Universidade Federal do Rio Grande do Sul
Instituto de Matemática e Estatística
Programa de Pós-Graduação em Estatística

Estimação de Processos com Longa Dependência na Presença de Muitos Dados Faltantes

Gladys Choque Ulloa

Porto Alegre, Março de 2023.

CIP - Catalogação na Publicação

Ulloa, Gladys Choque
Estimação de Processos com Longa Dependência na Presença de Muitos Dados Faltantes / Gladys Choque Ulloa. -- 2023.
73 f.
Orientador: Guilherme Pumi.

Dissertação (Mestrado) -- Universidade Federal do Rio Grande do Sul, Instituto de Matemática e Estatística, Programa de Pós-Graduação em Estatística, Porto Alegre, BR-RS, 2023.

1. Processos com Longa Dependência. 2. Dados Faltantes. 3. Análise de Séries Temporais. 4. Simulações de Monte Carlo. I. Pumi, Guilherme, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

Dissertação submetida por Gladys Choque Ulloa como requisito parcial para a obtenção do título de Mestre em Estatística pelo Programa de Pós-Graduação em Estatística da Universidade Federal do Rio Grande do Sul.

Orientador(a):

Prof. Dr. Guilherme Pumi

Comissão Examinadora:

Profa. Dra. Simone Maffini Cerezer (IFRS)

Profa. Dra. Taiane Schaedler Prass (PPGEst - UFRGS)

Prof. Dr. Marcio Valk (PPGEst - UFRGS)

Data de Apresentação: 31 de Março de 2023

Agradecimentos

Agradeço primeiramente a Deus pela vida, pelas oportunidades que ele coloca na minha vida e por me permitir concluir meu mestrado nesta prestigiada universidade e programa de pós-graduação em estatística, ao meu tio Wilbert que está no céu e sei que de lá me apoio para terminar esta etapa profissional na minha vida, aos meus pais Maria Flor e Amilcar, que sempre me deram o seu apoio incondicional para conseguir alcançar todos os meus objetivos pessoais e acadêmicos. São eles que, com seu amor, sempre me incentivaram a perseguir meus objetivos e nunca abandoná-los diante das adversidades. São também eles que me deram o apoio material e financeiro para poder me concentrar nos meus estudos e nunca abandoná-los. À minha irmã Yuliana, que sempre me deu seus conselhos, palavras motivacionais e amizade sincera.

Ao meu orientador, prof. Dr. Guilherme Pumi, exemplo de pessoa e profissional, que sempre me ajudou desde o primeiro momento que entrei no mestrado, pela sua dedicação, paciência e ensinamentos, sem as suas palavras e correções precisas, não teria conseguido chegar a esta tão esperada instância. Obrigada por sua orientação e todos os seus conselhos, vou carregá-los para sempre em minha memória na minha vida profissional. Tenho grande admiração por ele. Muito obrigada por tudo!. A prof. Dra. Taiane Prass, exemplo de pessoa e profissional, obrigada pelas importantes e valiosas sugestões, conselhos e dicas que contribuíram para fazer esta dissertação, é inevitável não sentir admiração por ela.

Aos professores do programa PPGEST que fizeram parte da minha formação acadêmica, obrigada por seus ensinamentos e conhecimento compartilhado. Aos meus colegas, muitos dos quais se tornaram meus amigos. Obrigada pelas horas compartilhadas, pelos trabalhos realizados em conjunto e pelas experiências vividas. Finalmente, gostaria de agradecer à universidade UFRGS e ao programa de Pós-graduação em Estatística (PPGEST) da UFRGS que tanto exigiu de mim, mas ao mesmo tempo me permitiu concluir o mestrado.

Resumo

Entre os modelos mais importantes para séries temporais com longa dependência está a classe de modelos ARFIMA(p, d, q) (processo autoregressivo fracionariamente integrado de média móvel). Embora a estimação do parâmetro de longa dependência d em modelos ARFIMA é um problema bastante estudado, o mesmo não se pode dizer sobre a estimação de d na presença de dados faltantes. Podemos abordar este problema de duas maneiras: os dados faltantes podem ser imputados usando algum método plausível e então procedemos com a estimativa como se nenhum dado estivesse faltando; ou podemos aplicar uma metodologia especialmente adaptada para estimar d na presença de dados faltantes. Neste trabalho, revisamos alguns dos métodos disponíveis para ambas as abordagens e os comparamos por meio de um estudo de simulação de Monte Carlo. Apresentamos uma comparação entre 35 configurações diferentes para estimar d, em dezenas de diferentes cenários, considerando porcentagens de dados faltantes variando entre 10% até 70% e vários níveis de dependência. A velocidade computacional de cada método de estimação foi avaliada medindo-se o tempo necessário para executar várias tarefas diferentes.

Palavras-chave: Longa Dependência, Análise de Séries Temporais, Dados Faltantes, Estimação Semi-paramétrica, Cópulas.

Abstract

Among the most important models for long-range dependent time series is the class of ARFIMA(p, d, q) models (fractionally integrated moving average autoregressive process). Estimating the long-dependency parameter d in ARFIMA models is a well-studied problem, but the literature on estimating d in the presence of missing data is very sparse. To solve this problem there are two basic ways to deal with the problem: the missing data can be imputed using some plausible method and then we proceed with the estimation as if no data were missing or we can apply a specially adapted methodology to estimate d in the presence of missing data. In this work, we review some of the available methods for both approaches and compare them through a Monte Carlo simulation study. We present a comparison between 35 different configurations for estimating d, in tenths of different scenarios, considering percentages of missing data ranging from 10% to 70% and various dependence levels. The computational speed of each estimation method used in the simulations was evaluated by measuring the time required to perform several different tasks. The computational speed of each estimation method was evaluated by measuring the time required to perform several different tasks.

Keywords: Long Dependency, Time Series Analysis, Missing Data, Semi-parametric Estimation, Copulas.

Índice

Capítulo 1

Introdução

Processos com longa dependência possuem uma longa história e atualmente são partes fundamentais da teoria de séries temporais. Ao leitor interessado na teoria de processos com longa dependência, sugerimos o livro de Palma (2007), as compilações de Doukhan et al. (2003) e Robinson (2003) e o livro de Beran (1994), que contém um relato da história e dos primeiros desenvolvimentos na área. Neste trabalho estudaremos a classe de processos autoregressivos fracionariamente integrados de médias móveis, abreviado ARFIMA, introduzidos por Honsking (1981) e Granger and Joyeux (1980). Os modelos ARFIMA são a classe de modelos de longa dependência mais aplicados e estudados na literatura.

Dada sua longa história, existem muitos procedimentos para estimar o parâmetro de dependência de longa dependência d, incluindo métodos baseados em representação de estado-espaço, densidade espectral, aproximações de verossimilhança, wavelets, análise de flutuação destendenciada e cópulas (Hurst, 1951; Geweke and Porter-Hudak, 1983; Peng et al., 1994; Robinson, 1995a,b; Abry and Veitch, 1998; Chan and Palma, 1998; Palma, 2007; Faÿ et al., 2009; Pumi et al., 2023, e referências ali contidas.). Estudos de simulação de Monte Carlo comparando diferentes estimadores podem ser encontrados em Taqqu et al. (1995); Kokoszka and Bhansali (2001); Reisen et al. (2001); Rea et al. (2009) e Faÿ et al. (2009).

Um problema particular que é relativamente pouco estudado na literatura é a presença de dados faltantes em séries temporais. Dados faltantes são definidos como valores que por um motivo ou outro não estão a disposição para consulta ou uso. Existem três mecanismos geradores de dados faltantes: faltantes completamente ao acaso (do inglês, missing completely at random, MCAR), faltantes ao acaso (missing at random, MAR) e faltantes não ao acaso (missing not at random, MNAR), que podem ocorrer por vários motivos. O mecanismo gerador dos dados faltantes é decisivo para a análise dos dados, e tem por objetivo determinar se a perda é aleatória, ou seja, afeta todos os indivíduos igualmente, ou pode ser por um motivo ou razões específicas que podem introduzir vieses que invalidam os resultados.

Na área de séries temporais a presença de dados faltantes requer um tratamento diferenciado devido que a ausência de uma observação afeta a estrutura de dependência da série, que deve ser levada em conta na hora do tratamento dos dados. No contexto de séries temporais com longa dependência, existem poucos estudos relacionados a estimação desses modelos na presença de dados faltantes. No contexto de séries temporais, basicamente existem duas maneiras de lidar com dados

faltantes. A primeira é a imputação, que é a mais simples e utilizada. A ideia básica é substituir os dados faltantes por valores razoáveis e então prosseguir com a análise como se os dados faltantes nunca tivessem existido. Existem muitas maneiras de fazer a imputação. Um dos métodos mais simples, especialmente quando se trata de séries temporais estacionárias, é substituir os valores faltantes pela média ou mediana da amostra calculada sobre os casos não faltantes. Para séries não-estacionárias, os valores faltantes podem ser substituídos pela média calculada nas vicinitudes do ponto faltante.

Quando apenas uma pequena porcentagem do tamanho total da amostra estiver faltando, tipicamente qualquer método de imputação aplicado a uma série temporal estacionária vai gerar boas estimativas pontuais sempre que o estimador utilizado for razoável. No entanto, à medida que o número de valores faltantes aumenta, a qualidade da estimativa baseada em imputação vai se degradando ao ponto de ser inútil. Um outro ponto atraente da imputação, é que na maioria dos casos ela é rápida de calcular e fácil de implementar.

Apesar de seus atrativos, um ponto importante a mencionar é que a grande maioria dos métodos tradicionais de imputação apresentam três problemas quando aplicados no contexto de séries temporais. Primeiro, a variância tende a ser subestimada, influenciando outros parâmetros que dependem da variância (correlações). Por exemplo, a substituição pela média/mediana reduz a variância, visto que valores faltantes são substituídos por um valor único. O segundo problema é que ao imputar um valor exógeno à série temporal, alteramos a estrutura de dependência da mesma de maneiras difíceis de entender ou quantificar. Isso é problemático, pois a estimação do modelo de interesse prático leva em consideração a estrutura de dependência da série temporal. O terceiro problema é igualmente sério. Os cálculos de erro padrão assumem que todos os dados estão corretos. As incertezas intrínsecas nos valores calculados e a variabilidade da amostra não são levadas em consideração. Como resultado, erros padrão podem ser distorcidos, levando a intervalos de confiança e testes de hipóteses incorretos e assim ter um modelo mal especificado.

O segundo método para estimação em modelos de séries temporais na presença de dados faltantes é usar um estimador que foi modificado ou projetado especificamente para lidar com dados faltantes. Obviamente, um estimador desse grupo não apresenta nenhum dos três problemas que os métodos de imputação apresentam, visto que um estimador razoável desse grupo leva em conta a ausência de algumas observações e é capaz de ignorar suas implicações. O problema é que existem poucas opções na literatura para estimadores nessa categoria, e as opções disponíveis costumam ser mais complexas de implementar e mais lentas de executar em comparação com uma alternativa de imputação. No contexto de processos ARFIMA(p, d, q), entre os métodos nativos para estimar o parâmetro d na presença de dados faltantes temos o método Wavelet de Craigmile and Mondal (2020), o método baseado em cópula de Pumi et al. (2023), o método baseado em Wavelet Lifting de Knight et al. (2017), entre outros.

Neste trabalho, estudamos o problema de estimação do parâmetro de longa dependência d no contexto de modelos ARFIMA(p, d, q) na presença de uma quantidade grande de dados faltantes, através de um extenso estudo de simulação de Monte Carlo. Consideramos três estimadores diferentes projetados especificamente para dados faltantes em diferentes configurações, cinco estimadores semi-paramétricos usados com mais frequência na literatura combinados com três métodos de imputação diferentes, 28 cenários com diferentes intensidades de dependência e porcentagem de valores faltantes foram utilizados. Também propomos um método de imputação aleatória calibrado para manter a variância da série temporal observada sem introduzir valores atípicos e levando em conta a estrutura

de dependência local da série temporal.

Objetivos

Este trabalho tem por principal objetivo fazer o levantamento sistemático da literatura cobrindo dados faltantes no contexto de longa dependência para os modelos ARFIMA (p, d, q), fazer uma comparação sistemática das técnicas existentes através de simulações de Monte Carlo em contextos tradicionais, estudar os limites de quebra de cada técnica via simulações de Monte Carlo e propor uma nova metodologia para lidar com dados faltantes no contexto de longa dependência.

Novidades do trabalho

Neste trabalho apresentamos um extensivo estudo de simulação no contexto de séries temporais com longa dependência na presença de muitos dados faltantes. Estudamos tanto o comportamento de estimadores especialmente desenvolvidos para estimar o parâmetro de longa dependência em séries temporais com dados faltantes (chamados de nativos), quanto o comportamento de estimadores tradicionais aplicados às séries após terem os valores faltantes imputados usando três métodos distintos, um deles inédito. No total consideramos 35 configurações diferente de métodos de estimação, diversos níveis de dependência, diferentes tamanhos amostrais e proporção de dados faltantes entre 10% e 70%, totalizando dezenas de cenários. Até onde sabemos, este é o primeiro estudo sistemático tratando da estimação do parâmetro d utilizando estimadores clássicos em séries temporais com dados faltantes imputados. Além disso, este é o primeiro estudo relacionado ao comportamento dos estimadores nativos no contexto de muitos dados faltantes, uma vez que os artigos que propõem estes estimadores apresentam simulações com no máximo 20% de dados faltantes. Apresentamos também um estudo detalhado do tempo computacional utilizado por cada estimador em dezenas de cenários.

Suporte computacional

Na parte computacional, apresentamos a simulação de diferentes cenários de ajuste do modelo para estimar o parâmetro (d) de estudo. Todas as validações numéricas apresentadas no artigo foram implementadas em R (R Core Team, 2020), versão 4.0.3. Neste trabalho, diferentes pacotes estatísticos em R também são usados para implementar os métodos nativos, métodos tradicionais e métodos de imputação. Estes são detalhados na Seção 3.2 do paper em anexo.

Organização do trabalho

No Capítulo 2, apresentamos uma rápida revisão dos métodos mais comuns para tratamento de dados faltantes, incluindo classificação de dados faltantes. O desenvolvimento de métodos convencionais para preencher dados faltantes (em dados que não são séries temporais) também é apresentado neste capítulo. No Capítulo 3, apresentamos uma revisão simples da teoria de longa dependência no contexto de modelos ARFIMA(p, d, q). No Capítulo 4, apresentamos uma rápida revisão dos

métodos mais comuns para tratamento de dados faltantes no contexto específico de séries temporais. Apresentamos 3 métodos nativos: o método baseado em Wavelet de Craigmile and Mondal (2020), o método baseado em cópulas de Pumi et al. (2023) e um método baseado em wavelets lifting (LoMPE) de Knight et al. (2017). Também são apresentados alguns dos métods mais tradicionais para estimar o parâmetro de longa dependência d, a saber, o método R/S, o estimador de Geweke e Porter-Hudak (GPH), o estimador Local de Whittle (LW) e Local Exato de Whittle (ELW) e o estimador baseado em DFA. No Capítulo 5, apresentamos a proposta de trabalho e o que foi realizado. No Capítulo 6, apresentamos as conclusões e possíveis trabalhos futuros. No apêndice A é apresentado o artigo, "Estimation of Long-Range Dependent Models with Missing Data: to Input or not to Input?", sendo este o principal produto desta dissertação.

Capítulo 2

Longa Dependência

Nesta seção, apresentamos conceitos básicos sobre séries temporais com longa dependência assim como os modelos ARFIMA(p, d, q). A presença de longa dependência em séries temporais foi observada primeira vez em Hurst (1951). Em processos com longa dependência, a função de autocovariância geralmente exibe decaimento hiperbólico não sendo absolutamente somável, ao contrário de processos com curta dependência, como os tradicionais modelos ARMA.

Existem alguns modelos de séries temporais capazes de modelar longa dependência, entre eles, os modelos ARFIMA(p, d, q), que são uma generalização dos modelos ARIMA(p, d, q), permitindo que o parâmetro de integração d assuma valores fracionários. Diversos estimadores existem para o parâmetro de longa dependência e diversas aplicações para seus modelos, de modo que hoje em dia o assunto é bastante importante em determinadas áreas. No que segue apresentamos a definição de longa dependência, modelos ARFIMA, estacionaridade, causalidade e invertibilidade.

2.1 Longa Dependência e Processos ARFIMA

Neste trabalho adotamos a seguinte definição de processo com longa dependência.

Definição 2.1. Um processo fracamente estacionário $\{X_n\}_{n\in\mathbb{N}}$ é dito possuir *longa dependência* se

$$\gamma_X(h) \sim n^{-\beta} L(n), \quad \text{quando} \quad n \to \infty \tag{2.1}$$

para algum $\beta \in (0, 1)$ e alguma função L de variação lenta no infinito.

Observamos também que processos com longa dependência são comumente associados à presença de uma singularidade na função densidade espectral na origem e também à condição $\sum_{h\in\mathbb{Z}} |\gamma_X(h)| = \infty$. Uma das classes mais difundidas de modelos com longa dependência é a dos modelos ARFIMA, que revisamos a seguir.

Definição 2.2. Um processo estocástico $\{X_t\}_{t\in\mathbb{N}}$ é dito ser um processo ARFIMA(p, d, q) se $\{X_t\}_{t\in\mathbb{N}}$ for uma solução fracamente estacionária de

$$\phi(L)X_t = \theta(L)(1 - L)^{-d} Z_t, \tag{2.2}$$

onde $\phi(z) := 1 - \phi_1 z - \cdots - \phi_p z^p$ e $\theta(z) := 1 + \theta_1 z + \cdots + \theta_q z^q$ são polinômios que assumimos não

ter raízes em comum, $\{Z_t\}_{t\in\mathbb{Z}}$ é um ruído branco e $(1-L)^d$ é definido pela sua expansão binomial $(1-L)^d := \sum_{k\in\mathbb{N}} \pi_k L^k$, onde $\pi_0 := 1$ e $\pi_k := \prod_{j=1}^{k} \frac{j-1-d}{j}$, para todo $k \in \mathbb{N}$.

Pode ser mostrado que, para $d \in (-1, 0.5)$, se os polinômios ϕ e θ não têm raízes no disco unitário $\{z : |z| \leq 1\}$, então existe uma única solução fracamente estacionária, causal e invertível de (2.2). Neste caso, $\{X_t\}_{t\in\mathbb{N}}$ terá uma representação MA(∞) dada por $X_t = \sum_{k\in\mathbb{N}} c_k Z_{t-k}$, para todo $t \in \mathbb{N}$, onde a sequência $\{c_k\}_{k\in\mathbb{N}}$ é determinada através da expansão $(1-z)^{-d}\theta(z)/\phi(z) = \sum_{k\in\mathbb{N}} c_k z^k$ e satisfaz $\sum_{k\in\mathbb{N}} c_k^2 < \infty$. Além disso, $\gamma_X(h) \sim Kh^{2d-1}$, para h grande. Desta forma, se $d \in (0, 0.5)$, $\sum_{h\in\mathbb{Z}} |\gamma_X(h)| = \infty$ e o processo $\{X_t\}_{t\in\mathbb{N}}$ apresenta longa dependência no sentido da equação (2.1) com $\beta = 1 - 2d$. Se $d \in (-1, 0)$ então a função de autocovariância é absolutamente somável, $\sum_{k\in\mathbb{N}} |c_k| < \infty$ e dizemos que $\{X_t\}_{t\in\mathbb{N}}$ possui *dependência intermediária*. Se $d = 0$, então (2.2) se reduz a um processo ARMA(p, q), que possui *curta dependência*. Também pode-se mostrar que a densidade espectral de um processo ARFIMA(p, d, q) satisfaz $f_X(\lambda) \sim K\lambda^{-2d}$, quando λ tende a 0, e, desta forma, na presença de longa dependência a densidade espectral é ilimitada na origem.

Um estudo mais aprofundado desses processos pode ser encontrado em Beran (1994), Palma (2007) e Brockwell and Davis (1991). Há evidências de que processos de longa dependência ocorrem com bastante frequência em campos tão diversos como hidrologia e economia. Veja Hurst (1951), Lawrance and Kottegoda (1977), Hipel and McLeod (1978) e Granger and Joyeux (1980).

Capítulo 3

Dados Faltantes

Os dados faltantes são aqueles valores que não estão disponíveis em uma amostra. Eles podem acontecer por diversos motivos: como perda de informações, ausência de resposta a uma questão, falha de mensuração, etc. Durante a análise de dados, os dados faltantes podem ser um problema sério, podendo afetar estimativas, intervalos de confiança, testes de hipóteses, etc., sendo necessário o tratamento adequado dos mesmos para que a análise permita conclusões e resultados confiáveis. Portanto, é importante lidar adequadamente com os dados faltantes em qualquer estudo estatístico. Existem várias abordagens para lidar com dados faltantes no contexto de observações independentes. Neste capítulo, descrevemos alguns métodos para lidar com essa situação que também são aplicáveis no contexto de séries temporais. Iniciamos tratando dos mecanismos geradores de dados faltantes.

3.1 Mecanismos Geradores de Dados Faltantes

O mecanismo gerador dos dados faltantes é fundamental para a escolha correta da metodologia a ser usada na análise de dados que apresentam dados faltantes. De acordo com Little and Rubin (1987), são três os principais mecanismos geradores de dados faltantes, que descrevemos brevemente abaixo.

3.1.1 Dados faltantes de forma completamente aleatória (MCAR)

Os dados faltantes são ditos estar faltando de forma completamente aleatória, ou completamente ao acaso, se a probabilidade de que a variável $\boldsymbol{Y}$ faltante não depende da variável $\boldsymbol{X}$ ou da própria variável $\boldsymbol{Y}$. Considere um cenário, onde uma única variável, denominada $\boldsymbol{Y}$, contém dados faltantes, enquanto um conjunto diferente de variáveis, representado pela variável $\boldsymbol{X}$, possui dados observados em todos os momentos. Se a probabilidade de os dados faltantes em $\boldsymbol{Y}$ for independente das medições de $\boldsymbol{X}$, então dizemos os dados faltantes estão faltando completamente ao acaso. Podemos expressar isso matematicamente da seguinte forma. Se R representa a indicadora de um dado estar faltando, então

$$P(R = 1 \mid \boldsymbol{X}, \boldsymbol{Y}) = P(R = 1). \tag{3.1}$$

3.1.2 Dados faltantes de forma aleatória (MAR)

Os dados faltantes são ditos estar faltando de forma aleatória (MAR) quando a probabilidade de resposta depende apenas de $\boldsymbol{X}$ e não de $\boldsymbol{Y}$. A condição MAR é uma suposição muito mais fraca que MCAR, mas ainda forte. Suponha que uma variável, $\boldsymbol{Y}$, possui dados faltantes enquanto outro conjunto de variáveis, $\boldsymbol{X}$, está completo. Ao considerarmos $\boldsymbol{X}$, dizemos que os dados em $\boldsymbol{Y}$ estão faltando aleatoriamente se a probabilidade de que $\boldsymbol{Y}$ esteja faltante não depende de $\boldsymbol{Y}$, mas potencialmente depende de $\boldsymbol{X}$. O MAR permite que dados faltantes em $\boldsymbol{Y}$ dependa de outras variáveis que são observadas. Ele simplesmente não pode depender das suas próprias observações. Matematicamente se pode apresentar da seguinte forma,

$$P(R = 1 \mid \boldsymbol{X}, \boldsymbol{Y}) = P(R = 1 \mid \boldsymbol{Y}) \tag{3.2}$$

3.1.3 Dados faltantes de forma não aleatória (MNAR)

O mecanismo mais complexo de dados faltantes, conhecido como dados faltantes de forma não aleatória (MNAR), ocorre quando os valores faltantes são influenciados tanto por dados observados quanto por dados não observados, dificultando a determinação do mecanismo que causou os dados faltantes. Quando a probabilidade de $\boldsymbol{Y}$ esteja faltando depende do valor do próprio $\boldsymbol{Y}$ após o ajuste para $\boldsymbol{X}$, dizemos que os dados estão faltando de forma não aleatória, ou não aleatoriamente. O problema com o mecanismo MNAR é que ele não pode ser identificado sem conhecimento prévio do valor que está faltando. Quando os dados faltante são MNAR, é necessário incluir o mecanismo de perda de dados no processo de estimação para obter estimativas de parâmetros não-viesadas.

Neste trabalho consideraremos apenas o mecanismo MCAR, sendo este o mais utilizado em séries temporais.

3.2 Imputação

Existem vários métodos que são usados para lidar com valores faltantes em dados cujas observações são independentes. Muitos porém são utilizados especificamente no contexto de modelagem. O método geral mais comumente utilizado para lidar com dados faltantes é a imputação. A ideia básica é substituir os valores faltantes por valores plausíveis e então continuar com a análise como se nenhum dado estivesse faltando. Claro, existem muitas abordagens para se fazer isso. Apresentamos a seguir alguns métodos de imputação mais utilizados na literatura.

3.2.1 Imputação pela Média, mediana e moda

Um dos métodos mais simples de imputar para preencher valores faltantes e produzir uma amostra completa é a abordagem de imputação pela média, mediana ou moda incondicional. Este método substitui os valores faltantes pela média, mediana ou moda calculada a partir dos valores observados. O benefício de usar esse método é que ele é simples de implementar e não exclui nenhuma observação,

como faria uma exclusão em lista. No geral, esses métodos devem ser usados com cuidado em amostras provindas de populações com distribuição contínua, pois a imputação de valores faltantes utilizando-se apenas um único valor pode induzir um ponto de massa numa distribuição que do contrário não possuiria valores repetidos. A imputação pela média, para o tipo MCAR, por exemplo, é conhecida por fornecer estimativas distorcidas para a maioria dos parâmetros (Haitovsky, 1968).

3.2.2 Imputação por Regressão

Neste método, os valores faltantes são substituídos por respostas previstas por um modelo de regressão ajustado utilizando-se as observações completas presentes no banco. Casos completos são usados em uma análise multivariada para estimar um modelo de regressão no qual a variável incompleta é a resposta e as variáveis explicativas são algumas das variáveis completas. As respostas previstas para casos com dados faltantes podem ser estimadas usando o modelo de regressão estimado. Embora a ideia de obter informações das variáveis completas seja boa, a imputação por regressão também produz estimativas de parâmetros viesadas e é de utilidade limitada.

3.2.3 Imputação por Regressão Estocástica

A imputação por regressão estocástica substitui os dados faltantes por um valor previsto pela imputação de regressão, juntamente com um resíduo que mostra o grau de incerteza no valor previsto. Em modelos de regressão linear padrão, o resíduo será normal, com média zero e variância igual à variância residual da regressão. Em um resultado binário, como a regressão logística, os valores imputados têm uma probabilidade de 1 versus 0, e os valores projetados são 1 ou 0. A regressão estocástica é usada no método de dois estágios de Rubin (1988).

3.2.4 Imputação Múltipla

A imputação múltipla (IM), proposta em Rubin (1987) substitui os valores faltantes por um conjunto de m valores plausíveis a serem imputados, refletindo a incerteza causada pelos dados faltantes. Estes m valores são utilizados para gerar m estimativas dos parâmetros do modelo interesse a ser ajustado nos dados. Com essas m estimativas, pode-se calcular os erros padrão devido às imputações feitas. Usando regras simples estes valores são apropriadamente combinados de forma a gerar uma única estimativa dos parâmetros de interesse e de seu erro padrão, agregando a incerteza nas estimativas induzida pelos dados faltantes.

A ideia por trás da imputação múltipla é bem simples. O que é mais complexo é decidir quais serão os m valores a serem imputados para realizar o procedimento. Dois métodos comumente utilizados são baseados em $propensity\ score$ e $predictive\ mean\ matching$ (Molenberghs et al., 2014, capítulos 11 e 12). Uma das vantagens da imputação múltipla é, quando as imputações são sorteadas aleatoriamente tentando representar a distribuição dos dados, a imputação múltipla aumenta a eficiência da estimação, refletindo uma variabilidade adicional, simplesmente obtida pela combinação de inferências de dados completos de uma maneira direta. Outra vantagem é que facilita o estudo direto da sensibilidade de inferências de vários modelos usando métodos de dados completos. Além das vantagens apresentadas

acima, tem-se também que as inferências de erro padrão, p-valores, etc., obtidas a partir de IM são geralmente válidas porque incorporam incerteza devido a falta de dados, tornando IM atraente porque pode ser altamente eficientes mesmo para pequenos valores de m (Schafer and Olsen, 1998). Em Meng (1994), os autores mostram que os estimadores baseado em imputação múltipla são mais eficientes que a imputação simples, além de que conduzir inferências requer apenas repetir o mesmo padrão de análise de dados completos várias vezes. Outra vantagem deste método é evitar subestimação da verdadeira variância.

3.2.5 Interpolação Linear

A interpolação usando algum modelo simples é uma prática bastante comum na literatura. Isso pode ser obtido por uma simples interpolação linear nas proximidades dos dados faltantes. Se y_t é uma observação ausente, aplicamos uma interpolação linear simples entre os dois pontos observados mais próximos de y_t. Sejam y_{t_1} e y_{t_2} os dois pontos observados mais próximos no tempo satisfazendo $t_1 < t < t_2$. Imputamos y_t como

$$y_t = y_{t_1} + \left(\frac{y_{t_2} - y_{t_1}}{t_2 - t_1}\right)(t - t_1).$$

A interpolação linear também é um método muito simples que, ao contrário da substituição média, atribui valores diferentes para cada dado faltante, quando a distribuição subjacente é absolutamente contínua. No entanto, ainda subestima a variância e afeta a estrutura de dependência da série temporal. Ele também imputa lacunas amplas como uma linha reta, afetando potencialmente o cálculo do erro padrão (para estimadores).

3.2.6 Substituição Aleatória

A substituição aleatória é um método de imputação baseado na substituição de um determinado valor faltante por meio de uma distribuição predeterminada. O mais comumente aplicado é substituir um valor faltante por um valor aleatório obtido de uma distribuição uniforme, normalmente ao longo dos valores mínimos e máximos observados. Este método simples tem a vantagem de que não importa o tamanho lacuna a ser preenchida, os valores imputados nunca serão iguais. No entanto, como valores próximos aos mínimos e máximos observados ocorrem com a mesma probabilidade de qualquer outro intervalo de mesmo comprimento, esse método de imputação tende a inflar a variância da série temporal e também alterar sua distribuição subjacente, afetando ainda a estrutura de dependência da série temporal.

Neste trabalho, propomos um método de substituição aleatória que herda informações sobre a estrutura de dependência na vizinhança imediata do valor ausente sendo imputado. A ideia é um híbrido do método da última observação realizada, que consiste em substituir cada valor faltante pelo valor observado mais recente, e a substituição aleatória. Denotemos por $tN(\mu, \sigma^2, a, b)$ a distribuição normal truncada, truncada no intervalo (a, b), com média $\mu \in \mathbb{R}$ e variância $\sigma^2 > 0$. Se $Z \sim tN(\mu, \sigma^2, a, b)$, Z tem densidade

$$f(x; \mu, \sigma^2, a, b) = \frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\sigma\left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right]} I(a < x < b),$$

onde, ϕ e Φ denotam a densidade e a distribuição de uma distribuição normal padrão, respectivamente. Seja y_t, com $t > 1$ um valor faltante a ser imputado. Propomos imputar y_t por $y_t \sim tN(y_{t-1}, \sigma^2, y_{(1)}, y_{(n)})$, onde $y_{(1)} = \min\{y_i : i \in M^{\complement}\}$ e $y_{(n)} = \max\{y_i : i \in M^{\complement}\}$, $M := \{i : y_i \text{ is missing}\}$. O parâmetro de variação σ^2 pode ser ajustado para corresponder à variância na série temporal observada.

Capítulo 4

Métodos de Estimação

Neste capítulo, apresentamos os métodos nativos e tradicionais em séries temporais para estimar o parâmetro d. O problema é que existem poucas opções na literatura para estimadores nessa categoria, e as opções disponíveis costumam ser mais complexas de implementar e mais lentas de executar em comparação com uma alternativa de imputação. A seguir descrevemos alguns desses métodos.

4.1 Métodos tradicionais

Conforme mencionado na introdução, existem vários métodos para estimar d quando a série temporal não apresenta dados faltantes. Esses estimadores normalmente não conseguem lidar com dados faltantes naturalmente. No entanto, eles ainda podem ser usados após a adequada imputação da série. Nesta seção, apresentamos alguns estimadores tradicionais para a estimação do parâmetro d.

4.1.1 Método R/S

O método R/S foi introduzido por Hurst (1951). Seja, uma amostra $\{y_1, \cdots, y_n\}$ de um processo estacionário com longa dependência e sejam, $x_t := \sum_{j=1}^{t} y_j$ para $t \in \{1, \cdots, n\}$ as somas parciais dos y_j's e seja $s_n^2 := \frac{1}{n-1}\sum_{t=1}^{n}(y_t - \overline{y})^2$ a variância da amostra, onde $\overline{y} = x_n/n$. A *estatística R/S*, é definida por

$$R_n := \frac{1}{s_n}\left[\max_{1\leq t\leq n}\left\{x_t - \frac{t}{n}x_n\right\} - \min_{1\leq t\leq n}\left\{x_t - \frac{t}{n}x_n\right\}\right].$$

Denotando $Q_n := n^{-\frac{1}{2}-d}R_n$, pode ser mostrado que $\log(R_n) = \mathbb{E}(Q_n) + (d+\frac{1}{2})\log(n) + \log(Q_n) - \mathbb{E}(Q_n)$. O estimador do parâmetro de longa dependência d é obtido através de mínimos quadráticos ordinários. Se $R_{t,k}$ é a estatística R/S baseada em uma amostra de tamanho $k, \{y_t, \cdots, y_{t+k-1}\}$, para $1 \leq t \leq n-k+1$, então um estimador d pode ser obtido regredindo-se $\log(R_{t,k})$ em $\log(k)$ para $1 \leq t \leq n-k+1$, mais um intercepto. Alguns resultados assintóticos das estatísticas R/S são apresentados em Mandelbrot (1975).

4.1.2 Estimador de Geweke e Porter-Hudak (GPH)

O método GPH que foi introduzido por Geweke and Porter-Hudak (1983) tem como propósito estimar o parâmetro de longa dependência através do método de regressão linear baseado na função periodograma. Seja $\{Y_t\}_{t\in\mathbb{Z}}$ um processo estacionário de longa dependência com densidade espectral satisfazendo

$$f(\lambda) = f_0(\lambda)[2\sin(\lambda/2)]^{-2d}, \tag{4.1}$$

para alguma função contínua f_0. Aplicando o logaritmo a equação (4.1) avaliados nas sequências de Fourier $\lambda_j := 2\pi j/n$, obtemos

$$\log\big(f(\lambda_j)\big) = \log\big(f_0(0)\big) - 2d\log\big(2\sin(\lambda_j/2)\big) + \log\left(\frac{f_0(\lambda_j)}{f_0(0)}\right), \tag{4.2}$$

De (4.2), Geweke and Porter-Hudak (1983) sugeriu uma abordagem de regressão para estimar d. A ideia é estimar a densidade espectral usando o periodograma, definido por

$$I(\lambda) := \left|\frac{1}{\sqrt{2\pi n}}\sum_{t=1}^{n} e^{it\lambda}Y_t\right|^2, \tag{4.3}$$

e escrever

$$\log\big(I(\lambda_j)\big) = \log\left(\frac{I(\lambda_j)}{f(\lambda_j)}\right) + \log\big(f(\lambda_j)\big),$$

que combinado com (4.2), resulta

$$\log\big(I(\lambda_j)\big) = \log\big(f_0(0)\big) - 2d\log\big(2\sin(\lambda_j/2)\big) + \log\left(\frac{I(\lambda_j)[2\sin(\lambda/2)]^{2d}}{f_0(0)}\right). \tag{4.4}$$

A partir de (4.4), podemos estimar d considerando as primeiras m ordenadas do periodograma regredindo $\log\big(I(\lambda_1)\big), \cdots, \log\big(I(\lambda_m)\big)$ em $2\log\big(2\sin(\lambda_1/2)\big), \cdots, 2\log\big(2\sin(\lambda_m/2)\big)$ mais um intercepto. As propriedades assintóticas do GPH foram estudadas por Robinson (1995a) e Hurvich et al. (1998), entre outros. Sob certas condições de regularidade, o estimador do GPH é consistente e assintoticamente normal com uma taxa de convergência de $n^{4/5}$, independentemente de d.

4.1.3 Estimador Local de Whittle (LW)

O método LW tem o propósito de estimar o parâmetro de longa dependência. Seja $\{Y_t\}_{t\in\mathbb{Z}}$ um processo estacionário com longa dependência, com parâmetro d e densidade espectral f satisfazendo

$$f(\lambda) \sim G\lambda^{-2d},$$

quando $\lambda \to 0^+$ e $G > 0$. Para um processo ARFIMA(p,d,q), $G = \left[\frac{\sigma_\varepsilon\theta(1)}{2\pi\phi(1)}\right]^2$. O estimador semiparamétrico, conhecido como estimador local de Whittle, é definido por

$$\hat{d} = \underset{|d|<1/2}{\operatorname{argmin}}\{R(d)\}, \tag{4.5}$$

onde

$$R(d) := \log\big(\widehat{G}(d)\big) - 2d\frac{1}{m}\sum_{j=1}^{m}\log(\lambda_j), \quad \text{para} \quad \widehat{G}(d) := \frac{1}{m}\sum_{j=1}^{m}\lambda_j^{2d}I(\lambda_j), \tag{4.6}$$

I é o periodograma definido em (4.3) e $0 < m < n/2$ é um inteiro. O estimador (4.5) foi introduzido e estudado em Robinson (1995a), que provou sua consistência e normalidade assintótica sob suposições moderadas, com uma taxa de convergência $\sqrt{m}$.

4.1.4 *Estimador Local Exato de Whittle (ELW)*

O estimador semiparamétrico introduzido por Shimotsu and Phillips (2005), conhecido de estimador Local Exato de Whittle (ELW) é muito semelhante em natureza ao estimador Local de Whittle. A principal diferença é que o estimador é derivado da manipulação algébrica da verossimilhança de Whittle sem depender de aproximações do periodograma, sendo exato nesse sentido. O ELW também é adequado para ser aplicado à região não estacionária $d > 0,5$, mas vamos considerar o intervalo $|d| < 0,5$ para simplificar a exposição. Seja,

$$I_{\Delta^d}(\lambda) := \left| \frac{1}{\sqrt{2\pi n}} \sum_{t=1}^{n} e^{it\lambda}(1-L)^d(Y_t) \right|^2.$$

O estimador é definido da mesma forma que o estimador local de Whittle, porém utilizando I_{Δ^d} ao invés de I em (4.6), isto é,

$$\hat{d} = \underset{|d|<1/2}{\operatorname{argmin}}\{R(d)\},$$

onde,

$$R(d) := \log\left(\widehat{G}(d)\right) - 2d\frac{1}{m}\sum_{j=1}^{m}\log(\lambda_j), \quad \text{com} \quad \widehat{G}(d) := \frac{1}{m}\sum_{j=1}^{m}\lambda_j^{2d} I_{\Delta^d}(\lambda_j).$$

A teoria assintótica do ELW é muito semelhante ao local Whittle, mas as condições exigidas em m são ligeiramente mais fortes. Computacionalmente, segundo os autores, o ELW é cerca de 10 vezes mais lento que o Local de Whittle, mas ainda é um método muito rápido em comparação com outros. Uma discussão sobre as vantagens e desvantagens do ELW sobre o GPH e o estimador Local de Whittle, bem como mais detalhes, podem ser encontrados em Shimotsu and Phillips (2005).

4.1.5 *Estimador baseado em DFA*

O método conhecido como *detrended fluctuation analysis* (DFA), introduzido por Peng et al. (1994) e é baseado no comportamento da variância destendenciada em séries temporais com longa dependência. Antes de prosseguir com a definição do estimador, introduzimos algumas notações, seguindo o trabalho mais geral de Prass and Pumi (2021).

Seja, $\{Y_t\}_{t\in\mathbb{Z}}$ um processo estacionário com longa dependência e parâmetro d e seja $\{y_1, \cdots, y_n\}$ uma amostra dele. Seja $R_t := \sum_{j=1}^{t} Y_j$ para $t \in \{1, \cdots, n\}$, os sinais integrados. Divida os sinais integrados em $k = \lfloor n/(m+1) \rfloor$ caixas não sobrepostas, cada uma contendo $m+1$ valores, que denotamos por $\boldsymbol{R}_i := (R_{(m+1)(i-1)+1}, \cdots, R_{i(m+1)})'$, para $i \in \{1, \cdots, k\}$. A seguir, em cada caixa ajustamos um polinômio de grau $\nu+1$ via mínimos quadrados, considerando

$$D_{m+1}^{\top} := \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 2 & \dots & m+1 \\ \vdots & \vdots & \ddots & \vdots \\ 1^{\nu+1} & 2^{\nu+1} & \dots & (m+1)^{\nu+1} \end{pmatrix}, \qquad \begin{aligned} P_{m+1} &:= D_{m+1}(D_{m+1}^{\top}D_{m+1})^{-1}D_{m+1}^{\top}, \\ Q_{m+1} &:= I_{m+1} - P_{m+1}. \end{aligned}$$

Define-se

$$\boldsymbol{\mathcal{E}}_i := Q_{m+1}\boldsymbol{R}_i = (\mathcal{E}_i, \cdots, \mathcal{E}_{m+i})^{\top},$$

o vetor de resíduos na i-ésima caixa. Seja $f^2_{DFA}(m,i) := \frac{1}{m+1}\boldsymbol{\mathcal{E}}_i'\boldsymbol{\mathcal{E}}_i$, i.e. a variância amostral do resíduo no i-ésimo ajuste. A variância destendenciada F^2_{DFA} é definida por

$$F^2_{DFA}(m) := \frac{1}{k}\sum_{i=1}^{k} f^2_{DFA}(m,i).$$

Como descrito em Peng et al. (1994), para um processo com longa dependência,

$$\sqrt{F^2_{DFA}(m)} \sim cm^{d+1/2},$$

para m grande e alguma constante c independente de d. Denotando por $L_m := \log\left(\sqrt{F^2_{DFA}(m)}\right)$, e tomando o logaritmo, obtemos

$$L_m \sim \log(c) + \left(d + \frac{1}{2}\right)\log(m). \tag{4.7}$$

De (4.7), podemos estimar d regredindo $L_{s+1}, \cdots, L_{s+l}$ em $\log(s+1), \cdots, \log(s+l)$, para $s, l > 0$ mais um intercepto. A teoria assintótica do estimador baseado na DFA é apresentado em Bardet and Kammoun (2008).

4.2 Métodos Nativos

Nesta seção apresentaremos métodos nativos que foram criados por diferentes autores focados na estimação do parâmetro d de um processo de longa dependência na presença de dados faltantes.

4.2.1 Método Wavelet de Craigmile e Modal

Seja $\{X_t\}_{t\in\mathbb{Z}}$ uma série temporal gaussiana de interesse, estacionária, de média 0 e com longa dependência. O estimador de Craigmile and Mondal (2020) é baseado na análise de wavelets não decimadas de X_t usando a classe de wavelets de Daubechies, caracterizada pela largura do filtro $L > 2$. Seja, $\{h_{j,l}\}_{j,l}$ para $j \in \{0, 1, \cdots\}$ e $l \in \{0, \cdots, L-1\}$ um filtro wavelet de Daubechies de largura para L. Seja $H_1(f)$ a transformada de Fourier para o filtro $\{h_{1,l}\}$ e $G_1(f) = e^{-i2\pi f(L-1)}H_1(1/2 - f)$ onde $f \in [-1/2, 1/2]$. Cada um dos filtros lineares (j), apresentam uma transformada de Fourier da forma $G_1(f), G_1(2f), \cdots, G_1(2^{j-2}f)$ e $H_1(2^{j-1}f)$. A largura de $\{h_{j,l}\}$ é igual a $L_j = (2^j - 1)(L-1) + 1$.

A representação wavelet de $\{X_t\}_{t\in\mathbb{Z}}$ tem coeficientes dados por

$$W_{j,t} = \sum_{l=0}^{L-1} h_{j,l}X_{t-l}.$$

Como $\{X_t\}_{t\in\mathbb{Z}}$ é um processo gaussiano estacionário de média zero, $\{W_{j,t}\}_{t\in\mathbb{Z}}$ é também um processo gaussiano estacionário de média zero para cada j. Seja, $v_j^2 := \mathrm{Var}(W_{j,t})$, então, pode-se mostrar que para $d \in [0, 1/2)$,

$$\log(v_j^2) \approx C + (2d-1)j\log(2),$$

para j grande e alguma constante C. Dado um estimador de v_j^2, $\hat{v}_j^2$ para $j \in \{j_0, \cdots, j_0 + m\}$, j_0 e m positivos, d pode ser estimado regredindo-se $\log(\hat{v}^2_{j_0}), \cdots, \log(\hat{v}^2_{j_0+m})$ em $j_0, \cdots, j_{0+m}$, mais

um intercepto. A principal contribuição em Craigmile and Mondal (2020) é propor um estimador não-viesado, consistente e assintoticamente normal para v_j^2 em processos com longa dependência na presença de dados faltantes. Mais detalhes podem ser encontrados em Craigmile and Mondal (2020).

4.2.2 Um método baseado em cópula

Nesta seção, revisamos o estimador baseado em cópula de d introduzido em Pumi et al. (2023) no caso particular em que é aplicado aqui. Para mais detalhes e toda a generalidade da metodologia, remetemos o leitor ao artigo original. Mais detalhes sobre a teoria de cópulas podem ser encontrados em Nelsen (2013). Seja, $\{X_t\}_{t\in\mathbb{Z}}$ uma série temporal estacionária com longa dependência e seja $\{C_\theta\}_{\theta\in\Theta}$ uma família paramétrica de cópulas para a qual existe um ponto $a \in \Theta$ tal que C_a é a cópula independência. Suponha que a cópula associada a (X_{n_0}, X_{n_0+h}), para $n_0 \in \mathbb{Z}$ e para todo $h > 0$ é C_{θ_h}. Em, Pumi et al. (2023) os autores apresentam uma relação entre o comportamento da sequência θ_h e o decaimento de $\gamma(h) = \mathrm{Cov}(X_t, X_{t+h})$, conforme h aumenta, e com base nessa relação, propõem um estimador baseado em cópula para o parâmetro de longa dependência d. Este é o primeiro (e até hoje o único disponível) estimador de d baseado em cópulas no contexto de séries temporais univariadas na literatura.

Mais especificamente, sejam $x_1, \cdots, x_n$ uma amostra de uma série temporal com longa dependência com parâmetro d, $\hat{F}_n$ a distribuição empírica calculada a partir da amostra, $\hat{F}_n'$ um estimador da densidade de X_t (como um estimador de densidade via kernel) e

$$\hat{K} = \iint_{(0,1)^2} \frac{1}{\hat{F}_n'\big(\hat{F}_n^{(-1)}(u)\big)\hat{F}_n'\big(\hat{F}_n^{(-1)}(v)\big)} \lim_{\theta\to a} \frac{\partial C_\theta(u,v)}{\partial\theta} du dv.$$

Seja $y_k := F_n^{-1}(x_k)$, $k \in \{1, \cdots, n\}$. Para $h \in \{1, \cdots, n-1\}$, formamos uma nova série temporal bivariada $\{\boldsymbol{u}_k^{(h)}\}_{k=1}^{n-h}$ tomando $\boldsymbol{u}_i^{(h)} := (y_i, y_{i+h})$, $i = 1, \cdots, n-h$. Observe que $\{\boldsymbol{u}_k^{(h)}\}_{k=1}^{n-h}$ pode ser considerado uma amostra (correlacionada) de C_{θ_h}, pelo teorema de Sklar. A partir dessas pseudo-observações, θ_h pode ser estimado utilizando-se um método razoável, como a inversão do τ de Kendall ou ρ de Spearman, ou por pseudo-máxima verossimilhança. Escolhemos dois inteiros positivos $0 < s < m < n$ e definimos o estimador de d como

$$\hat{d} := \underset{|d|<0.5}{\mathrm{argmin}} \left\{ \sum_{h=s}^{m} \left[\hat{K}\hat{\theta}_h - \frac{\Gamma(1-d)}{\Gamma(d)} h^{2d-1} \right]^2 \right\}. \tag{4.8}$$

Para séries temporais que apresentam dados faltantes, a ideia é estimar o parâmetro das cópulas de lag h considerando apenas pares de pseudo-observações presentes, que são usadas para obter $\{\hat{\theta}_s, \cdots, \hat{\theta}_m\}$. Uma vez obtida essa sequência, o procedimento de estimação permanece o mesmo. Os autores mostram que sob certas condições de regularidade, o estimador é consistente e satisfaz um teorema central do limite, embora com uma taxa de convergência mais lenta do que $\sqrt{n}$ e distribuição limite não-gaussiana. Observamos que a mesma prova pode ser usada para mostrar a consistência e obter um teorema central do limite na presença de dados faltantes, desde que o número de dados faltantes aumente mais lentamente que o tamanho da amostra. A distribuição limite e a taxa de convergência são as mesmas do caso de dados completos.

Para aplicar o estimador (4.8) algumas escolhas precisam ser feitas. Primeiro, devemos escolher a família paramétrica de cópulas a aplicar: um problema comum em metodologias baseadas em

cópulas. No entanto, os resultados da simulação apresentados em Pumi et al. (2023) mostram que o procedimento é robusto contra a especificação da cópula. Também é necessário escolher os estimadores para a densidade, a função de distribuição, a função quantílica e o estimador para o parâmetro de cópula. Os autores mostram que, desde que sejam escolhidos estimadores consistentes, a metodologia produz bons resultados. Os autores fornecem uma simulação comparando três tipos de estimadores de cópula, os métodos baseados na inversão de τ de Kendall, ρ de Spearman e o pseudo-máxima verossimilhança. Os resultados mostram pouca diferença entre os três. Eles também fornecem uma comparação entre o uso da distribuição marginal correta e o uso da distribuição empírica para $\hat{F}_n$, concluindo que a diferença é quase imperceptível. Quanto aos valores de s e m, os resultados da simulação apresentados em Pumi et al. (2023) sugerem que $s = 1$ e $m = 24$ rendem estimativas geralmente boas. Na presença de dados faltantes, os estimadores de F, F' e F^{-1} são calculados considerando apenas os valores observados.

4.2.3 Método baseado em Wavelet Lifting (LoMPE)

Nesta seção, discutiremos a abordagem baseada em Wavelet Lifting para a estimação do parâmetro de longa dependência. O método de estimação proposto por Knight et al. (2017) faz uso de uma transformada de elevação chamada de coeficiente de elevação (LOCAAT) proposta por Jansen et al. (2001). Seja, $\{X_t\}_{t\in\mathbb{Z}}$ uma série temporal estacionária com longa dependência de interesse. Para a estimativa do parâmetro d de longa dependência, é usada o método de (LoMPE). A descrição detalhada do algoritmo requer a introdução de vários detalhes que fogem do escopo deste trabalho. Tais detalhes podem ser encontrados em Knight et al. (2017).

Capítulo 5

Proposta de Trabalho

Neste trabalho propomos um estudo aprofundado do problema de estimação do parâmetro de longa dependência d no contexto de processos ARFIMA gaussianos, na presença de muitos dados faltantes. Para isso planejamos e executamos um extenso estudo de simulação de Monte Carlo, considerando 35 configurações diferentes para estimar d, avaliados em diversos cenários, incluindo diversos níveis de dependência e porcentagem de dados faltantes entre 10% e 70%. Foram utilizados três diferentes estimadores especialmente projetados para dados faltantes em diferentes configurações, também foram usados cinco dos estimadores semi-paramétricos mais amplamente utilizados na literatura, que foram combinados com 3 diferentes abordagens de imputação. Os resultados obtidos com o estudo proposto são apresentados em formato de artigo no Anexo A.

Capítulo 6

Conclusões e trabalhos futuros

Nesta dissertação, apresentamos um estudo de simulação no contexto de modelos ARFIMA para estimação do parâmetro de longa dependência em presença de muitos dados faltantes. Pontualmente, consideramos estimadores especialmente projetados para lidar com dados faltantes e estimadores que só podem ser usados após da imputação de valores faltantes, o que foi feito considerando três métodos diferentes. Avaliamos os métodos sob dezenas de cenários, incluindo porcentagens de dados faltantes variando de 10% a 70%, em diferentes tamanhos de amostra e valores para d.

Nossas descobertas sugerem que, em um ambiente de séries temporais com longa dependência, a imputação com a média deve ser evitada. Uma vez que a força da dependência é baixa, a aplicação de um estimador nativo (exceto o LoMPE) geralmente produz a melhor resposta, embora requer maior tempo computacional. Caso haja necessidade de utilizar um método tradicional de estimação e um método de imputação, o DFA combinado com a estimativa aleatória (ou linear) é a única alternativa aceitável, pois os outros mostram um desempenho muito ruim.

Trabalhos futuros incluem replicar os experimentos considerando modelos ARFIMA não-Gaussianos. A classe de modelos SYMARFIMA (do inglês, ARFIMA simétricos) (Benaduce and Pumi, 2023) pode ser uma alternativa viável, já que a geração de modelos ARFIMA com distribuições de caudas pesadas é desafiante em geral.

Referências Bibliográficas

Abry, P. and Veitch, D. (1998). Wavelet analysis of long-range-dependent traffic. *IEEE transactions on information theory*, 44(1):2–15.

Bardet, J.-M. and Kammoun, I. (2008). Asymptotic properties of the detrended fluctuation analysis of long range dependence processes. *IEEE Transactions on Information Theory*, 54(5):2041–2052.

Benaduce, H. and Pumi, G. (2023). SYMARFIMA: a dynamical model for conditionally symmetric time series with long range dependence mean structure. *Journal of Statistical Planning and Inference*, 225:71–88.

Beran, J. (1994). *Statistics for Long Memory Processes*. Chapman and Hall.

Brockwell, P. J. and Davis, R. A. (1991). *Time series: theory and methods*. Springer Science & Business Media.

Chan, N. H. and Palma, W. (1998). State space modeling of long-memory processes. *The Annals of Statistics*, 26(2):719 – 740.

Craigmile, P. F. and Mondal, D. (2020). Estimation of long-range dependence in gappy Gaussian time series. *Statistics and computing*, 30(1):167–185.

Doukhan, P., Oppenheim, G., and Taqqu, M. S. (2003). *Theory and Applications of Long-Range Dependence*. Birkhäuser Boston, MA.

Faÿ, G., Moulines, E., Roueff, F., and Taqqu, M. S. (2009). Estimators of long-memory: Fourier versus wavelets. *Journal of Econometrics*, 151(2):159–177.

Geweke, J. and Porter-Hudak, S. (1983). The estimation and application of long memory time series models. *Journal of time series analysis*, 4(4):221–238.

Granger, C. W. J. and Joyeux, R. (1980). An introduction to long memory time series and fractional differencing. *Journal of Time Series Analysis*, 1:15–30.

Haitovsky, Y. (1968). Missing data in regression analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(1):67–82.

Hipel, K. W. and McLeod, A. I. (1978). Preservation of the rescaled adjusted range: 3. fractional gaussian noise algorithms. *Water Resources Research*, 14(3):517–518.

Honsking, J. R. M. (1981). Fractional differencing. *Biometrika*, 1(68):165–176.

Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Transactions of the American society of civil engineers*, 116(1):770–799.

Hurvich, C. M., Deo, R., and Brodsky, J. (1998). The mean squared error of Geweke and Porter-Hudak's estimator of the memory parameter of a long-memory time series. *Journal of Time Series Analysis*, 19(1):19–46.

Jansen, M., Nason, G. P., and Silverman, B. W. (2001). Scattered data smoothing by empirical bayesian shrinkage of second-generation wavelet coefficients. In *Wavelets: Applications in Signal and Image Processing IX*, volume 4478, pages 87–97. SPIE.

Knight, M. I., Nason, G. P., and Nunes, M. A. (2017). A wavelet lifting approach to long-memory estimation. *Statistics and Computing*, 27(6):1453–1471.

Kokoszka, P. S. and Bhansali, R. J. (2001). Estimation of the long memory parameter: A review of recent developments and an extension. In Basawa, I. V., Heyde, C. C., and Taylor, R. L., editors, *Proceedings of the Symposium on Inference for Stochastic Processes*, IMS Lecture Notes, pages 125–150, Athens, Greece.

Lawrance, A. and Kottegoda, N. (1977). Stochastic modelling of riverflow time series. *Journal of the Royal Statistical Society: Series A (General)*, 140(1):1–31.

Little, R. J. and Rubin, D. B. (1987). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Mandelbrot, B. B. (1975). Limit theorem on the self-normalized range for weakly and strongly dependent process. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 31:271–285.

Meng, X.-L. (1994). Multiple-imputation inferences with uncongenial sources of input. *Statistical Science*, pages 538–558.

Molenberghs, G., Fitzmaurice, G., Kenward, M. G., Tsiatis, A., and Verbeke, G. (2014). *Handbook of Missing Data Methodology*. Handbook of Modern Statistical Methods. CRC Press, Boca Raton.

Nelsen, R. (2013). *An Introduction to Copulas*. Lecture Notes in Statistics. Springer New York, 2nd. edition.

Palma, W. (2007). *Long-memory time series: theory and methods*. John Wiley & Sons.

Peng, C. K., Buldyrev, S. V., Havlin, S., Simons, M., Stanley, H. E., and Goldberger, A. L. (1994). Mosaic organization of DNA nucleotides. *Phys. Rev. E*, 49:1685–1689.

Prass, T. S. and Pumi, G. (2021). On the behavior of the DFA and DCCA in trend-stationary processes. *Journal of Multivariate Analysis*, 182:104703.

Pumi, G., Prass, T. S., and Lopes, S. R. C. (2023). A novel copula-based approach for parametric estimation of univariate time series through its covariance decay. *Statistical Papers*, forthcoming.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rea, W., Oxley, L., Reale, M., and Brown, J. (2009). Estimators for long range dependence: An empirical study.

Reisen, V., Abraham, B., and Lopes, S. R. C. (2001). Estimation of parameters in ARFIMA processes: a simulation study. *Communications in Statistics - Simulation and Computation*, 30(4):787–803.

Robinson, P. M. (1995a). Gaussian semiparametric estimation of long range dependence. *The Annals of statistics*, pages 1630–1661.

Robinson, P. M. (1995b). Log-periodogram regression of time series with long range dependence. *The annals of Statistics*, pages 1048–1072.

Robinson, P. M. (2003). *Time Series with Long Memory*. Oxford University.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. Wiley.

Rubin, D. B. (1988). An overview of multiple imputation. In *Proceedings of the survey research methods section of the American statistical association*, volume 79, page 84. Citeseer.

Schafer, J. L. and Olsen, M. K. (1998). Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate behavioral research*, 33(4):545–571.

Shimotsu, K. and Phillips, P. C. B. (2005). Exact local Whittle estimation of fractional integration. *The Annals of Statistics*, 33(4):1890 – 1933.

Taqqu, M. S., Teverovsky, V., and Willinger, W. (1995). Estimators for long-range dependence: An empirical study. *Fractals*, 03(04):785–798.

Anexo A

Artigo Pumi et al. (2023)

Autores: Guilherme Pumi, Gladys Choque Ulloa e Taiane Schaedler Prass

Título: Estimation of Long-Range Dependent Models with Missing Data: to Input or not to Input?

Revista: a definir

Ano: 2023

Estimation of Long-Range Dependent Models with Missing Data: to Input or not to Input?

Guilherme Pumi[a,*], Gladys Choque Ulloa[a] and Taiane Schaedler Prass[a]

Abstract

Among the most important models for long-range dependent time series is the class of ARFIMA(p, d, q) (Autoregressive Fractionally Integrated Moving Average) models. Estimating the long-range dependence parameter d in ARFIMA models is a well-studied problem, but the literature regarding the estimation of d in the presence of missing data is very sparse. There are two basic approaches to dealing with the problem: missing data can be imputed using some plausible method, and then the estimation can proceed as if no data were missing, or we can use a specially tailored methodology to estimate d in the presence of missing data. In this work, we review some of the methods available for both approaches and compare them through a Monte Carlo simulation study. We present a comparison among 35 different setups to estimate d, under tenths of different scenarios, considering percentages of missing data ranging from as few as 10% up to 70% and several levels of dependence.

Keywords: Long-range dependence; time series analysis; missing data; semiparametric estimation; copulas.

1 Introduction

Long-range dependent processes have a long history and through time the subject has evolved into an essential component of time series analysis. We refer the reader to the book by Palma (2007), the compilations by Doukhan et al. (2003) and Robinson (2003), and, for an account of the history and early days' developments, the book by Beran (1994). In this work we are interested in the class of ARFIMA(p, d, q) models introduced by Honsking (1981) and Granger and Joyeux (1980), which is one of the most applied and studied classes of long-range dependent models in the literature. There are numerous estimation procedures for the long-range dependence parameter d, including methods based on state-space representation, spectral density, approximations to the likelihood, wavelets, detrended fluctuation analysis (DFA), and copulas (Hurst, 1951; Geweke and Porter-Hudak, 1983; Peng et al., 1994; Robinson, 1995a,b; Abry and Veitch, 1998; Chan and Palma, 1998; Palma, 2007; Faÿ et al., 2009; Pumi et al., 2023, and references therein). Monte Carlo simulation studies comparing different estimators have also been conducted (Taqqu et al., 1995; Kokoszka and Bhansali, 2001; Reisen et al., 2001; Rea et al., 2009; Faÿ et al., 2009).

Estimating d in the presence of missing data, on the other hand, is a much less studied problem with sparse literature. There are two approaches to handling missing data in time series. The first one, imputation, is the most widely used. The basic idea is to replace missing data with plausible values, and then proceed with the analysis as if no data were missing. Of course, there are many ways to do this. One of the simplest is to replace the missing values with the sample mean or median calculated for the non-missing cases, especially when in the context of stationary time series. For nonstationary time series, a missing value can be

*Corresponding author. This Version: February 28, 2023

[a]Programa de Pós-Graduação em Estatística - Universidade Federal do Rio Grande do Sul.

E-mails: guilherme.pumi@ufrgs.br (G. Pumi), gladyschoqueulloa7@gmail.com (G. U. Choque) and taianeprass@ufrgs.br (T. S. Prass)

replaced with the average of previous values up to that point, or by calculating the mean locally using a sliding window approach. When there are just a few missing values corresponding to a small percentage of the total sample size, any reasonable imputation method applied to a stationary time series tends to yield good results in the sense that estimated quantities should be close to those obtained if no data were missing, especially for point estimation. However, as the number of missing values increases, the quality of imputation-based estimation becomes poor. Imputation has the advantage of being quick and simple to implement, which adds to its appeal.

Despite their strengths, almost all conventional deterministic imputation methods suffer from three problems in the context of time series. First, variances tend to be underestimated, leading to biases in other parameters (like correlations) that depend on variances. Mean/median substitution, for example, replaces the presumably different missing values with a single value, reducing variance. It also has an effect on stationary distributions, artificially creating a point of mass in an otherwise continuous distribution. The second problem is that imputing an exogenous value into a time series changes the dependence structure in ways difficult to quantify or even understand. This is especially problematic considering that estimation in models of practical interest takes into account the time series' dependence structure. The third problem is equally serious. Standard error calculations presume that all data is accurate. The inherent uncertainty and sampling variability in the imputed values are not taken into account. As a result, standard errors and p-values may be distorted, leading to incorrect confidence intervals and hypothesis testing, as well as potentially misspecified models.

The second method for estimating d in the presence of missing data is to use an estimator that has been modified or designed specifically to handle missing data. An estimator in this group obviously has none of the three problems that imputation methods have since any reasonable estimator in this group should account for the absence of some observations and bypass their implications. The problem is that there are few options in the literature for estimators in that category, and the available options are usually more complex to implement and slower to run when compared to an imputation alternative.

It is intuitive that if the percentage of missing data is too high, the methodologies should break down in the sense that the estimated values are either so biased that they are practically useless, or cannot be computed due to numerical instability. In this direction, an important question is how high the percentage of missing data must be so that we cannot trust (or compute) a given methodology. Does the strength of the dependence affect this percentage at all? Must we always use an estimator adapted or specially made for missing data, or can we use the much simpler and faster approach of imputation? Is this answer influenced by the percentage of missing and/or strength of the dependence? In this work, we present a Monte Carlo simulation study to shed some light on these questions. We consider three different estimators specially tailored for missing data in different configurations, five of the most commonly used semiparametric estimators in the literature, paired with three different approaches to imputation, under 28 different scenarios of dependence strength and percentage of missing values. We also introduce a random imputation method especially tailored to closely mimic the original variance of the time series without introducing any potential outliers and while taking into account the time series' local dependence structure.

The paper is organized as follows. In the next Section, we introduce the estimators and the imputation methods applied in the simulation and discuss some of their properties. In Section 3 we present an extensive Monte Carlo simulation study to answer the questions posed in the introduction. Section 4 discusses our findings and presents our conclusions.

2 Framework

We say that $\{Y_t\}_{t\in\mathbb{Z}}$ is an ARFIMA(p, d, q) if it satisfies the difference equation

$$\phi(L)Y_t = \theta(L)(1-L)^{-d}\varepsilon_t, \tag{2.1}$$

where $\{\varepsilon_t\}$ is a zero mean white noise with $\sigma_\varepsilon^2 := \mathrm{Var}(\varepsilon_t) < \infty$, L is the backward shift operator, $\phi(L) := 1 - \phi_1 L - \cdots - \phi_p L^p$ and $\theta(L) := 1 + \theta_1 L + \cdots + \theta_q L^q$ are the AR and MA operators, respectively; $(1-L)^{-d}$ is a fractional differencing operator defined by the binomial expansion

$$(1-L)^{-d} = \sum_{j=0}^{\infty} \eta_j L^j, \qquad \text{with} \quad \eta_j := \frac{\Gamma(j+d)}{\Gamma(j+1)\Gamma(d)},$$

for $-1 < d < \frac{1}{2}$. It is usual to assume that ϕ and θ have no common roots and all roots of ϕ lie outside the unitary circle $\{z \in \mathbb{C} : |z| = 1\}$. Under these conditions, there exists a unique stationary solution for (2.1) with autocovariance satisfying

$$\gamma(h) \sim \kappa_d |h|^{2d-1}, \qquad \text{with} \quad \kappa_d := \sigma_\varepsilon^2 \left|\frac{\theta(1)}{\phi(1)}\right|^2 \frac{\Gamma(1-2d)}{\Gamma(1-d)\Gamma(d)}, \quad \text{as } h \to \infty. \tag{2.2}$$

For $0 < d < 1/2$, the covariance in (2.2) decays at a hyperbolic rate so that the autocorrelation is not absolutely summable.

In this work, our main interest is estimating the long-range parameter d under the presence of missing values. We assume that missing values occur completely at random, in the sense that the epoch at which a missing value occurs is independent of the time series' past and future values. We also assume, without loss of generality, that the time series' first and last values are never missing. We consider two approaches for dealing with missing values. The first is by employing a method specifically designed to estimate d without the need for missing data completion. Estimators in this class will be called *native estimators*. The second approach involves using an estimator that requires the time series to have no missing data. We proceed by imputing the missing values and then applying the estimator as if no values were missing. In this work, we shall consider five of the most commonly applied estimators for d, besides the native estimators, which can also be used when the time series is complete, and three methods for data imputation. In total, 35 different estimation procedures will be compared. In the following sections, we will briefly review each procedure.

2.1 Native methods

To the best of our knowledge, the earliest native estimator for d in the presence of missing values is Chan and Palma (1998), which introduces a state-space representation of an ARFIMA model and a modification of the Kalman filter to approximate the likelihood function in the presence of missing data. Although the methodology performed reasonably well in the authors' simulations, it suffers from being very slow, especially when compared to other alternatives, and as a result, it was not included in our simulation study.

In this work, we consider three different native estimators for d. The semiparametric estimators proposed by Knight et al. (2017) and Craigmile and Mondal (2020) are both wavelet-based, relying on a relationship between the undecimated wavelet variance and the parameter d. The former estimates the wavelet variance using wavelet lifting while the latter uses a specially designed estimator. The third method was proposed by Pumi et al. (2023), which is the only copula-based estimator for the long-range parameter d for univariate time

series in the literature. In Pumi et al. (2023), the authors derive a novel relationship between the decay of covariance $\gamma(h)$ in a time series $\{X_t\}_{t\in\mathbb{Z}}$ and the behavior of the copulas of pairs (X_0, X_h) as h increases. The relationship is used to propose an estimation procedure for any quantity of interest identifiable through the covariance decay. One special case is the parameter d in view of (2.2). The estimator is defined in the context of complete time series, but, being copula-based, the authors explain how the procedure naturally adapts to the missing data case; no simulation results in the case of missing data are provided in the paper. In the next two sections, we present details regarding these native estimators.

2.1.1 Craigmile and Mondal (2020)'s wavelet method

Let $\{X_t\}_{t\in\mathbb{Z}}$ be a stationary Gaussian long-range dependent time series of interest. The idea behind Craigmile and Mondal (2020)'s estimator is based on undecimated wavelet analysis of X_t using the Daubechies' class of wavelets, characterized by the filter width $L > 2$. Let $\{h_{j,l}\}_{j,l}$ for $j \in \{0, 1, \cdots\}$ and $l \in \{0, \cdots, L-1\}$ be a Daubechies' wavelet filter of even width L. The undecimated wavelet representation of $\{X_t\}_{t\in\mathbb{Z}}$ has coefficients given by

$$W_{j,t} = \sum_{l=0}^{L-1} h_{j,l} X_{t-l}.$$

Since $\{X_t\}_{t\in\mathbb{Z}}$ is a Gaussian process, for each j, $\{W_{j,t}\}_{t\in\mathbb{Z}}$ is a zero mean stationary Gaussian process. Let $v_j^2 := \mathrm{Var}(W_{j,t})$. It can be shown that in this case, for $d \in (0, 1/2)$,

$$\log(v_j^2) \approx C + (2d-1)j\log(2), \tag{2.3}$$

for large j and some constant C. Given an estimator of v_j^2, $\hat{v}_j^2$ for $j \in \{j_0, \cdots, j_0 + m\}$ for positive j_0 and m, d can be estimated by regressing $\log(\hat{v}_{j_0}^2), \cdots, \log(\hat{v}_{j_0+m}^2)$ in $j_0, \cdots, j_{0+m}$. The main contribution in Craigmile and Mondal (2020) is to propose an unbiased, consistent, and asymptotically normally distributed estimator for v_j^2 in long-range dependent processes containing missing data. However, to estimate d from $\hat{v}_j^2$, using a regression approach will depend on an unknown dispersion matrix. The authors propose an approximation for this matrix, leading to an estimator of d called the full estimator (called C.full here). A second estimator is also inspired by the work of Abry and Veitch (1998) for complete long-range dependent processes, using only the diagonal elements of the full dispersion matrix to estimate d. The authors called it the diagonal estimator (called C.abry here). More details can be found in Craigmile and Mondal (2020).

2.1.2 A copula-based method

In this section, we review the copula-based estimator of d, introduced by Pumi et al. (2023), in the particular case in which it is applied here. For more details and the full generality of the methodology, we refer the reader to the original paper. For a comprehensive introduction to copulas, as well as more details about the theory, we refer the reader to Nelsen (2013). Let $\{X_t\}_{t\in\mathbb{Z}}$ be a stationary long-range dependent time series of interest and let $\{C_\theta\}_{\theta\in\Theta}$ be a parametric family of copulas such that there exists a point $a \in \Theta$ for which C_a is the independent copula. Suppose that the copula associated to (X_{n_0}, X_{n_0+h}), for $n_0 \in \mathbb{Z}$ and all $h > 0$ is C_{θ_h}. In Pumi et al. (2023), the authors uncover a relationship between the behavior of θ_h and the decay of $\gamma(h) = \mathrm{Cov}(X_t, X_{t+h})$ as h increases, and based on this relationship, propose a copula-based estimator for the long-range parameter d. To the best

of our knowledge, this is the first (and, as of today, the only one available) copula-based estimator of d in the context of univariate time series in the literature.

More specifically, let $x_1, \cdots, x_n$ be a sample from a long-range dependent time series with parameter d and let $\hat{F}_n$ be the empirical distribution calculated from the sample, $\hat{F}_n'$ be an estimator of the density of X_t (such as a kernel density estimator) and let

$$\hat{K} = \iint_{(0,1)^2} \frac{1}{\hat{F}_n'\big(\hat{F}_n^{(-1)}(u)\big)\hat{F}_n'\big(\hat{F}_n^{(-1)}(v)\big)} \lim_{\theta \to a} \frac{\partial C_\theta(u,v)}{\partial \theta} du dv.$$

Let $y_k := F_n^{-1}(x_k)$, $k \in \{1, \cdots, n\}$. For $s \in \{1, \cdots, n-1\}$, we form a new bivariate time series $\{\boldsymbol{u}_k^{(s)}\}_{k=1}^{n-s}$ by setting $\boldsymbol{u}_i^{(s)} := (y_i, y_{i+s})$, $i = 1, \cdots, n-s$. Observe that $\{\boldsymbol{u}_k^{(s)}\}_{k=1}^{n-s}$ can be regarded as a (correlated) sample from C_{θ_s}, by Sklar's theorem. From these pseudo observations, θ_s can be estimated by a reasonable method, such as the inversion of Kendall's τ or Spearman's ρ, or by maximum pseudo-likelihood. We choose two positive integers $0 < s < m < n$ and define the estimator of d as

$$\hat{d} := \underset{|d|<0.5}{\operatorname{argmin}} \left\{ \sum_{h=s}^{m} \left[\hat{K}\hat{\theta}_h - \frac{\Gamma(1-d)}{\Gamma(d)} h^{2d-1} \right]^2 \right\}. \tag{2.4}$$

For time series presenting missing data, the estimator is defined by applying the copula estimator only to complete pseudo-observations, which are used to obtain $\{\hat{\theta}_s, \cdots, \hat{\theta}_m\}$. Once this sequence is obtained, the estimation procedure remains the same. The authors show that under mild regularity conditions, the estimator is consistent and satisfies a central limit theorem, although with a slower convergence rate than $\sqrt{n}$ and non-Gaussian limiting distribution. We observe that the same proof can be used to show consistency and obtain a central limit theorem under missing data, as long as the number of missing data increases slower than the sample size. The limiting distribution and convergence rate are the same as in the complete data case.

To apply the estimator (2.4) a few choices need to be made. First, we must choose the parametric family of copulas to apply: a common problem in copula-based methodologies. However, simulation results presented in Pumi et al. (2023) show that the procedure is robust against copula misspecification. It is also necessary to choose estimators for the density, the distribution function, the quantile function, and the estimator for the copula parameter. The authors show that as long as consistent estimators are chosen, under mild smoothness conditions, the methodology yields good results. The authors provide a simulation comparing three types of copula estimators, the methods based on the inversion of Kendall's τ and Spearman's ρ and the pseudo maximum likelihood. The results show little difference between the three. They also provide a comparison between using the correct marginal distribution against using the empirical distribution for $\hat{F}_n$, concluding that the difference is barely noticeable. As for the values of s and m, the simulation results presented in Pumi et al. (2023) suggest that $s = 1$ and $m = 24$ should yield generally good estimates. In the presence of missing data, the estimators of F, F', and F^{-1} are calculated considering only observed values.

2.1.3 Knight et al. (2017)'s LoMPE estimator

Another wavelet-based estimator for the long memory parameter d is presented in Knight et al. (2017). The method is based on a multiscale lifting transformation known as LOCAAT (lifting one coefficient at a time) proposed by Jansen et al. (2001) and it is similar in nature to Craigmile and Mondal (2020)'s method. The LoMPE's idea is to apply the LOOCAT method to obtain a collection of lifting coefficients, which, after suitable normalization, are used to

estimate the wavelet's coefficient variance. To estimate d, a regression approach similar to (2.3) is applied. Bootstrapped lifting trajectories can be used to improve estimation. The authors present a simulation considering small missing data percentages, from 5% to 20% using 50 bootstrapped trajectories. The method's main problem is its computational cost, which is high compared to other methods considered here. More details can be found in Jansen et al. (2001) and Knight et al. (2017).

2.2 Traditional methods

As mentioned in the introduction, there are several estimators to estimate d when the time series is complete. These estimators typically can't handle missing data naturally. However, they can still be used after the imputation of the missing data. In this Section, we review five well-known semiparametric estimators for d.

2.2.1 Rescaled Range Method (R/S)

Consider a sample $\{y_1, \cdots, y_n\}$ from a stationary long-memory process and let $x_t := \sum_{j=1}^{t} y_j$ for $t \in \{1, \cdots, n\}$ be the partial sums of the y_j's and let $s_n^2 := \frac{1}{n-1}\sum_{t=1}^{n}(y_t - \overline{y})^2$ be the sample variance, where $\overline{y} = x_n/n$. The *rescaled range statistic* (R/S), introduced by Hurst (1951), is defined by

$$R_n := \frac{1}{s_n}\left[\max_{1\leq t\leq n}\left\{x_t - \frac{t}{n}x_n\right\} - \min_{1\leq t\leq n}\left\{x_t - \frac{t}{n}x_n\right\}\right].$$

Denoting $Q_n := n^{-\frac{1}{2}-d}R_n$, it can be shown that $\log(R_n) = \mathbb{E}(Q_n) + (d+\frac{1}{2})\log(n) + \log(Q_n) - \mathbb{E}(Q_n)$. So that we can obtain an estimator of the long-memory parameter d by least squares. For instance, if $R_{t,k}$ is the R/S statistic based on a sample of size $k, \{y_t, \cdots, y_{t+k-1}\}$, for $1 \leq t \leq n-k+1$, then an estimator of d can be obtained by regressing $\log(R_{t,k})$ on $\log(k)$ for $1 \leq t \leq n-k+1$. Some asymptotic results of the R/S statistics are presented in Mandelbrot (1975).

2.2.2 Geweke and Porter-Hudak (1983)'s estimator (GPH)

Let $\{Y_t\}_{t\in\mathbb{Z}}$ be a stationary long-range dependent process with spectral density satisfying

$$f(\lambda) = f_0(\lambda)[2\sin(\lambda/2)]^{-2d}, \tag{2.5}$$

for some continuous function f_0. Taking logarithms on both sides of (2.5) evaluated at the Fourier sequences $\lambda_j := 2\pi j/n$, we obtain

$$\log\big(f(\lambda_j)\big) = \log\big(f_0(0)\big) - 2d\log\big(2\sin(\lambda_j/2)\big) + \log\left(\frac{f_0(\lambda_j)}{f_0(0)}\right). \tag{2.6}$$

From (2.6), Geweke and Porter-Hudak (1983) suggested a regression approach to estimate d. We start by estimating the spectral density using the periodogram, defined by

$$I(\lambda) := \left|\frac{1}{\sqrt{2\pi n}}\sum_{t=1}^{n} e^{it\lambda}Y_t\right|^2. \tag{2.7}$$

Upon writing

$$\log\big(I(\lambda_j)\big) = \log\left(\frac{I(\lambda_j)}{f(\lambda_j)}\right) + \log\big(f(\lambda_j)\big),$$

and combining it with (2.6), we obtain

$$\log\big(I(\lambda_j)\big) = \log\big(f_0(0)\big) - 2d\log\big(2\sin(\lambda_j/2)\big) + \log\left(\frac{I(\lambda_j)[2\sin(\lambda/2)]^{2d}}{f_0(0)}\right). \tag{2.8}$$

From (2.8), we can estimate d considering the first m ordinates of the periodogram by regressing $\log\big(I(\lambda_1)\big), \cdots, \log\big(I(\lambda_m)\big)$ in $2\log\big(2\sin(\lambda_1/2)\big), \cdots, 2\log\big(2\sin(\lambda_m/2)\big)$. The asymptotic properties of the GPH have been studied by Robinson (1995b) and Hurvich et al. (1998), among others. Under mild conditions, the GPH's estimator is consistent and asymptotically normally distributed with a convergence rate of $n^{4/5}$, independently of d.

2.2.3 Local Whittle estimator (LW)

Let $\{Y_t\}_{t\in\mathbb{Z}}$ be a stationary long-range dependent process with parameter d and spectral density f satisfying

$$f(\lambda) \sim G\lambda^{-2d}, \tag{2.9}$$

as $\lambda \to 0^+$ with $G > 0$. For an ARFIMA(p, d, q) process, $G = \left[\frac{\sigma_\varepsilon \theta(1)}{2\pi\phi(1)}\right]^2$. The semiparametric estimator, known as the local Whittle estimator, is defined by

$$\hat{d} = \underset{|d|<1/2}{\operatorname{argmin}}\{R(d)\}, \tag{2.10}$$

where

$$R(d) := \log\big(\widehat{G}(d)\big) - 2d\frac{1}{m}\sum_{j=1}^{m}\log(\lambda_j), \quad \text{for} \quad \widehat{G}(d) := \frac{1}{m}\sum_{j=1}^{m}\lambda_j^{2d} I(\lambda_j), \tag{2.11}$$

I is the periodogram defined in (2.7) and $0 < m < n/2$ is an integer. Estimator (2.10) was introduced and studied in Robinson (1995a), which proved its consistency and asymptotic normality under mild assumptions, with a $\sqrt{m}$ convergence rate.

2.2.4 Exact Local Whittle estimator (ELW)

The semiparametric estimator introduced by Shimotsu and Phillips (2005), called the exact local Whittle estimator (ELW) is very similar in nature to the local Whittle. The main difference is that the estimator is derived from algebraic manipulation of the Whittle likelihood without relying on approximations of the periodogram, being exact in this sense. The ELW is also applicable to be applied to the non-stationary region $d > 0.5$, but for simplicity, we will consider the range $|d| < 0.5$. Let

$$I_{\Delta^d}(\lambda) := \left|\frac{1}{\sqrt{2\pi n}}\sum_{t=1}^{n} e^{it\lambda}(1-L)^d(Y_t)\right|^2.$$

The estimator is defined in the same fashion as the local Whittle, but using I_{Δ^d} instead of I in (2.11), that is

$$\hat{d} = \underset{|d|<1/2}{\operatorname{argmin}}\{R(d)\},$$

where

$$R(d) := \log\big(\widehat{G}(d)\big) - 2d\frac{1}{m}\sum_{j=1}^{m}\log(\lambda_j), \quad \text{for} \quad \widehat{G}(d) := \frac{1}{m}\sum_{j=1}^{m}\lambda_j^{2d} I_{\Delta^d}(\lambda_j),$$

The asymptotic theory of the ELW is very similar to the local Whittle's but the conditions required on m are slightly stronger. Computationally, according to the authors, the ELW is about 10 times slower than the local Whittle, but it is still a very fast method in comparison to others (see Section 3.5). A discussion of the advantages and disadvantages of the ELW over the GPH and local Whittle as well as more details can be found in Shimotsu and Phillips (2005).

2.2.5 DFA-based estimator

The so-called *detrended fluctuation analysis* (DFA) method was introduced by Peng et al. (1994) and is based on the behavior of the detrended variance in long-range dependent time series. Before proceeding with the definition of the estimator, we introduce some notation, following the more general work of Prass and Pumi (2021).

Let $\{Y_t\}_{t\in\mathbb{Z}}$ be a stationary long-range dependent process with parameter d and let $Y_1, \cdots, Y_n$ be a sample from it. Let $R_t := \sum_{j=1}^{t} Y_j$ for $t \in \{1, \cdots, n\}$ denote the integrated signals. Divide the integrated signals into $k = \lfloor n/(m+1) \rfloor$ non-overlapping boxes each containing $m+1$ values, which we denote by $\boldsymbol{R}_i := (R_{(m+1)(i-1)+1}, \cdots, R_{i(m+1)})'$, for $i \in \{1, \cdots, k\}$. Next, in each box we fit a polynomial of degree $\nu+1$ via least squares, considering

$$D_{m+1}^{\top} := \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 2 & \dots & m+1 \\ \vdots & \vdots & \ddots & \vdots \\ 1^{\nu+1} & 2^{\nu+1} & \dots & (m+1)^{\nu+1} \end{pmatrix}, \qquad \begin{aligned} P_{m+1} &:= D_{m+1}(D_{m+1}^{\top}D_{m+1})^{-1}D_{m+1}^{\top}, \\ Q_{m+1} &:= I_{m+1} - P_{m+1}. \end{aligned}$$

Define

$$\boldsymbol{\mathcal{E}}_i := Q_{m+1}\boldsymbol{R}_i = (\mathcal{E}_i, \cdots, \mathcal{E}_{m+i})^{\top},$$

the vector of residuals at the i-th box. Let $f_{DFA}^2(m,i) := \frac{1}{m+1}\boldsymbol{\mathcal{E}}_i'\boldsymbol{\mathcal{E}}_i$, i.e. the sample variance of the residual in the i-th fit. The detrended variance F_{DFA}^2 is defined by

$$F_{DFA}^2(m) := \frac{1}{k}\sum_{i=1}^{k} f_{DFA}^2(m,i).$$

As described by Peng et al. (1994), for a long-range dependent process,

$$\sqrt{F_{DFA}^2(m)} \sim cm^{d+1/2},$$

for large m. Consequently, taking logarithms and denoting $L_m := \log\left(\sqrt{F_{DFA}^2(m)}\right)$, we obtain

$$L_m \sim \log(cm^{1/2}) + d\log(m), \tag{2.12}$$

from which we can estimate d by regressing $L_{s+1}, \cdots, L_{s+l}$ in $\log(s+1), \cdots, \log(s+l)$, for $s, l > 0$. In Bardet and Kammoun (2008), the consistency of the DFA-based estimator for d in the context of fractional Brownian motion is demonstrated under mild conditions, with a convergence rate of $\sqrt{n/m}$ and some control over how m increases. The authors also claim that the DFA-based estimator satisfies a CLT, but provide no proof of it.

2.3 Imputation methods

In this section, we review some of the imputation methods applied in the Monte Carlo simulation and introduce a new one. For a review of some of the `R` packages available for time series imputation and their performance in the estimation of ARMA models, see Moritz et al. (2015) and Moritz and Bartz-Beielstein (2017). An interesting alternative time series imputation method based on genetic (evolutionary) algorithms is presented in García et al. (2010). The proposed algorithm treats every missing point as a parameter to be estimated in order to minimize an objective depending on the mean, variance, and covariance structure of the time series. Its goal is to obtain a set of values to replace the missing data such that the observed time series' mean, variance, and covariance structure are preserved. The method, however, is complex and extremely slow in comparison to other approaches, so it was not considered here.

2.3.1 Mean Substitution

The mean substitution is among the simplest imputation methods available. It is based on substituting missing data with the mean calculated over the observed time series values. In other words, if $y_1, \cdots, y_n$ is a sample from a time series for which there are m missing values and denoting by $M := \{i : y_i \text{ is missing}\}$ the set containing the time epochs for which a value is missing (observed), we substitute a missing value y_i by

$$y_i = \frac{1}{\text{card}(M^{\complement})} \sum_{k \in M^{\complement}} y_k, \quad i \in M,$$

where for a set A, $\text{card}(A)$ denotes the number of elements (cardinality) in A. As mentioned in the introduction, the mean substitution's strength lies in its simplicity, but since a single value replaces all missing data, it induces some problems: points of masses are produced which affects the time series' variance, dependence structure, and standard errors calculation (for estimators). It also drastically affects estimators based on distributions, such as the estimator in subsection 2.1.2. Alternatively, it could be used the median instead of the mean to impute missing values. Since in this work, we are considering only symmetric time series, the median and mean should perform the same, so we choose the last since it is faster to calculate.

2.3.2 Linear interpolation

Interpolation using some simple model is quite a common practice in the literature. This can be achieved by a simple linear interpolation in the vicinity of the missing data. If y_t is missing, we apply a simple linear interpolation between the two nearest observed points. Let y_{t_1} and y_{t_2} be the two closest observed points in time satisfying $t_1 < t < t_2$. We impute y_t as

$$y_t = y_{t_1} + \left(\frac{y_{t_2} - y_{t_1}}{t_2 - t_1} \right)(t - t_1).$$

Linear interpolation is also a very simple method that, contrary to the mean substitution, imputes different values for each missing data, when the underlying distribution is absolutely continuous. However, it still underestimates the variance and affects the time series dependence structure. It also imputes wide gaps as a straight line, potentially affecting standard error calculation (for estimators).

2.3.3 Random Substitution

Random substitution is an imputation method based on drawing from a predetermined distribution to substitute a given missing value. The most common is to replace a missing value with a random value drawn from a uniform distribution, typically between the minimum and maximum observed values. This simple method has the advantage that no matter the size of a gap, the imputed values will never be equal. However, since values near the minimum and maximum observed values occur with the same probability as any other interval of the same length, this imputation method tends to inflate the variance of the time series, altering its underlying distribution and affecting its dependence structure.

In what follows, we propose a random substitution method that inherits information regarding the dependence structure on the immediate vicinity of the missing value being imputed. The proposed method is a hybrid of the last observation carried forward method, which consists in substituting each missing value with the most recent observed value, and the random substitution. Let $tN(\mu, \sigma^2, a, b)$ denote the truncated normal distribution, truncated in the interval (a, b), with mean $\mu \in \mathbb{R}$ and variance $\sigma^2 > 0$. If $Z \sim tN(\mu, \sigma^2, a, b)$, Z has density

$$f(x; \mu, \sigma^2, a, b) = \frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\sigma\left[\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)\right]} I(a < x < b),$$

where ϕ and Φ denote the density and distribution of a standard normal distribution, respectively. Let y_t, $t > 1$ be a missing value to be imputed. We propose imputing y_t by drawing $y_t \sim tN(y_{t-1}, \sigma^2, y_{(1)}, y_{(n)})$, where $y_{(1)} = \min\{y_i : i \in M^{\complement}\}$ and $y_{(n)} = \max\{y_i : i \in M^{\complement}\}$, $M := \{i : y_i \text{ is missing}\}$. The variance parameter σ^2 can be tuned in order to match the variance in the observed time series (see Section 3.2).

3 Simulation

In this section, we present a Monte Carlo simulation study to compare the different approaches to estimate the long-range dependent parameter d in the context of ARFIMA processes. We consider a total of 35 different combinations of estimation and imputation methods.

3.1 Data generation process (DGP)

In the Monte Carlo simulation study, we consider Gaussian ARFIMA processes generated using `R` package `arfima` (Veenstra, 2012), which, according to the package's documentation, generates "a sample from a multivariate normal distribution that has a covariance structure defined by the autocovariances generated for given parameters". The innovation variance was taken as 1. We generate samples of length 2,000 and discarded the first 1,000 observations as burn-in.

To generate the time series containing missing values, we first generate the complete time series, as described above, and then sample the appropriate number of time indexes for which the data is to be missing from a discrete uniform distribution in the set $\{2, \cdots, 999\}$. By design, the first and last values of the time series are never missing. Finally, the observations related to the sampled time indexes are set to NA. This procedure is repeated for each replication, resulting in a distinct missing data pattern for each generated time series. However, because we used the same random seed in each session, the pattern of missing data for scenarios with identical n and the percentage of missing data is the same. The missing data

pattern for scenario $d = 0.1$ with 10%, for instance, is the same as the missing data pattern for scenario $d = 0.4$ with 10% missing data.

3.2 Implementation details

In this Section, we shall review some computational details about each estimator's implementation and the imputation method used in the simulations.

Native methods

The `R` code for the native Craigmile and Mondal's wavelet method presented in Section 2.1.1 can be found in Craigmile's Github: github.com/petercraigmile/GappyLRD. We have made a few changes to the code to suit our purposes better. We apply Daubechi's D4 wavelet considering levels from 1 to 7. A set of R orthonormal Slepian tapers is used to estimate a specific dispersion matrix in the presence of missing data. The argument R in the code was kept at its default value of 7. More information can be found in Craigmile and Mondal (2020).

An `R` package to estimate the copula-based estimator presented in Section 2.1.2 will be available at CRAN soon. We consider two configurations for the estimator. In both cases, the copula parameter is estimated using the inversion of Spearman's ρ, and, for the estimation window, we apply $s = 1$ and $m = 24$. For the first one, called PP.G, we consider the Gaussian copula as the underlying copula family, while for the second one, called PP.F, we consider Frank's family. This choice was made because since we are simulating Gaussian ARFIMA processes, the Gaussian family is correctly specified. In practice, however, we can never be sure about the copula choice. By considering the misspecified Frank's family, we examine a more realistic scenario.

As for the native method LoMPE presented in Section 2.1.3, the method is implemented in `R` package `liftLRD` (Knight and Nunes, 2018). We consider the implemented bias-corrected estimator in the simulation. The long-range dependence parameter is estimated using a weighted least squares approach, in which the slope of the log-linear relationship between the artificial scales and the log of the integrals is used to re-weight the estimates. The slope in the energy-scale relationship is calculated using all wavelet levels. The LoMPE method is slow compared to the other methods considered, so we only calculate the minimum integral lifting trajectory to estimate d, as bootstrapping makes the method too slow for simulation purposes.

Traditional methods

The R/S method of section 2.2.1 was implemented by the authors. The estimator is obtained through ordinary least squares using the `R` function `lm`. Geweke and Porter-Hudak's estimator presented in Section 2.2.2 is implemented in `R` package `LongMemoryTS` (Leschinski, 2019). The first $m = \lfloor 1 + \sqrt{n} \rfloor$ Fourier frequencies were considered in the estimation. The Local Whittle and Exact local Whittle estimators, presented in Section 2.2.3 and 2.2.4, respectively, are also implemented in the package `LongMemoryTS`. In both cases, we considered $m = \lfloor 1 + \sqrt{n} \rfloor$. In the ELW, the initial value, Y_0 is considered known (or estimated). To fulfill this requirement, we consider the time series $\tilde{Y}_{t-1} := Y_t - Y_1$ for $t \in \{2, \cdots, n\}$ so that $\tilde{Y}_0 = 0$ is used. See Remark 2 in Shimotsu and Phillips (2005) for more details and the package `LongMemoryTS`'s documentation.

The DFA method presented in section 2.2.5 was implemented by the authors. Function F^2_{DFA} is calculated using the `R` package `DCCA` (Prass and Pumi, 2020), considering non-overlapping windows. Regression (2.12) is estimated via ordinary least squares using `R` function `lm` and considering $h \in \{50, \cdots, 100\}$, (i.e., $l = 50$ and $s = 50$).

Imputation methods

The mean substitution method was implemented by the authors. The Linear interpolation method is implemented in `R` package `zoo` (Zeileis and Grothendieck, 2005) through function `na.approx`. The proposed random interpolation method was implemented by the authors. Hyperparameters a and b were taken as the minimum and maximum observed values, respectively. The variance hyperparameter σ is user-chosen. The goal in defining σ is to use a value that closely approximates the sample standard deviation calculated over the observed values (S) for a variety of values of d and missing data proportions.

We conducted a Monte Carlo Simulation to determine the best value for σ in the context of Gaussian ARFIMA$(0, d, 0)$ processes. We generate time series of length $n = 1{,}000$ for $d \in \{0.1, 0.2, 0.3, 0.4\}$. A proportion of 30%, 50%, and 70% of missing data is induced in each time series. In each case, we use the proposed method for imputation considering $\sigma = S/\varsigma$, for $\varsigma \in \{4, 6, 8, 10\}$. The standard deviation of the imputed time series is then computed. We repeat the experiment 1,000 times for each scenario.

The simulation results are summarized in Table 1, which shows the average standard deviation of the generated (complete) time series, the standard deviation calculated from the observed values after introducing missing data (S), and the standard deviation calculated from the data after imputation for each scenario. In parentheses, we present the standard deviation of the estimates alongside the estimated standard deviation. The average standard deviation obtained from the generated time series and S are very similar in all scenarios, a consequence of the model's stationarity. The results show that $\varsigma = 10$ produces the closest results to S while introducing no additional variability. As a result, in the simulation we set $\varsigma = 10$.

Table 1: Simulation results presenting the average standard deviation calculated from the complete generated time series (complete), the observed time series after introducing missing data (S), and the imputed data using the proposed methodology. Also presented are the standard deviation of the estimates (in parentheses).

d	Missing	S	imputed time series				complete
			$\varsigma = 4$	$\varsigma = 6$	$\varsigma = 8$	$\varsigma = 10$	
0.1	0.3	1.007 (0.027)	1.017 (0.031)	1.011 (0.030)	1.008 (0.030)	1.007 (0.030)	1.007 (0.023)
	0.5	1.007 (0.032)	1.032 (0.041)	1.017 (0.040)	1.012 (0.040)	1.009 (0.040)	
	0.7	1.009 (0.040)	1.059 (0.056)	1.028 (0.054)	1.018 (0.053)	1.012 (0.053)	
0.2	0.3	1.039 (0.030)	1.050 (0.034)	1.043 (0.033)	1.041 (0.033)	1.040 (0.033)	1.040 (0.026)
	0.5	1.039 (0.035)	1.065 (0.043)	1.050 (0.043)	1.045 (0.042)	1.042 (0.042)	
	0.7	1.040 (0.044)	1.096 (0.061)	1.062 (0.058)	1.050 (0.057)	1.045 (0.056)	
0.3	0.3	1.112 (0.042)	1.124 (0.045)	1.117 (0.045)	1.114 (0.044)	1.113 (0.044)	1.113 (0.040)
	0.5	1.112 (0.046)	1.140 (0.053)	1.125 (0.052)	1.119 (0.052)	1.116 (0.051)	
	0.7	1.111 (0.054)	1.170 (0.069)	1.137 (0.066)	1.125 (0.065)	1.120 (0.065)	
0.4	0.3	1.256 (0.076)	1.270 (0.079)	1.261 (0.078)	1.258 (0.077)	1.257 (0.077)	1.256 (0.075)
	0.5	1.256 (0.078)	1.289 (0.084)	1.271 (0.082)	1.264 (0.082)	1.261 (0.082)	
	0.7	1.255 (0.082)	1.320 (0.094)	1.282 (0.090)	1.270 (0.091)	1.262 (0.090)	

3.3 ARFIMA$(0, d, 0)$ scenario

In this section we discuss the estimation of the long-range dependence parameter d in the context of Gaussian ARFIMA$(0, d, 0)$ processes for $d \in \{0.1, 0.2, 0.3, 0.4\}$ and missing data proportions $\{0.1, 0.2, \cdots, 0.7\}$. The native estimators presented in section 2.1 are used to estimate parameter d for each generated time series with missing values. We then proceed with the estimation of d for each estimator considering the originally generated time series, with no missing data, henceforth referred to as the "original" time series. Next, the time series with missing data are imputed using the three methods discussed in section 2.3. We estimate d for each imputation method and percentage of missing values using the native and the five estimators discussed in Section 2.2. In total 35 different estimation methods were considered. We repeat the experiment 1,000 times. Due to space limitations, we only present the results for $d \in \{0.1, 0.4\}$. The other cases are presented in the supplementary material that accompanies the paper.

Case $d = 0.1$

Table 2 shows the simulation results for $d = 0.1$. The estimated value of d for each estimator is presented in blocks organized by the type of time series considered, namely, native

(methods applied to the time series with missing data), mean, linear, and random (methods applied to the time series imputed using the mean, linear and the proposed random methods, respectively). We present the estimation for each percentage of missing values along with the estimated values for the original time series, presented in column "0". These values are repeated throughout the blocks for convenience. The best estimate in each block is highlighted in bold.

Table 2: Simulation results for the ARFIMA$(0, 0.1, 0)$ scenario.

$d = 0.1$									
Type	Estimator	Missing							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Native	Full	0.076	0.077	0.086	0.088	0.109	0.113	**0.116**	0.152
	Abry	0.081	0.083	0.086	**0.089**	**0.096**	**0.103**	**0.116**	0.142
	PP.G	**0.092**	**0.091**	**0.090**	**0.089**	0.089	0.088	0.082	**0.074**
	PP.F	0.086	0.086	0.085	0.084	0.084	0.083	0.079	0.071
	LoMPE	0.084	0.079	0.076	0.071	0.070	0.063	0.060	0.057
Mean	Full	0.076	0.068	0.060	0.052	0.045	0.035	0.027	0.019
	Abry	0.081	0.073	0.064	0.056	0.049	0.039	0.031	0.022
	PP.G	**0.092**	0.074	0.058	0.044	0.067	0.050	0.010	0.006
	PP.F	0.086	0.069	0.052	0.037	0.055	0.049	0.008	0.004
	LoMPE	0.084	**0.077**	**0.070**	0.062	0.053	0.046	0.036	0.029
	DFA	0.073	0.066	0.059	0.052	0.043	0.030	0.023	0.016
	GPH	0.034	0.020	0.013	0.001	-0.013	-0.033	-0.054	-0.076
	LW	0.048	0.045	0.041	0.038	0.034	0.030	0.028	0.022
	ELW	0.046	0.044	0.039	0.041	0.039	0.042	0.059	0.073
	R/S	0.145	0.140	0.136	**0.131**	**0.125**	**0.118**	**0.112**	**0.101**
Linear	Full	0.076	0.133	0.197	0.269	0.350	0.441	0.546	0.673
	Abry	0.081	0.131	0.188	0.254	0.331	0.418	0.521	0.650
	PP.G	**0.092**	0.120	0.143	0.163	0.182	0.201	0.221	0.247
	PP.F	0.086	**0.114**	0.139	0.161	0.183	0.206	0.233	0.266
	LoMPE	0.084	0.135	0.192	0.256	0.327	0.411	0.511	0.631
	DFA	0.073	0.073	**0.073**	**0.081**	**0.081**	**0.087**	**0.104**	**0.142**
	GPH	0.034	0.028	0.027	0.024	0.023	0.025	0.026	0.046
	LW	0.048	0.044	0.041	0.037	0.034	0.033	0.032	0.035
	ELW	0.046	0.042	0.039	0.038	0.034	0.033	0.036	0.043
	R/S	0.145	0.151	0.159	0.168	0.178	0.193	0.212	0.237
Random	Full	0.076	0.128	0.184	0.244	0.311	0.380	0.454	0.538
	Abry	0.081	0.126	0.177	0.233	0.296	0.365	0.442	0.531
	PP.G	**0.092**	0.118	0.141	0.162	0.184	0.207	0.231	0.262
	PP.F	0.086	**0.112**	0.137	0.159	0.183	0.209	0.237	0.272
	LoMPE	0.084	0.130	0.181	0.235	0.292	0.358	0.431	0.511
	DFA	0.073	0.070	**0.067**	**0.072**	**0.075**	**0.080**	**0.096**	**0.135**
	GPH	0.034	0.025	0.024	0.019	0.021	0.023	0.027	0.039
	LW	0.048	0.043	0.038	0.035	0.031	0.030	0.028	0.032
	ELW	0.046	0.041	0.037	0.035	0.034	0.031	0.035	0.043
	R/S	0.145	0.148	0.151	0.156	0.165	0.179	0.197	0.224

There is a lot to discuss from Table 2. First, we look at the best results. Interestingly, when there is no missing data, the native methods produce the best results. The copula-based methods presented the best results for complete data among the native estimators, followed closely by the LoMPE. When missing data is taken into account, for all percentages up to 50% the native estimators produce the best results overall by a wide margin. The behavior is somewhat wild in extreme cases (60% and 70%).

The effects of using the mean imputation method are very noticeable for all estimators. Among the imputation methods, the mean performs the worst for all percentages of missing, except for 70% for which it presented the best overall result. To begin, all methods underestimate d, with the exception of the R/S, which does the opposite. For all estimators, with the exception of the R/S, using the mean imputation method degrades the estimated values as the percentage of missing data increases. Most estimated values present a relative bias of over 50% in this case. The methods that suffer the most are the frequency domain ones, namely GPH, LW, and ELW. The R/S, however, behaves exactly the opposite and produces the best results when the mean imputation method is applied, even when the percentage of missing data increases.

When the best results from the linear and random imputation methods are compared, we find that they perform similarly, with the linear method having a slight advantage. The estimated values of native methods and the R/S for both imputation methods greatly overestimate the parameter and always increase with the percentage of missing data, rendering them effectively useless when more than 20% of the data is missing. With 10% of missing data, the copula-based ones (PP.F and PP.G) present the best overall performance. With 20% of missing data or more, the DFA is the one that performs the best for both imputation methods. The frequency domain estimators (GPH, LW, and ELW) consistently underestimate d and perform poorly, with relative bias exceeding 50%.

Boxplots of the results are presented in Figure 1. The columns represent the percentage of missing data (10%, 40%, and 70%). The first row presents the native methods applied to the data with missing values. From the second row on, we present the boxplot of the estimated values using each imputation method. As expected, the variability of the native estimators in the case of missing data increases as the percentage of missing data increases. The bias is also affected but to a lesser degree.

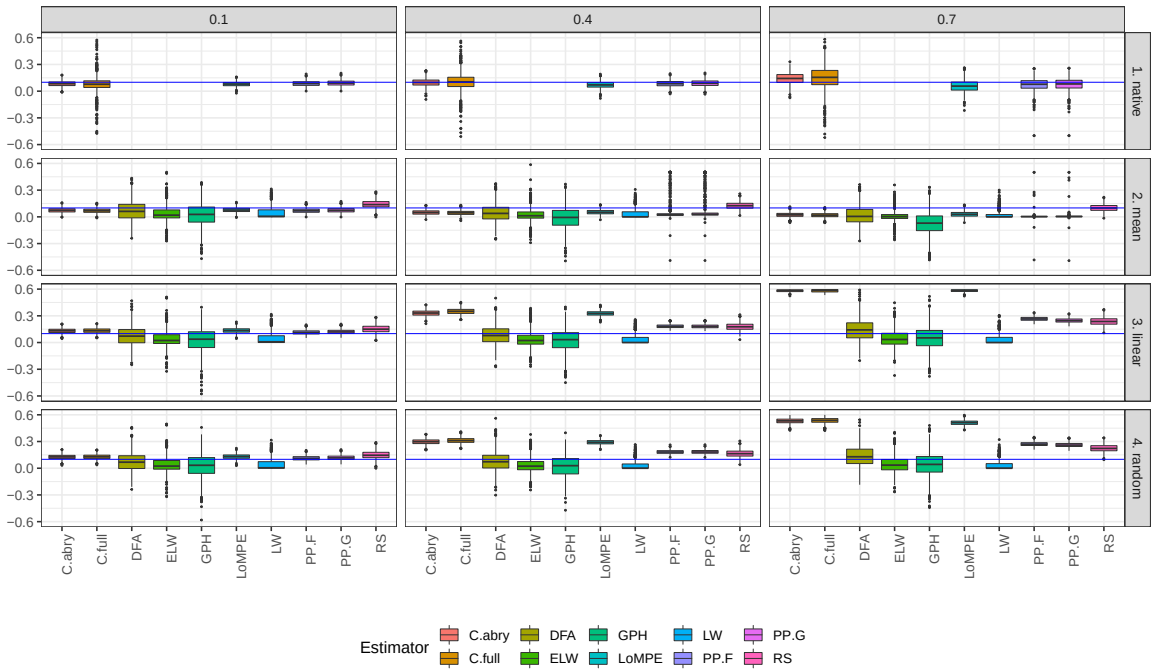

Figure 1: Box plot of the fitted ARFIMA $(0, 0.1, 0)$ model.

When data imputation is in place, the variability of the estimators does not seem to be significantly impacted by the percentage of missing data. The bias, however, is very much so.

The DFA, ELW, and GPH are the methods presenting the highest overall variability. It is also noteworthy that, for 10% of missing, the imputation method applied makes little difference in the boxplot within each method, but at 40% and 70% the imputation method plays an important role.

Case $d = 0.4$

Table 3 shows the simulation results for $d = 0.4$. When comparing the five native estimators, the copula-based ones consistently outperform the others, presenting a very good performance even when the percentage of missing data is 70%. When the original time series is taken into account, the GPH estimator is the best performer, followed by the copula-based ones. When there is no missing data, Craigmile and Mondal's (Full and Abry), LoMPE, and LW estimators all perform poorly.

The effects of using the mean imputation method are even more severe than in the case of $d = 0.1$ with the mean imputation method performing the worst of all imputation methods for all percentages of missing data. As the percentage of missing data increases, we observe a degradation of the estimated values with the mean imputation for all estimators, which uniformly underestimate d. The GPH was the best performer overall for mean imputation. Regardless of the method, when missing data reaches 30% or above, the estimated values are uniformly poor, to the point that they are useless for practical purposes.

The linear and random imputation methods perform about the same, just as in the case of $d = 0.1$. For percentages of missing data of 30% or above, the copula-based estimators are the best performers by a good margin, while the other native methods, the DFA, LW, ELW, and R/S increasingly degrade as the percentage of missing data increases beyond 30%, yielding useless estimations. The GPH also degrades as the percentage of missing data increases beyond 30% but to a lower degree.

Boxplots of the results are presented in Figure 2. From the boxplots, we observe that the variability of the spectral density-based estimators (GPH, EL, and ELW) are considerably greater than in the case $d = 0.1$ for all imputation methods. Comparing both cases, $d = 0.1$ and $d = 0.4$, we observe somewhat similar behavior for all estimators when the percentage of missing values is 10%. For missing data percentage of 70%, all imputation methods severely affect the wavelet-based estimators (Abry, Full, and LoMPE), causing such a bias that the estimated values are useless. In the case of linear imputation, for instance, all estimated values for these estimators are higher than 0.6, so the boxplots don't even appear in the plotting region of Figure 2.

The copula-based estimators perform well and consistently in the case of linear and random imputation as the percentage of missing values increases. In the case of mean imputation, however, the estimated values are useless for percentages of missing data above 10%. The DFA, GPH, ELW, and LW methods present a somewhat comparable overall performance, with a slight advantage for the GPH in most cases. The R/S performs stably for all percentages of missing values, with considerable bias, especially for the mean imputation case.

Table 3: Simulation results for the ARFIMA$(0, 0.4, 0)$ scenario.

$d = 0.4$									
Type	Estimator	Missing							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Native	Full	0.332	0.341	0.347	0.346	0.353	0.358	0.366	0.365
	Abry	0.344	0.350	0.352	0.355	0.358	0.360	0.362	0.366
	PP.G	**0.382**	**0.381**	**0.381**	**0.381**	**0.380**	**0.380**	**0.378**	**0.376**
	PP.F	0.379	0.379	0.378	0.379	0.378	0.378	0.376	0.375
	LoMPE	0.333	0.326	0.322	0.314	0.335	0.325	0.316	0.336
Mean	Full	0.332	0.294	0.260	0.227	0.196	0.165	0.133	0.102
	Abry	0.344	0.307	0.274	0.241	0.209	0.178	0.144	0.112
	PP.G	0.382	0.352	0.316	0.276	0.249	0.096	0.113	0.056
	PP.F	0.379	0.346	0.305	0.260	0.225	0.083	0.090	0.041
	LoMPE	0.333	0.297	0.265	0.235	0.205	0.176	0.147	0.116
	DFA	0.357	0.344	0.334	0.318	0.295	0.276	0.248	0.211
	GPH	**0.389**	**0.374**	**0.361**	**0.342**	**0.321**	**0.292**	**0.262**	0.215
	LW	0.314	0.307	0.300	0.292	0.278	0.262	0.243	0.215
	ELW	0.372	0.355	0.340	0.323	0.297	0.266	0.235	0.198
	R/S	0.342	0.332	0.321	0.309	0.291	0.273	0.250	**0.221**
Linear	Full	0.332	0.375	0.424	0.481	0.544	0.617	0.702	0.807
	Abry	0.344	0.381	0.423	0.473	0.531	0.599	0.681	0.786
	PP.G	0.382	0.383	0.384	0.385	0.388	0.391	0.393	**0.400**
	PP.F	0.379	0.381	0.384	**0.386**	**0.391**	**0.397**	**0.403**	0.416
	LoMPE	0.333	0.371	**0.415**	0.464	0.519	0.586	0.667	0.766
	DFA	0.357	0.352	0.351	0.345	0.340	0.335	0.336	0.344
	GPH	**0.389**	**0.384**	0.381	0.374	0.373	0.359	0.355	0.352
	LW	0.314	0.310	0.307	0.300	0.296	0.286	0.276	0.267
	ELW	0.372	0.367	0.363	0.354	0.351	0.341	0.329	0.322
	R/S	0.342	0.339	0.338	0.336	0.333	0.331	0.332	0.336
Random	Full	0.332	0.363	0.397	0.436	0.476	0.522	0.571	0.628
	Abry	0.344	0.369	**0.398**	0.432	0.470	0.513	0.565	0.626
	PP.G	0.382	0.382	0.383	**0.385**	**0.387**	0.391	0.394	**0.401**
	PP.F	0.379	0.380	0.382	0.384	**0.387**	**0.393**	**0.398**	0.409
	LoMPE	0.333	0.360	0.390	0.423	0.459	0.501	0.550	0.605
	DFA	0.357	0.349	0.344	0.335	0.324	0.317	0.307	0.313
	GPH	**0.389**	**0.383**	0.376	0.369	0.364	0.346	0.335	0.330
	LW	0.314	0.308	0.303	0.293	0.285	0.272	0.256	0.241
	ELW	0.372	0.365	0.358	0.347	0.338	0.325	0.310	0.298
	R/S	0.342	0.335	0.328	0.323	0.316	0.312	0.309	0.315

3.4 ARFIMA$(1, d, 1)$ scenario

In this section we consider the estimation of the long-range dependence parameter d in the context of Gaussian ARFIMA$(1, d, 1)$ processes for $d \in \{0.1, 0.2, 0.3, 0.4\}$, $\phi = 0.5$, $\theta = 0.6$ and missing data proportions $\{0.1, 0.2, \cdots, 0.7\}$. We follow the same steps as in Section 3.3. Tables and plots presenting the results can be found in the supplementary material that accompanies the paper. The results for the $(1, d, 1)$ case are very similar to the $(0, d, 0)$'s and the same remarks presented in Section 3.3 apply here case by case. The similarity between the results is expected since the estimators applied in the simulation are all semiparametric, focusing only on the long-range dependence structure in the time series, regardless of short-range nuisances.

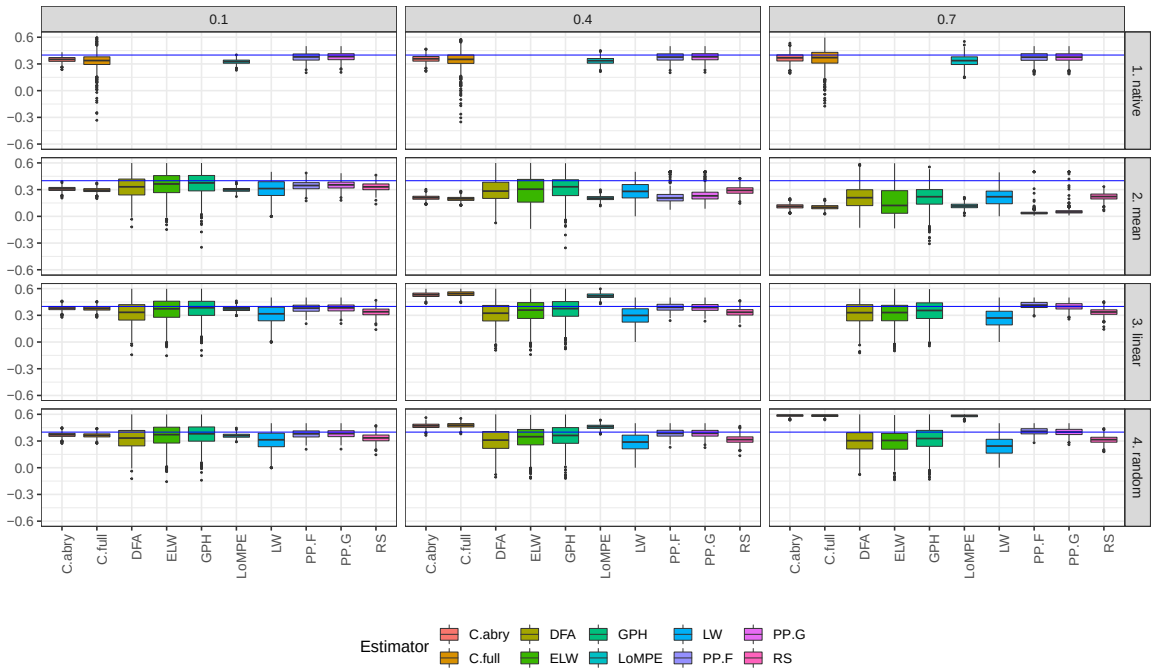

Figure 2: Box plot of the fitted ARFIMA$(0, 0.4, 0)$ model.

3.5 Time benchmarking

In this section, we compare the computational speed of each estimator considered in the simulations. Besides which estimator is the fastest to compute, there are a few other questions regarding computational speed that are of interest. For instance, does doubling the length of the time series double the time required to estimate d? Is the computational time required to estimate d affected by the strength of the dependence? Is the percentage of missing data a factor in estimation times? What about the imputation method applied? In this section, we study these questions through a series of Monte Carlo simulations.

3.5.1 Setup

We perform a series of routines with each estimator involving several different subtasks, measuring the time spent performing them for each estimator. The routine is divided into two main tasks. The first main task is intended only for the native estimators and consists of estimating d in Gaussian ARFIMA$(0, d, 0)$ generated with 20% and 70% of missing data, considering $d \in \{0.1, 0.4\}$ data and sample size $n \in \{1000, 2000\}$. The time spent executing each subtask 1,000 times for each estimator was recorded. The time series were all generated and prepared beforehand, so the recorded times reflect the time spent actually performing the task, and nothing else.

The second main task involves all estimators. The estimators are used to estimate d in the original time series. Next, the estimators are applied to estimate d from the time series with 20% and 70% of induced time series after imputation using each of the three methods presented in section 2.3 is performed. The exercise is performed 1,000, and the time spent performing it is recorded for each subtask and each estimator. Again, time series were all generated and prepared beforehand.

3.5.2 Results

The complete results are shown in Table 4 (time series with missing data) and Table 5 (original and imputed time series). For the Pumi et al. (2023)'s copula-based estimator, there is a significant difference in speed between the two variants considered. When considering the imputed (or original) time series, using Frank's family (PP.F) to perform the subtasks takes approximately 4.1 times longer than using the Gaussian's (PP.G) for $n = 1{,}000$, and approximately 3.2 when $n = 2{,}000$. Considering time series with missing data, these ratios increase to 4.3 and 3.4, respectively. This distinction is justified by the fact that the relationship between Spearman's ρ and Frank's copula parameter is dependent on special functions (c.f. Nelsen, 2013, p. 171) so estimating the parameter via inversion of Spearman's ρ requires a somewhat lengthy numerical inversion. On the other hand, Spearman's ρ has a closed formula for the Gaussian family, given by the classical relationship $\rho = \frac{6}{\pi}\arcsin(r)$, where r denotes the (Pearson's) correlation.

Table 4: Time spent to complete the simulation task for the native estimators. Presented is the total time spent, in seconds, performing the respective subtask.

Native Estimator	$n = 1{,}000$				$n = 2{,}000$			
	$d = 0.1$		$d = 0.4$		$d = 0.1$		$d = 0.4$	
	20%	70%	20%	70%	20%	70%	20%	70%
PP.G	94.2	88.4	93.4	87.7	130.8	120.7	129.5	120.0
PP.F	392.8	386.3	394.7	390.3	427.5	415.2	431.8	423.5
Abry	392.7	392.4	388.2	388.2	1367.6	1342.7	1334.9	1335.2
Full	392.8	392.4	388.2	388.3	1353.8	1341.8	1335.1	1335.0
LoMPE	244.8	68.8	238.1	68.1	745.6	162.4	713.2	161.3

Does doubling the time series' length double the time spent in estimating d?

It depends on the estimator and the setup. Considering only native estimators (Table 4), increasing the sample size from $n = 1{,}000$ to 2,000 took, on average, only 8.5% longer for PP.F to complete the tasks and 37.7% longer for PP.G. For the wavelet-based estimators, the scenario is very different. LoMPE and Craigmile and Mondal (2020)'s estimator took, on average, 2.7 and 3.4 times the amount of time to complete the task, respectively.

From Table 5, doubling the sample size from $n = 1{,}000$ to $n = 2{,}000$ requires less than twice the time to complete the task, on average, for the estimators GPH (1.81), LW (1.34) and DFA (1.65). The R/S and ELW took, on average, 4.3 and 9.3 times the amount of time to complete the task, respectively. Applying the native estimators to the original and imputed time series (Table 5) and doubling the sample size from $n = 1{,}000$ to $n = 2{,}000$ on average, produce an overall increase in the time spent to complete the tasks. For the PP.F and PP.G, this increase was small: 10% and 40.5% longer to complete the tasks, respectively. LoMPE and Craigmile and Mondal (2020)'s estimator took about 3.6 and 4.5 times the time spent to complete the tasks, on average, respectively.

Table 5: Time spent (in seconds) to complete the simulation task considering all estimators and imputed/original time series. Presented is the total time spent (in seconds) performing the subtasks.

n	d	%	input.	PP.G	PP.F	Abry	Full	LoMPE	GPH	LW	ELW	DFA	R/S
1,000	0.1	original		97.0	396.3	37.1	37.1	745.6	0.09	0.14	2.63	3.36	8.97
		0.2	mean	97.7	397.3	36.9	37.1	745.9	0.11	0.16	2.81	3.26	8.94
			linear	97.5	396.2	37.2	37.0	745.2	0.10	0.16	2.64	3.21	8.97
			rand	96.3	394.4	36.9	36.9	744.4	0.11	0.15	2.65	3.22	9.07
		0.7	mean	98.6	395.6	36.9	36.9	747.7	0.11	0.16	2.73	3.22	8.94
			linear	96.4	400.3	36.9	36.9	746.8	0.10	0.15	2.57	3.22	9.04
			rand	95.5	398.0	36.9	36.9	782.8	0.11	0.16	2.58	3.36	9.28
	0.4	original		95.9	397.3	36.6	36.6	728.3	0.11	0.11	2.19	3.19	9.03
		0.2	mean	96.6	395.4	36.5	36.5	733.1	0.11	0.11	2.39	3.20	8.88
			linear	95.5	403.3	36.6	37.0	735.8	0.11	0.10	2.17	3.12	8.95
			rand	95.4	398.0	36.5	36.5	732.4	0.10	0.11	2.20	3.18	9.00
		0.7	mean	97.7	392.8	36.5	36.4	732.2	0.11	0.11	2.65	3.19	8.87
			linear	95.0	408.6	36.5	36.5	733.5	0.11	0.11	2.33	3.18	8.87
			rand	94.6	402.0	36.5	36.5	735.7	0.11	0.11	2.23	3.18	8.87
2,000	0.1	original		142.3	448.5	171.1	171.1	2784.4	0.21	0.18	25.5	5.37	39.4
		0.2	mean	141.0	442.5	170.6	168.7	2808.4	0.20	0.18	25.6	5.38	38.7
			linear	136.5	431.9	168.8	166.3	2633.2	0.19	0.18	25.1	5.28	38.4
			rand	134.4	430.8	166.6	167.8	2636.9	0.19	0.21	25.0	5.30	38.4
		0.7	mean	135.4	429.6	166.7	167.6	2645.8	0.18	0.22	25.9	5.28	38.4
			linear	134.2	434.9	166.9	166.5	2661.0	0.21	0.22	25.1	5.65	39.7
			rand	138.7	453.9	170.5	169.9	2690.8	0.19	0.26	24.4	5.37	38.4
	0.4	original		133.6	435.4	165.7	165.5	2638.1	0.17	0.14	20.4	5.27	38.4
		0.2	mean	134.3	433.3	165.8	165.4	2652.5	0.20	0.14	20.9	5.26	38.5
			linear	133.5	438.4	165.8	165.4	2650.5	0.20	0.15	20.6	5.27	38.5
			rand	133.3	435.9	165.6	165.6	2641.4	0.19	0.15	20.4	5.25	38.4
		0.7	mean	134.8	429.0	165.7	165.8	2635.3	0.19	0.14	23.2	5.29	38.4
			linear	132.7	447.3	165.6	166.0	2638.0	0.18	0.14	20.8	5.25	38.4
			rand	132.0	441.4	165.3	165.7	2637.2	0.19	0.16	20.7	5.38	38.3

Does the dependence strength affect computational times?

It depends on the estimator. Completing the full task in the case $d = 0.1$ takes on average longer compared to $d = 0.4$ for all classical estimators. More precisely, comparing $d = 0.1$ versus $d = 0.4$, it takes about 42% longer for the LW, 18% for the ELW, 2% for the DFA, 1% for the R/S, and 0.25% for the GPH to complete the task. On the other hand, the time spent to complete the task using the native estimators is not significantly affected by the dependence strength - the difference in completing the full task when $d = 0.1$ takes no more than 2.3% in absolute value compared to $d = 0.4$.

Does the percentage of missing data affect computational times?

It depends on the estimator and whether the time series contains missing data or not. When missing values are considered, performing the tasks is not significantly affected by the percentage of missing values for estimators PP.G, PP.F Full, and Abry. For these estimators, when 20% of the data is missing, performing the task takes no more than 2% longer when compared to 70%. However, for LoMPE, performing the task when 20% of the data is missing takes about 4 times the time spent when 70% is missing.

When the time series is the original or imputed, the Whittle estimators LW and ELW are slightly affected by the percentage of missing value, taking about 6.6% and 2% longer to complete the task when 70% of the data is missing compared to 20%. The other estimators are affected for no more than 1.2%.

Does the imputation method applied to affect computational times?

The time spent completing the task after imputation does not depend on the percentage of missing values prior to imputation nor on the imputation method applied.

Which estimator is the fastest to compute?

It depends on the metrics and scenario. The native estimators are capable of handling missing data but they are more involved to calculate than the classic estimators. Hence it is expected that the classical estimators can be computed faster.

Table 6 presents the total time spent by each estimator to complete the whole routine, along with the minimum and maximum amount of time spent in a single subtask. Both Craigmile and Mondal (2020)'s variants, Full and Abry, spent about the same amount of time performing the full task. This is expected since the only difference in the estimators is the way a certain matrix is used to calculate the estimator (whole matrix or the diagonal entries only), at the end of the estimation algorithm. If we only consider the total amount of time

Table 6: Time spent to complete the simulation task considering all estimators and imputed/original time series. Presented is the average time spent (in seconds) over the percentage of missing and imputation methods to perform the subtasks.

metric	Original/imputed time series									
	PP.G	PP.F	Abry	Full	LoMPE	GPH	LW	ELW	DFA	R/S
total	3246.3	11708.2	2855.4	2852.4	47743.3	4.18	4.31	358.3	119.7	665.8
max	142.3	453.9	171.1	171.1	2808.4	0.21	0.26	25.9	5.65	39.7
min	94.6	392.8	36.5	36.4	728.3	0.09	0.10	2.17	3.12	8.87
metric	Time series with missing data									
total	864.8	3262.1	6941.9	6927.5	2402.3					
max	130.8	431.8	1367.6	1353.8	745.6					
min	87.7	386.3	388.2	388.2	68.1					

spent performing the task, when missing data is considered, the fastest estimator is the PP.G followed by LoMPE, which takes about 2.8 times the time PP.G takes to perform the whole

task. In this scenario, Craigmile and Mondal (2020)'s ones were the slowest, taking about 8 times the time spent by the PP.G to complete the task. Among the classical estimators, the fastest is the GPH followed closely by the LW. The third fastest is the DFA, but taking about 27.8 times the time to complete the task of the second one. The fastest among the native when the original/imputed time series is considered are the Craigmile and Mondal (2020)'s ones, followed closely by the PP.G with LoMPE being the slowest. The fastest native estimator (Full) took an astonishing 682 times GPH's total time to complete the task.

However, looking only at totals may not be ideal. For instance, in Table 4, we observe that when $n = 1{,}000$ and the percentage of missing is 20%, the PP.G is the fastest native estimator, while LoMPE is the fastest when 70% of the data is missing. When $n = 2{,}000$, the PP.G is uniformly faster though. Looking at the results presented in Table 5, we observe that the GPH and LW are the fastest estimators by far. For $n = 1{,}000$, the GPH is as fast or faster than LW in all but one subtask, while for $n = 2{,}000$, the LW is faster in 10 out of 14 subtasks.

A curiosity is that in Shimotsu and Phillips (2005), the authors claim (page 1891), based on their simulation experience, that the ELW is about ten times more expensive to compute than the LW. In the presented simulations, we found this number to be about 83 times more expensive, according to Table 6. Looking at Table 5, we found that the ELW is never less expensive than 16.1 times the LW, with a top value of 166 times. On average, the ELW is about 76.8 times more expensive to compute than the LW. This discrepancy could be due to the efficiency of the implementation applied in the original paper and the one used here.

3.5.3 Convergence

The classical semiparametric estimators applied are known to be computationally stable. The copula-based estimator is computationally stable as well, as is the LoMPE. Only Craigmile and Mondal (2020)'s presented computational issues. For most scenarios, the estimator failed in about 6% of the trials. Numerical instability is accentuated when $d = 0.1$ and the percentage of missing data is higher than 40%. In this scenario, when the percentage of missing is 70%, a third of all attempts to estimate d with Craigmile and Mondal (2020)'s estimators fail.

4 Discussion

In this work, we presented an extensive Monte Carlo simulation study regarding the estimation of parameter d in long-range dependent time series in the presence of missing data. We considered estimators especially tailored to deal with missing data, and estimators that can only be used after the imputation of the missing values, which was done considering three different methods. A variety of scenarios were considered, including percentages of missing data ranging from 10% to 70%, different sample sizes, and values of d.

Our findings show that in the context of long-range dependent time series, mean imputation should be avoided, in favor of the linear or random methods. When the dependence strength is low applying a native estimator (other than LoMPE) usually yields the best results, with a small advantage for the copula-based estimators, especially the PP.G given its numerical stability, comparatively low computational cost, bias, and variance. In this context, if a classical estimator is to be used, the DFA paired with the random or linear imputation is the only acceptable choice as the others present an extremely poor performance.

Under strong long-range dependence, the copula-based estimators applied to the gappy

time series present the overall best results, with a small advantage for the Gaussian variant, PP.G. If imputation is to be used, the copula-based estimators paired with the random or linear imputation yield the most consistent results, with a small advantage to the PP.F.

Our findings show that increasing the sample size has different effects on different estimators, for LoMPE, Full, Abry, R/S and ELW, doubling the sample size from $n = 1{,}000$ to $n = 2{,}000$ requires more than twice the time to complete the task, on average, while for the others, it requires less. We found that for most estimators, the value of d has a negligible effect on the time required to perform the estimation (exceptions: LW and ELW). The overall fastest estimators are by far the GPH and LW. Among the native estimator, the PP.G was the estimator that perform the fastest for $n = 2{,}000$. All estimators are very stable with exception of Craigmile and Mondal (2020)'s, which failed in producing an estimated value for d up to 33% of the time, depending on the context.

References

Abry, P. and Veitch, D. (1998). Wavelet analysis of long-range-dependent traffic. *IEEE Transactions on Information Theory*, 44(1):2–15.

Bardet, J.-M. and Kammoun, I. (2008). Asymptotic properties of the detrended fluctuation analysis of long range dependence processes. *IEEE Transactions on Information Theory*, 54(5):2041–2052.

Beran, J. (1994). *Statistics for Long Memory Processes*. Chapman and Hall.

Chan, N. H. and Palma, W. (1998). State space modeling of long-memory processes. *The Annals of Statistics*, 26(2):719 – 740.

Craigmile, P. F. and Mondal, D. (2020). Estimation of long-range dependence in gappy Gaussian time series. *Statistics and Computing*, 30(1):167–185.

Doukhan, P., Oppenheim, G., and Taqqu, M. S. (2003). *Theory and Applications of Long-Range Dependence*. Birkhäuser Boston, MA.

Faÿ, G., Moulines, E., Roueff, F., and Taqqu, M. S. (2009). Estimators of long-memory: Fourier versus wavelets. *Journal of Econometrics*, 151(2):159–177.

García, J. C. F., Kalenatic, D., and Bello, C. A. L. (2010). An evolutionary approach for imputing missing data in time series. *Journal of Circuits, Systems and Computers*, 19(01):107–121.

Geweke, J. and Porter-Hudak, S. (1983). The estimation and application of long memory time series models. *Journal of time series analysis*, 4(4):221–238.

Granger, C. W. J. and Joyeux, R. (1980). An introduction to long memory time series and fractional differencing. *Journal of Time Series Analysis*, 1:15–30.

Honsking, J. R. M. (1981). Fractional differencing. *Biometrika*, 1(68):165–176.

Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, 116(1):770–799.

Hurvich, C. M., Deo, R., and Brodsky, J. (1998). The mean squared error of Geweke and Porter-Hudak's estimator of the memory parameter of a long-memory time series. *Journal of Time Series Analysis*, 19(1):19–46.

Jansen, M., Nason, G. P., and Silverman, B. W. (2001). Scattered data smoothing by empirical bayesian shrinkage of second-generation wavelet coefficients. In *Wavelets: Applications in Signal and Image Processing IX*, volume 4478, pages 87–97. SPIE.

Knight, M. and Nunes, M. (2018). *liftLRD: wavelet lifting estimators of the Hurst exponent for regularly and irregularly sampled time series*. R package version 1.0-8.

Knight, M. I., Nason, G. P., and Nunes, M. A. (2017). A wavelet lifting approach to long-memory estimation. *Statistics and Computing*, 27(6):1453–1471.

Kokoszka, P. S. and Bhansali, R. J. (2001). Estimation of the long memory parameter: A review of recent developments and an extension. In Basawa, I. V., Heyde, C. C., and Taylor, R. L., editors, *Proceedings of the Symposium on Inference for Stochastic Processes*, IMS Lecture Notes, pages 125–150, Athens, Greece.

Leschinski, C. (2019). *LongMemoryTS: long memory time series*. R package version 0.1.0.

Mandelbrot, B. B. (1975). Limit theorem on the self-normalized range for weakly and strongly dependent process. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 31:271–285.

Moritz, S. and Bartz-Beielstein, T. (2017). imputeTS: time series missing value imputation in R. *The R Journal*, 9(1):207–218.

Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M., and Stork, J. (2015). Comparison of different methods for univariate time series imputation in R.

Nelsen, R. (2013). *An Introduction to Copulas*. Lecture Notes in Statistics. Springer New York, 2nd. edition.

Palma, W. (2007). *Long-memory time series: theory and methods*. John Wiley & Sons.

Peng, C. K., Buldyrev, S. V., Havlin, S., Simons, M., Stanley, H. E., and Goldberger, A. L. (1994). Mosaic organization of DNA nucleotides. *Phys. Rev. E*, 49:1685–1689.

Prass, T. S. and Pumi, G. (2020). *DCCA: Detrended Fluctuation and Detrended Cross-Correlation Analysis*. R package version 0.1.1.

Prass, T. S. and Pumi, G. (2021). On the behavior of the DFA and DCCA in trend-stationary processes. *Journal of Multivariate Analysis*, 182:104703.

Pumi, G., Prass, T. S., and Lopes, S. R. C. (2023). A novel copula-based approach for parametric estimation of univariate time series through its covariance decay. *Statistical Papers*, forthcoming.

Rea, W., Oxley, L., Reale, M., and Brown, J. (2009). Estimators for long range dependence: An empirical study.

Reisen, V., Abraham, B., and Lopes, S. R. C. (2001). Estimation of parameters in ARFIMA processes: a simulation study. *Communications in Statistics - Simulation and Computation*, 30(4):787–803.

Robinson, P. M. (1995a). Gaussian semiparametric estimation of long range dependence. *The Annals of statistics*, pages 1630–1661.

Robinson, P. M. (1995b). Log-periodogram regression of time series with long range dependence. *The annals of Statistics*, pages 1048–1072.

Robinson, P. M. (2003). *Time Series with Long Memory*. Oxford University.

Shimotsu, K. and Phillips, P. C. B. (2005). Exact local Whittle estimation of fractional integration. *The Annals of Statistics*, 33(4):1890 – 1933.

Taqqu, M. S., Teverovsky, V., and Willinger, W. (1995). Estimators for long-range dependence: An empirical study. *Fractals*, 03(04):785–798.

Veenstra, J. Q. (2012). *Persistence and anti-persistence: theory and software*. PhD thesis, Western University.

Zeileis, A. and Grothendieck, G. (2005). zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, 14(6):1–27.

Estimation of Long-Range Dependent Models with Missing Data: to Input or not to Input?

Supplementary Material

Guilherme Pumi[a,*], Gladys Choque Ulloa[a] and Taiane Schaedler Prass[a]

This supplementary material is intended to present some complementary results to the ones presented in the parent paper. Henceforth, We shall apply the same notation, nomenclature, and abbreviations as in the main paper without further mention or reference. To the interested reader we refer the main paper for details.

The results presented here cover two main scenarios. The first one is the case of simulated ARFIMA$(0, d, 0)$, for which we present the cases the case $d \in \{0.1, 0.4\}$ in the main paper and $d \in \{0.2, 0.3\}$ here. The results presented here are very similar to the ones presented in the paper. The discussion presented in the paper for the case $d = 0.1$ apply viz-a-viz to the case $d = 0.2$ here. The results for $d = 0.3$ presented here are very similar to those for $d = 0.4$ discussed in the main paper.

The second main scenario is the case of simulated ARFIMA$(1, d, 1)$, which is presented here. Since the estimators applied in the paper are semiparametric, the AR and MA contribution to the estimation of d should, in principle, be negligible. It is indeed the case and the results of every scenario simulated for the ARFIMA$(1, d, 1)$ are similar to the respective ARFIMA$(0, d, 0)$ counterpart. Hence, the discussion presented in the paper for the ARFIMA$(0, d, 0)$ apply viz-a-viz to the $(1, d, 1)$ case presented here.

[*]Corresponding author. This Version: March 1, 2023

[a]Programa de Pós-Graduação em Estatística - Universidade Federal do Rio Grande do Sul.

E-mails: guilherme.pumi@ufrgs.br (G. Pumi), gladyschoqueulloa7@gmail.com (G. U. Choque) and taianeprass@ufrgs.br (T. S. Prass)

1 ARFIMA$(0, d, 0)$ scenario

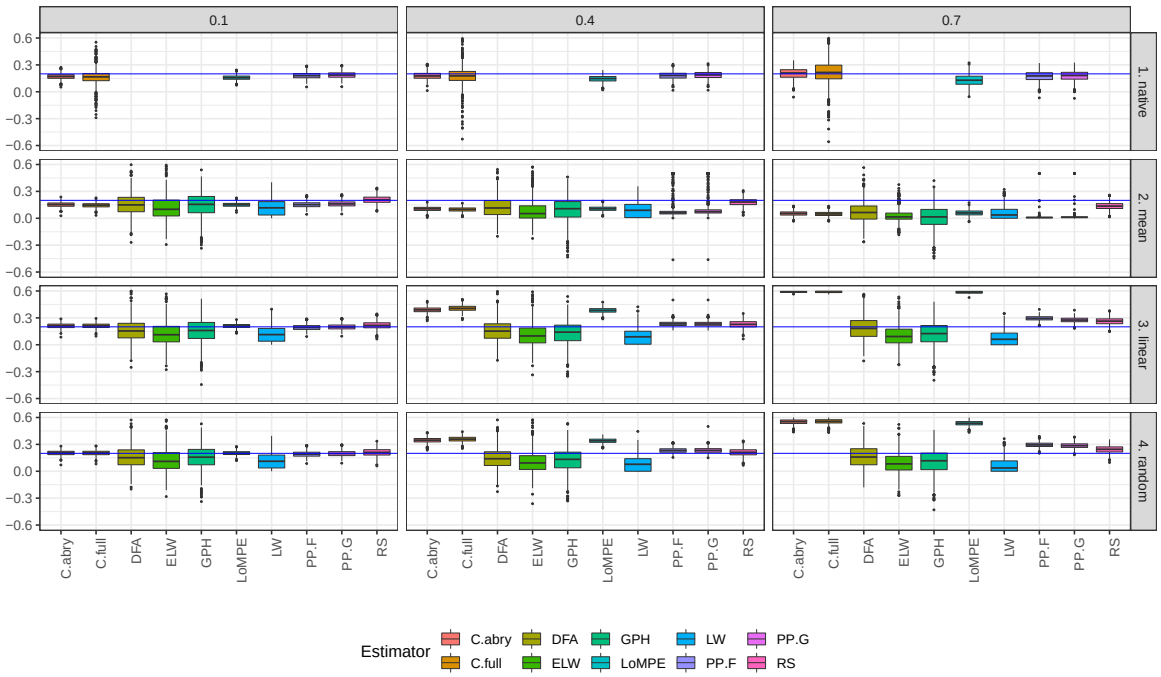

Figure 1: Box plot of the adjusted model ARFIMA $(0, d, 0)$ for $d = 0.2$.

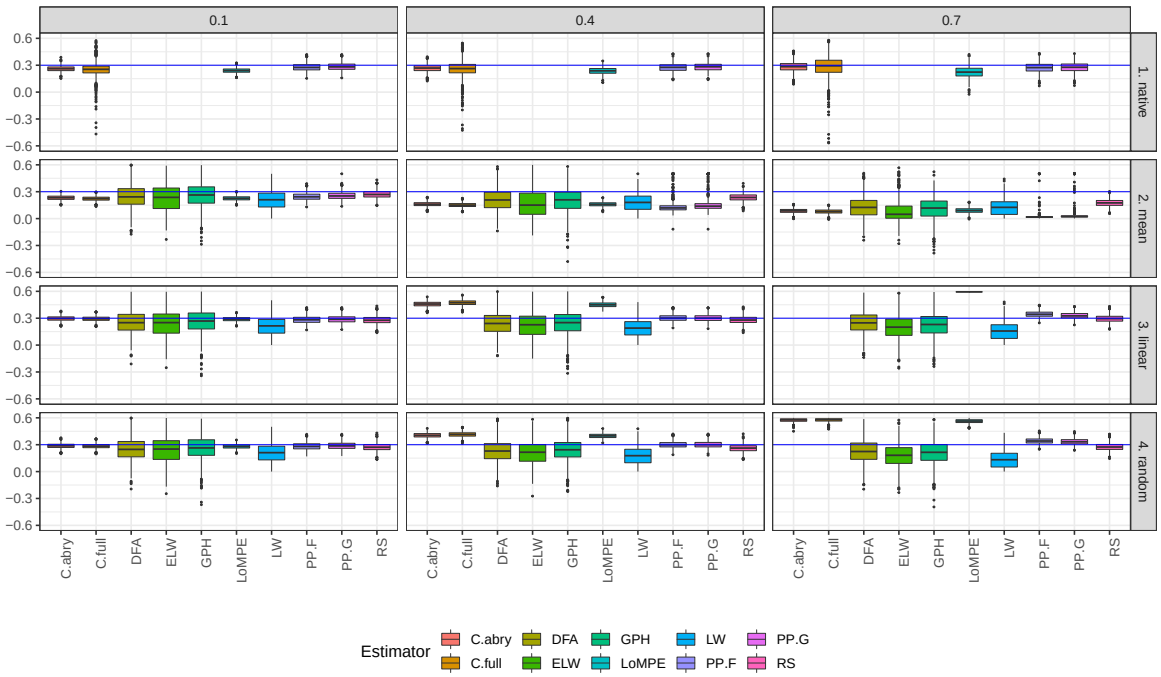

Figure 2: Box plot of the adjusted model ARFIMA $(0, d, 0)$ for $d = 0.3$.

Table 1: Simulation results for the ARFIMA $(0, 0.2, 0)$ scenario.

$d = 0.2$									
Type	Estimator	Missing							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Native	Full	0.160	0.164	0.164	0.170	0.181	**0.190**	**0.197**	0.225
	Abry	0.167	0.171	0.171	0.175	0.178	0.182	0.185	**0.203**
	PP.G	**0.188**	**0.188**	**0.187**	**0.186**	**0.187**	0.183	0.181	0.177
	PP.F	0.181	0.181	0.180	0.179	0.180	0.177	0.175	0.173
	LoMPE	0.164	0.157	0.151	0.144	0.145	0.136	0.127	0.127
Mean	Full	0.160	0.146	0.130	0.114	0.098	0.083	0.064	0.048
	Abry	0.167	0.152	0.136	0.121	0.105	0.089	0.070	0.053
	PP.G	**0.188**	0.161	0.131	0.102	0.114	0.092	0.030	0.015
	PP.F	0.181	0.152	0.120	0.090	0.095	0.066	0.023	0.011
	LoMPE	0.164	0.151	0.136	0.121	0.106	0.091	0.074	0.060
	DFA	0.164	0.156	0.143	0.132	0.121	0.104	0.087	0.066
	GPH	0.158	0.148	0.128	0.111	0.094	0.069	0.042	0.011
	LW	0.123	0.118	0.108	0.102	0.096	0.082	0.070	0.058
	ELW	0.130	0.120	0.106	0.094	0.085	0.072	0.071	0.081
	R/S	0.216	**0.207**	**0.200**	**0.189**	**0.181**	**0.168**	**0.154**	**0.136**
Linear	Full	0.160	0.212	0.268	0.333	0.407	0.494	0.587	0.707
	Abry	0.167	0.212	0.262	0.321	0.389	0.471	0.563	0.685
	PP.G	**0.188**	**0.200**	0.210	0.221	0.233	0.243	0.258	0.277
	PP.F	0.181	0.194	**0.206**	**0.219**	0.234	0.249	0.269	0.295
	LoMPE	0.164	0.210	0.260	0.318	0.383	0.461	0.551	0.669
	DFA	0.164	0.160	0.156	0.155	0.156	0.152	**0.162**	**0.184**
	GPH	0.158	0.155	0.143	0.135	0.132	0.118	0.124	0.116
	LW	0.123	0.118	0.109	0.104	0.096	0.084	0.080	0.077
	ELW	0.130	0.127	0.120	0.112	0.109	0.096	0.096	0.101
	R/S	0.216	0.217	0.219	0.222	**0.228**	**0.234**	0.246	0.263
Random	Full	0.160	0.204	0.250	0.302	0.358	0.420	0.485	0.563
	Abry	0.167	0.205	0.245	0.293	0.346	0.407	0.474	0.557
	PP.G	**0.188**	**0.198**	0.209	0.220	0.234	0.247	0.265	0.286
	PP.F	0.181	0.192	**0.204**	0.217	0.232	0.249	0.270	0.295
	LoMPE	0.164	0.203	0.243	0.290	0.340	0.397	0.460	0.536
	DFA	0.164	0.156	0.148	0.143	0.143	0.139	0.144	**0.163**
	GPH	0.158	0.153	0.136	0.130	0.123	0.110	0.111	0.107
	LW	0.123	0.116	0.103	0.098	0.087	0.074	0.070	0.065
	ELW	0.130	0.124	0.115	0.109	0.103	0.091	0.090	0.090
	R/S	0.216	0.211	0.210	**0.208**	**0.212**	**0.216**	**0.226**	0.244

Table 2: Simulated results for the ARFIMA$(0, 0.3, 0)$ scenario

$d = 0.3$									
Type	Estimator	Missing							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Native	Full	0.247	0.255	0.264	0.277	0.241	0.272	0.269	**0.296**
	Abry	0.256	0.261	0.263	0.264	0.266	0.270	0.273	0.283
	PP.G	**0.284**	**0.283**	**0.284**	**0.282**	**0.282**	**0.280**	**0.278**	0.275
	PP.F	0.278	0.277	0.278	0.277	0.277	0.275	0.274	0.271
	LoMPE	0.248	0.240	0.237	0.227	0.237	0.226	0.215	0.223
Mean	Full	0.247	0.223	0.200	0.175	0.151	0.128	0.103	0.079
	Abry	0.256	0.233	0.210	0.185	0.161	0.138	0.112	0.086
	PP.G	**0.284**	0.255	0.220	0.182	0.167	0.123	0.058	0.033
	PP.F	0.278	0.245	0.207	0.166	0.144	0.102	0.046	0.023
	LoMPE	0.248	0.226	0.204	0.181	0.159	0.138	0.115	0.090
	DFA	0.263	0.251	0.243	0.225	0.211	0.190	0.161	0.129
	GPH	0.273	0.257	0.247	0.227	0.203	0.179	0.149	0.108
	LW	0.216	0.208	0.203	0.190	0.179	0.167	0.147	0.125
	ELW	0.252	0.235	0.219	0.199	0.176	0.159	0.138	0.121
	R/S	0.280	**0.271**	**0.261**	**0.250**	**0.236**	**0.220**	**0.200**	**0.175**
Linear	Full	0.247	0.293	0.346	0.406	0.473	0.551	0.643	0.755
	Abry	0.256	**0.296**	0.342	0.395	0.457	0.531	0.621	0.733
	PP.G	**0.284**	0.288	**0.292**	**0.297**	**0.302**	**0.308**	0.316	0.328
	PP.F	0.278	0.283	0.289	0.295	0.303	0.312	0.326	0.345
	LoMPE	0.248	0.289	0.337	0.388	0.450	0.522	0.607	0.712
	DFA	0.263	0.258	0.256	0.250	0.246	0.241	0.242	0.254
	GPH	0.273	0.266	0.262	0.256	0.248	0.238	0.234	0.225
	LW	0.216	0.210	0.206	0.197	0.189	0.178	0.167	0.157
	ELW	0.252	0.246	0.242	0.236	0.224	0.215	0.207	0.198
	R/S	0.280	0.279	0.279	0.279	0.280	0.281	**0.286**	**0.295**
Random	Full	0.247	0.284	0.324	0.367	0.415	0.468	0.527	0.593
	Abry	0.256	**0.287**	0.322	0.360	0.405	0.458	0.518	0.589
	PP.G	**0.284**	**0.287**	**0.291**	**0.296**	**0.301**	**0.309**	**0.319**	0.333
	PP.F	0.278	0.282	0.287	0.293	**0.301**	0.310	0.323	0.342
	LoMPE	0.248	0.280	0.317	0.354	0.398	0.448	0.503	0.567
	DFA	0.263	0.254	0.249	0.239	0.231	0.223	0.221	0.227
	GPH	0.273	0.263	0.256	0.250	0.238	0.223	0.217	0.209
	LW	0.216	0.207	0.200	0.188	0.176	0.164	0.151	0.136
	ELW	0.252	0.245	0.236	0.228	0.212	0.203	0.190	0.180
	R/S	0.280	0.274	0.269	0.265	0.262	0.260	0.263	**0.276**

2 ARFIMA$(1, d, 1)$ scenario

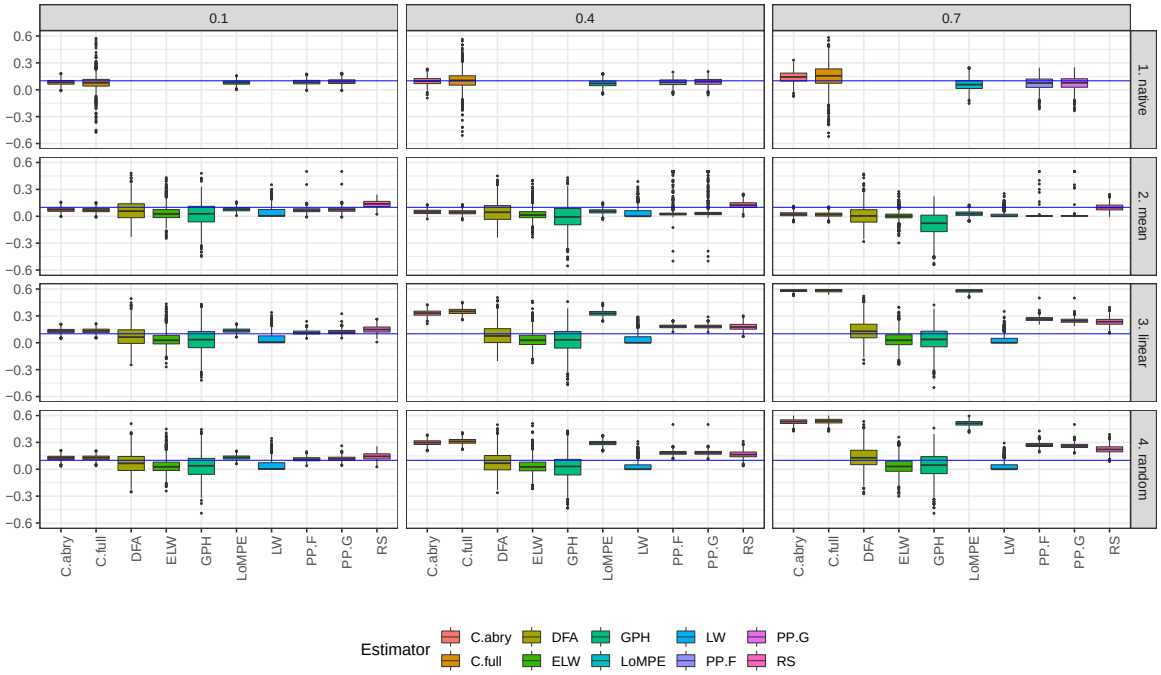

Figure 3: Box plot of the adjusted model ARFIMA $(1, d, 1)$ for $d = 0.1$.

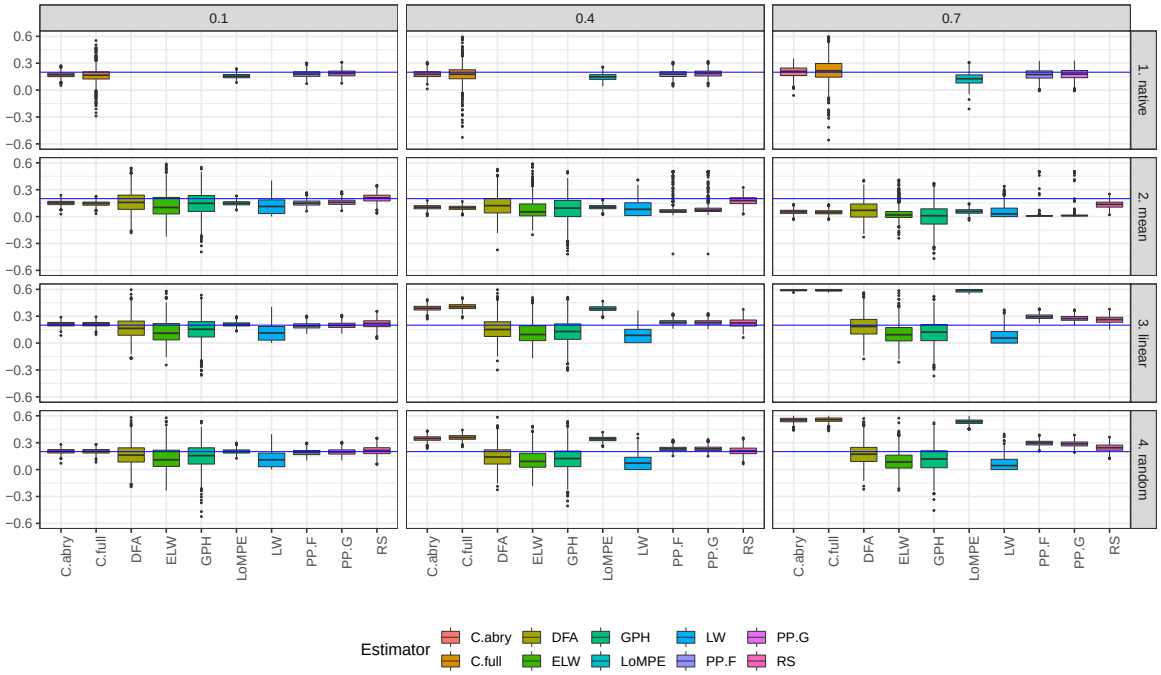

Figure 4: Box plot of the adjusted model ARFIMA $(1, d, 1)$ for $d = 0.2$.

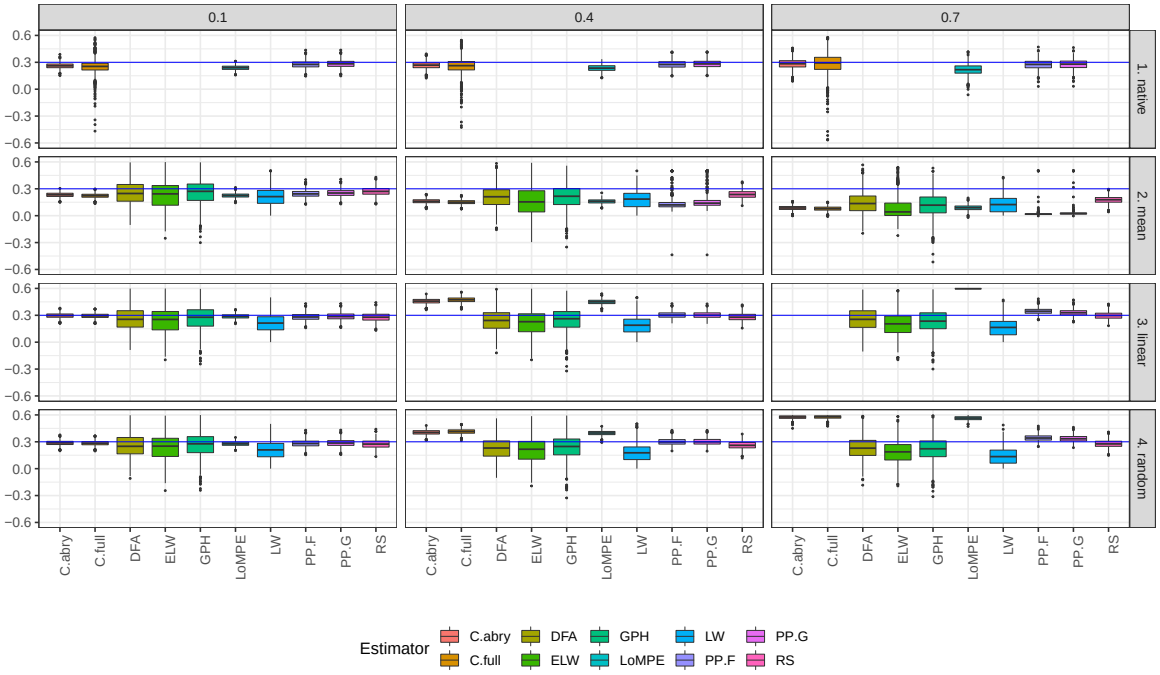

Figure 5: Box plot of the adjusted model ARFIMA $(1, d, 1)$ for $d = 0.3$.

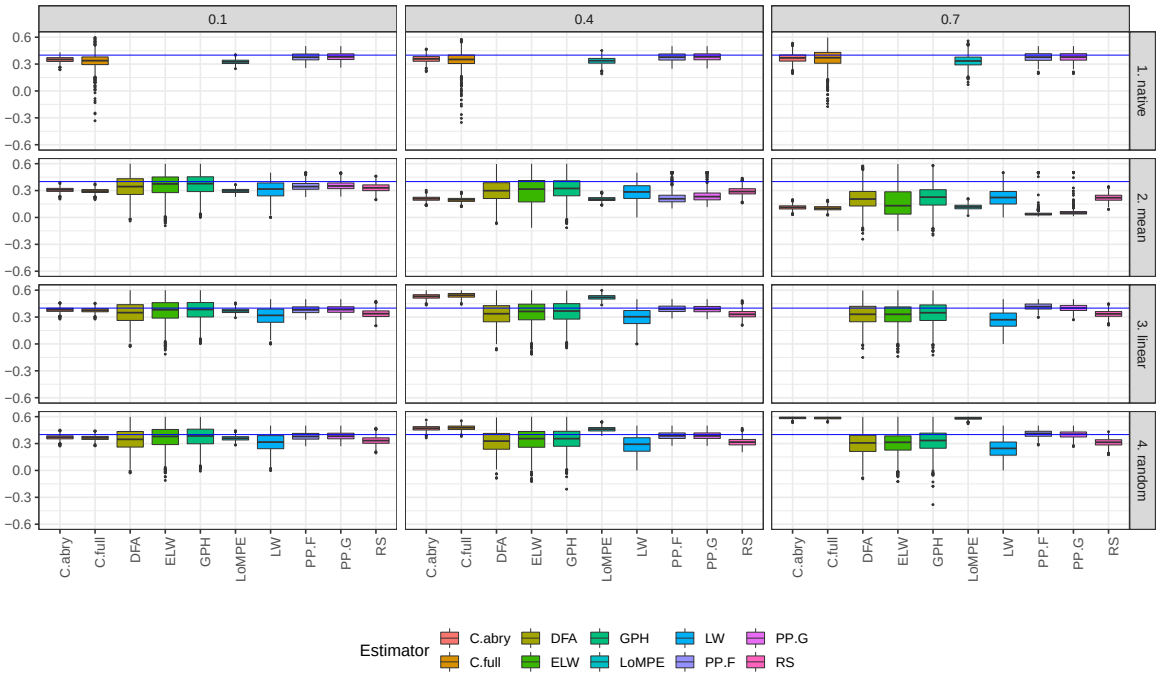

Figure 6: Box plot of the adjusted model ARFIMA $(1, d, 1)$ for $d = 0.4$.

Table 3: Simulation results for the ARFIMA$(1, 0.1, 1)$ scenario.

$d = 0.1$									
Type	Estimator	Missing							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Native	Full	0.076	0.077	0.086	0.088	0.109	0.113	**0.116**	0.152
	Abry	0.081	0.083	0.086	0.089	**0.096**	**0.103**	**0.116**	0.142
	PP.G	**0.092**	**0.091**	**0.091**	**0.090**	0.088	0.086	**0.084**	**0.073**
	PP.F	0.086	0.086	0.086	0.085	0.083	0.081	0.080	0.071
	LoMPE	0.085	0.080	0.076	0.071	0.070	0.066	0.062	0.057
Mean	Full	0.076	0.068	0.060	0.052	0.045	0.035	0.027	0.019
	Abry	0.081	0.073	0.064	0.056	0.049	0.039	0.031	0.022
	PP.G	**0.092**	0.075	0.057	0.043	0.062	0.054	0.012	0.007
	PP.F	0.086	0.069	0.051	0.037	0.053	0.055	0.009	0.005
	LoMPE	0.085	**0.078**	**0.070**	0.063	0.055	0.047	0.041	0.029
	DFA	0.072	0.066	0.057	0.053	0.047	0.037	0.024	0.008
	GPH	0.034	0.025	0.012	0.002	-0.010	-0.030	-0.047	-0.086
	LW	0.047	0.045	0.041	0.039	0.038	0.032	0.030	0.020
	ELW	0.044	0.041	0.037	0.036	0.039	0.044	0.050	0.065
	R/S	0.143	0.138	0.134	**0.130**	**0.125**	**0.117**	**0.109**	**0.098**
Linear	Full	0.076	0.133	0.197	0.269	0.350	0.441	0.546	0.673
	Abry	0.081	0.131	0.188	0.254	0.331	0.418	0.521	0.650
	PP.G	**0.092**	0.120	0.143	0.164	0.182	0.200	0.221	0.247
	PP.F	0.086	**0.115**	0.139	0.161	0.183	0.205	0.233	0.266
	LoMPE	0.085	0.135	0.192	0.256	0.329	0.412	0.512	0.632
	DFA	0.072	0.072	**0.072**	**0.078**	**0.083**	**0.094**	**0.108**	**0.134**
	GPH	0.034	0.032	0.029	0.026	0.029	0.028	0.031	0.036
	LW	0.047	0.046	0.040	0.038	0.039	0.033	0.032	0.031
	ELW	0.044	0.041	0.037	0.036	0.039	0.033	0.034	0.036
	R/S	0.143	0.149	0.157	0.167	0.178	0.192	0.209	0.234
Random	Full	0.076	0.128	0.184	0.244	0.311	0.380	0.454	0.538
	Abry	0.081	0.126	0.177	0.233	0.296	0.365	0.442	0.531
	PP.G	**0.092**	0.118	0.141	0.163	0.184	0.206	0.231	0.261
	PP.F	0.086	**0.112**	0.136	0.160	0.183	0.208	0.236	0.271
	LoMPE	0.085	0.131	0.180	0.235	0.293	0.359	0.431	0.512
	DFA	0.072	0.068	**0.067**	**0.070**	**0.072**	**0.083**	**0.101**	**0.130**
	GPH	0.034	0.030	0.023	0.021	0.025	0.021	0.029	0.041
	LW	0.047	0.044	0.038	0.035	0.033	0.029	0.028	0.031
	ELW	0.044	0.040	0.036	0.035	0.035	0.029	0.031	0.037
	R/S	0.143	0.145	0.150	0.156	0.164	0.177	0.193	0.222

Table 4: Simulation results for the ARFIMA$(1, 0.2, 1)$ scenario.

$d = 0.2$									
Type	Estimator	Missing							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Native	Full	0.160	0.164	0.164	0.170	0.181	**0.190**	**0.197**	0.225
	Abry	0.167	0.171	0.171	0.175	0.178	0.182	0.185	**0.203**
	PP.G	**0.188**	**0.187**	**0.186**	**0.185**	**0.185**	0.182	0.179	0.177
	PP.F	0.180	0.180	0.180	0.179	0.178	0.176	0.174	0.172
	LoMPE	0.164	0.156	0.151	0.142	0.146	0.138	0.130	0.125
Mean	Full	0.160	0.146	0.130	0.114	0.098	0.083	0.064	0.048
	Abry	0.167	0.152	0.136	0.121	0.105	0.089	0.070	0.053
	PP.G	**0.188**	0.161	0.131	0.105	0.108	0.068	0.027	0.018
	PP.F	0.180	0.152	0.120	0.092	0.091	0.072	0.021	0.013
	LoMPE	0.164	0.149	0.136	0.121	0.106	0.091	0.075	0.057
	DFA	0.172	0.164	0.152	0.138	0.125	0.109	0.089	0.069
	GPH	0.154	0.141	0.126	0.106	0.087	0.068	0.043	0.003
	LW	0.124	0.118	0.110	0.102	0.094	0.082	0.072	0.057
	ELW	0.135	0.126	0.112	0.097	0.088	0.079	0.076	0.086
	R/S	0.214	**0.207**	**0.198**	**0.189**	**0.178**	**0.167**	**0.152**	**0.135**
Linear	Full	0.160	0.212	0.268	0.333	0.407	0.494	0.587	0.707
	Abry	0.167	0.212	0.262	0.321	0.389	0.471	0.563	0.685
	PP.G	**0.188**	**0.199**	0.210	0.220	0.231	0.243	0.257	0.277
	PP.F	0.180	0.193	**0.206**	**0.218**	0.232	0.248	0.268	0.295
	LoMPE	0.164	0.208	0.259	0.317	0.385	0.463	0.554	0.666
	DFA	0.172	0.170	0.164	0.159	0.160	0.161	**0.165**	**0.189**
	GPH	0.154	0.148	0.142	0.133	0.127	0.120	0.115	0.117
	LW	0.124	0.119	0.110	0.103	0.096	0.087	0.080	0.076
	ELW	0.135	0.131	0.124	0.118	0.113	0.102	0.098	0.101
	R/S	0.214	0.216	0.218	0.220	**0.225**	**0.232**	0.242	0.261
Random	Full	0.160	0.204	0.250	0.302	0.358	0.420	0.485	0.563
	Abry	0.167	0.205	0.245	0.293	0.346	0.407	0.474	0.557
	PP.G	**0.188**	**0.198**	0.209	0.219	0.232	0.246	0.264	0.287
	PP.F	0.180	0.191	**0.204**	0.216	0.231	0.248	0.269	0.297
	Knight	0.164	**0.202**	0.243	0.290	0.341	0.399	0.463	0.535
	DFA	0.172	0.166	0.157	0.149	0.145	0.145	0.150	**0.172**
	GPH	0.154	0.146	0.137	0.130	0.115	0.114	0.103	0.110
	LW	0.124	0.116	0.105	0.097	0.086	0.076	0.069	0.066
	ELW	0.135	0.129	0.121	0.113	0.105	0.095	0.089	0.092
	R/S	0.214	0.211	0.208	**0.207**	**0.208**	**0.212**	**0.225**	0.244

Table 5: Simulation results for the ARFIMA$(1, 0.3, 1)$ scenario.

$d = 0.3$									
Type	Estimator	Missing							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Native	Full	0.247	0.255	0.264	0.277	0.241	0.272	0.269	**0.296**
	Abry	0.256	0.261	0.263	0.264	0.266	0.270	0.273	0.283
	PP.G	**0.283**	**0.282**	**0.282**	**0.281**	**0.282**	**0.280**	**0.277**	0.276
	PP.F	0.277	0.277	0.276	0.276	0.276	0.275	0.273	0.273
	LoMPE	0.247	0.239	0.234	0.225	0.235	0.224	0.210	0.218
Mean	Full	0.247	0.223	0.200	0.175	0.151	0.128	0.103	0.079
	Abry	0.256	0.233	0.210	0.185	0.161	0.138	0.112	0.086
	PP.G	**0.283**	0.253	0.217	0.180	0.171	0.094	0.058	0.029
	PP.F	0.277	0.244	0.205	0.164	0.149	0.082	0.046	0.021
	LoMPE	0.247	0.225	0.202	0.181	0.159	0.136	0.113	0.088
	DFA	0.268	0.257	0.243	0.229	0.212	0.196	0.167	0.138
	GPH	0.273	0.259	0.246	0.225	0.208	0.176	0.150	0.112
	LW	0.215	0.208	0.199	0.189	0.178	0.164	0.146	0.126
	ELW	0.250	0.235	0.218	0.192	0.177	0.150	0.135	0.120
	R/S	0.280	**0.271**	**0.261**	**0.249**	**0.237**	**0.221**	**0.201**	**0.177**
Linear	Full	0.247	0.293	0.346	0.406	0.473	0.551	0.643	0.755
	Abry	0.256	**0.296**	0.342	0.395	0.457	0.531	0.621	0.733
	PP.G	**0.283**	0.287	**0.292**	**0.295**	**0.301**	**0.307**	0.315	0.328
	PP.F	0.277	0.282	0.288	0.293	0.303	0.312	0.325	0.345
	LoMPE	0.247	0.288	0.333	0.386	0.448	0.520	0.603	0.711
	DFA	0.268	0.262	0.258	0.252	0.247	0.247	0.247	0.258
	GPH	0.273	0.269	0.265	0.252	0.250	0.235	0.227	0.233
	LW	0.215	0.210	0.204	0.193	0.186	0.173	0.165	0.160
	ELW	0.250	0.245	0.240	0.227	0.222	0.208	0.204	0.201
	R/S	0.280	0.279	0.279	0.278	0.280	0.282	**0.286**	**0.297**
Random	Full	0.247	0.284	0.324	0.367	0.415	0.468	0.527	0.593
	Abry	0.256	**0.287**	0.322	0.360	0.405	0.458	0.518	0.589
	PP.G	**0.283**	0.286	**0.291**	**0.294**	0.301	**0.308**	**0.318**	0.334
	PP.F	0.277	0.281	0.286	0.291	**0.300**	0.309	0.323	0.343
	LoMPE	0.247	0.279	0.313	0.352	0.398	0.446	0.500	0.567
	DFA	0.268	0.259	0.252	0.238	0.231	0.227	0.227	0.233
	GPH	0.273	0.266	0.263	0.245	0.239	0.221	0.215	0.216
	LW	0.215	0.207	0.200	0.186	0.175	0.158	0.149	0.139
	ELW	0.250	0.242	0.237	0.219	0.210	0.195	0.191	0.182
	R/S	0.280	0.274	0.269	0.263	0.261	0.261	0.264	**0.277**

Table 6: Simulation results for the ARFIMA$(1, 0.4, 1)$ scenario.

$d = 0.4$									
Type	Estimator	Missing							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Native	Full	0.332	0.341	0.347	0.346	0.353	0.358	0.366	0.365
	Abry	0.344	0.350	0.352	0.355	0.358	0.360	0.362	0.366
	PP.G	**0.384**	**0.383**	**0.383**	**0.383**	**0.382**	**0.380**	**0.380**	**0.378**
	PP.F	0.381	0.380	0.380	0.381	0.380	0.378	0.379	**0.378**
	LoMPE	0.332	0.325	0.322	0.314	0.336	0.325	0.312	0.334
Mean	Full	0.332	0.294	0.260	0.227	0.196	0.165	0.133	0.102
	Abry	0.344	0.307	0.274	0.241	0.209	0.178	0.144	0.112
	PP.G	0.384	0.354	0.317	0.277	0.253	0.102	0.114	0.062
	PP.F	0.381	0.348	0.307	0.261	0.229	0.086	0.092	0.046
	LoMPE	0.332	0.297	0.265	0.234	0.205	0.175	0.145	0.117
	DFA	0.367	0.355	0.344	0.326	0.305	0.274	0.251	0.213
	GPH	**0.393**	**0.379**	**0.363**	**0.346**	**0.323**	**0.292**	**0.267**	**0.220**
	LW	0.317	0.311	0.303	0.295	0.280	0.263	0.246	0.218
	ELW	0.380	0.368	0.353	0.332	0.303	0.276	0.234	0.204
	R/S	0.341	0.332	0.321	0.308	0.291	0.271	0.249	**0.220**
Linear	Full	0.332	0.375	0.424	0.481	0.544	0.617	0.702	0.807
	Abry	0.344	0.381	0.423	0.473	0.531	0.599	0.681	0.786
	PP.G	0.384	0.385	**0.386**	0.388	0.389	0.391	**0.396**	**0.402**
	PP.F	0.381	0.383	**0.386**	**0.389**	**0.392**	**0.397**	0.406	0.417
	LoMPE	0.332	0.371	**0.414**	0.462	0.520	0.587	0.664	0.765
	DFA	0.367	0.364	0.362	0.355	0.350	0.342	0.345	0.347
	GPH	**0.393**	**0.391**	0.385	0.382	0.375	0.363	0.359	0.353
	LW	0.317	0.314	0.311	0.305	0.298	0.286	0.278	0.269
	ELW	0.380	0.376	0.372	0.364	0.356	0.345	0.336	0.329
	R/S	0.341	0.339	0.336	0.335	0.332	0.330	0.332	0.335
Random	Full	0.332	0.363	0.397	0.436	0.476	0.522	0.571	0.628
	Abry	0.344	0.369	**0.398**	0.432	0.470	0.513	0.565	0.626
	PP.G	0.384	0.384	0.386	**0.387**	0.388	0.390	0.397	**0.404**
	PP.F	0.381	0.382	0.384	0.386	**0.389**	**0.393**	**0.401**	0.412
	LoMPE	0.332	0.359	0.389	0.422	0.460	0.502	0.547	0.605
	DFA	0.367	0.361	0.355	0.344	0.334	0.323	0.321	0.315
	GPH	**0.393**	**0.388**	0.382	0.374	0.363	0.354	0.343	0.331
	LW	0.317	0.312	0.307	0.298	0.287	0.273	0.260	0.244
	ELW	0.380	0.373	0.369	0.358	0.345	0.329	0.318	0.305
	R/S	0.341	0.334	0.328	0.321	0.316	0.310	0.311	0.314