

# Statistical analysis of autoregressive fractionally integrated moving average models in R

Javier E. Contreras-Reyes · Wilfredo Palma

Received: 19 August 2012 / Accepted: 11 February 2013  
© Springer-Verlag Berlin Heidelberg 2013

**Abstract** The autoregressive fractionally integrated moving average (ARFIMA) processes are one of the best-known classes of long-memory models. In the package `arfmitools` for R, we have implemented a number of statistical tools for analyzing ARFIMA models. In particular, this package contains functions for parameter estimation, exact autocovariance calculation, predictive ability testing and impulse response function computation, among others. Furthermore, the implemented methods are illustrated with applications to real-life time series.

**Keywords** ARFIMA models · Long-memory time series · Whittle estimation · Exact variance matrix · Impulse response functions · Forecasting · R

## 1 Introduction

Long-memory processes, introduced by [Granger and Joyeux \(1980\)](#) and [Hosking \(1981\)](#), are playing a key role in the time series literature and have become a useful tool for modeling data arising in biology, economics, geophysics and many other fields, cf. [Palma \(2007\)](#) and references therein. As a consequence, a number of techniques for analyzing these processes have been developed and implemented in statistical packages. For example, packages about long-memory processes have been developed

---

J. E. Contreras-Reyes (✉)  
Departamento de Estadística, Universidad de Valparaíso,  
Gran Bretaña 1111, Playa Ancha, 2360102 Valparaíso, Chile  
e-mail: jecontr@uc.cl

W. Palma  
Departamento de Estadística, Pontificia Universidad Católica de Chile,  
Av. Vicuña Mackena 4860, Macul, 7820436 Santiago, Chile  
e-mail: wilfredo@mat.puc.cl

in R (R Development Core Team 2012): the `longmemo` package produces a Whittle estimation for fractional gaussian noise and fractional ARIMA models via an approximate MLE using the Beran (1994) algorithm and performs spectral density of fractional gaussian noise and periodogram estimate. In addition, the `fracdiff` package simulates ARFIMA time series, estimates ARFIMA parameters using an approximate MLE approach (Haslett and Raftery 1989), and calculates their variances with the Hessian method. Recently, Hyndman and Khandakar (2008) describe the `forecast` package to automatically predict univariate time series via state space models with exponential smoothing for ARIMA models. In addition, the `forecast` package offers a forecast function for ARFIMA models estimated using the algorithm proposed by Peiris and Perera (1988). The `afmtools` package requires the `polynom`, `hypergeo`, `sandwich` and the aforementioned `fracdiff` and `longmemo` packages.

Unfortunately, many of these computational implementations have important shortcomings. For instance, there is a severe lack of algorithms for calculating exact autocovariance functions (ACVF) of ARFIMA models, for computing precise estimator variances, and for forecasting performance tests (Giacomini and White 2006), and impulse response functions (Hassler and Kokoszka 2010), as well as for other aspects. In order to circumvent some of these problems, this paper discusses the package `afmtools` developed by Contreras-Reyes et al. (2011). This package aims to provide functions for computing ACVFs by means of the Sowell (1992) algorithm, ARFIMA fitting through an approximate estimation scheme via Whittle algorithm (Whittle 1953), asymptotic parameter estimate variances and several other tasks mentioned before. Hence, the aims of this paper are to analyze the `afmtools` package and to illustrate its theoretical and practical performance, which complements the existing development packages related to ARFIMA models mentioned above. Specifically, we implement our findings in a meteorological application about tree ring growth.

The remainder of this paper is structured as follows. Section 2 is devoted to describing the ARFIMA processes and their properties. This section includes an analysis of the spectral density, autocovariance function, parameter variance-covariance matrix estimation, impulse response function, and a model parameters estimation method. In addition, this section provides a test for assessing the predictive ability of a time series model. Finally, Sect. 3 addresses the performance of the functions implemented in the `afmtools` package. Apart from describing the methodologies implemented in this package, we also illustrate their applications to real-life time series data.

## 2 ARFIMA processes

Recent statistical literature has been concerned with the study of long-memory models that go beyond the presence of random walks and unit roots in the univariate time series processes. The autoregressive fractionally integrated moving-average (ARFIMA) process is a class of long-memory models (Granger and Joyeux 1980; Hosking 1981), the main objective of which is to explicitly account for persistence to incorporate the long-term correlations in the data. The general expression for ARFIMA processes  $\{y_t\}$  may be defined by the equation

$$\Phi(B)y_t = \Theta(B)(1 - B)^{-d}\varepsilon_t, \tag{1}$$

where  $\Phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$  and  $\Theta(B) = 1 + \theta_1 B + \dots + \theta_q B^q$  are the autoregressive and moving-average operators, respectively;  $\Phi(B)$  and  $\Theta(B)$  have no common roots,  $B$  is the backward shift operator and  $(1 - B)^{-d}$  is the fractional differencing operator given by the binomial expansion

$$(1 - B)^{-d} = \sum_{j=0}^{\infty} \frac{\Gamma(j + d)}{\Gamma(j + 1)\Gamma(d)} B^j = \sum_{j=0}^{\infty} \eta_j B^j, \tag{2}$$

for  $d \in (-1, 1/2)$  and  $\{\varepsilon_t\}$  is a white noise sequence with zero mean and innovation variance  $\sigma^2$ . An asymptotic approximation of

$$\eta_j = \frac{\Gamma(j + d)}{\Gamma(j + 1)\Gamma(d)} \tag{3}$$

for large  $j$  is

$$\eta_j \sim \frac{j^{d-1}}{\Gamma(d)}, \tag{4}$$

where  $\Gamma$  is the usual gamma function.

**Theorem 2.1** Consider the ARFIMA process defined by (1) and assume that the polynomials  $\Phi(\cdot)$  and  $\Theta(\cdot)$  have no common zeros and that  $d \in (-1, \frac{1}{2})$ . Then,

- (a) If the zeros of  $\Phi(\cdot)$  lie outside the unit circle  $\{z : |z| = 1\}$ , then there is a unique stationary solution of (1) given by  $y_t = \sum_{j=-\infty}^{\infty} \psi_j \varepsilon_{t-j}$  where  $\psi_j$  are the coefficients of the following polynomial  $\psi(z) = (1 - z)^{-d}\Theta(z)/\Phi(z)$ .
- (b) If the zeros of  $\Phi(\cdot)$  lie outside the closed unit disk  $\{z : |z| \leq 1\}$ , then the solution  $\{y_t\}$  is causal.
- (c) If the zeros of  $\Theta(\cdot)$  lie outside the closed unit disk  $\{z : |z| \leq 1\}$ , then the solution  $\{y_t\}$  is invertible.

For a proof of Theorem 2.1, see e.g. Palma (2007). Recall that, according to the representation theorem of Wold (1938), any stationary process is the sum of a regular process and a singular process; these two processes are orthogonal and the decomposition is unique. Thus, a stationary purely nondeterministic process may be expressed as

$$y_t = \psi(B)\varepsilon_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}. \tag{5}$$

The spectral measure of the purely nondeterministic process (5) is absolutely continuous with respect to the Lebesgue measure on  $[-\pi, \pi]$ , where the spectral density of the process (1) can be written as

$$\begin{aligned}
 f(\lambda) &= \frac{\sigma^2}{2\pi} |\psi(e^{-i\lambda})|^2 \\
 &= \frac{\sigma^2}{2\pi} |1 - e^{-i\lambda}|^{-2d} \frac{|\Theta(e^{-i\lambda})|^2}{|\Phi(e^{-i\lambda})|^2} \\
 &= \frac{\sigma^2}{2\pi} \left(2 \sin \frac{\lambda}{2}\right)^{-2d} \frac{|\Theta(e^{-i\lambda})|^2}{|\Phi(e^{-i\lambda})|^2}.
 \end{aligned}
 \tag{6}$$

where  $i$  denotes the imaginary unit. A special case of ARFIMA models is the fractionally differenced process described by Hosking (1981), in which the polynomials are  $\Phi(B) = \Theta(B) = 1$  and the spectral density is given by  $f(\lambda) = \frac{\sigma^2}{2\pi} |1 - e^{-i\lambda}|^{-2d}$ .

### 2.1 Whittle estimation

The methodology to approximate MLE is based on the calculation of the periodogram  $I(\lambda)$  by means of the *fast Fourier transform* (FFT); e.g., Singleton (1979), and the use of the approximation of the Gaussian log-likelihood function due to Whittle (1953) and by Bisaglia and Guégan (1998). Suppose that the sample vector  $Y = (y_1, y_2, \dots, y_n)$  is normally distributed with zero mean and autocovariance given by (11) as

$$\gamma(k - j) = \int_{-\pi}^{\pi} f(\lambda) e^{i\lambda(k-j)} d\lambda,$$

where  $f(\lambda)$  is defined as in (6) and is associated with the parameter set  $\Omega$  of the ARFIMA model defined in (1). The log likelihood function of the process  $Y$  is given by

$$L(\Omega) = -\frac{1}{2n} [\log |\Delta| - \mathbf{Y}^T \Delta^{-1} \mathbf{Y}].
 \tag{7}$$

where  $\Delta = [\gamma(k - j)]$  with  $k, j = 1, \dots, n$ . For calculating (7), two asymptotic approximations are made for the terms  $\log(|\Delta|)$  and  $\mathbf{Y}^T \Delta^{-1} \mathbf{Y}$  to obtain

$$L(\Omega) \approx -\frac{1}{4\pi} \left[ \int_{-\pi}^{\pi} \log[2\pi f(\lambda)] d\lambda + \int_{-\pi}^{\pi} \frac{I(\lambda)}{f(\lambda)} d\lambda \right],
 \tag{8}$$

as  $n \rightarrow \infty$ , where  $I(\lambda) = |\sum_{j=1}^n y_j e^{i\lambda j}|^2 / (2\pi n)$ , is the periodogram indicated before. Thus, a discrete version of (8) is actually the Riemann approximation of the integral and is

$$L(\Omega) \approx -\frac{1}{2n} \left[ \sum_{j=1}^n \log f(\lambda_j) + \sum_{j=1}^n \frac{I(\lambda_j)}{f(\lambda_j)} \right],
 \tag{9}$$

where  $\lambda_j = 2\pi j/n$  are the Fourier frequencies. Now, to find the estimator of the parameter vector  $\Omega$ , we use the minimization of  $L(\Omega)$  produced by the `nlm` function.

This non-linear minimization function carries out a minimization of  $L(\Omega)$  using a Newton-type algorithm. Under regularity conditions according to Theorem 2.2 (see Sect. 2.2), the Whittle estimator  $\hat{\Omega}$  that maximizes the log-likelihood function given in (9) is consistent and distributed normally (e.g. Dahlhaus 1989).

### 2.2 Parameter variance-covariance matrix

Here, we discuss a method for calculating the exact asymptotic variance-covariance matrix of the parameter estimates. This is a useful tool for making statistical inferences about exact and approximate maximum likelihood estimators, such as the Haslett and Raftery (1989) and Whittle methods (see Sect. 2.1). An example of this calculation for an ARFIMA(1,  $d$ , 1) model is given by Palma (2007, pp. 105–108). This calculation method of the Fisher information matrix is an alternative to the numerical computation using the Hessian matrix.

This proposed method is based on the explicit formula obtained by means of the derivatives of the parameters log-likelihood gradients. From the spectral density defined in (6), we define the partial derivatives  $\nabla_{\Phi} = (\partial/\partial\phi_i)$  and  $\nabla_{\Theta} = (\partial/\partial\theta_j)$ , with  $i = 1, \dots, p$  and  $j = 1, \dots, q$ .

**Theorem 2.2** *Under the assumptions that  $y_t$  is a stationary Gaussian sequence, the densities  $f(\lambda)$ ,  $f^{-1}(\lambda)$ ,  $\partial/\partial\mu_i f^{-1}(\lambda)$ ,  $\partial^2/\partial\mu_i\partial\mu_j f^{-1}(\lambda)$  and  $\partial^3/\partial\mu_i\partial\mu_j\partial\mu_k f^{-1}(\lambda)$  are continuous in  $(\lambda, \mu)$  for a parameter set  $\mu = \{d, \phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q\}$ ; we have the convergence in distribution for an estimated parameter  $\hat{\mu}$  and the true parameter  $\mu_0$  about a Gaussian ARFIMA model with*

$$\sqrt{n}(\hat{\mu}_n - \mu_0) \xrightarrow[n \rightarrow \infty]{d} N(0, \Sigma^{-1}(\mu_0)),$$

where

$$\Sigma(\mu) = \frac{1}{4\pi} \int_{-\pi}^{\pi} [\nabla \log f_{\mu}(\lambda)][\nabla \log f_{\mu}(\lambda)]^{\top} d\lambda. \tag{10}$$

For a proof of Theorem 2.2, see e.g. Palma (2007). Thus, if we consider the model (1) with spectral density (6) where  $\{\varepsilon_t\}$  is an independent and identically distributed  $N(0, \sigma^2)$ , we have that the parameter variance-covariance matrix  $\Gamma$  may be calculated in the following proposition.

**Proposition 2.3** *If  $\{y_t\}$  is stationary, then*

$$\begin{aligned} \frac{\partial}{\partial d} \log f(\lambda) &= -\log [2(1 - \cos \lambda)], \\ \frac{\partial}{\partial \phi_u} \log f(\lambda) &= \frac{\sum_{j=1}^p \phi_j \cos[\lambda(u - j)]}{\sum_{j=1}^p \sum_{k=1}^p \phi_j \phi_k \cos[\lambda(j - k)]}, \\ \frac{\partial}{\partial \theta_v} \log f(\lambda) &= \frac{\sum_{j=1}^q \theta_j \cos[\lambda(v - j)]}{\sum_{j=1}^q \sum_{k=1}^q \theta_j \theta_k \cos[\lambda(j - k)]}. \end{aligned}$$

for  $u = 1, \dots, p$  and  $v = 1, \dots, q$ .

*Proof* First, from the spectral density given in (6) we have that

$$\log f(\lambda) = \log \left( \frac{\sigma^2}{2\pi} \right) - d \log [2(1 - \cos \lambda)] + \log |\Theta(e^{i\lambda})|^2 - \log |\Phi(e^{i\lambda})|^2.$$

By Theorems 2.1 and 2.2, we observe that  $\Phi(e^{i\lambda}) = \sum_{j=1}^p \phi_j e^{i\lambda j}$ , this yields

$$\begin{aligned} |\Phi(e^{i\lambda})|^2 &= \sum_{j=1}^p \sum_{k=1}^p \phi_j \phi_k e^{i\lambda(j-k)} \\ &= 2 \sum_{j=1}^p \sum_{k=1}^p \phi_j \phi_k \cos[(j - k)\lambda], \end{aligned}$$

and

$$\begin{aligned} \frac{\partial}{\partial \phi_u} |\Phi(e^{i\lambda})|^2 &= 2\phi_u + \sum_{j \neq u} \phi_j e^{i\lambda(u-j)} + \sum_{k \neq u} \phi_k e^{i\lambda(u-k)} \\ &= \sum_{j=1}^p \phi_j e^{i\lambda(u-j)} + \sum_{k=1}^p \phi_k e^{i\lambda(u-k)} \\ &= 2 \sum_{j=1}^p \phi_j \cos[(u - j)\lambda]. \end{aligned}$$

Analogously, for MA components we have that

$$\frac{\partial}{\partial \theta_v} |\Theta(e^{i\lambda})|^2 = 2 \sum_{j=1}^q \theta_j \cos[\lambda(v - j)].$$

Then, this implies the results for  $\partial/\partial \phi_u \log f(\lambda)$  and  $\partial/\partial \theta_v \log f(\lambda)$ . For  $\partial/\partial d \log f(\lambda)$  is direct.

### 2.3 Autocovariance function

We illustrate a method to compute the exact autocovariance function for the general ARFIMA( $p, d, q$ ) process using the Sowell algorithm. Considering the parameterizations of the autocovariance function derived by writing the spectral density (6) in terms of parameters of the model given by Sowell (1992), the autocovariance function of a general ARFIMA( $p, d, q$ ) process is given by

$$\gamma(h) = \frac{1}{2\pi} \int_0^{2\pi} f(\lambda) e^{-i\lambda h} d\lambda. \tag{11}$$

Particularly, the autocovariance and autocorrelation functions of the fractionally differenced ARFIMA(0,  $d$ , 0) process are given by

$$\gamma_0(h) = \sigma^2 \frac{\Gamma(1 - 2d)}{\Gamma(1 - d)\Gamma(d)} \frac{\Gamma(h + d)}{\Gamma(1 + h - d)} \quad \text{and} \quad \rho_0(h) = \frac{\Gamma(1 - d)}{\Gamma(d)} \frac{\Gamma(h + d)}{\Gamma(1 + h - d)},$$

respectively. Then, the polynomial  $\Phi(B)$  in (1) may be written as

$$\Phi(B) = \prod_{i=1}^p (1 - \rho_i B).$$

Under the assumption that all the roots of  $\phi(B)$  have multiplicity one, it can be deduced from (11) that

$$\gamma(h) = \sigma^2 \sum_{i=-q}^q \sum_{j=1}^p \psi(i) \xi_j C(d, p + i - h, \rho_j).$$

with

$$\begin{aligned} \xi_j &= \left[ \rho_j \prod_{i=1}^p (1 - \rho_i \rho_j) \prod_{k \neq j} (\rho_j - \rho_k) \right]^{-1}, \\ C(d, h, \rho) &= \frac{\gamma_0(h)}{\sigma^2} [\rho^{2p} \beta(h) + \beta(-h) - 1], \\ \beta(h) &= F(d + h, 1, 1 - d + h, \rho), \\ F(a, b, c, x) &= 1 + \frac{a \cdot b}{c \cdot 1} x + \frac{a \cdot (a + 1) \cdot b \cdot (b + 1)}{c \cdot (c + 1) \cdot 1 \cdot 2} x^2 + \dots \end{aligned}$$

where  $F(a, b, c, x)$  is the Gaussian hypergeometric function (e.g. [Gradshteyn and Ryzhik 2007](#)). The term  $\psi(i)$  presented here and in [Palma \(2007, pp. 47–48\)](#) is a corrected version with respect to [Sowell \(1992\)](#) and is

$$\psi(i) = \sum_{k=\max(0,i)}^{\min(q,q+i)} \theta_k \theta_{k-i}.$$

In the absence of AR parameters the formula for  $\gamma(h)$  reduces to

$$\gamma(h) = \sigma^2 \sum_{i=-q}^q \psi(i) \frac{\Gamma(1 - 2d)\Gamma(h + d - i)}{\Gamma(1 - d)\Gamma(d)\Gamma(1 + i - d - h)}.$$

On the other hand, the findings of [Hassler and Kokoszka \(2010\)](#) describe the asymptotic behavior of the autocovariance function  $\gamma(h)$  as

$$\gamma(h) \sim c_\gamma |h|^{2d-1}, \tag{12}$$

where  $c_\gamma = \sigma^2 \pi^{-1} \Gamma(1 - 2d) \sin(\pi d) \left( \sum_{j=0}^\infty \psi_j \right)^2$  for large  $|h|$ . Let  $\{y_1, y_2, \dots, y_n\}$  be a sample from the process in (1) and let  $\bar{y}$  be the sample mean. The exact variance of  $\bar{y}$  is given by

$$\text{Var}(\bar{y}) = \frac{1}{n} \left[ 2 \sum_{j=1}^{n-1} \left( 1 - \frac{j}{n} \right) \gamma(j) + \gamma(0) \right].$$

By (12) and for large  $n$ , we have the asymptotic variance formula  $\text{Var}(\bar{y}) \sim n^{2d-1} c_\gamma / d(2d + 1)$ . Additionally, the method developed by Lieberman and Phillips (2008) considers an infinite-order asymptotic expansion for the autocovariance function of a general stationary long-memory process.

### 2.4 Impulse response functions

The impulse response functions (IRF) is the most commonly used tool to evaluate the effect of shocks on time series. Among the several approximations to compute this, we consider the theory proposed by Hassler and Kokoszka (2010) to find the IRF of a process  $\{y_t\}$  following an ARFIMA( $p, d, q$ ) model. The properties of these approximations, depend on whether the series are assumed to be stationary according to Theorem 2.1. Under the assumption that the roots of the polynomials  $\Phi(B)$  and  $\Theta(B)$  are outside the closed unit disk and  $d \in (-1, 1/2)$ , the process  $\{y_t\}$  is stationary, causal and invertible. In this case, we can write  $y_t = \Psi(B)\varepsilon_t$  where  $\Psi(B)$  represents the expansion of the MA( $\infty$ ) coefficients denoted as  $\psi_j$  with  $j > 1$ . These coefficients satisfy the asymptotic relationship  $\psi_j \sim \Theta(1)j^{d-1}/[\Phi(1)\Gamma(d)]$  as  $j \rightarrow \infty$  (Kokoszka and Taqu 1995), where  $\Theta(1) = 1 + \sum_{i=1}^q \theta_i$  and  $\Phi(1) = 1 - \sum_{i=1}^p \phi_i$ . As a particular case, we have that the  $\psi_j$  coefficients for an ARFIMA(0,  $d$ , 0) are given in closed form by  $\psi_j = \Gamma(j+d)/[\Gamma(j+1)\Gamma(d)]$ . Now, from (2) and the Wold expansion (5), the process (1) has the expansion  $(1 - B)^{-d}y_t = \sum_{j=0}^\infty \eta_j y_{t-j} = \sum_{j=0}^\infty R_j \varepsilon_{t-j}$ , where  $R_j$  is the so-called IRF and is given by

$$R_j = \sum_{i=0}^j \psi_i \eta_{j-i}. \tag{13}$$

The terms  $\eta_j$  can be represented in recursive form using (3) as  $\eta_j = \eta_{j-1}(1 + (d - 1)/j)$ , for  $j \geq 1$  and  $\eta_0 = 1$ . From the asymptotic expression given in (4) and assuming that  $\sum_{j=0}^\infty \psi_j < \infty$ , we have the following asymptotic representation

$$R_j \sim \frac{j^{d-1}}{\Gamma(d)} \sum_{i=0}^\infty \psi_i \tag{14}$$

as  $j \rightarrow \infty$  and  $\psi_j / (j^{d-1}) \rightarrow 0$ .



### 2.5 Predictive ability test

One approach to compare prediction models is through their root mean square error (RMSE). Under this paradigm and given two forecasting methods, the one that presents the lower RMSE is the better of the two. To compare statistically the differences of predictive ability among two proposed models, we focus here on the evaluation paradigm proposed by White (2006, GW). This test aims to evaluate a prediction method but not to carry out a diagnostic analysis. Therefore, it does not consider the parametric uncertainty, which is useful if we want to compare nested models from an ARFIMA model. The GW test attributed to Diebold and Mariano (1995) is based on the differences  $\Delta L_i = |\hat{x}_i - y_i| - |\hat{z}_i - y_i|$ , where  $\hat{x}_i$  and  $\hat{z}_i$  are the forecasted observations of the first and second model respectively, for  $i = 1, \dots, n$ . The null hypothesis for the GW test associated with expected difference  $E[\Delta L]$  of the two prediction models is  $H_0: E[\Delta L] = 0$ , whereas the alternative is  $H_1: E[\Delta L] \neq 0$ . These hypotheses are tested by means of the statistic,

$$\Delta \hat{L}(N) = \frac{1}{N \sqrt{\frac{\hat{\sigma}_N^2}{N}}} \sum_{i=t_0}^{n-\tau} \Delta L_i,$$

where  $N = n - \tau - t_0 + 1$ ,  $n$  is the total size of the sample,  $\tau$  is the prediction horizon, and  $t_0$  is the observation at which the moving windows start. Note that under  $H_0$ , the statistic  $\Delta \hat{L}(N)$  is asymptotically normal. For  $\tau = 1$ , an estimator of  $\hat{\sigma}_N$  can be obtained from the estimation of  $\hat{\sigma}_{\hat{a}}$  from a simple regression  $\Delta \hat{L}(N) = \hat{a} + \epsilon$ , where  $\hat{a}$  is the estimated intercept and  $\epsilon$  is the error term of the simple regression model. However, for horizons  $\tau > 1$ , it is possible to apply a heteroscedasticity and autocorrelation consistent (HAC) estimator; for example, Newey and West (1987) or Andrews (1991).

### 3 Statistical software

The estimation of the fractionally, autoregressive, and moving-average parameters has been studied by several authors (Haslett and Raftery 1989; Beran 1994; Hyndman and Khandakar 2008). A widely used method is the approximate MLE method of Haslett and Raftery (1989). In our study, estimation of the ARFIMA( $p, d, q$ ) model using the corresponding Whittle method is described in Sect. 2.1 and this model is fitted by using the `arfima.whittle()` function.

The implementation of the `arfima.whittle()` function is presented in Sect. 3.3. In Sect. 3.1 we present some Monte-Carlo simulations to the `afmtools` package for R statistical environment (R Development Core Team 2012). Later, in Sect. 3.2, in order to test the performance of the estimators proposed in Sect. 2, we consider an application to real-life time series.

### 3.1 Monte-Carlo analysis

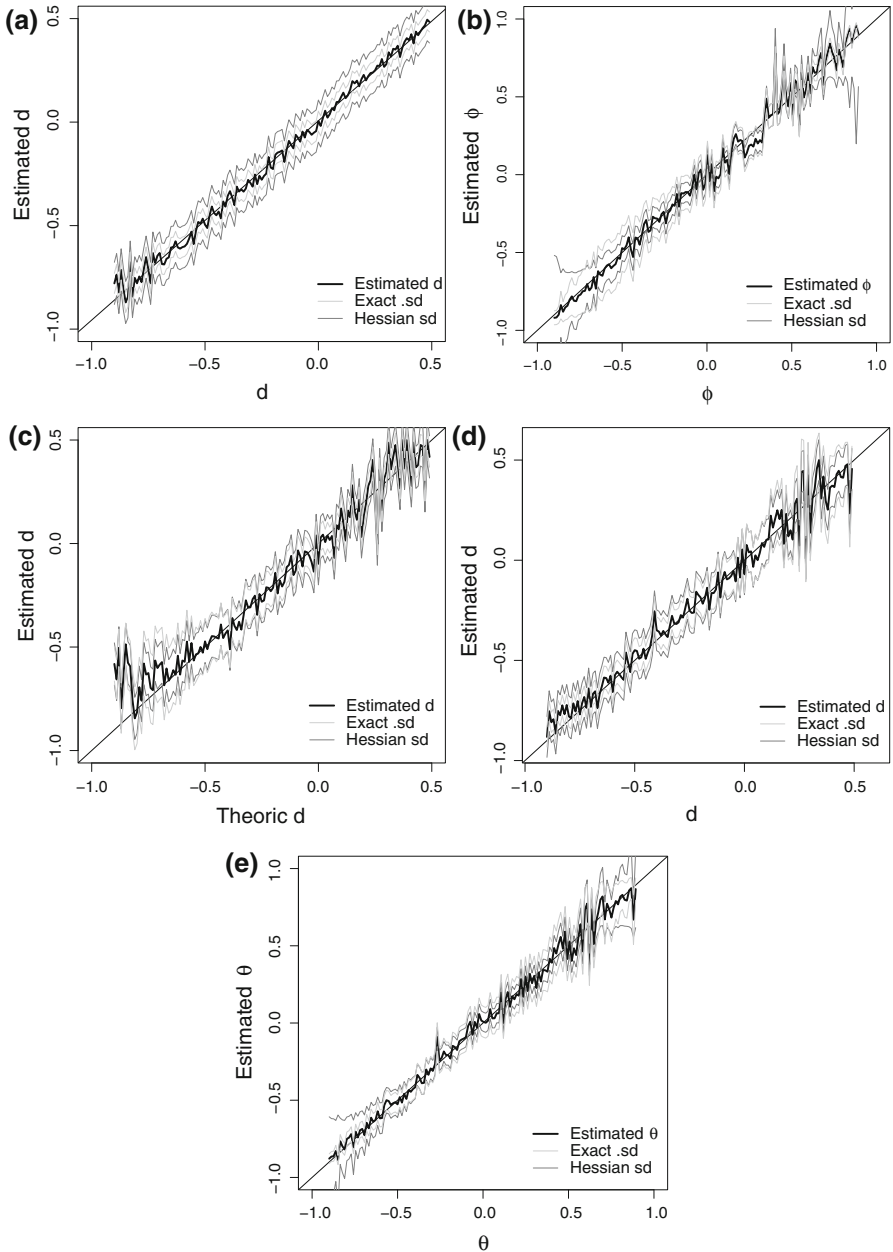
Figure 1 shows several simulation results assessing the performance of the Whittle estimators of  $d$ , AR, and the MA parameters, for different ARFIMA models. These plots include the exact and Hessian standard deviations. According to the definition of the ARFIMA model, the simulations are run in the interval  $(-1, 0.5)$  for  $d$ . The plots show a similar behavior for the estimators with respect to the theoretical parameters, except for the extreme values of the ARMA parameters near  $-1$  and  $1$ . Consequently, the confidence intervals tend to be larger than the other values of  $\phi$  and  $\theta$  parameters for plots (b) and (e). Figure 2 shows some simulation results regarding the log-likelihood behavior for the cases  $d = \{-0.9, -0.6, -0.3, 0, 0.25, 0.45\}$  with a rectangular grid  $\phi \times \theta = (-0.9, 0.9) \times (-0.9, 0.9)$  for the ARFIMA(1,  $d$ , 1) model using the `arfima.whittle.loglik()` function. The plots present low values of the likelihood function for the values of  $\phi$  and  $\theta$  closed to 0, especially for the plots (c)–(f) when  $d = \{-0.3, 0, 0.25, 0.45\}$ . However, the plots (a)–(d) show high values of the likelihood function when this is evaluated for the points near  $\phi = -0.9$  and  $\theta = 0.9$ . For the plots (e)–(f), the behavior is inverse, i.e., the likelihood function tends to be higher for values near  $\phi = 0.9$  and  $\theta = -0.9$ .

These heatmaps plots are performed by the function `heatmap.2()` (`gplots` package) using the vectors `phi`, `theta`, and the `A` matrix, which contain the estimated log-likelihoods in each simulation and is obtained by the instruction

```
R> phi = theta = seq(-0.9, 0.9, 0.2) # grid for ARMA parameters
R> d = 0.45 # fractional differential parameter
R> A = matrix(NA, length(phi), length(theta))
R> colnames(A) = phi
R> rownames(A) = theta
+
R> for (j in 1:length(phi)) {
+   for (k in 1:length(theta)) {
+     parameters = c(d, phi[j], theta[k]) # simulation:
+     sample = fracdiff.sim(n = 1000, ar = phi[j],
+     ma = theta[k], d = d) # loglikelihood estimation:
+     loglik = arfima.whittle.loglik(theta = parameters,
+     series = sample$series, nar = 1, nma = 1)
+     A[j, k] = loglik$L
+   }
+ }
+
R> heatmap.2(A, dendrogram = 'none', Colv = FALSE, Rowv = FALSE,
+ col = gray(1:20/20), scale = 'none', key = TRUE, density.info = 'none',
+ breaks = seq(-10,10,1), trace = 'none', symm = F, symkey = T,
+ symbreaks = T, main = '(f) d=0.45')
+ text(0.55, 0.9, expression(phi), ps = 2, cex = 1.5)
+ text(0.1, 0.4, expression(theta), ps = 2, cex = 1.5)
```

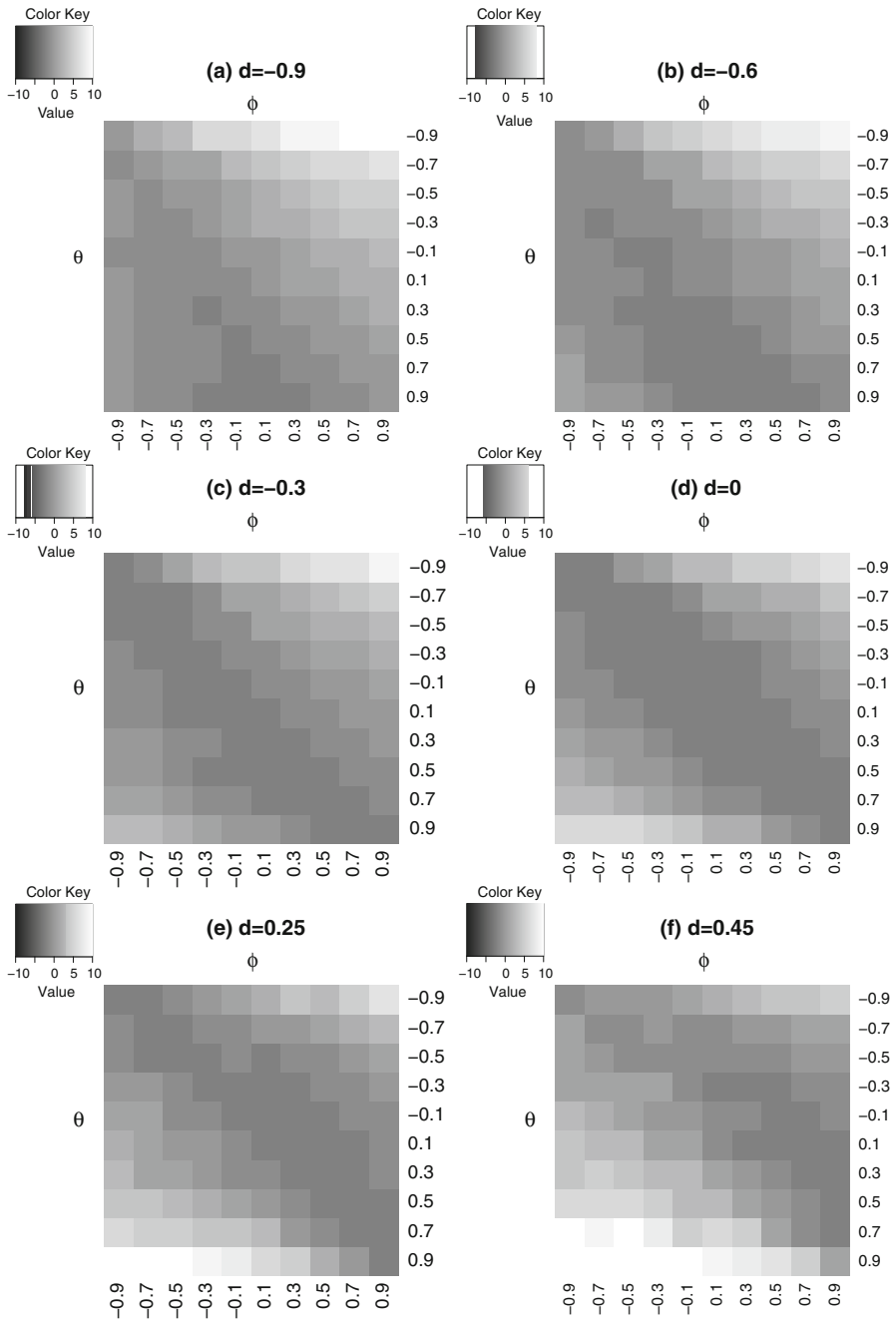
### 3.2 Application

We illustrate the performance of the `afmtools` package by applications to real-life time series `TreeRing` (Statlib Data Base, <http://lib.stat.cmu.edu/>) displayed in Fig. 3

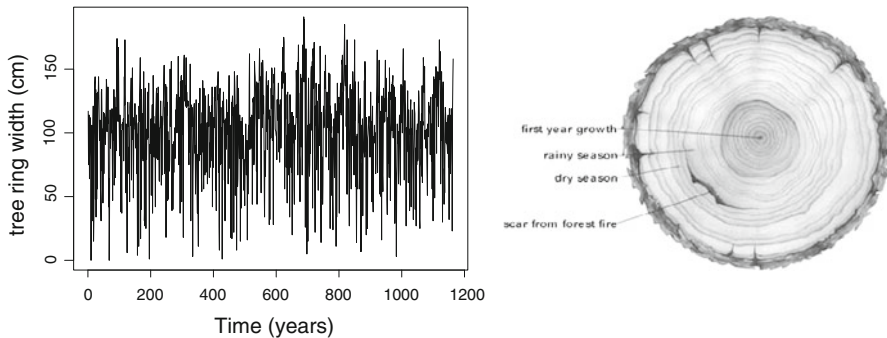


**Fig. 1** Dispersion plots between estimated and theoretical parameters of **a** ARFIMA(0,  $d$ , 0), **b**, **c** ARFIMA(1,  $d$ , 0) and **d**, **e** ARFIMA(0,  $d$ , 1)

left. It is very important to analyze this kind of data because this allows us to explore rainy and dry seasons in the study area. [Hipel and McLeod \(1994\)](#) have studied this time series to determine the range of possible growths for the upcoming years of the



**Fig. 2** Log-Likelihood heatmaps for ARFIMA(1,  $d$ , 1) model using a grid  $\phi \times \theta = (-0.9, 0.9) \times (-0.9, 0.9)$



**Fig. 3** *Left* Tree Ring Data Base. *Right* Illustration of the observed climatic episodes and tree ring growth. Image source:<http://summitcountyvoice.com>

trees using ARMA and ARIMA models. On the other hand, this time series displays a high persistence in its observations and has been analyzed by Palma and Olea (2010) and Palma et al. (2013) with a locally stationary approach. The illustrated growth of the trees represented by the number of the rings, displays a long-range dependence of its observations along the observations for ages and seasons (see Fig. 3, right). For these reasons, we model the Tree Ring widths time series using long-memory models; specifically, the ARFIMA models are used to estimate, diagnose, and compare forecasts of the number of tree rings for upcoming years.

In order to illustrate the usage of package functions, we consider a fitted ARFIMA(1,  $d$ , 1) model. For this model, we have implemented the Whittle algorithm and computed the exact variance-covariance matrix to compare with the Hessian method. Afterward, we compare the Sowell method for computing the ACVF function with the sample ACVF. Other functions have also been implemented and illustrated in this section.

### 3.3 Whittle algorithm

To apply the Whittle algorithm to the `TreeRing` time series as an example, we use the following command considering an ARFIMA(1,  $d$ , 1) model:

```
R> data(TreeRing)
R> y = TreeRing
R> model = arfima.whittle(series = y, nar = 1, nma = 1, fixed = NA)
```

Note that the option `fixed` (for fixing parameters to a constant value) has been implemented. This option allows the user to fix the parameters  $d$ ,  $\phi_1$ , or  $\theta_1$ , in order of occurrence. For example, in our ARFIMA(1,  $d$ , 1) model, we can set the parameter  $d$  to be equal to zero. Consequently, we obtain the estimation of a simple ARMA(1, 1) model. The object `model` is of class `arfima` and provides the following features:

- estimation of  $d$  and ARMA parameters;
- standard errors obtained by the Hessian method and the respective `t` value and `Pr(>|t|)` terms;

- the log-likelihood function performed in the `arfima.whittle.loglik()` function;
- innovation standard deviation estimated by the ratio of the theoretical spectrum;
- residuals from the fitted model.

The commands `plot()`, `residuals()`, `summary()`, `tsdiag()` and `print()` have been adapted to this model class of S3 method. The `summary()` option shows the estimated parameters, the Hessian standard deviations, the  $t$ -statistic, and their respectively  $p$  values. The computation of the long-memory parameter  $d$  as well as the autoregressive  $\{\phi_1, \dots, \phi_p\}$  and moving average  $\{\theta_1, \dots, \theta_q\}$  parameters can be handled quickly for moderate sample sizes. Printing the `model` object by the `summary()` function shows the items mentioned before as

```
R> summary(model)
$call
arfima.whittle(series = y, nar = 1, nma = 1)

$coefmat
      Estimate Std. Error  t value  Pr(>|t|)
d      0.1058021  0.04813552  2.198004 0.02794879
phi 1   0.3965915  0.03477914 11.403142 0.00000000
theta 1 -0.2848590  0.03189745 -8.930462 0.00000000

$sd.innov
[1] 35.07299

$method
[1] 'Whittle'

attr(,"class")
[1] 'summary.arfima'
```

### 3.4 Exact variance-covariance matrix

The `var.afm()` function shows the exact variance-covariance matrix and the standard deviations. The computation of the integrals of (10) is carried out by using the Quadpack numeric integration method (Piessens et al. 1983) implemented in the `integrate()` function (`stats` package). Note that the functions involved in these integrals diverge in the interval  $\lambda = [-\pi, \pi]$ . However, they are even functions with respect to  $\lambda$ . Thus, we integrate over  $[0, \pi]$  and then multiply the result by two. Now, by using the central limit theorem discussed in Sect. 2.2, we can obtain the asymptotic approximation of the estimated parameters standards errors  $SE(\hat{\Omega})_i = (n^{-1}[\hat{\Sigma}^{-1}]_{ii})^{1/2}$  of an ARFIMA model, where  $\hat{\Omega} = (\hat{d}, \hat{\phi}_1, \dots, \hat{\phi}_p, \hat{\theta}_1, \dots, \hat{\theta}_q)$  and  $[\hat{\Sigma}^{-1}]_{ii}$  corresponds to the  $i$ th diagonal components of the matrix  $\hat{\Sigma}^{-1}$  for  $i = \{1, \dots, p + q + 1\}$ .

By using the Whittle estimators, we search for the lowest AIC (Akaike Information Criterion, Akaike 1974) given by  $AIC(\hat{\omega}) = -2[\log L(\hat{\omega}) - (p + q + 1)]$  over a class of ARFIMA models with  $p, q \in \{0, 1, 2\}$ , where  $\hat{\omega}$  is a subset of  $\hat{\Omega}$  and  $L(\hat{\omega})$  is the likelihood associated with  $\hat{\omega}$ . From Table 2, we can see that the fractionally differenced

model ARFIMA(0,  $d$ , 0) has the lowest AIC. Candidate models are marked in bold in Table 2. This table is obtained by the instruction

```
R> for (i in 0:2) {
R>   for (j in 0:2) {
+     mod = arfima.whittle(y, nar = i, nma = j)
+     theta = c(mod$d, mod$ar, mod$ma)
+     mll = arfima.whittle.loglik(theta, series = y, nar = i, nma = j)
+     AIC = - 2 * ( mll$L - (1+i+j) )
+     coef = summary.arfima(mod)$coefmat[1, 4]
+     res = c(i, j, AIC, mod$d, coef)
+   }
+ }
```

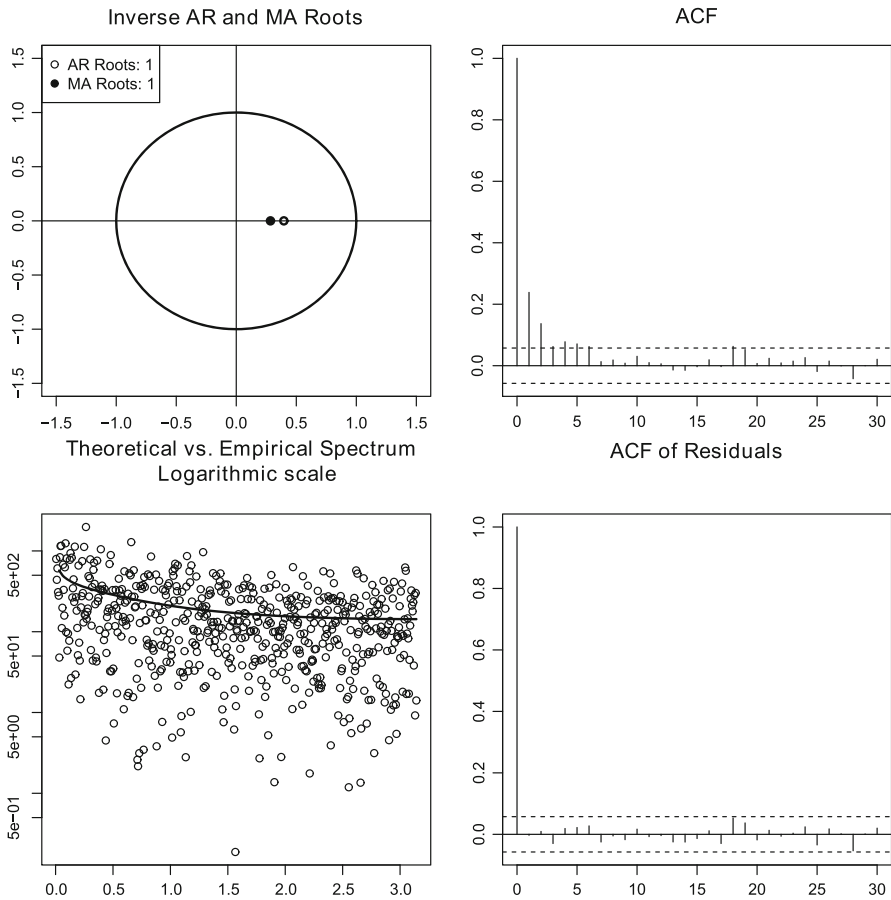
Additionally, we propose a technique for obtaining the spectral density associated with ARFIMA and ARMA processes in `spectrum.arfima()` and `spectrum.arma()`, respectively. This is done by using the `polyroot()` function of the `polynom` package to compute the roots of the polynomials  $\Phi(e^{-i\lambda})$  and  $\Theta(e^{-i\lambda})$ . Both functions need the estimated ARFIMA parameters and the estimation of the innovation standard deviation (`sd.innov`) given by an object of `arfima` class. For the spectrum density and periodogram, see Sects. 2 and 2.1, respectively. Since the calculation of the FFT has a numerical complexity of the order  $O[n \log_2(n)]$ , this approach produces a very fast algorithm to estimate the parameters. It is possible to obtain the FFT through the `fft()` function based on the method proposed by Singleton (1979).

### 3.5 Diagnostic functions

We have also implemented a very practical function called `check.parameters.arfima()`. This verifies whether the long-memory parameter  $d$  belongs to the interval  $(-1, 0.5)$  and whether the roots of the fitted ARMA parameters lie outside the unit disk. This function was incorporated in the `plot()` command. In the first plot of Fig. 4, we can see that the roots of the AR and MA polynomials lie outside the unit disk, according to the assumptions of stationarity solutions of (1) presented in Theorem 2.1 (see Sect. 2). Alternatively, the `check.parameters.arfima()` that takes an `arfima`-class object, gives TRUE/FALSE-type results indicating whether the parameters pass the requirement for a stationary process.

Additionally, an adaptation of the function `tsdiag()` can be found in this package. This is implemented in an S3-type `arfima` class method and shows three plots for analyzing the behavior of the residual from the fitted ARFIMA model. This function has additional arguments such as the number of observations `n` for the standardized residuals and critical  $p$  value `alpha`. Figure 5 illustrates these results, where, the residuals are white noise at a confidence level of  $\alpha = 0.05$ .

The exact Sowell autocovariance computation obtained by `rho.sowell()` and the sample autocorrelation obtained by the `ACF()` command are applied to tree ring time series (see Fig. 6). The function `rho.sowell()` requires the specification of an object of class `arfima` in the `object` option that, by default, is `NULL`. But, if `object=NULL`, the user can incorporate the ARFIMA parameters and the

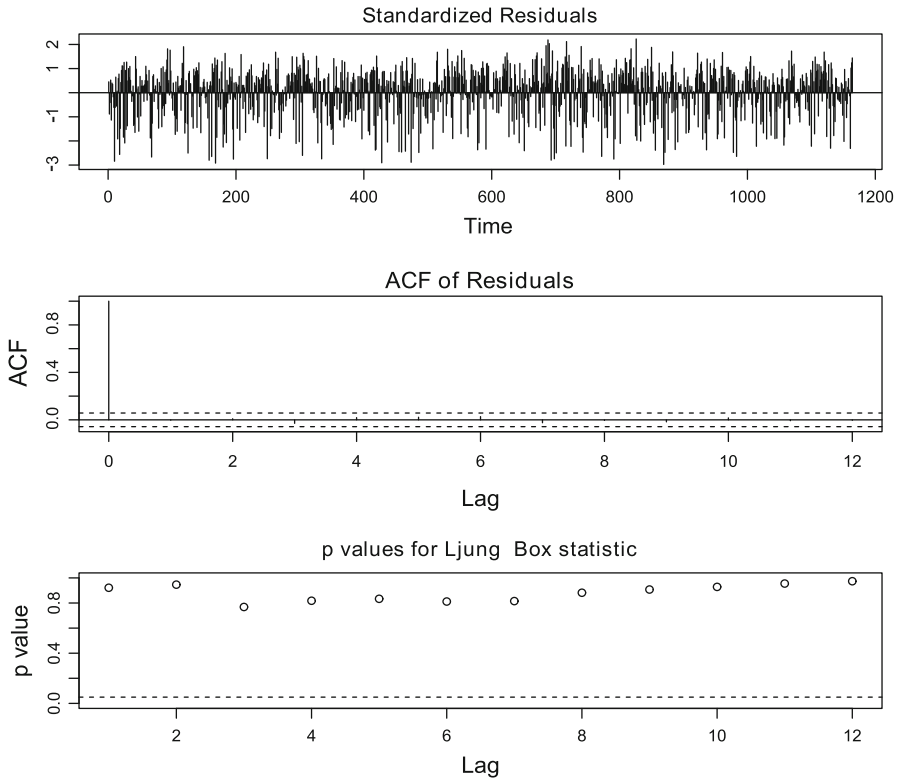


**Fig. 4** Diagnostic Plots made by the `plot()` command. *Left* Plots of unit roots circle along with the root provided by AR and MA polynomials and Theoretical (thick line) versus Empirical Spectrum (black points) plot. *Right* ACF plots of Tree Rings and ARFIMA(1,  $d$ , 1) model residuals. The dotted lines correspond to the  $\{\mp 2/\sqrt{n}\}$  significance level for the autocorrelations

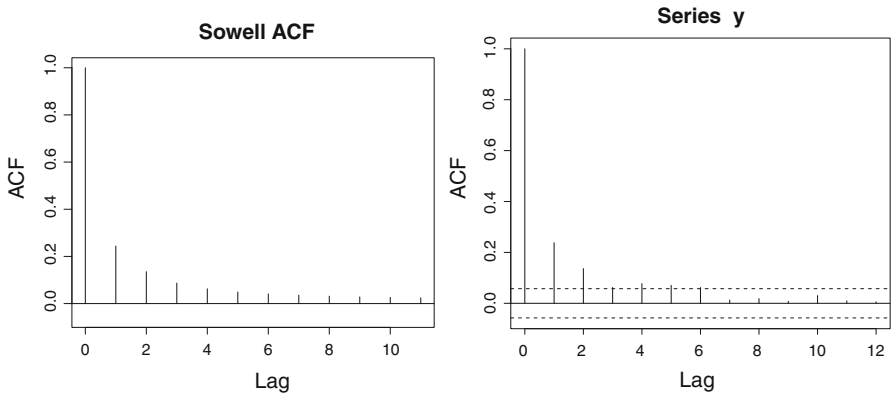
innovation variance. Alternatively, the implemented `plot` option gives a graphical result similar to the `ACF()` command in the sample autocorrelation. We can see the similarity of both results for the discussed model. The ACVF implementation is immediate but, for the calculation of the Gaussian hypergeometric functions, we use the `hypergeo()` function from the `hypergeo` package. For values of  $h > 50$ , we use the approximation (12) reducing considerably the computation time as compared to the Sowell algorithm. On the other hand, the `rho.sowell()` function is required by the `smv.afm()` function. The `smv.afm()` function calculates the variance of the sample mean of an ARFIMA process. When the argument `comp` is `TRUE`, the exact variance of the sample mean is calculated, and when `comp` is `FALSE`, the asymptotic variance is calculated.

The  $R_j$  function (IRF) is illustrated in Fig. 7. This decays exponentially fast, at a rate of  $j^{d-1}$  because, these functions inherit the behavior of  $\eta_j$ . This behavior is



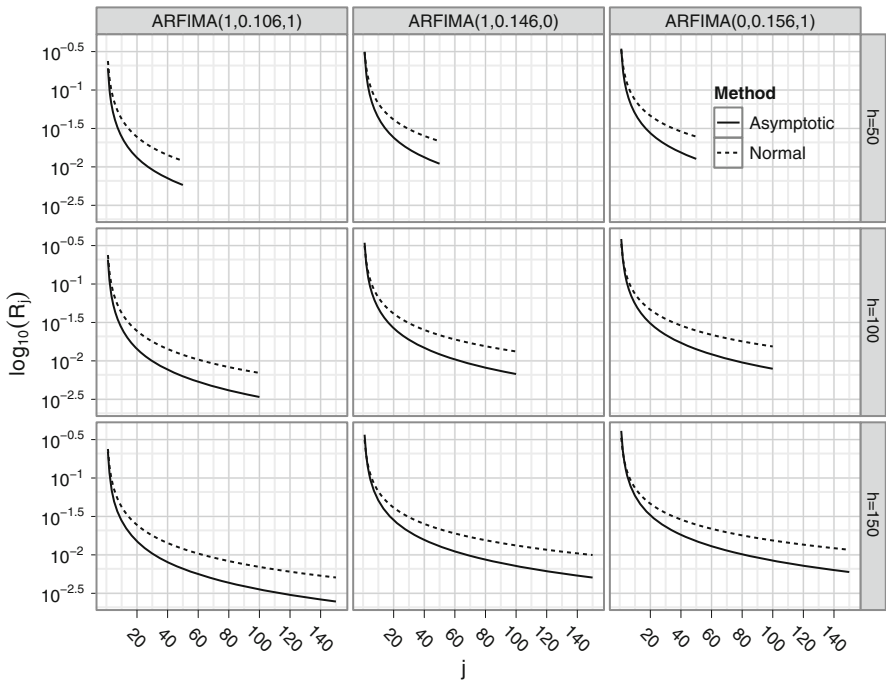


**Fig. 5** Plots of residuals analysis from `tsdiag` standard command adapted to ARFIMA model residuals. The dotted lines of the second plot correspond to the  $\{\pm 2/\sqrt{n}\}$  significance level for the autocorrelations. The dotted line of the third plot is the probability (0.05) related to a 5% confidence level



**Fig. 6** Theoretical (left) and Empirical (right) ACF of selected ARFIMA model. The dotted lines correspond to the  $\{\pm 2/\sqrt{n}\}$  significance level for the autocorrelations

typical for ARFIMA models, as reported by Hassler and Kokoszka (2010), Kokoszka and Taquu (1995) and Hosking (1981). Figure 7 shows some  $R_j$  curves associated with the three models considered in Table 1 for the asymptotic method by formula



**Fig. 7** Plots of  $\log_{10}(R_j)$  functions for  $h = 50, 100$  and  $150$  lags

**Table 1** Summary of estimated parameters for several ARFIMA models

Model	Parameter	Estimates	Hessian	Exact
FN( $d$ )	$d$	0.195	0.048	0.023
ARFIMA( $1,d,0$ )	$d$	0.146	0.048	0.038
	$\phi$	0.072	0.029	0.049
ARFIMA( $0,d,1$ )	$d$	0.156	0.048	0.035
	$\theta$	0.059	0.029	0.045
ARFIMA( $1,d,1$ )	$d$	0.106	0.048	0.063
	$\phi$	0.397	0.035	0.282
	$\theta$	-0.285	0.032	0.254

(14) (labeled *Asymptotic* in the plot) and the counterpart method by formula (13) (labeled *Normal* in the plot). Note that for a large value of  $j \approx 50$ , both methods tend to converge, and, the curves make an inflexion in the value  $j \approx 10$ . Note that the *Asymptotic* approximation tends to be equal to the *Normal* method in the measure that the input  $h$  lag increases (see plots for  $h=150$  in Fig. 7). These IRFs are available in the function `ir.arfima()`, with arguments  $h$  to evaluate the IRFs over a particular  $h$  lag and, model for an object `arfima.whittle`. The `ir.arfima()` function produces the vectors `RE` and `RA` for *Normal* and *Asymptotic* IRFs.

**Table 2** Akaike’s criterion for several ARFIMA models with  $p$  values obtained for the Hessian standard deviation

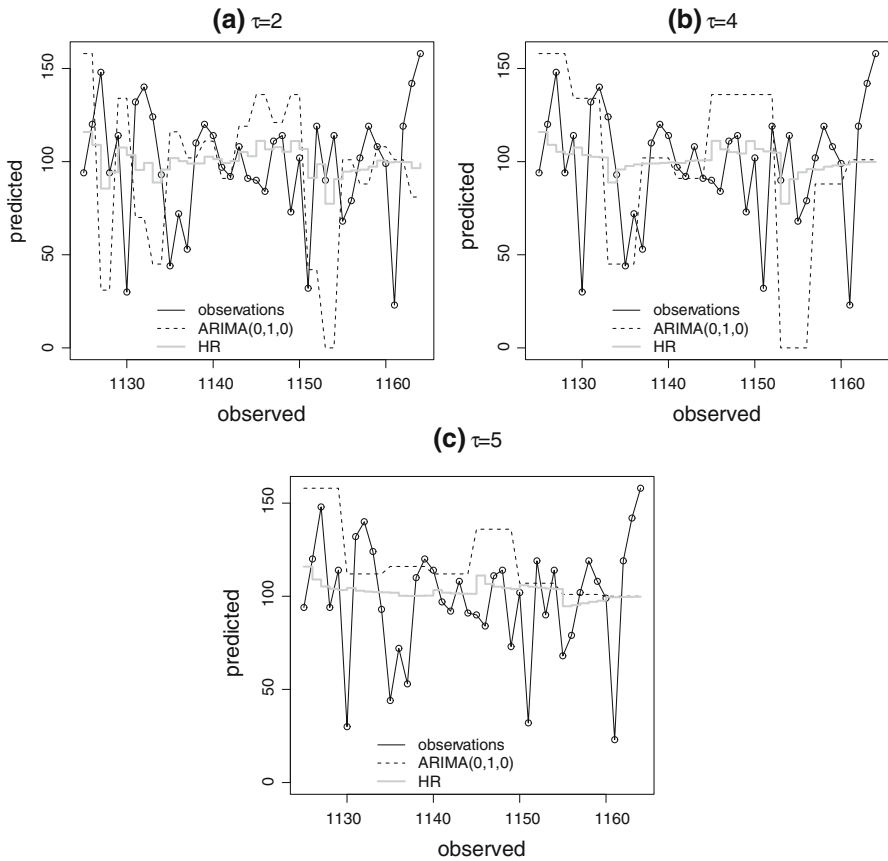
$p$	$q$	AIC	$\hat{d}$	$p$ value
0	0	−37.44	0.196	0
0	1	−35.44	0.156	0.001
0	2	−33.44	0.113	0.018
1	0	−35.44	0.146	0.002
1	1	−33.44	0.106	0.028
1	2	−31.44	0.142	0.003
2	0	−33.44	0.111	0.021
2	1	−31.44	0.130	0.007
2	2	−29.44	0.191	0

### 3.6 Forecasting evaluations

The GW method implemented in `gw.test()` for evaluating forecasts proposed by [Giacomini and White \(2006\)](#) compares two vectors of predictions,  $x$  and  $y$ , provided by two time series models and a data set  $p$ . We consider that it is relevant to implement this test to determine if the predictions produced by a time series model (e.g., ARFIMA) process good forecasting qualities. This *test for predictive ability* is of particular interest since it considers the `tau` prediction horizon parameter or ahead in the case of `pred.arfima()` function. Alternative methods are discussed, for instance, by [Diebold and Mariano \(1995\)](#). If `tau=1`, the *standard statistic simple regression estimator* method is used. Otherwise, for values of `tau` larger than 1, the method chosen by the user is used in the `method` option. The available methods for selection are described below. They include several Matrix Covariance Estimation methods but, by default, the HAC estimator is used in the test. The user can select between the several estimators of the `sandwich` package mentioned before:

- HAC: Heteroscedasticity and Autocorrelation Consistent (HAC) Covariance Matrix Estimation by `vcovHAC()` function ([Zeileis 2004, 2006](#)).
- NeweyWest: Newey-West HAC Covariance Matrix Estimation by `NeweyWest()` function ([Newey and West 1987](#)).
- LumleyHeagerty: Weighted Empirical Adaptive Variance Estimation by `weave()` function ([Lumley and Heagerty 1999](#)).
- Andrews: Kernel-based HAC Covariance Matrix Estimation by `kernHAC()` function ([Andrews 1991](#); [Andrews and Monahan 1992](#)).

This test gives the usual results of the general test implemented in such as the GW statistic in `statistic`, the alternative hypothesis in `alternative`, the  $p$  value in `p.value`, others such as the method mentioned before in `method`, and the name of the data in `data.name`. In some studies, the GW test is used to compare selected models versus benchmark models such as ARMA, ARIMA, or SARIMA models (e.g. [Contreras-Reyes and Idrovo 2011](#)). To illustrate the GW test performance, we simulate the out-of-sample prediction exercise through moving windows for `TreeRing` data sets considering the first 1124 observations and, later, forecasting 40 observations using three models: the ARIMA(0,1,0) benchmark model, the ARFIMA( $p, d, q$ ) with



**Fig. 8** Plots of out-of-sample predictions for two models: ARIMA(0,1,0) and ARFIMA(1,  $d$ , 1) with Haslett and Raftery estimator (HR)

MLE estimator, and the Haslett & Raftery estimator (HR) using the algorithm of the automatic `forecast()` function implemented in the `forecast` package by Hyndman and Khandakar (2008) algorithm. In Fig. 8, the GW test compares the out-of-sample predictions of the ARIMA(0,1,0) and ARFIMA( $p,d,q$ ) model. It is important to note that the goal of this test is only to compare prediction abilities between models.

Finally, we study a more general simulation, comparing the three predictors vectors with 40 real observations using `gw.test()` function considering the hypotheses testing `alternative="two.sided"` to contrast significant differences between predictions. In addition, we consider the four HAC estimators mentioned in the beginning of this section and prediction horizon parameters  $\tau = \{2, 4, 5\}$ . The results are summarized in Table 3 and Fig. 8. We can see that the differences in the prediction ability between B versus MLE and B vs. HR are significant for  $\tau = 2$  and 4 but, between MLE and HR they are not unequal for the three considered values of  $\tau$ . Given the non-significance of the MLE-HR test  $p$  value, the MLE model is not considered in Fig. 8. Note that when the prediction horizon increases, the difference in forecasting

**Table 3** Summary of  $p$  values of the GW test for each HAC estimator,  $\tau = \{2, 4, 5\}$  prediction horizon parameters and for the estimator methods B (Benchmark model), ML (ARFIMA models using Maximum Likelihood estimator) and HR (ARFIMA models using Haslett & Raftery estimator) over 40 observations of samples from the TreeRing data set

Estimator	$\tau$	Model	B	ML	HR
HAC	2	B	–	<b>0.001</b>	<b>0.001</b>
		ML	<b>0.001</b>	–	0.307
		HR	<b>0.001</b>	0.307	–
	4	B	–	<b>0.031</b>	<b>0.031</b>
		ML	<b>0.031</b>	–	0.559
		HR	<b>0.031</b>	0.559	–
	5	B	–	0.126	0.126
		ML	0.126	–	0.689
		HR	0.126	0.689	–
Newey and West	2	B	–	<b>0.002</b>	<b>0.002</b>
		ML	<b>0.002</b>	–	<b>0.005</b>
		HR	<b>0.002</b>	<b>0.005</b>	–
	4	B	–	0.051	0.051
		ML	0.051	–	0.281
		HR	0.051	0.281	–
	5	B	–	0.174	0.174
		ML	0.174	–	0.479
		HR	0.174	0.479	–
Lumley and Heagerty	2	B	–	<b>0.001</b>	<b>0.001</b>
		ML	<b>0.001</b>	–	0.486
		HR	<b>0.001</b>	0.486	–
	4	B	–	<b>0.023</b>	<b>0.023</b>
		ML	<b>0.023</b>	–	0.652
		HR	<b>0.023</b>	0.652	–
	5	B	–	0.096	0.096
		ML	0.096	–	0.762
		HR	0.096	0.762	–
Andrews	2	B	–	<b>0.003</b>	<b>0.003</b>
		ML	<b>0.003</b>	–	<b>0.018</b>
		HR	<b>0.003</b>	<b>0.018</b>	–
	4	B	–	<b>0.041</b>	<b>0.041</b>
		ML	<b>0.041</b>	–	0.283
		HR	<b>0.041</b>	0.283	–
	5	B	–	0.206	0.206
		ML	0.206	–	0.456
		HR	0.206	0.456	–

The  $p$  values marked in bold are lower than the probability (0.05) related to a 5% confidence level

performance between both models, measured in terms of mean absolute error, tends to decrease. Consequently, the individual prediction performance of each model is not considered by the test.

## 4 Conclusions

We developed the `afmtools` package with the goal of incrementing the necessary utilities to analyze the ARFIMA models and, consequently, it is possible to execute several useful commands already implemented in the long-memory packages. In addition, we have provided the theoretical results of the Whittle estimator, which were applied to the Tree Ring data base. Furthermore, we have performed a brief simulation study for evaluating the estimation method used herein and also have evaluated the properties of its log-likelihood function. The numerical examples shown here illustrate the different capabilities and features of the `afmtools` package; specifically, the estimation, diagnostic, and forecasting functions. The `afmtools` package would be improved by incorporating other functions related to change-point models and tests of unit roots, as well as other important features of the models related to long-memory time series.

**Acknowledgments** Wilfredo Palma would like to thank the support from Fondecyt Grant 1120758. The authors thank the editor, the co-editor and an anonymous referee for their helpful comments and suggestions.

## References

- Akaike H (1974) A new look at the statistical model identification. *IEEE Trans Automat Contr* 19(6):716–723
- Andrews D (1991) Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 59(3):817–858
- Andrews D, Monahan JC (1992) An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica* 60(4):953–966
- Beran J (1994) *Statistics for long-memory processes*. Chapman & Hall, New York
- Bisaglia L, Guégan D (1998) A comparison of techniques of estimation in long-memory processes. *Comput Stat Data Anal* 27:61–81
- Contreras-Reyes JE, Goerg G, Palma W (2011) Estimation, diagnostic and forecasting Functions for ARFIMA models. R package version 0.1.8. <http://cran.r-project.org/web/packages/afmtools/index.html>
- Contreras-Reyes J, Idrovo B (2011) En busca de un modelo Benchmark Univariado para predecir la tasa de desempleo de Chile. *Cuad Econ* 30(55):105–125
- Dahllhaus R (1989) Efficient parameter estimation for self-similar processes. *Ann Stat* 17(4):1749–1766
- Diebold F, Mariano R (1995) Comparing predictive accuracy. *J Bus Econ Stat* 13(3):253–263
- Giacomini R, White H (2006) Tests of conditional predictive ability. *Econometrica* 74(6):1545–1578
- Gradshteyn IS, Ryzhik IM (2007) *Table of Integrals, Series, and Products*, 7th edn. Elsevier, Amsterdam
- Granger CW, Joyeux R (1980) An introduction to long-memory time series models and fractional differencing. *J Time Ser Anal* 1:15–29
- Haslett J, Raftery AE (1989) Space-time modelling with long-memory dependence: assessing Ireland's wind power resource (with Discussion). *Appl Stat* 38(1):1–50
- Hassler U, Kokoszka PS (2010) Impulse responses of fractionally integrated processes with long memory. *Economet Theor* 26(6):1861–1885
- Hipel KW, McLeod AI (1994) *Time series modelling of water resources and environmental systems*. Elsevier, Amsterdam
- Hosking JR (1981) Fractional differencing. *Biometrika* 68(1):165–176

- Hyndman RJ, Khandakar Y (2008) Automatic time series forecasting: the forecast Package for R. *J Stat Softw* 27(3):1–22
- Kokoszka PS, Taqqu MS (1995) Fractional ARIMA with stable innovations. *Stoch Proc Appl* 60(1):19–47
- Lieberman O, Phillips PCB (2008) A complete asymptotic series for the autocovariance function of a long memory process. *J Econometrics* 147(1):99–103
- Lumley A, Heagerty P (1999) Weighted Empirical adaptive variance estimators for correlated data regression. *J Roy Stat Soc B* 61(2):459–477
- Newey WK, West KD (1987) A simple, positive semidefinite, Heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica* 55(3):703–708
- Palma W (2007) Long-memory time series, theory and methods. Wiley, Hoboken
- Palma W, Olea R (2010) An efficient estimator for locally stationary Gaussian long-memory processes. *Ann Stat* 38(5):2958–2997
- Palma W, Olea R, Ferreira G (2013) Estimation and forecasting of locally stationary processes. *J Forecast* 32(1):86–96
- Peiris M, Perera B (1988) On prediction with fractionally differenced ARIMA models. *J Time Ser Anal* 9(3):215–220
- Piessens R, de Doncker-Kapenga E, Uberhuber C, Kahaner D (1983) Quadpack: a subroutine package for automatic integration. Springer, New York
- R Development Core Team (2012) R: A language and environment for statistical computing. Foundation for Statistical Computing, Vienna. ISBN 3-900051-07-0. <http://www.R-project.org>
- Singleton RC (1979) Mixed radix fast fourier transforms. In: IEEE Digital Signal Processing Committee (eds) *Programs for Digital Signal Processing*. IEEE Press, New York
- Sowell F (1992) Maximum likelihood estimation of stationary univariate fractionally integrated time series models. *J Econom* 53(1–3):165–188
- Whittle P (1953) Estimation and information in stationary time series. *Ark Mat* 2:423–434
- Wold H (1938) A study in the analysis of stationary time series, vol 53. Almqvist and Wiksell, Uppsala, pp 165–188
- Zeileis A (2004) Econometric Computing with HC and HAC Covariance Matrix Estimators. *J Stat Softw* 11(10):1–17
- Zeileis A (2006) Object-oriented Computation of Sandwich Estimators. *J Stat Softw* 16(9):1–16