to} \chi^2(q^2), \quad 0 < b < 1/2. \tag{8}$$

On the other hand, when $1/2 < b \leq d$ ("strong cointegration"), asymptotic theory is non-standard and

$$\text{LR}_T(q) \overset{D}{\to} \text{Tr}\left\{\int_0^1 \mathsf{d}W(s)F(s)' \left(\int_0^1 F(s)F(s)'\mathsf{d}s\right)^{-1} \int_0^1 F(s)\mathsf{d}W(s)'\right\}, \quad b > 1/2, \tag{9}$$

where the vector process $\mathsf{d}W$ is the increment of ordinary (non-fractional) vector standard Brownian motion of dimension $q = p - r$. The vector process F depends on the deterministics in a similar way as in the CVAR model in Johansen (1995), although the fractional orders complicate matters. The following cases have been derived in the literature:

1. When no deterministic term is in the model, $F(u) = W_b(u)$, where $W_b(u) = \Gamma(b)^{-1}\int_0^u (u - s)^{b-1}\mathsf{d}W(s)$ is vector fractional Brownian motion of type II, see Johansen and Nielsen (2012, Theorem 11(i)).

2. When only the restricted constant term is included in model (2), $F(u) = (W_b(u)', u^{-(d-b)})'$, see Johansen and Nielsen (2012, Theorem 11(iv)) for the result with $d = b$ and an earlier working paper version for the general result.

3. In model (3) the same result as in bullet 2. holds because $\beta'\mu = \rho'$ is the restricted constant and $\beta'_{\perp}\mu$ has no influence on the asymptotic distribution (in a similar way to X_0 in a random walk).

4. When both the restricted and unrestricted constants are included in model (2) with $d = 1$,

$$F_i(u) = W_{b,i}(u) - \int_0^1 W_{b,i}(u)\mathsf{d}u, \; i = 1, ..., q-1,$$
$$F_q(u) = u^b - \int_0^1 u^b \mathsf{d}u = u^b - 1/(b+1),$$
$$F_{q+1}(u) = u^{b-1} - \int_0^1 u^{b-1}\mathsf{d}u = u^{b-1} - 1/b,$$

see Dolatabadi *et al.* (2016).

Importantly, the asymptotic distribution (9) of the test statistic $\mathrm{LR}_T(q)$ depends on both b and $q = p - r$. The dependence on the unknown (true value of the) scalar parameter b complicates empirical analysis compared to the CVAR model. Generally, the distribution (9) would need to be simulated on a case-by-case basis. However, for model (1) and for model (2) with $d = b$ and $\xi = 0$, and hence also for model (3) with $d = b$ in light of bullet 3. above, computer programs for computing asymptotic critical values and asymptotic P values for the LR cointegration rank tests based on numerical distribution functions, are made available by MacKinnon and Nielsen (2014). Their computer programs are incorporated in the present program with a companion R package **fracdist** for the relevant cases/models as discussed and illustrated below.

2.4. Restricted models

Note that a reduced rank restriction has already been imposed on models (1)–(3), where the coefficient matrix $\Pi = \alpha\beta'$ has been restricted to rank $r \leq p$. Other restrictions on the model parameters can be considered as in Johansen (1995). The most interesting restrictions from an economic theory point of view would likely be restrictions on the adjustment parameters α and cointegration vectors β.

We formulate hypotheses as

$$R_\psi \psi = r_\psi, \tag{10}$$
$$R_\alpha \mathrm{vec}(\alpha) = 0, \tag{11}$$
$$R_\beta \mathrm{vec}(\beta^*) = r_\beta, \tag{12}$$

with $\beta^* = (\beta', \rho')'$, and use the switching algorithm in (Boswijk and Doornik 2004, p. 455) to optimize the likelihood numerically subject to the restrictions. The switching algorithm can be improved by adding a line search, see Doornik (2018). This is done by setting the option `opt$LineSearch <- 1`, which is the default setting.

The only limitation on the linear restrictions that can be imposed on (d, b, α, β^*) in (10)–(12) is that only homogenous restrictions can be imposed on $\mathrm{vec}(\alpha)$ in (11). Otherwise, any combination of linear restrictions can be imposed on these parameters. For now, the remaining parameters cannot be restricted.

Note that, when the restricted constant term ρ is included in the model, restrictions on β and ρ must be written in the form given by (12). This is without loss of generality.

The restrictions in (10)–(12) above can be implemented individually or simultaneously in the **FCVAR** package The next section provides an example session illustrating the use of the package with a step-by-step description of a typical empirical analysis, including several restricted models in Section 3.6.

2.5. Forecasting from the FCVAR model

Because the FCVAR model is autoregressive, the best linear predictor takes a simple form and is relatively straightforward to calculate. Consider, for example, the model with level parameter in (3). We first note that

$$\Delta^d(X_{t+1}-\mu) = X_{t+1}-\mu-(X_{t+1}-\mu)+\Delta^d(X_{t+1}-\mu) = X_{t+1}-\mu-L_d(X_{t+1}-\mu)$$

and then rearrange (3) as

$$X_{t+1} = \mu + L_d(X_{t+1}-\mu) + \alpha\beta'\Delta^{d-b}L_b(X_{t+1}-\mu) + \sum_{i=1}^{k}\Gamma_i\Delta^d L_b^i(X_{t+1}-\mu) + \varepsilon_{t+1}. \quad (13)$$

Since $L_b = 1-\Delta^b$ is a lag operator, so that $L_b^i X_{t+1}$ is known at time t for $i \geq 1$, this equation can be used as the basis to calculate forecasts from the model.

We let conditional expectation given the information set at time t be denoted $E_t(\cdot)$, and the best linear predictor forecast of any variable Z_{t+1} given information available at time t be denoted $\hat{Z}_{t+1|t} = E_t(Z_{t+1})$. Clearly, we then have that the forecast of the innovation for period $t+1$ at time t is $\hat{\varepsilon}_{t+1|t} = E_t(\varepsilon_{t+1}) = 0$, and $\hat{X}_{t+1|t}$ is then easily found from (13). Inserting also coefficient estimates based on data available up to time t, denoted[4] $(\hat{d}, \hat{b}, \hat{\mu}, \hat{\alpha}, \hat{\beta}, \hat{\Gamma}_1, \ldots, \hat{\Gamma}_k)$, we have that

$$\hat{X}_{t+1|t} = \hat{\mu} + L_{\hat{d}}(X_{t+1}-\hat{\mu}) + \hat{\alpha}\hat{\beta}'\Delta^{\hat{d}-\hat{b}}L_{\hat{b}}(X_{t+1}-\hat{\mu}) + \sum_{i=1}^{k}\hat{\Gamma}_i\Delta^{\hat{d}}L_{\hat{b}}^i(X_{t+1}-\hat{\mu}). \quad (14)$$

This defines the one-step-ahead forecast of X_{t+1} given information at time t.

Multi-period-ahead forecasts can be generated recursively. That is, to calculate the h-step-ahead forecast, we first generalize (14) as

$$\hat{X}_{t+j|t} = \hat{\mu} + L_{\hat{d}}(\hat{X}_{t+j|t}-\hat{\mu}) + \hat{\alpha}\hat{\beta}'\Delta^{\hat{d}-\hat{b}}L_{\hat{b}}(\hat{X}_{t+j|t}-\hat{\mu}) + \sum_{i=1}^{k}\hat{\Gamma}_i\Delta^{\hat{d}}L_{\hat{b}}^i(\hat{X}_{t+j|t}-\hat{\mu}), \quad (15)$$

where $\hat{X}_{s|t} = X_s$ for $s \leq t$. Then forecasts are calculated recursively from (15) for $j = 1, 2, \ldots, h$ to generate h-step-ahead forecasts, $\hat{X}_{t+h|t}$.

Clearly, one-step-ahead and h-step-ahead forecasts for the model (2) with a restricted constant term, and possibly also an unrestricted constant term, instead of the level parameter, can be calculated entirely analogously.

[4]To emphasize that these estimates are based on data available at time t, they could be denoted by a subscript t. However, to avoid cluttering the notation we omit this subscript and let it be understood in the sequel.

3. Example session

A demonstration of analysis is shown in `FCVAR_replication_JNP2014.R` and it serves as an example of what a typical session of model specification, estimation and testing can include. This code replicates "Table 4: FCVAR results for Model 1" from Jones *et al.* (2014) and follows the empirical procedure developed in that paper. This procedure includes the following steps:

1. Importing data
2. Choosing estimation options
3. Lag selection
4. Cointegration rank selection
5. Model estimation
6. Hypothesis testing

3.1. Importing data

The first step is importing the data. In pactice, a user could assign a data frame from a saved dataset, using, e.g. `read.csv()`. In this example, executing the code shown below assigns data from the external dataset `votingJNP2014`, which is available with the package.

```
R> x1 <- votingJNP2014[, c("lib", "ir_can", "un_can")]
```

The columns of the full dataset contain the following variables: (1) aggregate support for the Liberal party, (2) aggregate support for the Conservative party, (3) Canadian 3-month T-bill rates, (4) US 3-month T-bill rates, (5) Canadian unemployment rate, and (6) US unemployment rate. This example uses the variables in the first, third and fifth columns.

3.2. Choosing options

Once the data are imported, the user sets the program options. The script contains two sets of options: the arguments of the estimation functions and an object comprising the settings for model specification and estimation. The first set of options is as follows.

```
R> p             <- ncol(x1)
R> kmax          <- 3
R> order         <- 12
R> printWNtest   <- 1
```

The variable `kmax` determines the highest lag order for the sequential testing that is performed in the lag selection, whereas `p` is the dimension of the system. The `order` specifies the number of lags used for the white noise test in lag selection.

The next set of initialization commands assign values to the variables contained in the object `opt`, defined by the function `FCVARoptions()`.

```
R> opt <- FCVARoptions(
 +    unrConstant  = 0,
 +    rConstant    = 0,
 +    levelParam   = 1,
 +    constrained  = 0,
 +    restrictDB   = 1,
 +    dbMin = c(0.01, 0.01),
 +    dbMax = c(2.00, 2.00)
 +    )
R> opt$db0 <- c(0.80, 0.80)
R> opt$gridSearch <- 0
```

The first assignment initializes the object `opt` and assigns all of the default options set in `FCVARoptions`, other than the arguments stated in the function call. The next two commands show how to easily change any of the default options after `opt` is defined. Defining the program options in this way allows the user to create and store several option objects with different attributes. This can be very convenient when, for example, performing the same hypothesis tests on different data sets, or when performing a series of hypothesis tests that share many settings with a base model.

The option settings within the call to `FCVARoptions()` define the model to be estimated. In the present example, a model estimated with options `opt` will include the level parameter μ but no restricted or unrestricted constant. Adding deterministics requires setting the variable corresponding to the type of deterministic component, either `rConstant` or `unrConstant`, to `1`. The user may want to set restrictions on the fractional integration parameters d and b. Setting the option `opt$constrained  <- 1` would impose the inequality restriction $d >= b$. In this example, the restriction $d = b$ is imposed.

The last two settings relate to the numerical optimization of the likelihood function, to expedite the estimation for this example. The option `db0` sets the optimization to begin at the starting values $d = b = 0.80$. The option `gridSearch` is set off, which specifies that a single optimization sequence will be performed with these starting values. In practice, the grid search would be set with `opt$gridSearch <- 1`, i.e. the default, particularly in an initial investigation of the data, so that several values of d and b can be tested as potential starting values. This is meant to alleviate the identification problem discussed in Johansen and Nielsen (2010, Section 2.3) and Carlini and de Magistris (2019) because the likelihood function may have several local optima. This functionality is described in detail in Section 4.5.

Another option related to optimization is a line search in the switching algorithm for estimation of models with restrictions on α and/or β, such as when conducting hypothesis tests. This is added via the option `opt$LineSearch <- 1` and is the default. See Doornik (2018, Section 2.2) for details.

The remaining options can be divided into several categories: numerical optimization, model deterministics and restrictions, output and grid search. These are explained in detail in the package documentation.

3.3. Lag-order selection

Once the options are set, the user moves to the next step, which involves choosing the appropriate lag order. The relevant information is obtained with a call to `FCVARlagSelect()` which performs estimation of models with lag-orders from 0 to `kmax`. The program performs lag selection on the full-rank unrestricted model.

```
R> FCVARlagSelectStats <- FCVARlagSelect(x1, kmax, p, order, opt)
```

```
--------------------------------------------------------------------------------
                        Lag Selection Results
--------------------------------------------------------------------------------
Dimension of system:       3     Number of observations in sample:         316
Order for WN tests:       12     Number of observations for estimation:    316
Restricted constant:      No     Initial values:                             0
Unrestricted constant:    No     Level parameter:                          Yes
--------------------------------------------------------------------------------
Parameter Estimates and Information Criteria:
--------------------------------------------------------------------------------
 k  r    d     b      LogL     LR    pv     AIC       BIC
 3  3 0.676 0.676  456.42   7.31 0.605  -832.85   -682.62
 2  3 0.581 0.581  452.77  20.59 0.015  -843.53*  -727.11
 1  3 1.043 1.043  442.47  56.99 0.000  -840.94   -758.31*
 0  3 1.036 1.036  413.97   0.00 0.000  -801.95   -753.12
--------------------------------------------------------------------------------
Tests for Serial Correlation of Residuals:
--------------------------------------------------------------------------------
 k   pmvQ  pQ1   pLM1  pQ2   pLM2  pQ3   pLM3
 3   0.94  0.72  0.46  0.49  0.89  0.51  0.47
 2   0.82  0.69  0.45  0.29  0.75  0.54  0.40
 1   0.34  0.75  0.52  0.15  0.58  0.34  0.18
 0   0.00  0.01  0.01  0.00  0.08  0.37  0.17
--------------------------------------------------------------------------------
```

Estimates of d and b are reported for each lag (k) with rank (r) set to the number of variables in the system. Note that in this example the restriction $d = b$ has been imposed. The log-likelihood for each lag is shown in column `LogL`. The likelihood ratio test-statistic `LR` is for the null hypothesis $\Gamma_k = 0$ with p value reported in column `pv`. This is followed by AIC and BIC information criteria. The columns in the next block provide p values for white noise tests on the residuals. The first p value, `pmvQ`, is for the multivariate Q-test followed by univarite Q-tests as well as LM tests on the p individual residuals; that is, `pQ1` and `pLM1` are the p values for the residuals in the first equation, `pQ2` and `pLM2` are for the residuals in the second equation, and so on.

3.4. Cointegration rank testing

The user now chooses the lag-order based on the information provided above and can move to the next step, which is cointegration rank testing. In the next code block, the user first assigns the lag augmentation, $k = 2$ in this case, and then calls the function `FCVARrankTests()`.

```
R> k <- 2
R> rankTestStats <- FCVARrankTests(x1, k, opt)
```

```
--------------------------------------------------------------------------------
             Likelihood Ratio Tests for Cointegrating Rank
--------------------------------------------------------------------------------
Dimension of system:       3     Number of observations in sample:        316
Number of lags:            2     Number of observations for estimation:   316
Restricted constant:      No     Initial values:                            0
Unestricted constant:     No     Level parameter:                         Yes
--------------------------------------------------------------------------------
Rank     d       b     Log-likelihood  LR statistic   P-value
 0     0.643   0.643        440.040         25.454     0.043
 1     0.569   0.569        451.174          3.186     0.820
 2     0.576   0.576        452.707          0.120     0.947
 3     0.581   0.581        452.767           ----      ----
--------------------------------------------------------------------------------
```

The first block of output provides a summary of the model specification. The second block provides the test results relevant for selecting the appropriate rank. These include likelihood ratio tests for a restriction to a cointegrating rank against an unrestricted model with full rank. The p values are calculated by the **fracdist** package, which obtains simulated p values from MacKinnon and Nielsen (2014). The table is meant to be read sequentially from lowest to highest rank, i.e. from top to bottom. Since we can reject the null of rank 0 against the alternative of rank 3 we move to the test of rank 1 against rank 3. This test fails to reject with a p value of 0.820, so this is the appropriate choice in this case.

3.5. Unrestricted model estimation

With the rank and lag selected, the user can now move to the next code section.

Here the user first specifies the choice for the rank based on the previously performed cointegrating rank tests (thus setting $r = 1$ in this example). Next, the default options set in the initialization, see Section 3.2, are assigned to `opt1`, which is used as an argument in the call to the function `FCVARestn()`. This function is the main part of the program since it performs the estimation of the parameters, obtains model residuals and standard errors, and calculates many other relevant components such as the number of free parameters and the roots of the characteristic polynomial. If `opt1$print2screen <- 1` then, in addition to storing all of these results in the list `m1`, the function outputs the estimation results to the command window. To see a list of variables stored in `m1`, the user can type `m1` in the command line.

The program output is shown below. It begins with a table summarizing relevant model specifications and then the coefficients and their standard errors. The roots of the characteristic polynomial are displayed at the bottom.

```
R> r <- 1
R> opt1 <- opt
```

```
R> opt1$gridSearch <- 1
R> opt1$plotLike <- 1
R> m1 <- FCVARestn(x1, k, r, opt1)

--------------------------------------------------------------------------------
                      Fractionally Cointegrated VAR: Estimation Results
--------------------------------------------------------------------------------
Dimension of system:       3     Number of observations in sample:       316
Number of lags:            2     Number of observations for estimation:  316
Restricted constant:      No     Initial values:                           0
Unrestricted constant:    No     Level parameter:                        Yes
Starting value for d:    0.569   Parameter space for d: (0.469 , 0.669)
Starting value for b:    0.569   Parameter space for b: (0.469 , 0.669)
--------------------------------------------------------------------------------
Cointegrating rank:          1  AIC:              -848.348
Log-likelihood:        451.174  BIC:              -746.943
log(det(Omega_hat)):   -11.369  Free parameters:        27
--------------------------------------------------------------------------------
    Fractional parameters:
--------------------------------------------------------------------------------
    Coefficient              Estimate                Standard error
--------------------------------------------------------------------------------
         d                      0.569                      0.049
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
    Cointegrating equations (beta):
--------------------------------------------------------------------------------
      Variable       CI equation 1
--------------------------------------------------------------------------------
        Var1           1.000
        Var2           0.111
        Var3          -0.241
--------------------------------------------------------------------------------
Note: Identifying restriction imposed.
--------------------------------------------------------------------------------
    Adjustment matrix (alpha):
--------------------------------------------------------------------------------
      Variable       CI equation 1
--------------------------------------------------------------------------------
        Var 1         -0.180
         SE 1       (   0.064  )
        Var 2          0.167
         SE 2       (   0.194  )
        Var 3          0.037
         SE 3       (   0.014  )
--------------------------------------------------------------------------------
Note: Standard errors in parenthesis.
```

```
--------------------------------------------------------------------------------
    Long-run matrix (Pi):
--------------------------------------------------------------------------------
      Variable         Var 1          Var 2          Var 3
--------------------------------------------------------------------------------
      Var 1           -0.180         -0.020          0.043
      Var 2            0.167          0.019         -0.040
      Var 3            0.037          0.004         -0.009
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
    Level parameter (mu):
--------------------------------------------------------------------------------
        Var 1            -0.345
         SE 1         (   0.069  )
        Var 2            11.480
         SE 2         (   0.548  )
        Var 3            -2.872
         SE 3         (   0.033  )
--------------------------------------------------------------------------------
Note: Standard errors in parenthesis (from numerical Hessian)
      but asymptotic distribution is unknown.
--------------------------------------------------------------------------------
    Lag matrix 1 (Gamma_1):
--------------------------------------------------------------------------------
      Variable         Var 1          Var 2          Var 3
--------------------------------------------------------------------------------
      Var 1            0.276         -0.032         -0.510
       SE 1      (     0.160  )  (    0.026  )  (    0.513  )
      Var 2           -0.148          1.126         -3.286
       SE 2      (     0.378  )  (    0.196  )  (    1.975  )
      Var 3           -0.052          0.008          0.711
       SE 3      (     0.022  )  (    0.005  )  (    0.170  )
--------------------------------------------------------------------------------
Note: Standard errors in parentheses.
--------------------------------------------------------------------------------
    Lag matrix 2 (Gamma_2):
--------------------------------------------------------------------------------
      Variable         Var 1          Var 2          Var 3
--------------------------------------------------------------------------------
      Var 1            0.566          0.106          0.609
       SE 1      (     0.182  )  (    0.045  )  (    0.612  )
      Var 2            0.493         -0.462          0.451
       SE 2      (     0.562  )  (    0.198  )  (    2.627  )
      Var 3           -0.039         -0.020          0.318
       SE 3      (     0.033  )  (    0.008  )  (    0.143  )
--------------------------------------------------------------------------------
```

```
Note: Standard errors in parentheses.
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
    Roots of the characteristic polynomial
--------------------------------------------------------------------------------
    Number     Real part    Imaginary part       Modulus
--------------------------------------------------------------------------------
       1        -2.894         -0.000             2.894
       2        -1.522         -0.000             1.522
       3         1.010         -0.927             1.371
       4         1.010          0.927             1.371
       5         1.108          0.000             1.108
       6         1.000          0.000             1.000
       7         1.000          0.000             1.000
       8         0.944         -0.261             0.980
       9         0.944          0.261             0.980
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Restrictions imposed on the following parameters:
- Psi. For details see "options$R_psi"
--------------------------------------------------------------------------------
```

At the end of the output, a notice is printed to remind the user that restrictions were imposed on `Psi`, i.e. on (d, b). In this case, this is the restriction $d = b$ imposed via `opt$restrictDB <- 1`.

In addition to the coefficient estimates, we are also interested in testing the model residuals for serial correlation. After the unrestricted model has been estimated, this code section concludes with a call to `MVWNtest()`, which performs a series of white noise tests on the residuals and prints the output in the command window. The results of the white noise tests are shown below. For each residual both the Q- and LM-test statistics and their p values are reported, in addition to the multivariate Q-test and p value in the first line of the table. From the output of this table we can conclude that there does not appear to be any problems with serial correlation in the residuals.

```
R> MVWNtest_m1 <- MVWNtest(m1$Residuals, order, printWNtest)

       White Noise Test Results (lag = 12)
---------------------------------------------
Variable |       Q  P-val |      LM  P-val  |
---------------------------------------------
Multivar |  97.868  0.747 |     ----  ----  |
Var1     |   9.302  0.677 |  11.238  0.509  |
Var2     |  14.442  0.273 |   8.566  0.740  |
Var3     |  10.595  0.564 |  12.269  0.424  |
---------------------------------------------
```

Because `opt$plotRoots <- 1` in the options, the roots of the characteristic polynomial are also plotted along with the unit circle and the transformed unit circle, $\mathbb{C}_{\hat{b}}$, see Johansen (2008). The plot is shown in Figure 1.

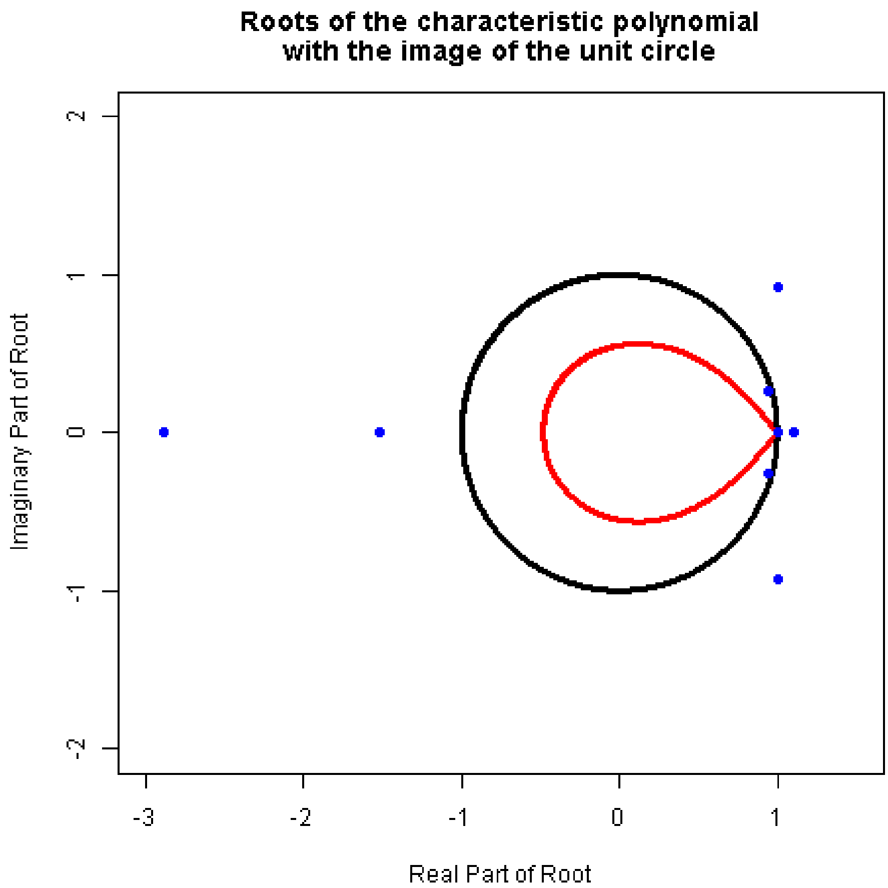


Figure 1: Roots of the characteristic polynomial.

Furthermore, the estimation was performed with the grid search and the plot option selected, i.e. with `opt$gridSearch <- 1` and `opt$plotLike <- 1`, which produces a plot of the log-likelihood. The plot for this model is shown in Figure 2. Note that the horizontal axis is the parameter ϕ, which represents both parameters d and b, which are constrained to be equal in this specification. Other such restrictions are described in more detail in Section 4.5.

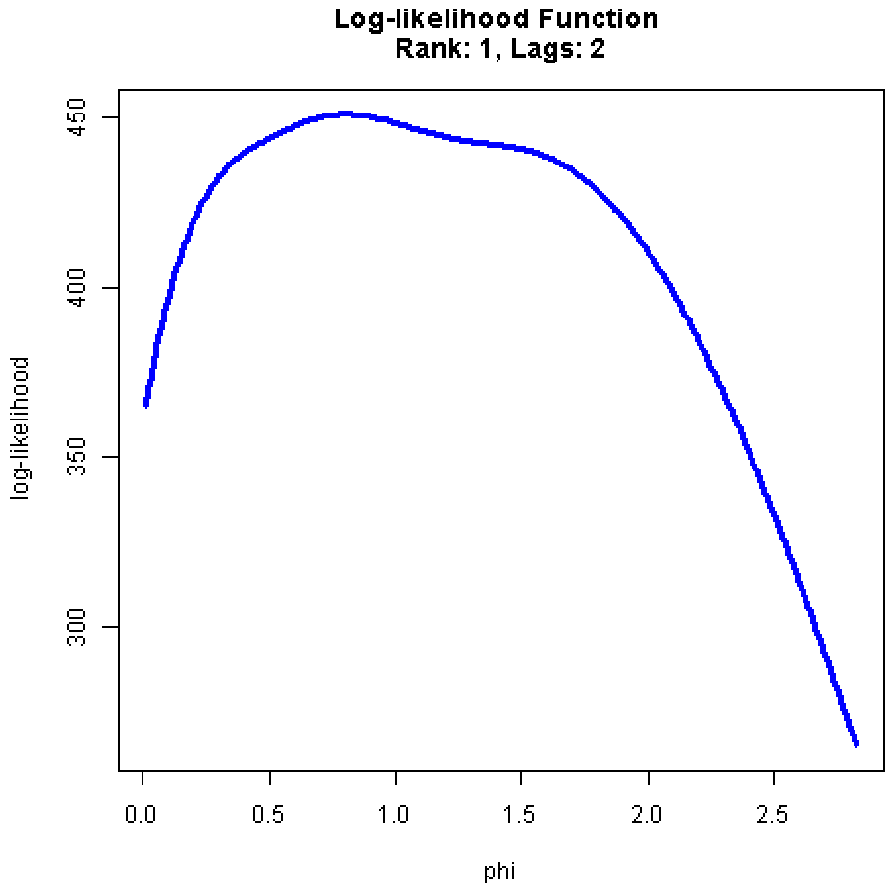


Figure 2: Plot of the log-likelihood function.

The complete results for the unrestricted model are stored in the S3 object `m1` of class `FCVAR_model` and can be accessed anytime. For instance, if the user would like to perform a more careful analysis of the residuals they are stored in `m1$Residuals`.

3.6. Hypothesis testing

We now move into the hypothesis testing section of the code where we can test several restricted models and perform inference. For restricted model estimation the grid search option is switched off because computation can be very slow, especially in the presence of the level parameter. However, if the user wishes to verify the accuracy of the results or if estimates are close to the upper or lower bound, the grid search option can resolve these issues and give the user additional insight about the behaviour of the likelihood, as shown in Section 4.5.

All hypotheses are defined as shown in (10)–(12). The first hypothesis test is $\mathscr{H}_d^1$ (for precise definitions of each hypothesis, please see Jones *et al.* (2014)), shown below.

```
R> opt1 <- opt
R> opt1$R_psi <- matrix(c(1, 0), nrow = 1, ncol = 2)
```

```
R> opt1$r_psi <- 1
R> m1r1 <- FCVARestn(x1, k, r, opt1)
R> MVWNtest_m1r1 <- MVWNtest(m1r1$Residuals, order, printWNtest)
R> Hdb <- FCVARhypoTest(m1, m1r1)
```

Here we test the CVAR model (null hypothesis $d = b = 1$) against the FCVAR model (alternative hypothesis $d = b \neq 1$). Since `opt$restrictDB <- 1` was selected in the choice of options, the restriction that $d = b$ is already imposed. Thus, the user needs to only impose an additional restriction that either d or b is equal to one. In this example, the restriction that $d = 1$ is imposed by setting `opt1$R_psi = [1 0]` and `opt1$r_psi = 1`, but the result would be the same if $b = 1$ were imposed instead. The restricted model is then estimated and the results are stored in the S3 object `m1r1` of class `FCVAR_model`. As before, the user can perform a series of white noise tests on the residuals by calling the `MVWNtest()` function. The next step is to perform the actual test. With the objects returned from the restricted and unrestricted models, the user can call the function `FCVARhypoTest()` and perform an LR test. This function takes the two model result objects as inputs, automatically compares the number of free parameters to obtain the degrees of freedom, computes the LR test statistic, and displays the output. The results of this test are then stored in the list `Hdb` and can be accessed at any time.

Since the output of the estimated model and the white noise tests are similar to the previous example, we only show the output from the hypothesis test.

```
Likelihood ratio test results:
Unrestricted log-likelihood: 451.174
Restricted log-likelihood:   442.027
Test results (df = 1):
LR statistic:   18.295
P-value:   0.000
```

The log-likelihoods from both models are reported, along with the degrees of freedom, the LR test statistic, and its p value. In this case the test clearly rejects the null hypothesis that the model is a CVAR. For more significant digits, or to access any of these values from the command window, the user can type `Hdb`.

The next hypothesis of interest is $\mathscr{H}_\beta^1$, which is a zero restriction on the first element of the cointegration vector.

```
R> opt1 <- opt
R> opt1$R_Beta <- matrix(c(1, 0, 0), nrow = 1, ncol = 3)
R> m1r2 <- FCVARestn(x1, k, r, opt1)
R> MVWNtest_m1r2 <- MVWNtest(m1r2$Residuals, order, printWNtest)
R> Hbeta1 <- FCVARhypoTest(m1, m1r2)
```

Since the object `opt1` has the restriction $d = b = 1$ stored, the first step is to reset the options to those of the base model. The restriction on β is then specified as in (12). There are two details to note here. First, the column length of R_β must equal $p_1 r$, where $p_1 = p + 1$ if a restricted constant is present and $p_1 = p$ otherwise; recall that p is the number of variables in the system and r is the number of cointegrating vectors. Second, zero restrictions are the default and automatically imposed when r_β is empty. Therefore, the user only needs to specify r_β if it includes non-zero elements. Recall that for restrictions on α only $r_\alpha = 0$ is allowed so that there is no need to specify r_α. As before, the restricted model is estimated with results stored in `m1r2`, the residuals are tested for white noise, and the model under the null is tested against the unrestricted model `m1` with results stored in `Hbeta1`.

Again, since the estimation output is similar to the first example, we only show the results of the hypothesis test here. With a p value close to zero, this hypothesis is also strongly rejected.

```
Likelihood ratio test results:
Unrestricted log-likelihood: 451.174
Restricted log-likelihood:   444.395
Test results (df = 1):
LR statistic:   13.557
P-value:   0.000
```

Next, we move to tests on α. In this case, we test the restriction that the political variable is long-run exogenous, i.e. that the adjustment coefficient on this variable is zero.

```
R> opt1 <- opt
R> opt1$R_Alpha <- matrix(c(1, 0, 0), nrow = 1, ncol = 3)
R> opt1$gridSearch <- 0
R> m1r3 <- FCVARestn(x1, k, r, opt1)
R> MVWNtest_m1r3 <- MVWNtest(m1r3$Residuals, order, printWNtest)
R> Halpha1 <- FCVARhypoTest(m1, m1r3)
```

Again we first reset `opt1` to the original options to clear previously imposed restrictions. Note that, if it were the case that we failed to reject $\mathscr{H}_\beta^1$ and wanted to leave it imposed while adding a restriction on α, we could either omit the first line `opt1 <- opt`, or we could replace it with `opt1 <- m1r2$options`. The latter assignment is preferred in this case because it is explicit about which model options we are leaving imposed.

The hypothesis $\mathscr{H}_\alpha^1$ is tested in the exact same way as before, only now we are changing the variable R_α instead of R_β. The results are shown below and we can see that this hypothesis is also rejected.

```
Likelihood ratio test results:
Unrestricted log-likelihood: 451.174
Restricted log-likelihood:   446.086
Test results (df = 1):
LR statistic:   10.176
P-value:   0.001
```

We next move to the remaining long-run exogeneity tests, $\mathscr{H}_\alpha^2$ and $\mathscr{H}_\alpha^3$, shown in the examples below. The hypothesis $\mathscr{H}_\alpha^2$ tests that the interest-rate is long-run exogenous.

```
R> opt1 <- opt
R> opt1$R_Alpha <- matrix(c(0, 1, 0), nrow = 1, ncol = 3)
```

```
R> opt1$gridSearch <- 0
R> m1r4 <- FCVARestn(x1, k, r, opt1)
R> MVWNtest_m1r4 <- MVWNtest(m1r4$Residuals, order, printWNtest)
R> Halpha2 <- FCVARhypoTest(m1, m1r4)

Likelihood ratio test results:
Unrestricted log-likelihood: 451.174
Restricted log-likelihood:   450.857
Test results (df = 1):
LR statistic:          0.633
P-value:          0.426
```

Next, we test the hypothesis $\mathscr{H}_\alpha^3$ that unemployment is long-run exogenous.

```
R> opt1 <- opt
R> opt1$gridSearch <- 0
R> opt1$R_Alpha <- matrix(c(0, 0, 1), nrow = 1, ncol = 3)
R> m1r5 <- FCVARestn(x1, k, r, opt1)
R> MVWNtest_m1r5 <- MVWNtest(m1r5$Residuals, order, printWNtest)
R> Halpha3 <- FCVARhypoTest(m1, m1r5)

Likelihood ratio test results:
Unrestricted log-likelihood: 451.174
Restricted log-likelihood:   446.184
Test results (df = 1):
LR statistic:          9.979
P-value:          0.002
```

The only hypothesis that we fail to reject is $\mathscr{H}_\alpha^2$, under which interest rates are long-run exogenous. After having estimated all of the restricted models of interest, we provide the full estimation output for the model `m1r4`, with the restriction imposed for $\mathscr{H}_\alpha^2$. Note from the output that $\alpha_2 = 0$ as imposed by the restriction.

```
--------------------------------------------------------------------------------
                      Fractionally Cointegrated VAR: Estimation Results
--------------------------------------------------------------------------------
Dimension of system:       3     Number of observations in sample:          316
Number of lags:            2     Number of observations for estimation:     316
Restricted constant:      No     Initial values:                              0
Unrestricted constant:    No     Level parameter:                           Yes
Starting value for d:    0.800   Parameter space for d: (0.010 , 2.000)
Starting value for b:    0.800   Parameter space for b: (0.010 , 2.000)
--------------------------------------------------------------------------------
Cointegrating rank:            1 AIC:              -849.715
Log-likelihood:          450.857 BIC:              -752.065
log(det(Omega_hat)):     -11.367 Free parameters:        26
```

```
--------------------------------------------------------------------------------
    Fractional parameters:
--------------------------------------------------------------------------------
    Coefficient                Estimate                Standard error
--------------------------------------------------------------------------------
         d                        0.575                      0.048
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
    Cointegrating equations (beta):
--------------------------------------------------------------------------------
      Variable        CI equation 1
--------------------------------------------------------------------------------
        Var1              0.994
        Var2              0.105
        Var3             -0.181
--------------------------------------------------------------------------------
    Adjustment matrix (alpha):
--------------------------------------------------------------------------------
      Variable        CI equation 1
--------------------------------------------------------------------------------
        Var 1            -0.189
         SE 1         (   0.065  )
        Var 2             0.000
         SE 2         (   0.000  )
        Var 3             0.039
         SE 3         (   0.014  )
--------------------------------------------------------------------------------
Note: Standard errors in parenthesis.
--------------------------------------------------------------------------------
    Long-run matrix (Pi):
--------------------------------------------------------------------------------
      Variable         Var 1          Var 2          Var 3
--------------------------------------------------------------------------------
      Var 1           -0.188         -0.020          0.034
      Var 2            0.000          0.000          0.000
      Var 3            0.039          0.004         -0.007
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
    Level parameter (mu):
--------------------------------------------------------------------------------
        Var 1            -0.310
         SE 1         (   0.067  )
        Var 2            11.538
         SE 2         (   0.553  )
        Var 3            -2.873
         SE 3         (   0.033  )
```

```
--------------------------------------------------------------------------------
Note: Standard errors in parenthesis (from numerical Hessian)
      but asymptotic distribution is unknown.
--------------------------------------------------------------------------------
    Lag matrix 1 (Gamma_1):
--------------------------------------------------------------------------------
      Variable        Var 1          Var 2          Var 3
--------------------------------------------------------------------------------
      Var 1           0.269         -0.032         -0.512
       SE 1       (   0.157  )  (   0.026  )  (   0.507  )
      Var 2          -0.013          1.115         -3.001
       SE 2       (   0.345  )  (   0.189  )  (   1.909  )
      Var 3          -0.053          0.008          0.694
       SE 3       (   0.022  )  (   0.005  )  (   0.164  )
--------------------------------------------------------------------------------
Note: Standard errors in parentheses.
--------------------------------------------------------------------------------
    Lag matrix 2 (Gamma_2):
--------------------------------------------------------------------------------
      Variable        Var 1          Var 2          Var 3
--------------------------------------------------------------------------------
      Var 1           0.570          0.104          0.585
       SE 1       (   0.184  )  (   0.044  )  (   0.606  )
      Var 2           0.685         -0.371          0.223
       SE 2       (   0.508  )  (   0.159  )  (   2.509  )
      Var 3          -0.043         -0.020          0.330
       SE 3       (   0.032  )  (   0.008  )  (   0.138  )
--------------------------------------------------------------------------------
Note: Standard errors in parentheses.
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
    Roots of the characteristic polynomial
--------------------------------------------------------------------------------
    Number     Real part    Imaginary part       Modulus
--------------------------------------------------------------------------------
       1        -2.710         -0.000             2.710
       2        -1.498         -0.000             1.498
       3         1.129         -0.939             1.469
       4         1.129          0.939             1.469
       5         1.098          0.000             1.098
       6         1.000          0.000             1.000
       7         1.000          0.000             1.000
       8         0.934         -0.281             0.976
       9         0.934          0.281             0.976
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
```

```
Restrictions imposed on the following parameters:
- Psi. For details see "options$R_psi"
- Alpha. For details see "options$R_Alpha"
--------------------------------------------------------------------------------


       White Noise Test Results (lag = 12)
---------------------------------------------
Variable |       Q  P-val |      LM  P-val   |
---------------------------------------------
Multivar |  97.665  0.752 |     ----  ----   |
Var1     |   9.084  0.696 |  11.267  0.506   |
Var2     |  14.931  0.245 |   9.338  0.674   |
Var3     |  10.729  0.552 |  12.241  0.426   |
---------------------------------------------
```

The model output is usually not normalized with respect to the user's variable of interest, for example, when restrictions are imposed on α or β. For this reason, we also include a code section that normalizes the output, i.e. imposes an identity matrix in the first $r \times r$ block of β. Suppose that $\tilde{\alpha}$ and $\tilde{\beta}$ are the particular estimates of α and β. Then, $\tilde{\alpha}$ can be post-multiplied by G^{-1}, where G' is the inverse of the upper $r \times r$ block of $\tilde{\beta}$. Then $\Pi = \tilde{\alpha}\tilde{\beta}' = (\tilde{\alpha}G^{-1})(G\tilde{\beta}')$, so that $\Pi = \hat{\alpha}\hat{\beta}'$ Of course, this code section should only be executed if it does not interfere with any restrictions imposed on the model.

```
R> modelRstrct <- m1r4
R> G <- solve(modelRstrct$coeffs$betaHat[1:r, 1:r])
R> betaHatR <- modelRstrct$coeffs$betaHat %*% G
R> alphaHatR <- modelRstrct$coeffs$alphaHat %*% t(solve(G))
R> print("betaHatR' = ")
R> print(t(betaHatR), print.gap = 5)
R> print("alphaHatR' = ")
R> print(t(alphaHatR), print.gap = 5)
```

As an example of when this feature can be useful, consider model $\mathscr{H}_\alpha^2$. In the output above, we notice that the cointegrating vector has not been normalized (because restrictions are imposed). The user assigns the model of interest to the variable `modelRstrct`, in this case `m1r4`, and executes the commands. The output is shown below.

```
[1] "betaHatR' = "
         [,1]          [,2]           [,3]
[1,]        1     0.1057173     -0.1824022
[1] "alphaHatR' = "
               [,1]     [,2]           [,3]
[1,]     -0.1876696        0     0.03856341
```

In the unrestricted case, however, this normalization is always imposed, since the reduced-rank matrix Π has only $(2p - r)r$ degrees of freedom and the restriction of the upper $r \times r$

block of $\hat{\beta}$, is enough to identify the remaining $(p-r)r+pr$ parameters in $\hat{\alpha}$ and $\hat{\beta}$. From a practical point of view, the user may want to make a deliberate choice of the order of the variables in the vector X_t, so that the cointegrating relations can be stated in the form $X_{j,t} = -\beta_{r+1,j}X_{r+1,t} - \beta_{r+2,j}X_{r+2,t} - \cdots - \beta_{p,j}X_{p,t}$, for $j = 1, \ldots, r$, if this form permits an easier interpretation.

One implication of this discussion of identification is that arbitrary restrictions on the matrices $\hat{\alpha}$ and $\hat{\beta}$ must overidentify the parameters in order to be tested. That is, the restrictions to be tested must go beyond the restriction that the upper $r \times r$ block of $\hat{\beta}$ is the identity matrix.[5]

4. Additional examples

To show some additional functionality of the FCVAR software package, this section contains several other examples, which are based on Jones *et al.* (2014), but are not part of that paper. These include forecasting, bootstrap tesing, simulation and plotting of the likelihood function.

4.1. Forecasting

This code block performs recursive one-step ahead forecasts for each of the variables as well as the equilibrium relation.

```
R> NumPeriods <- 12
R> modelF <- m1r4
R> xf <- FCVARforecast(x1, modelF, NumPeriods)
R> seriesF <- rbind(x1, xf)
R> equilF <- seriesF %*% modelF$coeffs$betaHat
```

[5]Otherwise, the optimized values of the likelihood function might be equal under the restricted and unrestricted models and the likelihood ratio statistic will have value zero. If the user observes this outcome, further restrictions are required to overidentify the restricted model. For this problem and others, the `FCVARtest()` function prints warning messages to guide the user to restate the restrictions.

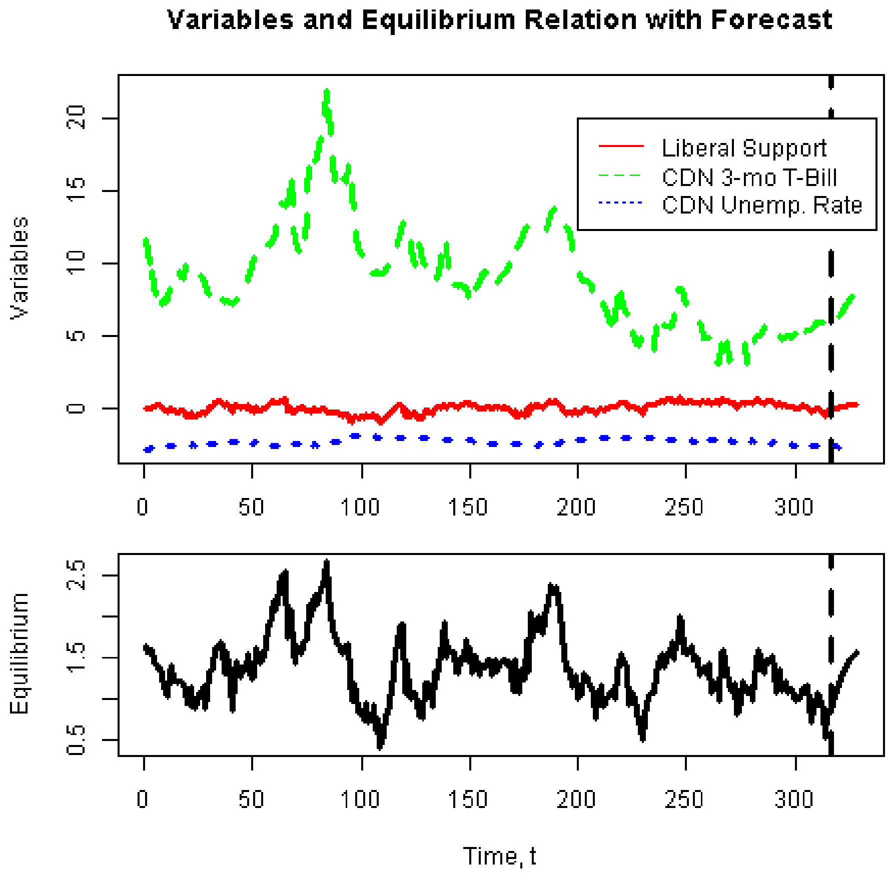


Figure 3: Forecast and equilibrium relationship of final model 12 steps ahead.

The user specifies the forecast horizon (`NumPeriods`) as well as the model (in this case, `modelF <- m1r4`). These two inputs, along with the data, are used in the call to the function `FCVARforecast()`. This function returns `xf`, a `NumPeriods` by p matrix of forecasted values of X, which are forecasted to take place after `x1`. Figure 3 plots the original series and the equilibrium relation $X\hat{\beta}$, stored in `equilF`, along with the forecasts.

4.2. Bootstrap hypothesis test

This code block demonstrates the use of the wild bootstrap for hypothesis tests on the parameters, as developed by Boswijk, Cavaliere, Rahbek, and Taylor (2016) for the CVAR model. The function `FCVARboot()` returns the results of the wild bootstrap. The user specifies two sets of options corresponding to two different nested models, the restricted model with `optRES` and the unrestricted model with `optUNR`. This particular example tests the restriction that political variables do not enter the cointegrating relation(s).

```
R> opt$plotRoots <- 0
R> optUNR <- opt
R> optRES <- opt
```

```
R> optRES$R_Beta <- matrix(c(1, 0, 0), nrow = 1, ncol = 3)
R> set.seed(42)
R> FCVARboot_stats <- FCVARboot(x1, k, r, optRES, optUNR, B = 999)
R> LRbs_density <- density(FCVARboot_stats$LRbs)
```

An example of the output is

```
Bootstrap likelihood ratio test results:
Unrestricted log-likelihood: 451.174
Restricted log-likelihood:   444.395
Test results (df <- 1):
LR statistic:   13.557
P-value:   0.000
P-value (BS):   0.021
```

The user might also be interested in comparing the bootstrap likelihood ratio test statistic distribution to the asymptotic one, a χ-squared distribution with `H$df` degrees of freedom. These objects can be used to produce a plot of the two distributions, shown in Figure 4. Note that, in the small sample, the statistic has probability mass over negative values, and the upper tail is much thicker. This is the reason for the larger bootstrap p value and a justification for conducting the bootstrap test.

4.3. Bootstrap rank test

This code block shows how to perform a wild bootstrap rank test, following the methodology of Cavaliere, Rahbek, and Taylor (2010) for the CVAR model. This procedure works in much the same way as the bootstrap hypothesis test described in Section 4.2. The difference is that, instead of providing two sets of estimation options, the user specifies two different ranks for comparison.

```
R> r1 <- 0
R> r2 <- 1
R> FCVARbootRank_stats <- FCVARbootRank(x1, k, opt, r1, r2, B = 999)
R> cat(sprintf('P-value (asy): \t %1.3f\n', rankTestStats$pv[1]))
```

The results are printed as

```
Bootstrap rank test results:
Unrestricted log-likelihood: 451.174
Restricted log-likelihood:   440.040
Test results:
LR statistic:   22.268
P-value (BS):   0.031
P-value (asy):   0.043
```

in which the last p value is printed to compare the bootstrap p value to that based on the asymptotic distribution.

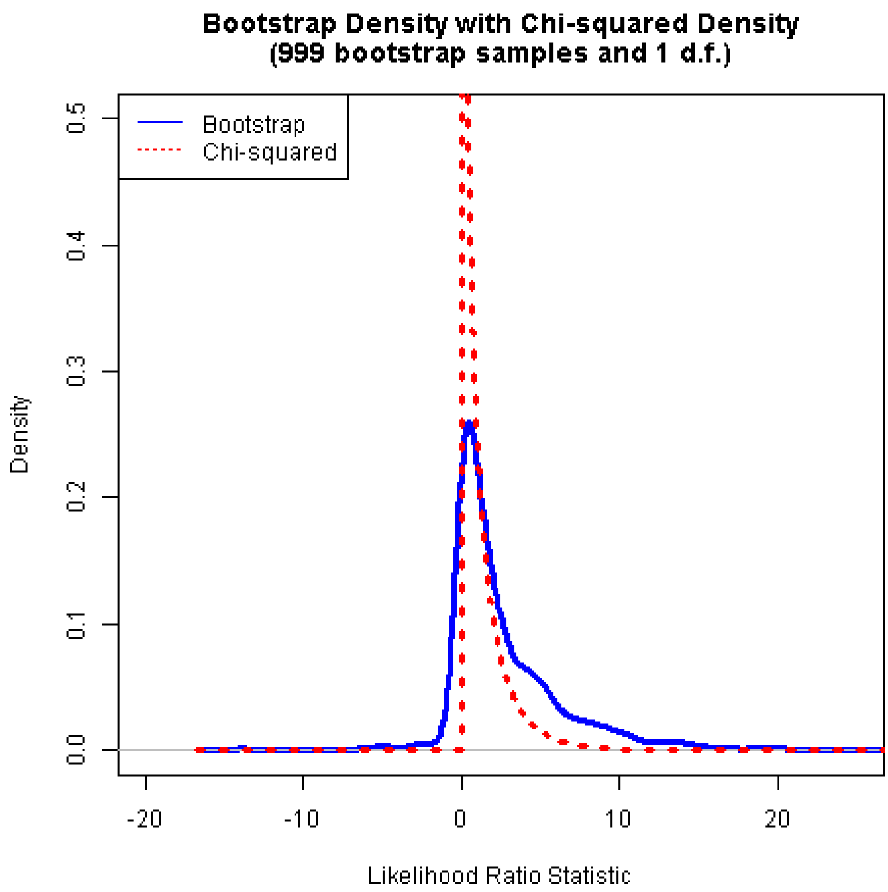


Figure 4: Density of bootstrap LR test statistic.

4.4. Simulation

Next, this example shows how to simulate an FCVAR model for a given set of parameters. The user provides data for starting values and a list containing model parameters for simulation as well as the number of periods to simulate. The simulated data are generated using Gaussian errors.

```
R> T_sim <- 100
R> xSim <- FCVARsim(x1, modelF, T_sim)
```

For the example above, using the same data as for the forecasting example above, the generated data is shown in Figure 5.

4.5. Plotting the likelihood function

Users should be aware that the likelihood function is sometimes badly behaved, in that there may be multiple local optima. This is a typical problem with models with fractionally integrated series, which is the reason the **arfima** package, designed for univariate series, employs

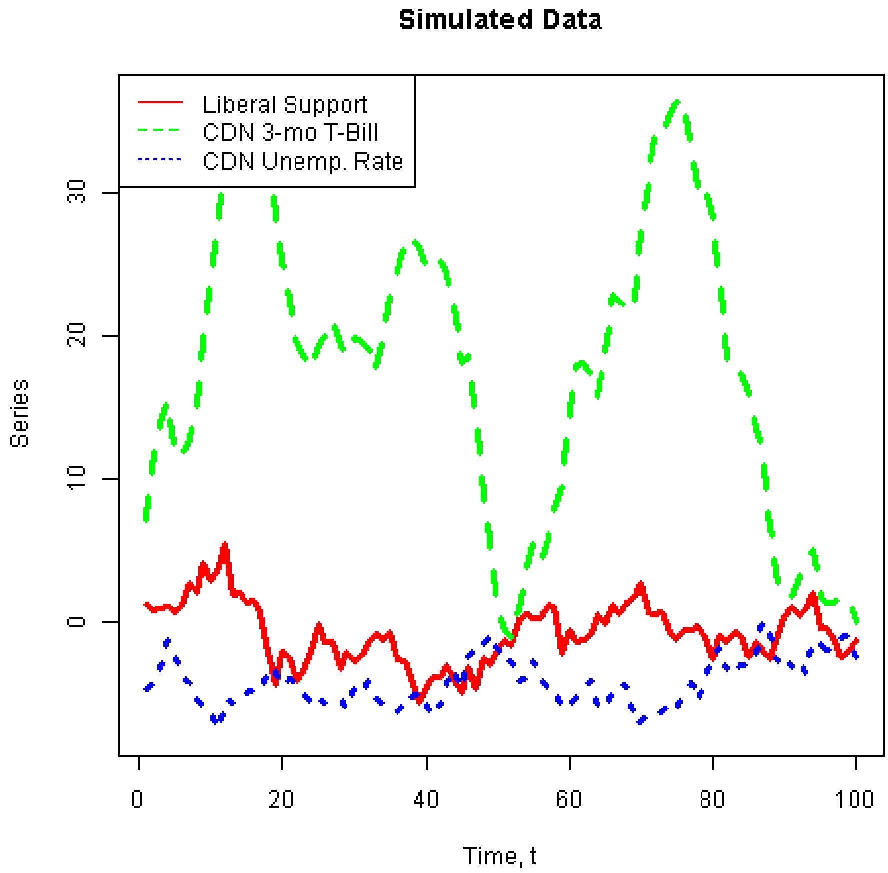
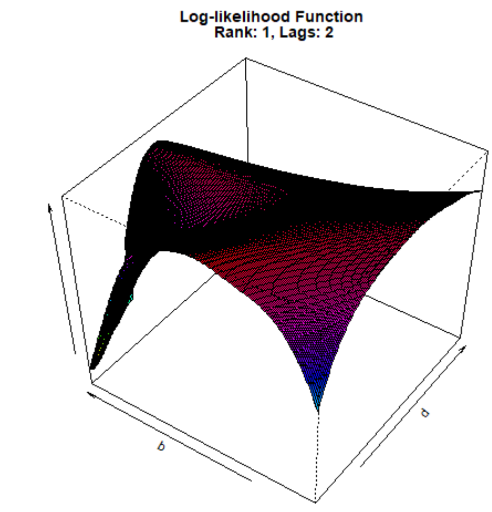


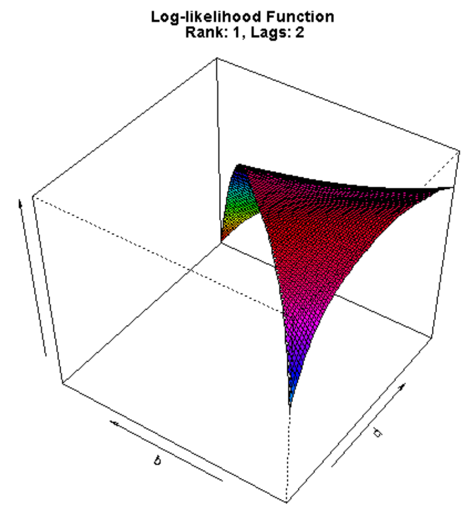
Figure 5: Simulated data

a series of optimization runs, each with different starting values, in an attempt to discover several local optima. To mitigate this problem in the FCVAR model, the **FCVAR** package includes functions for calculating the value of the likelihood function on a grid of values of the fractional integration parameters d and b.

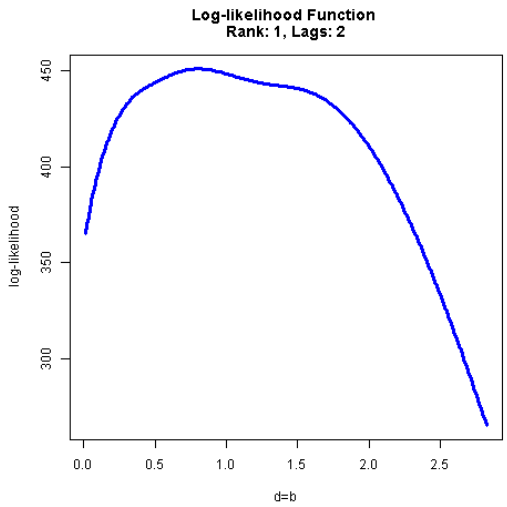
The function `FCVARlikeGrid()` conducts this grid search, which provides parameter values that are used as starting values for the optimization on d and b. It also allows for the possibility of overcoming an identification problem outlined in Johansen and Nielsen (2010, Section 2.3) and Carlini and de Magistris (2019). This identification problem occurs if the model lag order k is overspecified. In this case, there may be, asymptotically, several local maxima with identical maximal values, and the correct estimator is the one corresponding to the largest value of the parameter b. Note that this estimator is *not necessarily* the one that achieves the global optimum. For this reason, when the likelihood function has multiple local optima, the identification problem is alleviated if the estimator of the pair of parameters $[d, b]$ is chosen as the one with the highest value of b, among the local optima, and the optimization continues using that gridpoint as the starting values for d and b in `FCVARestn()`. This is the procedure that is followed if the user selects the option `opt$gridSearch <- 1`.



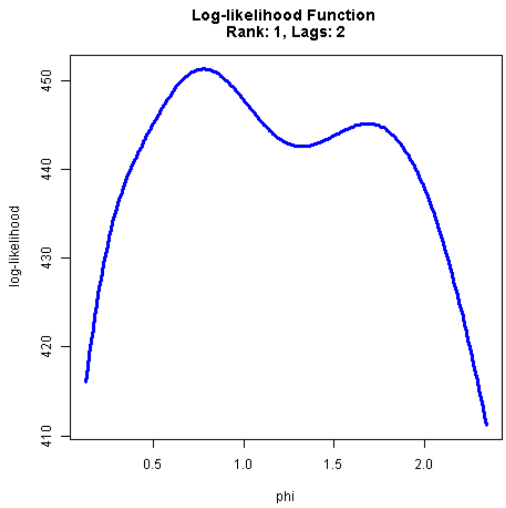
(a) (d, b) unconstrained



(b) $d \geq b$



(c) $d = b$



(d) $R_\psi(d, b)' = r_\psi, \quad (d, b)' = H\phi + h$

Figure 6: Plots of the log-likelihood function under alternative restrictions.

As an aid to users while distingushing between local and global optima, the output of the `FCVARlikeGrid()` is a class of S3 object called `FCVAR_grid` that is compatible with the generic `plot()` method. Together, these functions can produce four kinds of plots, corresponding to the four types of grid searches, under four alternative types of constaints on d and b. The first two cases are two-dimensional problems in d and b. In the first case, with d and b completely unconstrained, the grid search is over two dimensions within the bounds specified by `opt$dbMin` and `opt$dbMax`. An example of the likelihood obtained in an unconstrained grid search is shown in Figure 6(a). Next, if $d \geq b$ is imposed via `opt$constrained <- 1` (imposed in Johansen and Nielsen (2012) but relaxed in Johansen and Nielsen (2018b)) the computation time can be cut in half. An example of this likelihood is shown in Figure 6(b).

If the restriction $d = b$ is imposed, then the grid search is one-dimensional as shown in Figure 6(c). Finally, if a restriction is imposed on either d or b via R_ψ and r_ψ in (10), then the grid search is also one-dimensional. An example of this situation is shown in Figure 6(d). Note that the horizontal axis is over the parameter ϕ and the fractional parameters are found from

$$\begin{bmatrix} d \\ b \end{bmatrix} = H\phi + h, \tag{16}$$

where $H = (R'_\psi)_\perp$ and $h = R'_\psi(R_\psi R'_\psi)^{-1}r_\psi$. The bounds on ϕ are derived from `opt$dbMin` and `opt$dbMax` in a similar way. In the example in Figure 6(d), the parameters are chosen with $R_\psi = [2, -1]$ and $r_\psi = 0.5$, so that the restriction imposed is $[2, -1][d, b]' = 2d - b = 0.5$. In this case, $H = [1, 2]'/\sqrt{5}$ and $h = [0.2, -0.1]'$, and ϕ ranges from $\phi = 0.05 \times \sqrt{5} = 0.12$, where $b = 0$, to $\phi = 2.1 \times \sqrt{5} = 2.35$, where $b = 2$. This restriction, although somewhat artificial, produces a likelihood function with two local optima. The global optimum takes place at $\phi = 0.346 \times \sqrt{5} = 0.77$, which corresponds to $d = 0.546$ and $b = 0.591$, with a likelihood value of 451.36. A local maximum occurs at $\phi = 0.753 \times \sqrt{5} = 1.68$, which corresponds to $d = 0.953$ and $b = 1.405$, with a likelihood value of 445.19. According to the recommendation in Carlini and de Magistris (2019), users are advised to choose the local optimum with $b = 1.405$ under this particular constraint on d and b. With the functionality offered by `FCVARlikeGrid()`, the user can be assured that the optimization of the likelihood produces a consistent estimator.

Computational details

The results in this paper were obtained using R 4.0.5 with the **FCVAR** package, version 0.1.0. The p values for cointegrating rank tests are obtained using the **fracdist** package version 0.1.1. R itself, in addition to each package used in this document, is also available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/. The **FCVAR** package is available on CRAN, and can be loaded using the `install.packages()` function, as in

```
R> install.packages("FCVAR")
R> library("FCVAR")
```

A development version is stored in a GitHub repository and, with the **devtools** package and the supplementary toolkit **Rtools** installed, the **FCVAR** package can also be installed by entering

```
devtools::install_github("LeeMorinUCF/FCVAR")
```

The latest version of the MATLAB package **FCVARmodel.m** in Nielsen and Popiel (2016) can be downloaded from the Website of one of the authors at:

https://sites.google.com/view/mortennielsen/software

The version of MATLAB available at the release of the last version was MATLAB 9.0, R2016a, number 35.

Acknowledgments

We are grateful to Federico Carlini, Andreas Noack Jensen, Søren Johansen, Maggie Jones, James MacKinnon, Harry J. Paarsch, Jason Rhinelander, and Daniela Osterrieder for comments, and to the Canada Research Chairs program, the Social Sciences and Humanities Research Council of Canada (SSHRC), and the Center for Research in Econometric Analysis of Time Series (CREATES, funded by the Danish National Research Foundation DNRF78) for financial support.

References

Aschersleben P, Wagner M (2016). ***cointReg**: Parameter Estimation and Inference in a Cointegrating Regression.* R package version 0.2.0, URL https://CRAN.R-project.org/package=cointReg.

Beran J (1995). "Maximum Likelihood Estimation of the Differencing Parameter for Short and Long Memory Autoregressive Integrated Moving Average Models." *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**(4), 659–672.

Boswijk HP, Cavaliere G, Rahbek A, Taylor AMR (2016). "Inference on Co-integration Parameters in Heteroskedastic Vector Autoregressions." *Journal of Econometrics*, **192**, 64–85.

Boswijk HP, Doornik JA (2004). "Identifying, Estimating and Testing Restricted Cointegrated Systems: An Overview." *Statistica Neerlandica*, **58**, 440–465.

Carlini F, de Magistris PS (2019). "On the Identification of Fractionally Cointegrated VAR Models with the F(d) Condition." *Journal of Business & Economic Statistics*, **37**(1), 134–146.

Cavaliere G, Rahbek A, Taylor AMR (2010). "Testing for Co-integration in Vector Autoregressions with Non-stationary Volatility." *Journal of Econometrics*, **158**, 7–24.

Christensen BJ, Nielsen MØ (2006). "Asymptotic Normality of Narrow-band Least Squares in the Stationary Fractional Cointegration Model and Volatility Forecasting." *Journal of Econometrics*, **133**, 343–371.

Clegg M (2017). ***egcm**: Engle-Granger Cointegration Models.* R package version 1.0.12, URL https://CRAN.R-project.org/package=egcm.

Dolatabadi S, Nielsen MØ, Xu K (2016). "A Fractionally Cointegrated VAR Model with Deterministic Trends and Application to Commodity Futures Markets." *Journal of Empirical Finance*, **38B**, 623–639.

Doornik JA (2018). "Accelerated Estimation of Switching Algorithms: The Cointegrated VAR Model and other Applications." *Scandinavian Journal of Statistics*, **45**(2), 283–300.

Engle RF, Granger CWJ (1987). "Co-integration and Error Correction: Representation, Estimation and Testing." *Econometrica*, **55**(2), 251–276.

Groebe B (2019). ***nsarfima**: Methods for Fitting and Simulating Non-Stationary ARFIMA Models.* R package version 0.1.0.0, URL https://CRAN.R-project.org/package=nsarfima.

Hyndman RJ (2020). *CRAN Task View: Time Series Analysis.* URL https://CRAN.R-project.org/view=TimeSeries.

Jensen AN, Nielsen MØ (2014). "A Fast Fractional Difference Algorithm." *Journal of Time Series Analysis*, **35**, 428–436.

Johansen S (1995). *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models.* Oxford University Press, New York.

Johansen S (2008). "A Representation Theory for a Class of Vector Autoregressive Models for Fractional Processes." *Econometric Theory*, **24**, 651–676.

Johansen S, Nielsen MØ (2010). "Likelihood Inference for a Nonstationary Fractional Autoregressive Model." *Journal of Econometrics*, **158**, 51–66.

Johansen S, Nielsen MØ (2012). "Likelihood Inference for a Fractionally Cointegrated Vector Autoregressive Model." *Econometrica*, **80**, 2667–2732.

Johansen S, Nielsen MØ (2016). "The Role of Initial Values in Conditional Sum-of-Squares Estimation of Nonstationary Fractional Time Series Models." *Econometric Theory*, **32**, 1095–1139.

Johansen S, Nielsen MØ (2018a). "Nonstationary Cointegration in the Fractionally Cointegrated VAR Model." QED working paper 1405, Queen's University.

Johansen S, Nielsen MØ (2018b). "Testing the CVAR in the Fractional CVAR Model." Forthcoming in *Journal of Time Series Analysis*.

Jones M, Nielsen MØ, Popiel MK (2014). "A Fractionally Cointegrated VAR Analysis of Economic Voting and Political Support." *Canadian Journal of Economics*, **47**, 1078–1130.

MacKinnon JG, Nielsen MØ (2014). "Numerical Distribution Functions of Fractional Unit Root and Cointegration Tests." *Journal of Applied Econometrics*, **29**, 161–171.

Maechler M, Fraley C, Leisch F, Reisen V, Lemonte A, Hyndman R (2020). ***fracdiff**: Fractionally Differenced ARIMA aka ARFIMA(P,d,q) Models.* R package version 1.5-1, URL https://CRAN.R-project.org/package=fracdiff.

Marmol F, Velasco C (2004). "Consistent Testing of Cointegrating Relationships." *Econometrica*, **72**(6), 1809–1844.

Mayoral L (2007). "Minimum Distance Estimation of Stationary and Non-stationary ARFIMA Processes." *The Econometrics Journal*, **10**, 124–148.

Nielsen MØ (2010). "Nonparametric Cointegration Analysis of Fractional Systems with Unknown Integration Orders." *Journal of Econometrics*, **155**(2), 170–187.

Nielsen MØ, Popiel MK (2016). "A Matlab Program and User's Guide for the Fractionally Cointegrated VAR model." QED working paper 1330, Queen's University.

Pfaff B, Zivot E, Stigler M (2016). ***urca**: Unit Root and Cointegration Tests for Time Series Data.* R package version 1.3-0, URL https://CRAN.R-project.org/package=urcs.

Phillips P, Hansen B (1990). "Statistical Inference in Instrumental Variables Regression with I(1) Processes." *Review of Economic Studies*, **57**, 99–125.

Phillips P, Loretan M (1991). "Estimating Long Run Economic Equilibria." *Review of Economic Studies*, **58**, 407–436.

Phillips P, Ouliaris S (1990). "Asymptotic Properties of Residual Based tests for Cointegration." *Econometrica*, **58**(1), 165–193.

Phillips P, Perron P (1988). "Testing for a Unit Root in Time Series Regression." *Biometrika*, **75**(2), 335–346.

Qiu D (2015). ***aTSA**: Alternative Time Series Analysis.* R package version 3.1.2, URL https://CRAN.R-project.org/package=aTSA.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Robinson P (2008). "Diagnostic Testing for Cointegration." *Journal of Econometrics*, **143**(1), 206–225.

Robinson PM (1994). "Semiparametric Analysis of Long-memory Time Series." *Annals of Statistics*, **22**, 515–539.

Robinson PM, Marinucci D (2003). "Likelihood-Based Inference in Cointegrated Vector Autoregressive Models." In PM Robinson (ed.), *Time Series with Long Memory*, pp. 334–373. Oxford University Press, Oxford.

Saikkonen P (1991). "Asymptotically Efficient Estimation of Cointegrating Regressions." *Econometric Theory*, **7**, 1–21.

Souza IVM, Reisen VA, Franco GdC, Bondon P (2018). "The Estimation and Testing of the Cointegration Order based on the Frequency Domain." *Journal of Business and Economic Statistics*, **36**(4), 695–704.

Stock J, Watson M (1993). "A Simple Estimator of Cointegrating Vectors in Higher Order Integrated Systems." *Econometrica*, **61**, 783–820.

Trapletti A, Hornik K, LeBaron B (2019). ***tseries**: Time Series Analysis and Computational Finance.* R package version 0.10-47, URL https://CRAN.R-project.org/package=tseries.

Veenstra JQ, McLeod A (2018). ***arfima**: Fractional ARIMA (and Other Long Memory) Time Series Modeling.* R package version 1.7-0, URL https://CRAN.R-project.org/package=arfima.

Wang B, Wang M, Chan NH (2015). "Residual-based Test for Fractional Cointegration." *Economics Letters*, **126**, 43–46.

Affiliation:

Lealand Morin
Department of Economics
University of Central Florida
E-mail: Lealand.Morin@ucf.edu

Morten Ørregaard Nielsen
Department of Economics
Queen's University
and
CREATES
Aarhus University
E-mail: mon@econ.queensu.ca
URL: https://sites.google.com/view/mortennielsen

Michał Ksawery Popiel
Analysis Group
E-mail: Michal.Popiel@analysisgroup.com

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd