

GUSTAVO DE CARVALHO LANA

INTERVALOS DE PREVISÃO EM MODELOS  
ARFIMA UTILIZANDO A METODOLOGIA  
BOOTSTRAP

UNIVERSIDADE FEDERAL DE MINAS GERAIS

Fevereiro 2012

GUSTAVO DE CARVALHO LANA  
ORIENTADORA: GLAURA DA CONCEIÇÃO FRANCO

INTERVALOS DE PREVISÃO EM MODELOS  
ARFIMA UTILIZANDO A METODOLOGIA  
BOOTSTRAP

Dissertação de apresentada ao Programa de Pós-Graduação em Estatística da  
Universidade Federal de Minas Gerais para a obtenção do título de Mestre em  
Estatística.

Belo Horizonte  
23 de Fevereiro de 2012

## **AGRADECIMENTOS**

Aos meus pais, Maurício e Sandra, pelo amor e carinho.

Aos meus irmãos, Bruno, Raquel, Marcela e Fabiano, pela eterna amizade.

À minha sobrinha, Lavínia, pela alegria que nos trouxe.

À minha orientadora, Glaura, pelo apoio prestado.

A todos aqueles que me ajudaram ao longo dos últimos 25 anos.

## RESUMO

Os métodos tradicionais de construção de intervalos de previsão para séries temporais assumem que os parâmetros do modelo são conhecidos e os erros normais. Quando estas suposições não são verdadeiras, o intervalo de previsão possui cobertura abaixo da nominal. Este trabalho propõe a utilização da metodologia *bootstrap* para construir intervalos de previsão com cobertura mais próxima da nominal. Dois intervalos *bootstrap* são utilizados, o intervalo PRR e o intervalo EPB. O intervalo PRR é uma adaptação para o modelo ARFIMA do intervalo homônimo proposto por Pascual et al. (2004) para o modelo ARIMA. O intervalo EPB proposto neste trabalho é similar a intervalos propostos para outros modelos de séries temporais, como o de Masarotto (1990) para modelos autorregressivos. Correções *bootstrap* de vício, incluindo o vício no desvio padrão dos resíduos, são testadas como possíveis fontes de melhoras dos intervalos. Os métodos utilizados no trabalho também foram testados para séries ARMA. O trabalho conclui que os intervalos PRR e EPB melhoram significativamente a qualidade dos intervalos de previsão em comparação com o intervalo assintótico, e a correção *bootstrap* de vício no desvio padrão dos resíduos também pode ser útil neste sentido.

## ABSTRACT

The traditional methods of building prediction intervals for time series assume that the model parameters are known and the errors are Gaussian. When such assumptions are not true, the prediction intervals possess a coverage different from the nominal one. This work proposes the use of the bootstrap methodology to build prediction intervals with coverage closer to the nominal. Two bootstrap intervals are used, the PRR interval and the EPB interval. The PRR interval is an adaptation for the ARFIMA model of the homonymous interval proposed by Pascual et al. (2004) for the ARIMA model. The EPB interval proposed in this work is similar to intervals proposed for time series models, such as the one of Masarotto (1990) for autoregressive models. Bootstrap bias corrections, including the bias of the residuals standard deviation, are tested as possible sources of improvement for the intervals. The methods used in this work were also tested for ARMA series. The work concludes that the PRR and EPB intervals improve significantly the quality of the prediction intervals in comparison with asymptotic one, and the bootstrap bias corrections of the residuals standard deviation may also be useful on that goal.

## SUMÁRIO

1 INTRODUÇÃO .....	1
2 O MODELO ARFIMA .....	4
2.1 Introdução .....	4
2.2 Estacionariedade e Invertibilidade .....	5
2.3 Função de Densidade Espectral .....	5
2.4 Estimadores .....	6
2.4.1 O Estimador de Fox-Taququ .....	7
2.4.2 O Estimador GPH .....	8
2.4.3 O Estimador SPR .....	9
2.5 Previsão .....	9
3 INTERVALOS DE PREVISÃO BOOTSTRAP .....	11
3.1 O Método <i>Bootstrap</i> .....	11
3.2 O Intervalo Percentílico .....	11
3.3 <i>Bootstrap</i> Não-Paramétrico em Séries Temporais .....	12
3.4 Tipos de Intervalos de Previsão <i>Bootstrap</i> .....	13
3.4.1 Intervalo PRR .....	14
3.4.2 Intervalo Utilizando os Erros de Previsão <i>Bootstrap</i> .....	15
3.5 Correção <i>Bootstrap</i> de Vício .....	15
3.6 Correção do Vício dos Resíduos .....	17
4 RESULTADOS DE SIMULAÇÕES .....	19
4.1 Estudo Monte Carlo dos Estimadores .....	22
4.2 Resultados das Simulações sobre os Intervalos de Previsão .....	26
5 APLICAÇÃO A UMA SÉRIE REAL.....	34
6 CONSIDERAÇÕES FINAIS .....	37
REFERÊNCIAS .....	39
ANEXO – Programa das Simulações .....	41

## 1 INTRODUÇÃO

A construção de intervalos de previsão para séries temporais é um tópico de interesse na literatura além de ter grande utilidade na prática. Os métodos de construção de intervalos de previsão para séries temporais em geral fazem suposições como normalidade e conhecimento dos parâmetros. Estas suposições geralmente não são satisfeitas, o que afeta a cobertura dos intervalos. Entre os modelos de séries temporais mais utilizados, se encontram o modelo ARIMA (Box e Jenkins, 1976), o modelo de espaço de estados (West e Harrison, 1997 e Harvey, 1989), os modelos ARCH (Engle, 1982) e GARCH (Bollerslev, 1986) e o modelo ARFIMA (Granger e Joyeux, 1980 e Hosking, 1981). O modelo ARFIMA é uma generalização do modelo ARIMA para lidar com séries que apresentam a característica de memória longa, decaimento lento da função de autocorrelação. A característica de memória longa aparece com frequência em séries financeiras e climáticas.

O principal objetivo deste trabalho é a utilização de métodos *bootstrap* para a construção de intervalos de confiança em modelos ARFIMA. Como em outros modelos de séries temporais, os intervalos comumente utilizados para séries ARFIMA supõem normalidade e conhecimento dos parâmetros. A propriedade de memória longa presente no modelo ARFIMA implica no domínio da frequência que o espectro é não limitado na frequência zero, enquanto que no domínio do tempo, memória longa implica que as autocorrelações não são absolutamente somáveis.

No modelo ARFIMA o parâmetro de diferenciação assume valores reais, ao contrário do modelo ARIMA, onde tal parâmetro é um número natural. Existem vários métodos para se estimar o parâmetro de diferenciação. Neste trabalho serão utilizados três estimadores: o estimador paramétrico de Fox-Taquq (Fox e Taquq, 1986), que se utiliza de uma função de verossimilhança aproximada, e os estimadores semiparamétricos GPH (Geweke e Porter-Hudak, 1983) e SPR (Reisen, 1994). Os três estimadores são consistentes. A vantagem dos estimadores GPH e SPR sob o estimador de Fox-Taquq é a maior facilidade de implementação. Tanto o GPH quanto o SPR se utilizam do periograma, um estimador da função espectral, para estimar o parâmetro de diferenciação. A diferença está no fato de que o SPR se utiliza de um periodograma suavizado.

O método *bootstrap* (Efron, 1979) vem sendo frequentemente utilizado na literatura de séries temporais. O *bootstrap* pode ser utilizado para construir intervalos de

confiança para os parâmetros do modelo, construir intervalos de previsão para valores futuros da série, realizar testes de hipóteses, corrigir o vício de estimadores, entre outras utilidades.

Muitos trabalhos foram publicados onde se utiliza o método *bootstrap* para a construção de intervalos de predição em séries temporais, especialmente para modelos ARMA ou autorregressivos puros. Em um trabalho seminal nesta área, Thombs e Schucany (1990) constroem intervalos para modelos autorregressivos puros realizando replicações *bootstrap* utilizando a forma *backward* do modelo e fixando-se os últimos  $p$  valores da série, onde  $p$  é a ordem do maior coeficiente autorregressivo. Outros métodos para modelos autorregressivos podem ser encontrados em Masarotto (1990) e Grigoletto (1998). Em Masarotto não há a necessidade de se conhecer a forma *backward* do modelo para realizar as replicações *bootstrap*. Em Grigoletto, o método desenvolvido por Masarotto é refinado de modo a aumentar a eficiência computacional. Pascual, Romo e Ruiz (2004) propõem um método para a construção de intervalos de previsão via *bootstrap* para modelos ARIMA, também sem precisar utilizar a forma *backward* do modelo.

Os métodos mencionados anteriormente apresentam melhora em relação ao intervalo assintótico, mas ignoram o efeito da possível existência de vício nos estimadores, o que pode afetar os resultados. Kim (2001) mostra que a correção de vício em modelos autorregressivos causa uma melhora substancial nos intervalos de previsão. Neste trabalho, Kim utilizou a correção de vício *bootstrap* para séries temporais proposta por Kilian (1998), que por sua vez é uma adaptação para séries temporais da correção proposta por Efron e Tibshirani (1993). Neste trabalho, a correção *bootstrap* de vício proposta por Kilian será utilizada. Outro fator a ser considerado, por vezes negligenciado pela literatura, é a presença de vício no desvio padrão dos resíduos. Este trabalho também utilizará a correção de vício *bootstrap* para corrigir o valor do desvio padrão dos resíduos.

Ainda não há propostas na literatura para construção de intervalos de previsão *bootstrap* para modelos ARFIMA, embora muitos trabalhos já existam para a construção de intervalos de confiança para os parâmetros do modelo, especialmente para o parâmetro de diferenciação. Além do *bootstrap* não paramétrico de Efron e Tibshirani (1993), outros métodos *bootstraps* foram propostos para séries ARFIMA, como o *bootstrap* local (Paparoditis e Politis, 1999) e o *bootstrap* nos resíduos da



equação de regressão (Franco e Reisen, 2004). Os dois últimos, no entanto, não replicam a série original.

Este trabalho utiliza dois métodos de construção de intervalos de previsão para séries ARFIMA, além do assintótico. O primeiro é uma adaptação do método proposto por Pascual et al (2004) para séries ARIMA. O segundo método, uma proposta do trabalho, é similar ao método proposto por Masarotto para modelos autorregressivos.

Intervalos de previsão para modelos ARFIMA serão construídos para os três tipos de estimadores, para os dois métodos de construção de intervalo *bootstrap* e com ou sem correção *bootstrap* de vício dos estimadores. Os diferentes intervalos serão comparados quanto ao nível de cobertura médio e quanto ao erro quadrado médio em relação à cobertura nominal.

Este trabalho está organizado como segue. Na Seção 2, o modelo ARFIMA é apresentado. Na Seção 3 os métodos de construção de intervalos de previsão *bootstrap* utilizados são explicados. Na Seção 4 são apresentados os resultados de simulação. Na Seção 5 estão as considerações finais.

## 2 O MODELO ARFIMA

Nesta seção o modelo ARFIMA é apresentado. A Seção 2.1 traz a definição do modelo e algumas propriedades básicas. A Seção 2.2 traz as condições de estacionariedade e invertibilidade. A Seção 2.3 apresenta a função de densidade espectral e seu estimador, o periodograma. A Seção 2.4 apresenta os diferentes tipos de estimadores do modelo ARFIMA e suas propriedades. A Seção 2.5 ensina como fazer previsão pontual e intervalar pelo método assintótico em modelos ARFIMA.

### 2.1 Introdução

A série  $\{Y_t\}$ ,  $t = 1, \dots, N$ , segue um modelo ARFIMA(p,d,q), proposto por Granger e Joyeux (1980) e Hosking (1981), se satisfaz

$$\phi(B)(1 - B)^d Y_t = \theta(B)a_t,$$

onde  $\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ ,  $\theta(B) = 1 - \theta_1 B - \dots - \theta_q B^q$ ,  $(1 - B)^d = 1 - dB - \frac{1}{2!}d(1-d)B^2 - \frac{1}{3!}d(1-d)(2-d)B^3 - \dots$ ,  $a_t$  é uma sequência de variáveis aleatórias independentes e identicamente distribuídas com média zero, variância finita,  $\sigma_a^2$ , e função de distribuição  $F_a$ ,  $d$  é o parâmetro de memória do processo e  $B$  é o operador de translação para o passado que satisfaz  $B^k Y_t = Y_{t-k}$  e  $B^k a_t = a_{t-k}$ . Os coeficientes  $\phi_1, \dots, \phi_p$  e  $\theta_1, \dots, \theta_q$  são chamados respectivamente de coeficientes autorregressivos e de médias móveis.

O modelo ARFIMA é uma generalização do modelo ARIMA (Box e Jenkins, 1976) onde o parâmetro  $d$  pode assumir valores reais, enquanto no modelo ARIMA  $d$  necessita ser um número natural. Devido ao processo ARIMA não ser mais que um caso especial do processo ARFIMA, a Seção 2 focará na metodologia referente a processos ARFIMA, podendo a metodologia referente a processos ARIMA ser obtida imediatamente. Quando  $d \in (0; 0,5)$ , o processo ARFIMA é estacionário e dito ser de memória longa. Memória longa implica que a função de autocovariância,  $\gamma$ , possui decaimento lento e não é absolutamente somável, isto é,  $\lim_{n \rightarrow \infty} \sum_{k=0}^n \gamma(k) = \infty$ . Em processos ARIMA estacionários este limite é sempre finito.

## 2.2 Estacionariedade e Invertibilidade

Um processo ARFIMA é estacionário se  $d < \frac{1}{2}$  e as raízes de  $\phi(B)$  estão fora do círculo unitário. Dizer que as raízes de  $\phi(B)$  estão fora do círculo unitário significa dizer que substituindo o operador  $B$  por uma variável real, as raízes do polinômio  $\phi(B)$  em  $\mathbb{R}^p$  estão fora de um círculo com raio 1.

Uma série é dita invertível se podemos reparametrizá-la como uma série autorregressiva pura, ou seja, se podemos escrevê-la como  $\pi(B)Y_t = a_t$ , onde  $\pi(B) = 1 - \pi_1 B - \pi_2 B^2 - \dots$ . Uma série ARFIMA será invertível se  $d > -\frac{1}{2}$  e todas as raízes de  $\theta(B)$  estiverem fora do círculo unitário.

A forma autorregressiva pura de um modelo pode ser encontrada resolvendo a equação  $\theta(B)\pi(B) = \phi(B)(1 - B)^d$ . No caso de um processo ARFIMA(1,d,1), os termos de  $\pi(B)$  são dados por:

$$\begin{aligned}\pi_j &= \varphi_j - \theta_1, \quad j = 1 \\ \pi_j &= \varphi_j + \pi_{j-1}\theta_1, \quad j = 2, 3, \dots,\end{aligned}$$

onde  $\varphi_j$  são os termos de  $\varphi(B) = \phi(B)(1 - B)^d$ .

## 2.3 Função de Densidade Espectral

A função de densidade espectral de um processo estacionário,  $f$ , também chamada simplesmente de espectro, é a transformada de Fourier de sua função de autocovariância:

$$f(\lambda) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \gamma(k)e^{-i\lambda k}, \quad -\infty < \lambda < \infty$$

A função densidade espectral é par e periódica no período  $2\pi$ . Devido a estas propriedades, é comum representar a função densidade espectral apenas entre as frequências 0 e  $\pi$ . Ademais, devido à função de autocorrelação também ser par, podemos escrevê-la sem a parte imaginária:

$$f(\lambda) = \frac{1}{2\pi} \sum_{k=0}^{\infty} \gamma(k)2\cos(\lambda k), \quad -\infty < \lambda < \infty$$

Em séries com memória longa o espectro é não-limitado na frequência zero.

O periodograma é um estimador do espectro e é dado por:

$$I(\lambda) = \frac{1}{2\pi} \sum_{k=0}^{\infty} \hat{\gamma}(k) 2\cos(\lambda k),$$

onde  $\hat{\gamma}(k)$  é o estimador da função de autocovariância e é dada por:

$$\hat{\gamma}(k) = \sum_{t=k+1}^N \frac{(Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{N},$$

onde  $N$  é o tamanho da série. O periodograma é calculado para as frequências  $\frac{2\pi}{N}, \frac{4\pi}{N}, \dots, \pi$ . O periodograma é assintoticamente não-viciado, mas é não-consistente (veja Brillinger, 1975). Sua distribuição assintótica é dada por  $f(\lambda)\chi_1^2$ , se  $\lambda = 0, \pi$ , e  $\frac{1}{2}f(\lambda)\chi_2^2$ , caso contrário, onde  $\chi_k^2$  é uma variável aleatória qui-quadrado com  $k$  graus de liberdade.

Estimadores consistentes do espectro podem ser obtidos utilizando-se de suavizadores, da seguinte forma:

$$\hat{f}(\lambda) = \frac{1}{2\pi} \sum_{k=0}^{\infty} \omega(k) \hat{\gamma}(k) 2\cos(\lambda k),$$

onde  $\omega$  é uma função peso ou núcleo. Uma das funções peso mais utilizadas na literatura consiste no núcleo de Parzen (Parzen, 1962), dado por:

$$\omega(k) = \begin{cases} 1 - 6(k/M)^2 + 6(k/M)^3, & k \leq M/2 \\ 2(1 - k/M)^3, & M/2 \leq k \leq M \\ 0, & k > M \end{cases},$$

onde  $M$  é um número tal que  $M \rightarrow \infty$  e  $\frac{M}{N} \rightarrow 0$ , quando  $N \rightarrow \infty$ , condições sob as quais a consistência do estimador do espectro é garantida.

## 2.4 Estimadores

Existem vários estimadores para  $d$ , o parâmetros de memória, na literatura. Neste trabalho serão utilizados três estimadores: o estimador de Fox-Taqqu (Fox e Taqqu, 1986), o estimador GPH (Geweke e Porter-Hudak, 1983) e o estimador SPR (Reisen, 1994)

O estimador de Fox-Taqqu estima todos os parâmetros do modelo ao mesmo tempo, enquanto os estimadores GPH e SPR estimam apenas o parâmetro  $d$ . Os demais parâmetros devem ser estimados separadamente. Para tal, diferenciamos a série  $Y_t$ , encontrando:

$$U_t = (1 - B)^{\hat{d}} Y_t, \quad t = 1, \dots, N,$$

onde  $\hat{d}$  é a estimativa de  $d$ . Estando o modelo corretamente especificado,  $U_t$  será aproximadamente uma série ARMA(p,q). Para se estimar os demais parâmetros, deve-se utilizar os métodos usuais para séries ARMA. Neste trabalho, os parâmetros da série ARMA serão estimados por máxima verossimilhança (MV), supondo normalidade dos erros. Os procedimentos de estimação dos parâmetros do modelo que se iniciam pela estimação de  $d$  via GPH e SPR separadamente, e depois estimam os demais parâmetros por máxima verossimilhança, serão chamados de MV-GPH e MV-SPR respectivamente.

Uma vez devidamente estimados os parâmetros do modelo, é possível encontrar os resíduos do mesmo. Isto é feito resolvendo:

$$\hat{a}_t = \hat{\theta}^{-1}(B) \hat{\phi}(B) (1 - B)^{\hat{d}} Y_t = \hat{\pi}(B) Y_t, \quad t = 1, \dots, N,$$

onde  $\hat{\phi}(B) = 1 - \hat{\phi}_1 B - \dots - \hat{\phi}_p B^p$ ,  $\hat{\theta}(B) = 1 - \hat{\theta}_1 B - \dots - \hat{\theta}_q B^q$  e  $\hat{\phi}_1, \dots, \hat{\phi}_p, \hat{\theta}_1, \dots, \hat{\theta}_q$  são as estimativas de  $\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q$ . Para encontrar  $\hat{\pi}(B)$ , basta resolver  $\hat{\theta}(B) \hat{\pi}(B) = \hat{\phi}(B) (1 - B)^{\hat{d}}$ . Note que se o modelo não é invertível, não é possível encontrar os resíduos. Para se estimar o desvio padrão dos resíduos, os mesmos são centralizados pela equação  $\hat{a}'_t = \hat{a}_t - \frac{\sum_{j=1}^N \hat{a}_j}{N}$ ,  $t = 1, \dots, N$ . O desvio padrão dos erros,  $\hat{\sigma}_a$ , é então estimados como:

$$\hat{\sigma}_a = \sqrt{\frac{\sum_{t=1}^T \hat{a}'_t{}^2}{N}}$$

É importante notar, também, que na diferenciação, no cálculo de resíduos, entre outras tarefas, os valores de  $Y_0, Y_{-1}, \dots$  deveriam ser conhecidos. Neste trabalho suporemos sempre que  $Y_t = 0$  para  $t < 0$  quando necessário. Existem métodos mais sofisticados de se lidar com o problema, como o *backforecasting*. Mas, exceto para casos extremos muito próximos das regiões de não-estacionariedade ou não-invertibilidade, o método mais simples aqui adotado é suficiente.

#### 2.4.1 O estimador de Fox-Taquu (FT)

O estimador de Fox-Taquu (Fox e Taquu, 1986) consiste em minimizar uma função de máxima verossimilhança aproximada. A distribuição dos erros é suposta ser

normal. A função aproximada é preferível em relação à exata pela minimização da última ser muito lenta. A função de verossimilhança aproximada é dada por:

$$\ln \frac{1}{N} \sum_{j=1}^{N/2} \frac{I(\lambda_j)}{2\pi f(\lambda_j; \beta)}$$

onde  $\beta = \{d, \phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q, \sigma_a^2\}$  é o vetor de parâmetros do modelo,  $f(\lambda_j; \beta)$  é a função densidade espectral sob  $\beta$  na frequência  $\lambda_j$ , e  $\lambda_j = \frac{2\pi j}{N}$ . O valor de  $f(\lambda_j; \beta)$  é dado por:

$$f(\lambda_j; \beta) = \frac{\sigma_a^2 |1 - \theta_1 e^{-i\lambda_j} - \dots - \theta_q e^{-qi\lambda_j}|^2}{2\pi |1 - \phi_1 e^{-i\lambda_j} - \dots - \phi_p e^{-pi\lambda_j}|^2} |1 - e^{-i\lambda_j}|^{-2d}$$

Fox e Taqqu mostram que o estimador é consistente e para processos normais com forte dependência  $\sqrt{N}(\hat{d}_{FT} - d) \xrightarrow{d} N\left(0, \frac{6}{\pi^2}\right)$ , onde  $\hat{d}_{FT}$  é o estimador FT de  $d$ .

## 2.4.2 O Estimador GPH

O estimador GPH (Geweke e Porter-Hudak, 1983) consiste em resolver uma equação de regressão aproximada. A função de densidade espectral de uma série ARFIMA  $Y_t$  pode ser escrita como:

$$f_Y(\lambda) = |1 - e^{-i\lambda}|^{-2d} f_U(\lambda), \quad (2.1)$$

onde  $f_U(\lambda)$  é a função densidade espectral da série ARMA:  $U_t = (1 - B)^{-d} Y_t$ . Após alguma álgebra, substituindo  $\lambda$  por  $\lambda_j = \frac{2\pi j}{N}$ , e adicionando o periodograma de  $Y_t$ ,  $I(\lambda_j)$ , aos dois lados de (2.1), encontramos a seguinte igualdade:

$$\ln I_Y(\lambda) = \ln f_U(0) - d \ln \left( 4 \sin^2 \left( \frac{\lambda_j}{2} \right) \right) + \ln \left( \frac{f_U(\lambda_j)}{f_U(0)} \right) + \ln \left( \frac{I_Y(\lambda_j)}{f_Y(\lambda_j)} \right) \quad (2.2)$$

Em (2.2), o termo  $\ln \left( \frac{f_U(\lambda_j)}{f_U(0)} \right)$  pode ser ignorado para as frequências próximas de zero.

Assim, podemos reescrever a equação (2.1) como um modelo de regressão linear:

$$Z_j = a - dX_j + \varepsilon_j, \quad j = 1, \dots, m \quad (2.3)$$

onde:

$$Z_j = \ln I(\lambda)$$

$$a = \ln f_U(0)$$

$$X_j = \ln \left( 4 \operatorname{sen}^2 \left( \frac{\lambda_j}{2} \right) \right)$$

$$\varepsilon_j = \ln \left( \frac{I_Y(\lambda_j)}{f_Y(\lambda_j)} \right)$$

$$m = N^\eta, \quad 0 < \eta < 1$$

Os termos  $\varepsilon_j$  em (2.3) seguem aproximadamente distribuição Gumbel (Geweke e Porter-Hudak, 1983) com média zero e variância  $\pi^2/6$ . O valor de  $d$  é estimado por mínimos quadrados:

$$\hat{d}_{GPH} = - \frac{\sum_{j=1}^m (X_j - \bar{X})(Z_j - \bar{Z})}{\sum_{j=1}^m (X_j - \bar{X})^2}$$

Geweke e Porter-Hudak mostram que  $\hat{d}_{GPH} \xrightarrow{d} N \left( d, \frac{\pi^2}{6 \sum_{j=1}^m (X_j - \bar{X})^2} \right)$ . O valor de  $\eta$  utilizado neste trabalho será 0,5 (veja Robinson, 1995)

### 2.4.3 O Estimador SPR

O estimador SPR (Reisen, 1994) é bastante similar ao estimador GPH, mas ao invés de utilizar o periodograma de  $Y_t$  na equação de regressão, utiliza um estimador suavizado pelo núcleo de Parzen do espectro. O valor de  $M$  utilizado no núcleo de Parzen neste trabalho será o menor inteiro menor que  $N^{0,9}$  (Reisen, 1994). Para séries normais, o estimador SPR é assintoticamente normal (veja Reisen, 1994 e Doukham et al, 2003) com variância igual a  $0,53928 \frac{M^2}{N \sum_{j=1}^m (X_j - \bar{X})^2}$  e média  $d$ .

### 2.5 Previsão

Supondo que os parâmetros do modelo são conhecidos, a previsão de erro quadrado médio mínimo  $h$  passos à frente no instante  $T$  é dada por:

$$Y_T(h) = \sum_{j=1}^{\infty} \hat{\varphi}_j [Y_{T+h-j}] - \sum_{j=1}^q \hat{\theta}_j [a_{T+h-j}], \quad (2.4)$$

onde  $\hat{\varphi}_j$  são os termos do polinômio  $\hat{\varphi}(B) = 1 - \hat{\varphi}_1 B - \hat{\varphi}_2 B^2 - \dots = \hat{\varphi}(B)(1 - B)^d$  e:

$$[Y_{T+h-j}] = \begin{cases} Y_T(h-j), & \text{se } h-j > 0 \\ Y_{T+h-j}, & \text{se } h-j \leq 0 \end{cases}$$

$$[a_{T+h-j}] = \begin{cases} 0, & \text{se } h-j > 0 \\ \hat{a}_{T+h-j}, & \text{se } h-j \leq 0 \end{cases}$$

A forma na equação (2.4) de realizar previsões é chamada de forma de equação de diferenças. Uma forma equivalente de se fazer previsões é pela forma autorregressiva infinita:

$$Y_T(h) = \sum_{j=1}^{\infty} \hat{\pi}_j [Y_{T+h-j}], \quad (2.5)$$

onde  $\hat{\pi}(B) = \hat{\theta}^{-1}(B)\hat{\phi}(B)(1-B)^{\hat{d}}$ . Na prática, os termos na soma em (2.5) só são calculados até  $T+h-1$ .

Supondo que os parâmetros do modelo são conhecidos, e que os erros do modelo seguem distribuição normal, então o intervalo de previsão com nível de confiança  $\alpha$  para  $Y_{T+h}$  é dado por:

$$(Y_T(h) - z_{\alpha}\hat{\sigma}_{\alpha}\hat{V}(h); Y_T(h) + z_{\alpha}\hat{\sigma}_{\alpha}\hat{V}(h)), \quad (2.6)$$

onde  $z_{\alpha}$  é o percentil  $\frac{1+\alpha}{2}$  da normal padrão,  $\hat{V}(h) = \sqrt{1 + \hat{\psi}_1^2 + \dots + \hat{\psi}_{h-1}^2}$ , e  $\hat{\psi}_1, \dots, \hat{\psi}_{h-1}$  são termos do polinômio  $\hat{\psi}(B) = 1 - \hat{\psi}_1 B - \hat{\psi}_2 B^2 - \dots = \hat{\phi}^{-1}(B)(1-B)^{-\hat{d}}\hat{\theta}(B)$ , que pode ser encontrado resolvendo  $\hat{\phi}(B)(1-B)^{\hat{d}}\hat{\psi}(B) = \hat{\theta}(B)$ . Este intervalo foi proposto por Box e Jenkins (1976) e costuma ser chamado de intervalo de previsão de Box-Jenkins.

Obviamente, se as suposições de parâmetros conhecidos e normalidade não forem satisfeitas, a cobertura real do intervalo em (2.6) será diferente da nominal.



### 3 INTERVALOS DE PREVISÃO BOOTSTRAP

Nesta seção será explicado o método *bootstrap* e como utilizá-lo para construir intervalos de previsão. A Seção 3.1 introduz o método *bootstrap*. A Seção 3.2 introduz o intervalo percentílico. A Seção 3.3 descreve o *bootstrap* não paramétrico de séries temporais. A Seção 3.4 descreve os tipos de intervalos *bootstrap* que serão utilizados. A Seção 3.5 descreve a correção *bootstrap* de vício. A Seção 3.6 descreve a correção *bootstrap* do vício do desvio padrão dos erros.

#### 3.1 O Método Bootstrap

Proposto por Efron (1979), o método *bootstrap* é um método computacionalmente intensivo de reamostragem. Para o caso geral univariado, o objetivo do método *bootstrap* é conhecer a distribuição,  $F_R$ , de uma variável aleatória,  $R = g(X_1, \dots, X_n)$ , onde  $X_1, \dots, X_n$  são amostras aleatórias de uma distribuição qualquer,  $F_X$ , desconhecida, e  $g(\cdot)$  é uma função real. Dado uma amostra,  $x_1, \dots, x_n$ , de  $F_X$ , o método *bootstrap* consiste em três passos:

1. Estimar  $F_X$ , atribuindo probabilidade  $1/n$  para cada ponto  $x_i$ ,  $i = 1, \dots, n$ . Chame de  $\hat{F}_X$  a estimativa de  $F_X$ . A  $\hat{F}_X$  estimada desta maneira é chamada de distribuição empírica de  $X$ .
2. Gerar  $B$  amostras aleatórias de tamanho  $n$ ,  $X_1^*, \dots, X_n^* = x_1^*, \dots, x_n^*$ , de  $\hat{F}_X$ . Cada amostra obtida desta maneira é chamada de amostra *bootstrap*.
3. Calcular  $g(x_1^*, \dots, x_n^*)$ , para cada amostra *bootstrap*  $x_1^*, \dots, x_n^*$ , e estimar  $F_R$  atribuindo probabilidade  $1/B$  para cada  $g(x_1^*, \dots, x_n^*)$ .

#### 3.2 O Intervalo Percentílico

Seja  $r = g(x_1^*, \dots, x_n^*)$ , com  $x_1^*, \dots, x_n^*$  e  $g(\cdot)$  definidos como na Seção 3.1. Muitas vezes, ao se realizar replicações *bootstrap*, o interesse não está em estimar  $F_R$  diretamente, mas em encontrar determinados percentis de  $R$ . É o caso, por exemplo, da construção de intervalos de previsão ou de confiança.

Suponha que após  $B$  replicações *bootstrap*,  $B$  valores de  $r$  tenham sido obtidos, e seja  $r^{(1)}, \dots, r^{(B)}$  estes valores ordenados. Então o intervalo percentílico (Efron e Tibshirani, 1993) baseado em  $R$  com cobertura  $\alpha$  é dado por:

$$(r^{\{(1-\alpha)/2\}}, r^{\{(1+\alpha)/2\}}) \quad (3.1)$$

Onde  $r^{\{(1-\alpha)/2\}}$  e  $r^{\{(1+\alpha)/2\}}$  em (3.1) são os percentis  $100 \times (1 - \alpha)/2$  e  $100 \times (1 + \alpha)/2$  de  $r^{(1)}, \dots, r^{(B)}$ . A maneira exata de se calcular os percentis em (3.1) pode variar.

Os valores  $r^{(1)}, \dots, r^{(B)}$  não são exatamente uma amostra aleatória de  $F_R$ , uma vez que tipicamente não se conhece  $F_X$ , mas  $\hat{F}_X$ . Seja  $R^*$  a variável aleatória análoga a  $R$  advinda de  $\hat{F}_X$ , ao invés de  $F_X$ . Ou seja,  $R^* = g(X_1^*, \dots, X_n^*)$ , onde  $X_1^*, \dots, X_n^*$  são independentes entre si e com distribuição  $\hat{F}_X$ . Seja  $F_{R^*}$  a função de distribuição de  $R^*$ . Então  $r^{(1)}, \dots, r^{(B)}$  é uma amostra aleatória de  $R^*$ . O que se deseja ao se construir um intervalo percentílico é encontrar os valores  $r_{INF}$  e  $r_{SUP}$  tais que  $F_{R^*}(r_{INF}) = (1 - \alpha)/2$  e  $F_{R^*}(r_{SUP}) = (1 + \alpha)/2$ .

É trivial ver que  $F_{R^*}(R^*)$  segue distribuição uniforme com parâmetros 0 e 1. Então, dada uma amostra aleatória de tamanho  $B$  de  $R^*$ ,  $F_{R^*}(\cdot)$  aplicada ao  $k$ -ésimo posto desta amostra aleatória terá distribuição beta com esperança  $\frac{k}{B+1}$ .

Portanto, em replicações *bootstrap*, se o valor de  $B$  for escolhido de modo que  $(B + 1) \frac{(1-\alpha)}{2} = k_1$  e  $(B + 1) \frac{(1+\alpha)}{2} = k_2$ , onde  $k_1$  e  $k_2$  são inteiros, então:

$$\begin{aligned} & E\left[P\left(r^{((B+1)(1-\alpha)/2)} \leq R^* \leq r^{((B+1)(1+\alpha)/2)}\right)\right] = \\ & = E\left[F_{R^*}\left(r^{((B+1)(1+\alpha)/2)}\right) - F_{R^*}\left(r^{((B+1)(1-\alpha)/2)}\right)\right] = \alpha. \end{aligned}$$

### 3.3 Bootstrap Não-paramétrico em Séries Temporais

A aplicação imediata do método *bootstrap* conforme descrito na seção 3.1 para as observações de uma série temporal,  $Y_1, \dots, Y_T$  não é adequada, uma vez que estas são correlacionadas entre si. O *bootstrap* não paramétrico dos resíduos de uma série temporal (Efron e Tibshirani, 1993) supera este obstáculo ao realizar a replicação *bootstrap* dos resíduos centralizados do modelo,  $\hat{a}'_1, \dots, \hat{a}'_T$ , que são independentes se o modelo for corretamente especificado.

Considere  $\hat{F}_a$  a distribuição empírica dos resíduos padronizados, e  $\hat{\phi}_1, \dots, \hat{\phi}_p$ ,  $\hat{\theta}_1, \dots, \hat{\theta}_q$  e  $\hat{d}$  os parâmetros estimados da série original que será replicada. Considere

também  $\hat{\varphi}(B) = 1 - \hat{\varphi}_1 B - \hat{\varphi}_2 B^2 - \dots = \hat{\varphi}(B)(1 - B)^d$ . O *bootstrap* não paramétrico proposto por Efron e Tibshirani para modelos autorregressivos de ordem  $p$ , fixa os primeiros  $p$  valores da série original. No entanto, sendo uma série ARFIMA na prática uma série autorregressiva infinita, esta abordagem não pode ser imediatamente aplicada. Neste trabalho geramos uma série ARFIMA *bootstrap* sem fixar valores iniciais. Se queremos uma replicação de tamanho  $T'$ , geramos  $T' + K$  valores,  $Z_1^*, \dots, Z_{T'+K}^*$ , recursivamente da seguinte forma:

$$\begin{aligned} Z_1^* &= a_1^*, \\ Z_2^* &= \hat{\varphi}_1 Z_1^* - \hat{\theta}_1 a_1^* + a_2^*, \\ &\vdots \\ Z_{T'+K}^* &= \sum_{j=1}^{T'+K-1} \hat{\varphi}_j Z_{T'+K-j}^* - \sum_{j=1}^q \hat{\theta}_j \varepsilon_{T'+K-j}^* + a_{T'+K}^*, \end{aligned}$$

onde  $a_t^*$ ,  $t = 1, \dots, T' + K$ , são amostras aleatórias de  $\hat{F}_a$ . A série *bootstrap*,  $Y_t^*$ , é então definida como  $Y_t^* = Z_{t+K}^*$ ,  $t = 1, \dots, T'$ .

É importante notar que uma replicação *bootstrap* realizada desta maneira só faz sentido se os parâmetros estimados da série original estão nas regiões de estacionariedade e invertibilidade. Se os parâmetros são não-estacionários, então o processo não possui distribuição estacionária, e portanto não existe replicação (a não ser que se fixe alguns valores iniciais, como no caso autorregressivo mencionado anteriormente). Se os parâmetros são não-invertíveis, então não é possível estimar os resíduos. É importante observar que nenhuma suposição é feita sob a distribuição de  $F_a$  a priori.

### 3.4 Tipos de Intervalos de Previsão Bootstrap

O intervalo de Box-Jenkins (BJ) pode ter uma cobertura diferente da nominal, pelas razões já discutidas na Seção 2.5. Os intervalos apresentados nesta seção procuram lidar com este problema ao levar em consideração a variabilidade na estimação dos parâmetros do modelo e ao não fazerem suposições sobre a distribuição dos erros do mesmo.

Todos os intervalos *bootstrap* descritos aqui possuem duas etapas iniciais em comum: estimação dos parâmetros e resíduos da série e construção de séries *bootstrap*

conforme descrito em 3.3. Todos os intervalos são baseados no intervalo percentílico de Efron e Tibshirani (1993).

### 3.4.1 Intervalo PRR

O intervalo PRR foi proposto por Pascual, Romo e Ruiz para modelos ARIMA (Pascual et al, 2004), mas neste trabalho foram adaptados para modelos ARFIMA. Após se estimarem os parâmetros da série original, a construção do intervalo PRR com cobertura  $\alpha$  para  $Y_{T+h}$ , no instante  $T$ , obedece às seguintes etapas:

1. São geradas  $B$  séries bootstrap de tamanho  $T$ , o tamanho da série original.
2. Para cada série *bootstrap*,  $Y_1^*, \dots, Y_T^*$ , são estimados os seus parâmetros  $\phi_1^*, \dots, \phi_p^*$ ,  $\theta_1^*, \dots, \theta_q^*$  e  $d^*$ .
3. Para cada série *bootstrap*, é calculado um valor futuro *bootstrap*  $h$  passos à frente recursivamente da seguinte maneira:

$$Y_{T+h}^* = \sum_{j=1}^{\infty} \phi_j^* [Y_{T+h-j}^*] - \sum_{j=1}^q \theta_j^* [a_{T+h-j}^*] + a_{T+h}^*$$

onde:

$\phi_j^*$  são os termos do polinômio  $\varphi^*(B) = \phi^*(B)(1 - B)^{d^*}$

$$[Y_{T+h-j}^*] = \begin{cases} Y_{T+h-j}^* & \text{se } h - j > 0 \\ Y_{T+h-j} & \text{se } h - j \leq 0 \end{cases}$$

$$[a_{T+h-j}^*] = \begin{cases} a_{T+h-j}^* & \text{se } h - j > 0 \\ \hat{a}_{T+h-j} & \text{se } h - j \leq 0 \end{cases}$$

$a_t^*$  são amostras aleatórias de  $\hat{F}_a$ .

4. Os  $B$  valores bootstrap futuros são ordenados. Seja  $Y_{T+h}^{*(j)}$  o  $j$ -ésimo valor bootstrap futuro em ordem crescente. O intervalo é dado por:

$$\left[ Y_{T+h}^{*((B+1)(1-\alpha)/2)}; Y_{T+h}^{*((B+1)(1+\alpha)/2)} \right]$$

Portanto, no cálculo de cada valor *bootstrap* futuro, os valores conhecidos da série original  $Y_1, \dots, Y_T$ , e os resíduos da mesma  $\hat{a}_1, \dots, \hat{a}_T$ , são mantidos fixos, enquanto os parâmetros e os erros nos instantes  $T + 1$  até  $T + h$  variam, procurando capturar os efeitos dos erros de estimação e dos erros do modelo.

### 3.4.2 Intervalo Utilizando os Erros de Previsão Bootstrap (EPB)

O intervalo EPB é similar ao intervalo proposto por Masarotto em 1990 para modelos autorregressivos puros, embora não similar o bastante para ser chamado de uma adaptação para séries ARFIMA. Após se estimarem os parâmetros da série original, a construção do intervalo EPB com cobertura  $\alpha$  para  $Y_{T+h}$ , no instante  $T$ , obedece às seguintes etapas:

1. São geradas  $B$  séries bootstrap de tamanho  $T + h$ , o tamanho da série original mais o número de passos à frente da previsão.

2. Para cada série *bootstrap*, os  $T$  primeiros valores,  $Y_1^*, \dots, Y_T^*$ , são utilizados para estimar os parâmetros  $\phi_1^*, \dots, \phi_p^*$ ,  $\theta_1^*, \dots, \theta_q^*$  e  $d^*$ .

3. Os resíduos da série *bootstrap*,  $a_1^*, \dots, a_T^*$ , são calculados exatamente como na Seção 2.4, substituindo  $Y_1, \dots, Y_T$  por  $Y_1^*, \dots, Y_T^*$  e  $\hat{\phi}_1, \dots, \hat{\phi}_p, \hat{\theta}_1, \dots, \hat{\theta}_q$  e  $\hat{d}$  por  $\phi_1^*, \dots, \phi_p^*, \theta_1^*, \dots, \theta_q^*$  e  $d^*$ .

4. Para cada série *bootstrap*, é realizada uma previsão *bootstrap*  $h$  passos à frente,  $Y_T^*(h)$ , que é calculada exatamente como na Seção 2.5. Calcula-se  $E_T^*(h) = Y_T^*(h) - Y_{T+h}^*$ . A idéia do intervalo EPB é construir uma distribuição empírica dos erros de previsão, utilizando-se do bootstrap. Chame  $E_T^*(h)$  de erro de previsão bootstrap.

5. Os  $B$  valores de erros de previsão bootstrap futuros são ordenados. Seja  $E_T^{*(j)}(h)$  o  $j$ -ésimo erro de previsão bootstrap em ordem crescente. O intervalo é dado por:

$$\left[ Y_T(h) - E_T^{*((B+1)(1+\alpha)/2)}(h); \quad Y_T(h) - E_T^{*((B+1)(1-\alpha)/2)}(h) \right]$$

O objetivo do intervalo EPB é construir a distribuição empírica dos erros de previsão dado o tamanho da série e seus parâmetros estimados, e utilizar o conhecimento sobre esta distribuição para construir o intervalo de previsão.

### 3.5 Correção Bootstrap de Vício

O vício nos estimadores dos parâmetros pode afetar a cobertura real de intervalos de previsão (veja Kim, 2001). Efron e Tibshirani (1993) propõem o uso do bootstrap para a correção de vício. Suponha um vetor de parâmetros,  $\beta = (\beta_1, \dots, \beta_k)$

que se deseje estimar, e seja  $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_k)$  o estimador de  $\beta$ . O vício de  $\hat{\beta}$  é definido como  $\Psi = E(\hat{\beta}) - \beta$ .

A correção bootstrap de vício consiste em gerar  $B$  replicações bootstrap dos dados sob  $\hat{\beta}$  e em cada replicação estimar  $\beta^*$ , usando o mesmo estimador  $\hat{\beta}$ . O passo seguinte é calcular a média aritmética dos  $\beta^*$ ,  $\bar{\beta}^*$ . O vício de  $\hat{\beta}$ ,  $\Psi$ , é então estimado como sendo  $\hat{\Psi} = \bar{\beta}^* - \hat{\beta}$ , uma aproximação para o vício real de  $\hat{\beta}$ . O estimador corrigido de vício,  $\tilde{\beta}$ , é então calculado como  $\tilde{\beta} = \hat{\beta} - \hat{\Psi}$ .

A correção de vício *bootstrap* pode levar  $\tilde{\beta}$  para regiões de não-estacionariedade ou não-invertibilidade. Para resolver este problema, Kilian (1998) sugere encolher o valor de  $\hat{\Psi}$  em direção a zero até que  $\tilde{\beta}$  volte à região de estacionariedade e invertibilidade. O algoritmo iterativo descrito por Kilian para ajustar  $\tilde{\beta}$  segue os seguintes passos:

1. Estabeleça  $\hat{\Psi}_1 = \hat{\Psi}$  e  $\tilde{\beta}_1 = \tilde{\beta}$ . Estabeleça também  $\delta_1 = 1$ .
2. Verifique se  $\tilde{\beta}_i$  está na região de estacionariedade. Se sim, faça  $\tilde{\beta} = \tilde{\beta}_i$  e encerre a iteração. Se não, faça  $\delta_{i+1} = \delta_i - 0,01$ ,  $\hat{\Psi}_{i+1} = \hat{\Psi}_1 \delta_{i+1}$  e  $\tilde{\beta}_{i+1} = \hat{\beta} - \hat{\Psi}_{i+1}$ .
3. Repita o passo 2 quantas vezes for necessário.

Killian demonstra a validade assintótica da correção *bootstrap* de vício com ou sem a correção de estacionariedade ou invertibilidade proposta por ele, desde que o estimador de  $\beta$  seja consistente.

O artigo de Kilian (1998) lida com modelos VAR, e o artigo de Kim (2001) com modelos AR. Séries ARFIMA possuem a peculiaridade de possuir três grupos de parâmetros, os autorregressivos, os de médias móveis e o de diferenciação. Cada grupo não interage com os demais para determinar se a série será estacionária ou invertível. Por exemplo, se temos  $d < 1/2$ , o valor de  $d$  é irrelevante para determinar os valores de  $\phi_1, \dots, \phi_p$  que geram uma série estacionária. Em contraste, em uma série estritamente autorregressiva, os valores de  $\phi_1, \dots, \phi_{p-1}$  são relevantes para saber o valor de  $\phi_p$  que levará a série à não estacionariedade. Devido a isto, cada um dos três grupos de parâmetros, conforme definidos anteriormente, são submetidos ao algoritmo de Kilian separadamente.

A correção *bootstrap* de vício será utilizada na construção dos intervalos de previsão. Após serem encontradas as estimativas dos parâmetros corrigidas de vício, os métodos descritos na Seção 3.4 são utilizados substituindo-se  $\hat{\phi}_1, \dots, \hat{\phi}_p, \hat{\theta}_1, \dots, \hat{\theta}_q$  e  $\hat{d}$

pelas estimativas corrigidas,  $\tilde{\phi}_1, \dots, \tilde{\phi}_p, \tilde{\theta}_1, \dots, \tilde{\theta}_q$  e  $\tilde{d}$ , e  $\hat{F}_a$  por  $\tilde{F}_a$ , que é a distribuição empírica dos resíduos calculados utilizando-se de  $\tilde{\phi}_1, \dots, \tilde{\phi}_p, \tilde{\theta}_1, \dots, \tilde{\theta}_q$  e  $\tilde{d}$ :

$$\tilde{a}_t = \tilde{\theta}^{-1}(B)\tilde{\phi}(B)(1-B)^{\tilde{d}}Y_t = \tilde{\pi}(B)Y_t, \quad t = 1, \dots, N,$$

e fazendo-se a devida centralização.

Ao se estimar os parâmetros das séries *bootstrap*, idealmente o método descrito nesta Seção para corrigir vício deveria se repetir, uma vez que os estimadores das séries *bootstrap* e os da série original devem ser os mesmos. No entanto, realizar replicações *bootstrap* dentro de cada replicação *bootstrap* teria um custo computacional proibitivo. Em vez disto, nas replicações *bootstrap*, a correção de vício é feita simplesmente subtraindo as estimativas encontradas por  $\hat{\Psi}$ , o vício estimado na série original. O algoritmo proposto por Kilian é utilizado quando necessário também nas séries *bootstrap*, isto é, quando o as estimativas dos parâmetros ficam fora das regiões de estacionariedade e invertibilidade.

### 3.6 Correção do Vício dos Resíduos

Na literatura, até o momento, pouca atenção vem sido prestada à possível presença de vício no desvio padrão dos resíduos. Isto acontece mesmo nos artigos que procuram corrigir o vício dos demais parâmetros. A possibilidade de presença de vício no desvio padrão dos resíduos não deveria ser ignorada, já que esta informação é utilizada seja no intervalo de Box-Jenkins, seja nos intervalos *bootstrap*. No intervalo de Box-Jenkins o desvio padrão dos resíduos é usado diretamente na construção do intervalo. Nos intervalos *bootstrap*, o desvio padrão dos resíduos é usado indiretamente através de  $\hat{F}_a$ , que idealmente deveria ser uma boa estimativa de  $F_a$ . Assim como se deseja que  $\hat{\phi}_1, \dots, \hat{\phi}_p, \hat{\theta}_1, \dots, \hat{\theta}_q$  e  $\hat{d}$  sejam boas estimativas de  $\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q$  e  $d$ . Neste trabalho se procurará corrigir o vício nas estimativas do desvio padrão dos resíduos, conforme descritos a seguir.

A correção do vício do desvio padrão dos resíduos é feita de forma idêntica aos demais parâmetros. Primeiro encontra-se  $\bar{\sigma}_a^*$ , a média dos desvios padrões dos resíduos nas replicações *bootstrap*. Calcula-se  $\hat{\Psi} = \bar{\sigma}_a^* - \hat{\sigma}_a$ , e  $\sigma_a^{CV} = \hat{\sigma}_a - \hat{\Psi}$ . Não se deve confundir o que aqui é chamado de  $\sigma_a^{CV}$  com  $\tilde{\sigma}_a$ :

$$\tilde{\sigma}_a = \sqrt{\frac{\sum_{t=1}^T \tilde{a}_t^2}{T}},$$

onde  $\tilde{a}'_t$ ,  $t = 1, \dots, T$ , são os  $\tilde{a}_t$  devidamente centralizados, a Seção 3.5 define  $\tilde{a}_t$ .

O novo valor de desvio padrão,  $\sigma_a^{CV}$ , pode ser imediatamente utilizado na construção do intervalo de Box-Jenkins. No caso dos intervalos bootstrap, o interesse em corrigir vícios na estimativa de desvio padrão é para se obter uma melhor estimativa de  $F_a$ . Isto é feito reescalando os resíduos  $\tilde{a}'_1, \dots, \tilde{a}'_T$  através da seguinte equação:

$$a_t^{CV} = \tilde{a}'_t \left( \frac{\sigma_a^{CV}}{\tilde{\sigma}_a} \right), \quad t = 1, \dots, T$$

É trivial ver que os resíduos assim gerados,  $a_1^{CV}, \dots, a_T^{CV}$ , terão desvio padrão  $\sigma_a^{CV}$  e média zero. Assim, utiliza-se  $F_a^{CV}$ , a distribuição empírica de  $a_1^{CV}, \dots, a_T^{CV}$ , para gerar os intervalos *bootstrap* da Seção 3.4.



## 4 RESULTADOS DE SIMULAÇÕES

Para testar as metodologias propostas, foram realizadas simulações Monte Carlo. Cinco modelos e três tipos de erros foram considerados. Nos modelos 1, 4 e 5 os erros seguem distribuição normal com desvio padrão 1. No modelo 2, os erros seguem distribuição exponencial centralizada com desvio padrão igual a 1. No modelo 3, os erros seguem uma distribuição bimodal. Neste caso, com probabilidade 0,9 os erros provêm de uma distribuição normal com média  $-1$  e desvio padrão 1 e com probabilidade 0,1 de uma distribuição normal com média 9 e desvio padrão 1. Os erros bimodais são divididos por  $\sqrt{10}$  para possuírem desvio padrão igual a 1. A Figura 1 mostra o a função densidade de probabilidade dos três tipos de erros utilizados. Os três tipos de erros utilizados são os mesmos utilizados em Pascual et al. (2004).

MODELO 1: ARFIMA(1,  $d$ , 1) com  $\phi = 0,3$ ,  $d = 0,3$ ,  $\theta = -0,3$  e  $F_\alpha$  com distribuição normal.

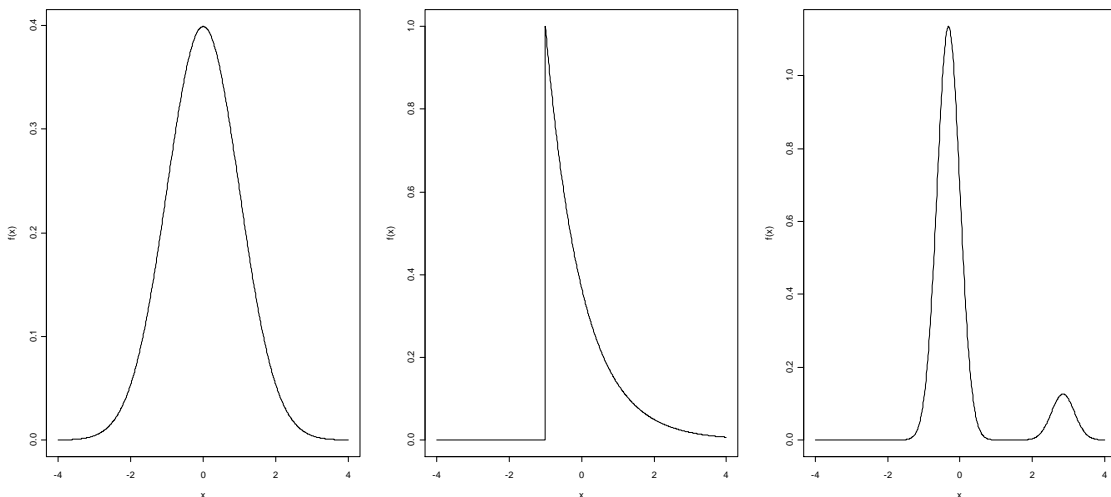
MODELO 2: ARFIMA(1,  $d$ , 1) com  $\phi = 0,3$ ,  $d = 0,3$ ,  $\theta = -0,3$  e  $F_\alpha$  com distribuição exponencial centralizada.

MODELO 3: ARFIMA(1,  $d$ , 1) com  $\phi = 0,3$ ,  $d = 0,3$ ,  $\theta = -0,3$  e  $F_\alpha$  com distribuição bimodal.

MODELO 4: ARFIMA(1,  $d$ , 0) com  $\phi = 0,3$ ,  $d = 0,3$  e  $F_\alpha$  com distribuição normal.

MODELO 5: ARMA(1,1) com  $\phi = 0,3$ ,  $\theta = -0,3$  e  $F_\alpha$  com distribuição normal.

**Figura 1: Função de densidade de probabilidade dos modelos 3, 4 e 5 (esquerda), 1 (centro) e 2 (direita)**



Também houve simulações com séries ARMA devido a algumas das metodologias propostas também serem inéditas também para este tipo de séries, como o intervalo EPB.

Em todas as simulações o número de replicações Monte Carlo foi 1000, o número de replicações *bootstrap* foi 999, tanto para a correção de vício quanto para a construção dos intervalos, a cobertura nominal dos intervalos de previsão foi 0,95 e o número de passos à frente da previsão foram 1 e 10. O tamanho da série foi 100 nos 4 primeiros modelos e 50 no último modelo, o único modelo ARMA. O tamanho das séries foi escolhido para não ser grande o suficiente para não se justificar o abandono da modelagem tradicional. O programa com o qual as simulações foram geradas foi escrito em linguagem Fortran 77. As simulações com os atributos descritos acima duraram 3 horas e 20 minutos para os Modelos 1 a 3 (em média), 2 horas e 34 minutos para o Modelo 4 e 36 minutos para o Modelo 5. No caso mais lento, o dos Modelos 1 a 3, cada replicação Monte Carlo com 999 replicações *bootstrap* durou, em média, 12 segundos.

O número de replicações *bootstrap* foi escolhido por  $(999 + 1) \frac{(1-0,95)}{2}$  e  $(999 + 1) \frac{(1+0,95)}{2}$  serem ambos inteiros, atendendo ao exposto na Seção 3.2.

O algoritmo para gerar as séries nas replicações Monte Carlo é similar ao algoritmo descrito na Seção 3.3 para gerar séries *bootstrap*, substituindo-se  $\hat{F}_a$  por  $F_a$  e os parâmetros estimados pelos parâmetros reais. O *burn-in*  $K$ , ou seja, o número de observações da série geradas para descarte (veja Seção 3.3), foi 500 em todas as simulações. Para se gerar as séries *bootstrap* também se utilizou  $K = 500$ . Na seção 3.3 é dito que para que uma replicação *bootstrap* seja possível é necessário que os parâmetros estimados sejam estacionários e invertíveis. Esta condição foi garantida nas estimativas de máxima verossimilhança ou de Fox-Taquq ao se limitar a procura do ponto de valor máximo às regiões de estacionariedade e invertibilidade. No caso das estimativas de GPH e SPR, isto foi garantido levando as estimativas encontradas para o ponto da região de estacionariedade mais próximo. Este procedimento também foi realizado nas estimativas *bootstrap*. Ainda que estimar parâmetros não-estacionários em uma série *bootstrap* não traz problemas adicionais, já que não se fará mais replicações, isto é feito para que os estimadores das séries originais e *bootstrap* sejam os mesmos.

Ao todo, para cada replicação Monte Carlo gerada foram utilizados 3 procedimentos de estimação (MV-GPH, MV-SPR e FT) e 3 tipos de intervalo (BJ, PRR, e EPB). Há também 3 abordagens possíveis quanto ao vício: nenhuma correção de

vício (SCV), correção de vício dos parâmetros mas não do desvio padrão dos erros (CCV) e correção de vício dos parâmetros e do desvio padrão dos erros (CCDP). No caso de séries ARMA, apenas o estimador de verossimilhança é usado (MV). O efeito da correção de vício nos intervalos será mostrado no Modelo 4, e a correção de vício no desvio padrão dos resíduos no Modelo 5.

Seja  $MC$  o número de replicações Monte Carlo,  $\alpha$  a cobertura nominal do intervalo de previsão,  $L_{INF}^{(i)}$  e  $L_{SUP}^{(i)}$  respectivamente o limite inferior e superior do intervalo na  $i$ -ésima replicação,  $Y_{T+h}^{(i)}$  o valor da série no instante  $T+h$  na  $i$ -ésima replicação e  $\chi_T^{(i)} = Y_T^{(i)}, Y_{T-1}^{(i)}, \dots$ . Para cada simulação e número de passos à frente, os intervalos construídos foram medidos quanto aos seguintes atributos:

1. Cobertura média:

$$C = \frac{\sum_{i=1}^{MC} P\left(L_{INF}^{(i)} < Y_{T+h}^{(i)} < L_{SUP}^{(i)} \mid \chi_T^{(i)}\right)}{MC}$$

2. Não-cobertura inferior média:

$$C_{INF} = \frac{\sum_{i=1}^{MC} P\left(Y_{T+h}^{(i)} \leq L_{INF}^{(i)} \mid \chi_T^{(i)}\right)}{MC}$$

3. Não-cobertura superior média:

$$C_{SUP} = \frac{\sum_{i=1}^{MC} P\left(Y_{T+h}^{(i)} \geq L_{SUP}^{(i)} \mid \chi_T^{(i)}\right)}{MC}$$

4. Distância das não-coberturas nominais:

$$S = |C_{INF} - (1 - \alpha)/2| + |C_{SUP} - (1 - \alpha)/2|$$

5. Tamanho médio do intervalo:

$$\text{Tamanho} = \frac{\sum_{i=1}^{MC} \left(L_{SUP}^{(i)} - L_{INF}^{(i)}\right)}{MC}$$

As probabilidades acima foram calculadas analiticamente em cada simulação quando os erros foram normais. Quando os erros foram não-normais, resultando em funções de distribuição mais complicadas, as probabilidades acima foram estimadas gerando-se 1000 valores de  $Y_{T+h}^{(i)}$  dado  $\chi_T^{(i)}$ .

Todos os intervalos são construídos com o objetivo de possuírem  $P\left(Y_{T+h}^{(i)} \leq L_{INF}^{(i)} \mid \chi_T^{(i)}\right) = 0,025$  e  $P\left(Y_{T+h}^{(i)} \geq L_{SUP}^{(i)} \mid \chi_T^{(i)}\right) = 0,025$ . Quando a distribuição dos erros não foi simétrica, o principal critério para decidir os melhores intervalos foi  $S$ , ou seja, a da soma das diferenças absolutas entre as não-coberturas reais e as nominais.

No caso de erros assimétricos essa quantia é preferível a avaliar simplesmente a distância da cobertura média em relação à nominal. Isto porque um intervalo pode ter cobertura melhor (no sentido de mais próxima da nominal) do que outro por errar nas estimativas tanto de  $L_{INF}$  quanto na de  $L_{SUP}$ . Por exemplo, um primeiro intervalo pode ter  $C = 0,94$ ,  $C_{INF} = 0,03$  e  $C_{SUP} = 0,03$ , enquanto um segundo intervalo pode ter  $C = 0,95$ ,  $C_{INF} = 0$  e  $C_{SUP} = 0,05$ . A cobertura mais próxima da nominal do segundo intervalo não se deve a algum mérito em relação ao primeiro intervalo, mas a dois erros, um compensando o outro. Em contraste, no primeiro intervalo,  $S = 0,01$ , e no segundo  $S = 0,05$ . Quando os erros são simétricos, no caso das simulações aqui realizadas somente quando estes são normais, o critério utilizado foi simplesmente a cobertura, já que nestes casos  $C_{INF} \cong C_{SUP}$ .

Além da análise dos intervalos de previsão, foi feita paralelamente uma simulação Monte Carlo para encontrar as propriedades dos estimadores utilizados (com e sem correção de vício) nos modelos apresentados nesta seção. Seja  $\beta$  um parâmetro qualquer e  $\hat{\beta}^{(i)}$  a estimativa de  $\beta$  na  $i$ -ésima replicação Monte Carlo. O Erro Quadrático Médio (EQM) é dado por:

$$EQM = \frac{\sum_{i=1}^{MC} (\hat{\beta}^{(i)} - \beta)^2}{MC}$$

A análise dos estimadores e dos erros estimados encontra-se na Seção 4.1, enquanto a Seção 4.2 analisa os intervalos construídos.

#### 4.1 Estudo Monte Carlo dos Estimadores

A Tabela 1 mostra a média e o EQM das estimativas dos parâmetros para os diferentes tipos de estimadores utilizados.

O valor dos parâmetros nos Modelos de 1 a 3 são todos os mesmos, e por isto as médias e EQMs de cada modelo são muito parecidas. Todos os estimadores do parâmetro  $d$  apresentam vício. O estimador GPH é o que apresenta o menor vício, mas o estimador SPR é o que apresenta o menor EQM. Os estimadores GPH e FT subestimam o valor de  $d$ , enquanto o estimador SPR superestima. A estimativa de  $\phi$  também possui um vício muito grande quando o estimador é MV-SPR, em comparação com os outros métodos de estimação, mas ainda assim o EQM é menor. O EQM das estimativas de  $\theta$  é parecida para os três métodos de estimação. O vício do estimador de

$\theta$  quando o estimador é MV-GPH, no entanto, é um pouco menor. O vício do desvio padrão dos erros foi pequeno para os três estimadores de  $d$  utilizados.

No Modelo 4 os valores médios das estimativas de  $d$  são parecidos com os encontrados nos modelos 1 e 2. No entanto, o EQM do estimador FT de  $d$  tem uma leve caída, a ponto do estimador GPH de  $d$  passar a ter o maior EQM. O estimador MV-GPH de  $\phi$  é o menos viciado, mas também o com maior EQM. E o estimador MV-SPR de  $\phi$  é o mais viciado mas com menor EQM. Todos os vícios de  $d$  e de  $\phi$  são não desprezíveis. O vício do desvio padrão dos erros foi pequeno para os três estimadores utilizados.

No Modelo 5 todos os parâmetros foram estimados somente por máxima verossimilhança. As estimativas de  $\theta$  e  $\phi$  foram pouco viciadas, mas o desvio padrão dos erros foi subestimado.

Como é possível ver pela Tabela 1, a estimativa do desvio padrão nos Modelos 1 a 4 tem vício insignificante. Por isto, somente no Modelo 5 o vício do desvio padrão foi corrigido via *bootstrap*. Apesar da existência de vício nos demais parâmetros em todos os modelos, as simulações mostram que a correção *bootstrap* de vício não foi capaz de melhorar os intervalos. Os efeitos da correção *bootstrap* de vício serão mostrados no Modelo 4.

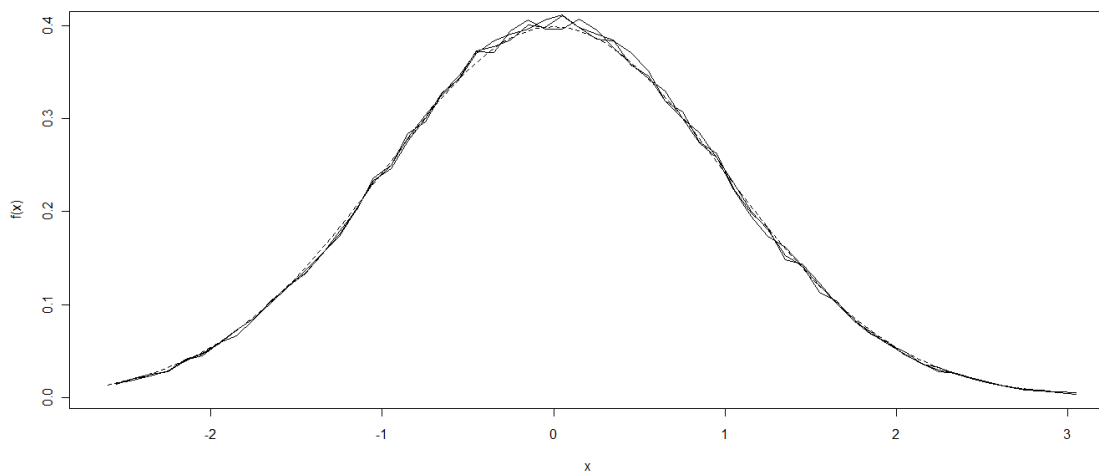
**Tabela 1: Média e EQM dos estimadores para os Modelos 1 a 5**

Estimador de d	Parâmetro	Estatística	Modelo				
			1	2	3	4	5
FT	d	Média	0,1978	0,1885	0,1954	0,2075	-
		EQM	0,0825	0,0879	0,0859	0,0682	-
	$\phi$	Média	0,3092	0,3324	0,3190	0,3585	-
		EQM	0,1246	0,1180	0,1170	0,0608	-
	$\theta$	Média	-0,3682	-0,3552	-0,3595	-	-
		EQM	0,0473	0,0460	0,0406	-	-
	$\sigma_a$	Média	0,9889	0,9869	0,9875	0,9830	-
		EQM	0,0099	0,0238	0,0149	0,0052	-
GPH	d	Média	0,2688	0,2670	0,2681	0,2514	-
		EQM	0,0733	0,0694	0,0739	0,0773	-
	$\phi$	Média	0,2956	0,3158	0,3045	0,3465	-
		EQM	0,1236	0,1146	0,1186	0,0795	-
	$\theta$	Média	-0,3380	-0,3201	-0,3301	-	-
		EQM	0,0432	0,0417	0,0383	-	-
	$\sigma_a$	Média	0,9911	0,9901	0,9946	0,9918	-
		EQM	0,0052	0,0188	0,0148	0,0051	-
SPR	d	Média	0,4004	0,4061	0,4032	0,3910	-
		EQM	0,0429	0,0395	0,0384	0,0406	-
	$\phi$	Média	0,1329	0,1367	0,1376	0,1987	-
		EQM	0,1065	0,0967	0,0992	0,0443	-
	$\theta$	Média	-0,3663	-0,3577	-0,3578	-	-
		EQM	0,0487	0,0466	0,0462	-	-
	$\sigma_a$	Média	0,9924	0,9910	0,9952	0,9929	-
		EQM	0,0052	0,0188	0,0147	0,0051	-
-	$\phi$	Média		-	-	-	0,2949
		EQM		-	-	-	0,0708
	$\theta$	Média		-	-	-	-0,3010
		EQM		-	-	-	0,0815
	$\sigma_a$	Média		-	-	-	0,9784
		EQM		-	-	-	0,0110

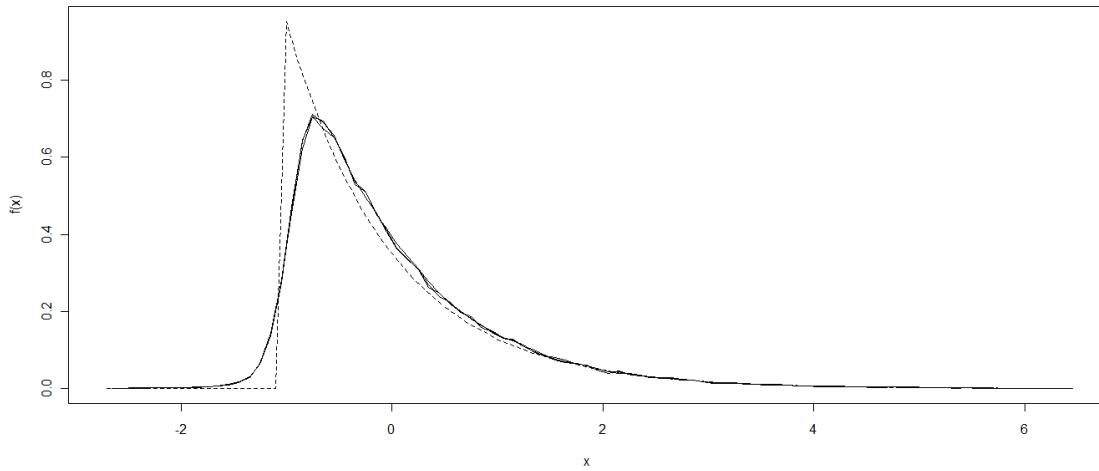
Uma das suposições mais importantes no *bootstrap* de séries temporais é a de que  $\hat{F}_a$  é uma boa aproximação de  $F_a$ , não só no que diz respeito ao desvio padrão, como também em forma. As Figuras 2, 3 e 4 mostram as funções de densidade dos erros dos modelos 1, 2 e 3 (linha tracejada) e as densidades estimados dos resíduos (linha contínua). As densidades dos resíduos também foram estimadas após 1000 replicações Monte Carlo. As densidades dos resíduos para cada estimador e em cada modelo são

muito parecidas entre si, fazendo com que suas densidades sejam indistinguíveis entre si nas figuras. A Figura 2 mostra que a densidade dos resíduos do modelo 1 é bem parecida com a densidade dos erros do modelo 1. Resultados quase idênticos aos encontrados na Figura 2 acontecem com os modelos 4 e 5, e por isto eles são omitidos aqui. Os Modelos 4 e 5 possuem erros com distribuição normal como o Modelo 1. A Figura 3 mostra que a densidade dos resíduos do modelo 2 possui uma cauda à esquerda que não existe na distribuição exponencial centralizada. A Figura 4 mostra que a densidade dos resíduos do modelo 3 é parecida com a densidade dos erros, mas os dois picos da densidade são mais suaves na densidade dos resíduos e o segundo pico está um levemente deslocado para a esquerda.

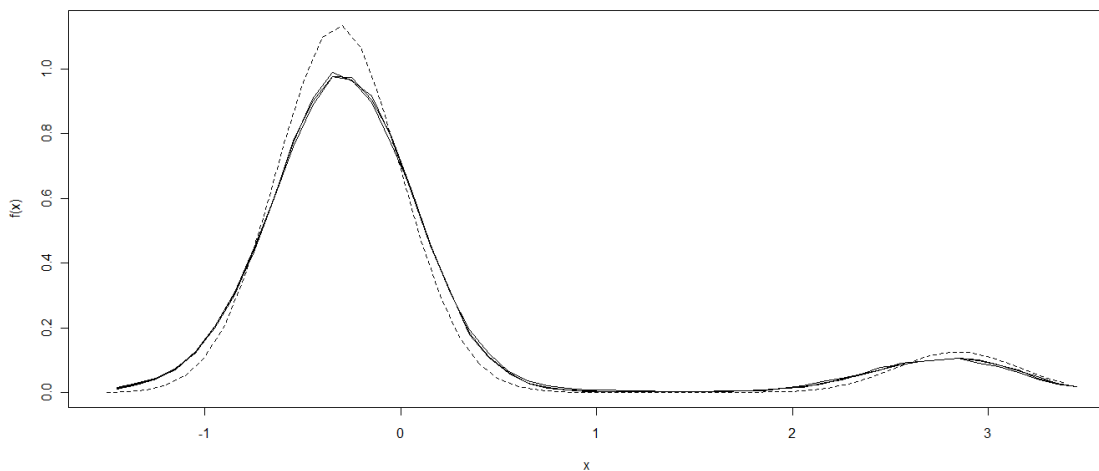
**Figura 2: Função de densidade dos erros do Modelo 1 (linha tracejada) e funções de densidade estimada dos resíduos do Modelo 1 (linhas contínuas).**



**Figura 3: Função de densidade dos erros do Modelo 2 (linha tracejada) e funções de densidade estimada dos resíduos do Modelo 2 (linhas contínuas).**



**Figura 4: Função de densidade dos erros do Modelo 3 (linha tracejada) e funções de densidade estimada dos resíduos do Modelo 3 (linhas contínuas).**



#### **4.2 Resultados das simulações sobre os intervalos de previsão**

As Tabelas 2 a 5 mostram os atributos dos intervalos construídos para os Modelos 1 a 4 sem correção de vício. A Tabela 6 mostra os atributos dos intervalos construídos para o Modelo 4 com correção de vício. A Tabela 7 mostra os atributos dos intervalos construídos para o Modelo 5. A coluna “Correção” na Tabela 7 especifica o tipo de correção de vício utilizado, se não houve correção de vício (SCV), se houve



correção de vício mas não do desvio padrão dos resíduos (CCV) e se houve correção de vício também do desvio padrão dos resíduos (CCDP).

A Tabela 2 mostra os resultados para o Modelo 1, uma série ARFIMA(1,  $d$ , 1) com  $\phi = 0,3$ ,  $d = 0,3$ ,  $\theta = -0,3$ , erros com distribuição normal e tamanho igual a 100.

Nas previsões 1 passo à frente com intervalo BJ, a cobertura real ficou abaixo da nominal para todos os métodos de estimação usados. Para os três métodos de estimação, os intervalos *bootstrap* se aproximaram mais da cobertura nominal do que o intervalo BJ, sendo que o intervalo PRR se aproximou mais que o intervalo EPB. Para os intervalos *bootstrap*, no entanto, a cobertura continuou abaixo da nominal. Nas previsões 10 passos à frente com método de estimação FT, o intervalo BJ apresentou cobertura ainda menor (0,9206). Os intervalos *bootstrap* melhoraram essa cobertura, principalmente o PRR (0,9413), embora esta tenha ficado ainda abaixo da nominal. Com os métodos de estimação MV-GPH e MV-SPR, a cobertura do intervalo BJ já é próxima da nominal (0,9476 e 0,9514). Nestes casos, a utilização dos intervalos *bootstrap* distanciou a cobertura da nominal, especialmente quando o intervalo PRR foi utilizado (0,9658 e 0,9643).

**Tabela 2: Resultados Monte Carlo para o Modelo 1 (ARFIMA(1,  $d$ , 1) com erros normais)**

Passos à Frente	Método de Estimação	Intervalo	C	$C_{INF}$	$C_{SUP}$	S	Tamanho
1	FT	BJ	0,9344	0,0344	0,0312	0,0156	3,8780
		PRR	0,9485	0,0263	0,0253	0,0016	4,2456
		EPB	0,9433	0,0298	0,0270	0,0068	4,1006
	MV-GPH	BJ	0,9416	0,0296	0,0287	0,0083	3,8974
		PRR	0,9467	0,0271	0,0261	0,0032	4,0264
		EPB	0,9439	0,0285	0,0276	0,0061	3,9960
	MV-SPR	BJ	0,9423	0,0294	0,0283	0,0077	3,9023
		PRR	0,9444	0,0288	0,0268	0,0056	3,9888
		EPB	0,9431	0,0293	0,0276	0,0069	3,9810
10	FT	BJ	0,9203	0,0419	0,0378	0,0297	6,0457
		PRR	0,9413	0,0301	0,0286	0,0087	6,3630
		EPB	0,9299	0,0368	0,0333	0,0201	6,3823
	MV-GPH	BJ	0,9476	0,0266	0,0258	0,0024	6,5189
		PRR	0,9658	0,0174	0,0169	0,0157	7,1207
		EPB	0,9535	0,0237	0,0228	0,0035	6,7457
	MV-SPR	BJ	0,9519	0,0243	0,0238	0,0019	6,5875
		PRR	0,9643	0,0181	0,0177	0,0142	7,0427
		EPB	0,9540	0,0234	0,0226	0,0040	6,6795

A Tabela 3 mostra os resultados para o Modelo 2, uma série ARFIMA(1,  $d$ , 1) com  $\phi = 0,3$ ,  $d = 0,3$ ,  $\theta = -0,3$ , erros com distribuição exponencial centralizada e tamanho igual a 100.

A Tabela 4 mostra os resultados para o Modelo 3, uma série ARFIMA(1,  $d$ , 1) com  $\phi = 0,3$ ,  $d = 0,3$ ,  $\theta = -0,3$ , erros com distribuição bimodal e tamanho igual a 100.

Os resultados dos Modelos 2 e 3 mostraram muitas similaridades, com a diferença de que a assimetria dos intervalos BJ foram maiores no Modelo 3. Os resultados dos dois modelos, apresentados nas Tabelas 3 e 4, são descritos nos parágrafos seguintes.

A cobertura dos intervalos de previsão BJ, tanto 1 quanto 10 passos à frente, possui uma assimetria muito grande, como se pode ver pelos valores  $C_{INF}$ ,  $C_{SUP}$  e  $S$ . Os intervalos *bootstrap* diminuíram o valor de  $S$  para todos os métodos de estimação e as duas quantidades de passos à frente. Na previsão 1 passo à frente, os intervalos EPB tiveram menor valor de  $S$  para os três métodos de estimação. Na previsão 10 passos à

frente, o intervalo PRR possuiu menor valor de  $S$  com o método de estimação FT, mas para os dois outros métodos de estimação o intervalo EPB foi novamente melhor.

No que diz respeito ao valor de cobertura, no Modelo 2, o intervalo BJ apresenta valores de cobertura competitivos com os valores dos intervalos *bootstrap*. No caso de previsões 10 passos à frente a com métodos de estimação MV-GPH e MV-SPR a cobertura do intervalo BJ é melhor, próxima da nominal, o mesmo acontece no Modelo 3 sob as mesmas condições. Nos demais casos a cobertura do intervalo BJ é bem mais distante da nominal em comparação com os intervalos *bootstrap*.

Apesar de diminuir a assimetria apresentada nos intervalos BJ, os intervalos *bootstrap* ainda apresentam alguma assimetria, mesmo no caso do intervalo EPB que apresentou melhores resultados. Pelo que se pode observar pelos valores de  $C_{INF}$  e  $C_{SUP}$  para o intervalo EPB nas Tabelas 3 e 4, é possível notar que o valor de  $C_{SUP}$  tende a ser maior que o de  $C_{INF}$ , especialmente quando o método de estimação escolhido é o MV-GPH ou o MV-SPR. Estes resultados podem ser consequência de dificuldade de estimar  $F_a$ , como sugerem as Figuras 2 e 3.

**Tabela 3: Resultados Monte Carlo para o Modelo 2 (ARFIMA(1,  $d$ , 1) com erros exponenciais centralizados)**

Passos à Frente	Método de Estimação	Intervalo	C	$C_{INF}$	$C_{SUP}$	S	Tamanho
1	FT	BJ	0,9369	0,0004	0,0627	0,0623	3,8266
		PRR	0,9581	0,0116	0,0303	0,0187	4,1984
		EPB	0,9407	0,0290	0,0303	0,0093	4,0219
	MV-GPH	BJ	0,9429	0,0005	0,0566	0,0561	3,8532
		PRR	0,9604	0,0107	0,0289	0,0182	3,9424
		EPB	0,9548	0,0160	0,0292	0,0132	3,9281
	MV-SPR	BJ	0,9422	0,0002	0,0576	0,0574	3,8525
		PRR	0,9600	0,0103	0,0297	0,0194	3,9026
		EPB	0,9553	0,0144	0,0303	0,0159	3,8811
10	FT	BJ	0,9226	0,0156	0,0618	0,0462	5,9782
		PRR	0,9437	0,0243	0,0320	0,0077	6,3641
		EPB	0,9204	0,0453	0,0343	0,0296	6,3050
	MV-GPH	BJ	0,9495	0,0054	0,0451	0,0397	6,5616
		PRR	0,9668	0,0105	0,0227	0,0168	7,2216
		EPB	0,9541	0,0194	0,0266	0,0072	6,8352
	MV-SPR	BJ	0,9509	0,0033	0,0458	0,0425	6,5162
		PRR	0,9651	0,0099	0,0249	0,0152	6,9707
		EPB	0,9542	0,0170	0,0289	0,0119	6,5899

**Tabela 4: Resultados Monte Carlo para o Modelo 3 (ARFIMA(1,  $d$ , 1) com erros bimodais)**

Passos à Frente	Método de Estimação	Intervalo	C	C <sub>INF</sub>	C <sub>SUP</sub>	S	Tamanho
1	FT	BJ	0,9001	0,0005	0,0994	0,0989	3,8545
		PRR	0,9497	0,0146	0,0357	0,0211	4,4981
		EPB	0,9402	0,0259	0,0339	0,0098	4,2400
	MV-GPH	BJ	0,9018	0,0007	0,0975	0,0968	3,8785
		PRR	0,9534	0,0158	0,0308	0,0150	4,1685
		EPB	0,9505	0,0191	0,0304	0,0113	4,1664
	MV-SPR	BJ	0,9020	0,0006	0,0974	0,0968	3,8814
		PRR	0,9524	0,0165	0,0311	0,0146	4,1354
		EPB	0,9513	0,0177	0,0310	0,0133	4,1378
10	FT	BJ	0,9216	0,0089	0,0695	0,0606	6,0191
		PRR	0,9487	0,0205	0,0309	0,0104	6,2594
		EPB	0,9146	0,0514	0,0339	0,0353	6,1788
	MV-GPH	BJ	0,9480	0,0025	0,0496	0,0471	6,5026
		PRR	0,9715	0,0083	0,0201	0,0216	6,9472
		EPB	0,9526	0,0230	0,0243	0,0027	6,5430
	MV-SPR	BJ	0,9494	0,0012	0,0494	0,0482	6,5545
		PRR	0,9696	0,0085	0,0219	0,0196	6,8336
		EPB	0,9576	0,0170	0,0254	0,0084	6,4313

A Tabela 5 mostra os resultados para o Modelo 4, uma série ARFIMA(1,  $d$ , 0) com  $\phi = 0,3$ ,  $d = 0,3$ , erros com distribuição normal e tamanho igual a 100.

Para previsões 1 passo à frente, novamente o intervalo BJ apresentou cobertura abaixo da nominal para todos os métodos de estimação. Para o método de estimação FT, os intervalos *bootstrap* tiveram cobertura melhor que a do BJ, especialmente o intervalo PRR, embora sem ultrapassar 0,9425. No caso dos métodos de estimação MV-GPH e MV-SPR, os intervalos *bootstrap* não surtiram efeito, e estes intervalos apresentam maior tamanho que o BJ. Para previsões 10 passos à frente e método de estimação FT, o intervalo BJ apresenta cobertura muito abaixo da nominal (0,9269). Esta cobertura é melhorada com a utilização dos intervalos *bootstrap*, especialmente o PRR, ainda que neste caso também sem ultrapassar 0,9450. No caso dos métodos de estimação MV-GPH e MV-SPR, o intervalo BJ já apresenta cobertura perto da nominal. O intervalo EPB não melhora esta cobertura, enquanto o intervalo PRR possui uma cobertura ainda mais distante da nominal.

**Tabela 5: Resultados Monte Carlo para o Modelo 4 (ARFIMA(1,  $d$ , 0) com erros normais) , sem correção de vício**

Passos à Frente	Método de Estimação	Intervalo	C	C <sub>INF</sub>	C <sub>SUP</sub>	S	Tamanho
1	FT	BJ	0,9355	0,0325	0,0320	0,0145	3,8551
		PRR	0,9424	0,0298	0,0278	0,0076	4,0028
		EPB	0,9398	0,0311	0,0291	0,0102	3,9902
	MV-GPH	BJ	0,9427	0,0290	0,0283	0,0073	3,8882
		PRR	0,9449	0,0278	0,0273	0,0051	3,9713
		EPB	0,9435	0,0288	0,0278	0,0066	3,9611
	MV-SPR	BJ	0,9434	0,0286	0,0280	0,0066	3,8923
		PRR	0,9438	0,0286	0,0276	0,0062	3,9598
		EPB	0,9428	0,0288	0,0284	0,0072	3,9478
10	FT	BJ	0,9269	0,0372	0,0359	0,0231	5,0032
		PRR	0,9428	0,0293	0,0278	0,0071	5,1891
		EPB	0,9339	0,0339	0,0322	0,0161	5,2040
	MV-GPH	BJ	0,9480	0,0264	0,0256	0,0020	5,3186
		PRR	0,9615	0,0196	0,0188	0,0116	5,6293
		EPB	0,9528	0,0242	0,0230	0,0028	5,4671
	MV-SPR	BJ	0,9529	0,0238	0,0233	0,0029	5,3891
		PRR	0,9607	0,0199	0,0194	0,0107	5,6111
		EPB	0,9534	0,0238	0,0228	0,0034	5,4284

A Tabela 6 também mostra os resultados para o Modelo 4, uma série ARFIMA(1,  $d$ , 0) com  $\phi = 0,3$ ,  $d = 0,3$ , erros com distribuição normal e tamanho igual a 100, mas desta vez os parâmetros sofreram correção de vício *bootstrap*.

O uso da correção de vício parece ter melhorado a cobertura de alguns intervalos, mas piorado o de outros, enquanto outros não foram afetados pelo procedimento. Não é possível identificar algum tipo de padrão na maneira como a correção de vício altera a cobertura dos intervalos de confiança.

**Tabela 6: Resultados Monte Carlo para o Modelo 4 (ARFIMA(1, d, 0) com erros normais), com correção de vício**

Passos à Frente	Método de Estimação	Intervalo	C	C <sub>INF</sub>	C <sub>SUP</sub>	S	Tamanho
1	FT	BJ	0,9384	0,0315	0,0301	0,0116	3,8782
		PRR	0,9424	0,0299	0,0277	0,0076	4,0016
		EPB	0,9426	0,0297	0,0276	0,0073	4,0104
	MV-GPH	BJ	0,9425	0,0297	0,0278	0,0075	3,9293
		PRR	0,9413	0,0303	0,0284	0,0087	4,0363
		EPB	0,9527	0,0246	0,0226	0,0028	4,1695
	MV-SPR	BJ	0,9432	0,0288	0,0280	0,0068	3,8946
		PRR	0,9425	0,0293	0,0283	0,0076	3,9571
		EPB	0,9443	0,0284	0,0274	0,0058	3,9747
10	FT	BJ	0,9346	0,0338	0,0315	0,0153	5,0973
		PRR	0,9403	0,0312	0,0285	0,0097	5,1288
		EPB	0,9402	0,0314	0,0284	0,0098	5,2611
	MV-GPH	BJ	0,9428	0,0293	0,0279	0,0072	5,1576
		PRR	0,9515	0,0244	0,0240	0,0016	5,8144
		EPB	0,9585	0,0216	0,0199	0,0085	5,6289
	MV-SPR	BJ	0,9502	0,0253	0,0244	0,0009	5,3097
		PRR	0,9592	0,0207	0,0201	0,0092	5,6690
		EPB	0,9515	0,0249	0,0236	0,0015	5,3725

A Tabela 7 mostra os resultados para o Modelo 5, uma série ARMA(1,1) com  $\phi = 0,3$ ,  $\theta = -0,3$ , erros normais, tamanho igual a 50 e previsão 1 passos à frente.

Sem nenhuma correção de vício, e com previsão 1 passo à frente, o intervalo BJ tem baixa cobertura (0,9319). Os intervalos *bootstrap* melhoram a cobertura, mas sem passar de 0,94. Com previsão 10 passos à frente a cobertura do intervalo BJ é maior (0,9427) e é melhorada com a utilização do intervalo PRR. A correção de vício *bootstrap* sem correção do desvio padrão dos resíduos não provoca melhora nas previsões 1 passo à frente. No caso das previsões 10 passos à frente, há um aumento da cobertura em direção à nominal para os três tipos de intervalos utilizados. Utilizando-se da correção de vício *bootstrap* do desvio padrão dos resíduos, os intervalos para a previsão 1 passo à frente apresentam todos uma melhora da cobertura, em comparação à utilização da correção de vício. No entanto, mesmo para estes casos a cobertura fica abaixo de 0,94. Para previsões 10 passos à frente, a correção de vício do desvio padrão dos resíduos também causa melhora na cobertura dos intervalos, exceto para o intervalo PRR.

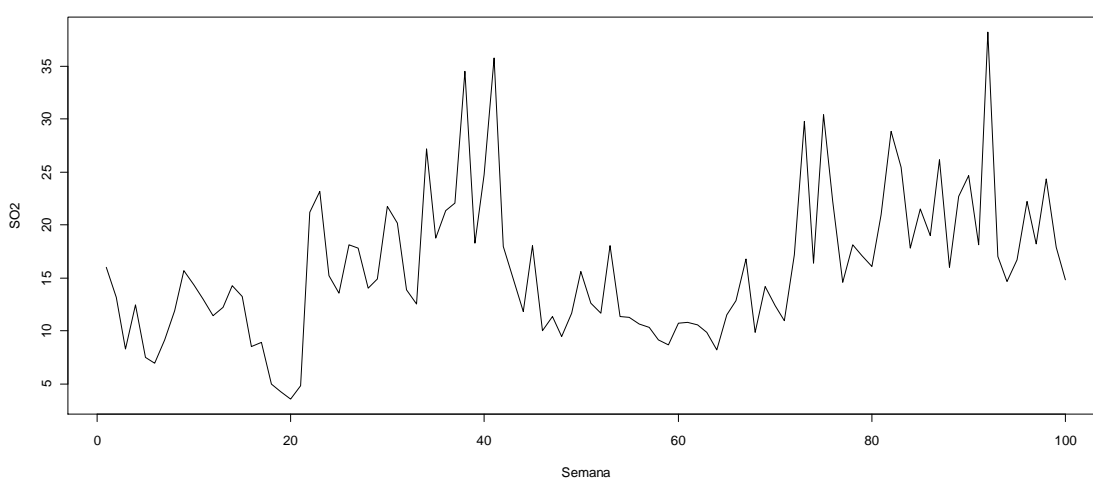
**Tabela 7: Resultados Monte Carlo para o Modelo 5 (ARMA(1,1) com erros normais)**

Correção de Vício	Passos à Frente	Intervalo	C	C <sub>INF</sub>	C <sub>SUP</sub>	S	Tamanho
SCV	1	BJ	0,9319	0,0344	0,0338	0,0182	3,8200
		PRR	0,9355	0,0321	0,0324	0,0145	3,9423
		EPB	0,9363	0,0319	0,0318	0,0137	3,9828
	10	BJ	0,9427	0,0287	0,0286	0,0073	4,6494
		PRR	0,9483	0,0260	0,0257	0,0017	4,8505
		EPB	0,9417	0,0294	0,0289	0,0083	4,7036
CCV	1	BJ	0,9306	0,0349	0,0345	0,0194	3,8305
		PRR	0,9333	0,0336	0,0331	0,0167	3,9518
		EPB	0,9360	0,0323	0,0317	0,0140	4,0623
	10	BJ	0,9446	0,0278	0,0276	0,0054	4,6964
		PRR	0,9508	0,0249	0,0243	0,0008	4,9519
		EPB	0,9447	0,0279	0,0274	0,0053	4,8381
CCDP	1	BJ	0,9343	0,0332	0,0325	0,0157	3,8985
		PRR	0,9368	0,0320	0,0312	0,0132	4,0174
		EPB	0,9395	0,0307	0,0298	0,0105	4,1313
	10	BJ	0,9486	0,0259	0,0255	0,0014	4,7730
		PRR	0,9543	0,0233	0,0224	0,0043	5,0322
		EPB	0,9488	0,0259	0,0253	0,0012	4,9110

## 5 APLICAÇÃO A UMA SÉRIE REAL

Nesta seção os métodos de construção de intervalos são utilizados em uma série real. A série em questão é a concentração semanal média de  $\text{SO}_2$  (em  $\mu\text{g}/\text{m}^3$ ) na cidade de São Paulo de 22 de novembro de 1998 (primeiro dia da primeira semana medida) a 21 de dezembro de 2010 (último dia da última semana medida). A série possui exatamente 100 observações. A Figura 5 mostra o gráfico de linhas da série de  $\text{SO}_2$ :

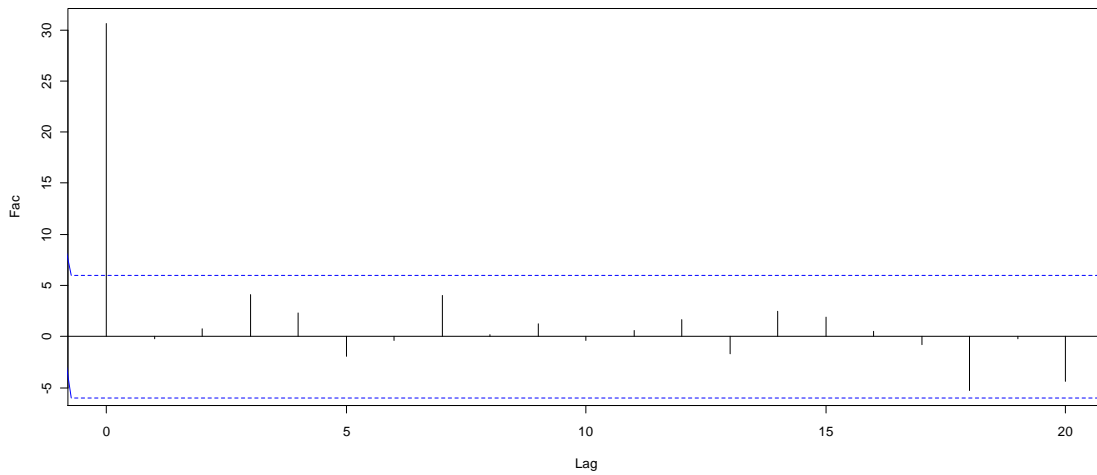
**Figura 5: Gráfico de linhas da série de  $\text{SO}_2$**



Primeiramente foi suposto um modelo  $\text{ARFIMA}(0, d, 0)$ . Utilizando-se o estimador FT, foi encontrado  $\hat{d} = 0,3757$ . A Figura 6 mostra a função de autocovariância estimada da série de  $\text{SO}_2$  diferenciada por  $\hat{d}$ . A função de autocovariância estimada da série  $\text{SO}_2$  não possui valores significativamente diferentes de zero, indicando que a série diferenciada. O modelo  $\text{ARFIMA}(0, d, 0)$  foi considerado adequado para os dados em questão.

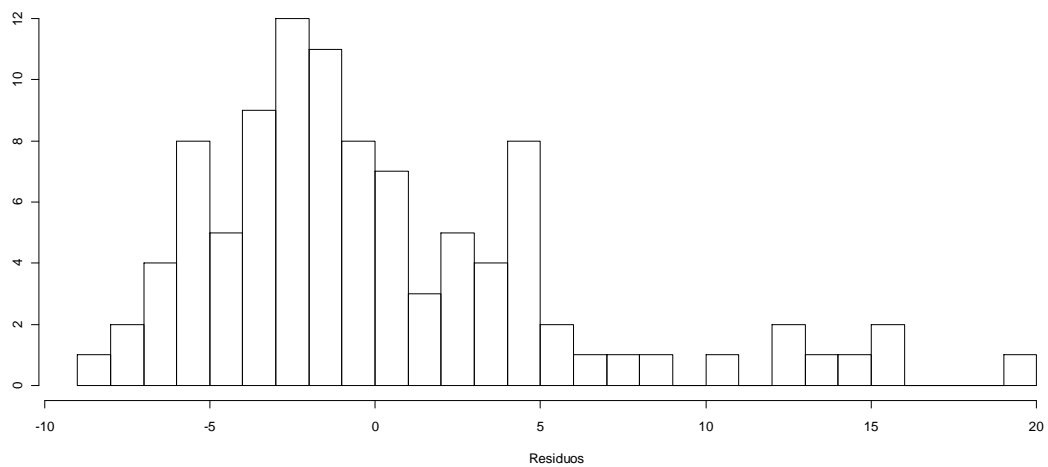


**Figura 6: Função de autocovariância estimada da série de SO<sub>2</sub> diferenciada e intervalo de confiança de 95% (linha tracejada)**



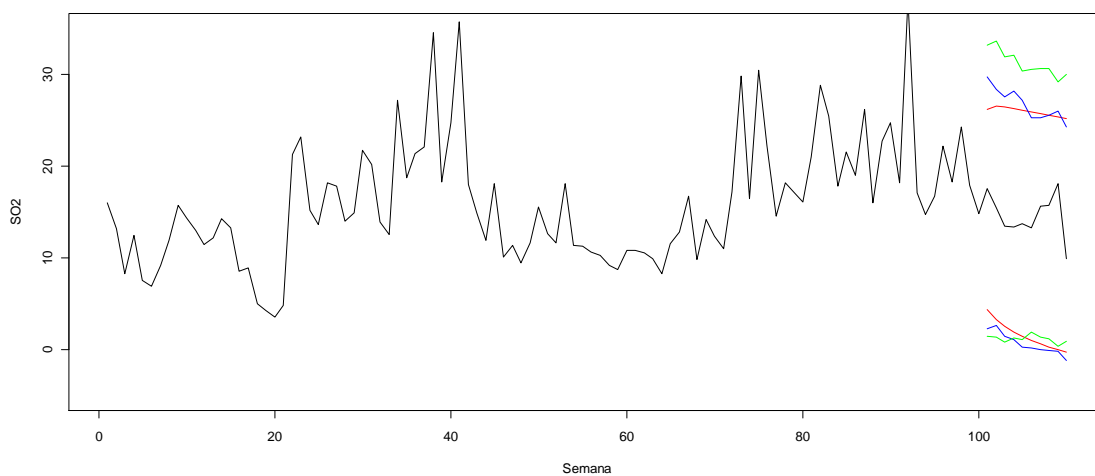
O histograma na Figura 7 indica que os resíduos não possuem distribuição normal.

**Figura 7: Histograma dos resíduos da série de SO<sub>2</sub>**



A Figura 8 mostra os intervalos de previsão BJ (vermelho), PRR (azul) e EPB (verde), de 1 a 10 semanas à frente construídos na 100<sup>a</sup> semana. Os intervalos em questão possuem nível de cobertura 0,95.

**Figura 8: Série de SO<sub>2</sub> e intervalos de previsão BJ (vermelho), PRR (azul) e EPB (verde), de 1 a 10 semanas à frente construídos na 100ª semana.**



## 6 CONSIDERAÇÕES FINAIS

Este trabalho utilizou 3 estimadores diferentes para o parâmetro de memória em modelos ARFIMA, a saber GPH, SPR e FT, e três intervalos diferentes, BJ, PRR e EPB, utilizando-se também da correção *bootstrap* de vício para se construir intervalos de previsão para modelos ARFIMA. Intervalos de previsão também foram construídos para modelos ARMA com o estimador de máxima verossimilhança. Dos três intervalos utilizados, dois são intervalos *bootstrap*: PRR e EPB. O autor do trabalho não tem conhecimento do uso do *bootstrap* na literatura para a construção de intervalos de previsão em modelos ARFIMA. O intervalo PRR é uma adaptação para séries ARFIMA do intervalo homônimo para séries ARIMA, proposto por Pascual et al (2004). O intervalo EPB é uma proposta do autor, baseado em propostas similares para outros modelos de séries temporais, como por exemplo o intervalo proposto em Masarotto (1990). O autor propõe também uma correção *bootstrap* do vício dos resíduos. Ainda que outros autores já tenham utilizado o *bootstrap* para corrigir vício no contexto de previsão em séries temporais, a devida atenção não foi dada ao desvio padrão dos resíduos.

Os intervalos *bootstrap* apresentaram resultados melhores que os do intervalo BJ. Mesmo com erros normais, suposição importante do intervalo BJ, os intervalos *bootstrap* costumam apresentar cobertura mais próxima da nominal. A diferença de cobertura nestes casos, no entanto, não é muito grande, e os intervalos BJ apresentam menor tamanho. No caso de previsões 10 passos à frente com estimadores MV-GPH e MV-SPR, o intervalo *bootstrap* apresenta melhor cobertura e menor tamanho. No caso de erros assimétricos, a diferença de desempenho a favor dos intervalos *bootstrap* foi maior. Os intervalos *bootstrap* apresentaram melhores resultados independentemente dos estimadores utilizados e do número de passos à frente da previsão. É interessante notar, no entanto, que os intervalos *bootstrap* ainda apresentam alguma assimetria, embora bem menor que a do intervalo BJ. Uma hipótese levantada pelo autor, com base em resultados de simulações, é de que isto se deve à má estimação da distribuição dos erros. No que diz respeito aos intervalos *bootstrap* utilizados, não foi possível encontrar sinais claros de superioridade de um sobre o outro. Em algumas situações o intervalo PRR é superior, em outras o EPB. Entre os Modelos analisados, o intervalo PRR se saiu melhor nos Modelos 1 e 4, e o intervalo EPB nos Modelos 2, 3 e 5.

Ainda que os estimadores utilizados sejam viciados para amostras pequenas, como se pode ver via simulação, a utilização de corretores *bootstrap* de vício não causou impacto positivo nas coberturas dos intervalos. O autor credita este fato mais à deficiência da correção de vício do que à inexistência de problemas causados pelo vício. Isto é um ponto que necessita ser investigado. O autor também propôs que corretores *bootstrap* de vício fossem utilizados para corrigir o desvio padrão dos resíduos. Nas séries ARFIMA analisadas o vício do desvio padrão era irrelevante, e por isto a técnica foi utilizada para a série ARMA tamanho 50 com resultados claramente positivos.

Propostas para trabalhos futuros incluem propor intervalos *bootstrap* para outros modelos de séries temporais, encontrar corretores de vício mais eficientes para pequenas amostras, melhorar a estimação da distribuição dos erros e avaliar os métodos com valores de  $d$  mais próximos da região de não estacionaridade.

## REFERÊNCIAS

- Bollerslev, T. (1986) Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31, 307-327.
- Box, G.E.P., Jenkins, G.M. (1976) *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.
- Brillinger, D. R. (1975) *Time Series: Data Analysis and Theory*. New York: Holt, Rinehart and Winston.
- Doukham, P., Oppenheim, G., Taqqu, M.S. (2003) *Theory and Applications of Long Range Dependency*. Basel: Birkhauser.
- Efron, B. (1979) Bootstrap methods: another look at the Jackknife. *The Annals of Statistics*. 7-1, 1-26.
- Efron, B., Tibshirani, R. (1993) *An Introduction to the Bootstrap*. New York: Chapman and Hall.
- Engle, R.F. (1982) Autoregressive conditional heteroskedasticity with the estimates of the variance of U. K. inflation. *Econometrica*, 50, 987-1008.
- Geweke, J. Porter-Hudak, S. (1983) The estimation and application of long memory time series models. *Journal of Time Series Analysis*, 4, 221-238.
- Grigoletto, M. (1998) Bootstrap prediction intervals for autoregression: some alternatives. *International Journal of Forecasting*, 14, 447-456.
- Fox, R., Taqqu, M.S. (1986) Large sample properties of parameter estimates for strongly dependent stationary Gaussian time series. *The Annals of Statistics*, 14, 517-532.
- Franco, G.C., Reisen, V.A. (2004) Bootstrap techniques in semiparametric estimation methods for ARFIMA models: a comparison study. *Computational Statistics*, 19, 243-259.
- Granger, C.M.G., Joyeux, R. (1980) An introduction to long memory time series models and fractional differencing. *Journal of Time Series Analysis*, 1, 15-29.
- Harvey, A.C. (1989) *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge: Cambridge University Press.
- Hosking, J.R.M. (1981) Fractional differencing. *Biometrika*, 68, 165-176.
- Kilian, L. (1998) Small sample confidence intervals for impulse response functions. *The Review of Economics and Statistics*, 80, 218-230.

- Kim, J.H. (2001) Bootstrap-after-bootstrap prediction intervals for autoregressive models. *Journal of Business & Economic Statistics*, 19, 117-128.
- Masarotto, G. (1990) Bootstrap prediction intervals for autoregressions. *International Journal of Forecasting*, 6, 229-239.
- Parapoditis, E., Politis, D.N. (1999) The local bootstrap for periodogram statistics. *Journal of Time Series Analysis*, 20-2, 193-222.
- Parzen, E. (1962) On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33, 1065-1076.
- Pascual, L., Romo, R., Ruiz, E. (2004) Bootstrap predictive inference for ARIMA processes. *Journal of Time Series Analysis*, 25-4, 449-465.
- Reisen, V.A. (1994) Estimation of the fractional difference parameter in the ARIMA(p,d,q) model using the smoothed periodogram. *Journal of Time Series Analysis*, 15-3, 335-350.
- Robinson, P.M. (1995) Log-periodogram regression of time series with long-range dependency. *The Annals of Statistics*, 23-3, 1048-1072
- Thombs, L.A, Schucany, W.R. (1990) Bootstrap prediction intervals for autoregression. *Journal of the American Statistical Association*, 85, 486-492
- West, M., Harrison, J. (1997) *Bayesian Forecasting and Dynamic Models*. New York: Springer.

## ANEXO – Programa das Simulações

Programa escrito em linguagem Fortran.

```
! PROGRAMA PRINCIPAL
USE MSFLIB
IMPLICIT NONE
REAL(8) A(1100),ALFA,ARF(3,1100),ARFMC(3,1100),ARFNP(3,1100),AVAR
REAL(8) CALERTB(30,3,1100),CALERTBM(30,3),CIP,CIPV(30),CNST
REAL(8) COV(10,10),CPSEC
REAL(8) D,DF(3,1100),DFMC(3,1100),DFNP(3,1100),DIF(3,1100)
REAL(8) DIFNP(3,1100),DP(3),DPBC(3),DPNP(3),DPPSI(100),DV(30)
REAL(8) EQM(2,30,3,4,100),EQMBC(2,30,3,4,100)
REAL(8) EST(3),ESTBC(3),ESTNP(3)
REAL(8) Fi,FIBC(3),FIEST(3),FIESTNP(3),FiV(30)
REAL(8) GEB(1000,100),GREB(1000,100),GUS
REAL IALFA2
REAL(8) LIMITE(30,2,3,4,100),LIMITEBC(30,2,3,4,100)
REAL(8) MBETA,MEB(3,4),MEDIAEST
REAL(8) NAOCOB(30,2,3,4,100),NAOCOBBC(30,2,3,4,100)
REAL(8) PI(3,1100),PINP(3,1100),PSIS(3,1100),PSIMC(3,1100)
REAL(8) R(1100),R2(100),RES(3,1100),RESNA(3,1100),RES2(1100)
REAL(8) RESNP(3,1100),RESNPNA(3,1100)
REAL(8) SCOV(3,0:1100),SCOVNP(3,0:1100)
REAL(8) T1,T2,TEMPOTOTAL,Teta,TETABC(3),TETAEST(3),TETAESTNP(3)
REAL(8) TetaV(30)
REAL(8) TOLBC,TOLSS,TESTE(2,1100),TESTE2(1100),TESTE3(3,4)
REAL(8) TESTE4(3,1100)
REAL(8) VESP(100),VICIO(3,4)
REAL(8) W(1100)
REAL(8) X1(3,1100),XF(3,100),XFBC(3,100),XFNP(3,2:4,100,1100)
REAL(8) XFNIBC(3,2:4,100,1000),XFUL(2,3,4,100),XFULBC(2,3,4,100)
REAL(8) YBOOT(3,1100)
REAL(8) ZN(2)
REAL(8) TESTEFT
INTEGER ALERT,ALERTB
INTEGER CALERT(30),CIPVB(30),CONTMC(30,4,3),CONTBC(30,3,3)
INTEGER CONTMCBC(30,3,3)
INTEGER DVB(30)
INTEGER ESTIMP(3),ESTIMPV(30,3)
INTEGER FiVB(30),FTPV(30),FTPVB(30)
INTEGER GHPV(30),GHPVB(30)
INTEGER IALFA,I3,INF,INT1,INT1E2(4),INT1E2V(30,4),INT1V(30)
INTEGER INT1VB(30),INT2,INT2V(30),INT2VB(30),INT3,INT3V(30)
INTEGER INT3VB(30),ISEED
INTEGER K,K1,K2,K3,K4,K5,KBC1,KBC2,KBC3,KT,KVET(1100)
INTEGER MAXBC,ME,MEV(30),MEVB(30),MKBC
INTEGER N,NA,NB,NBV(30),NBVB(30),NN,NN2,NN3,NT,NTV(30),NTVB(30)
INTEGER NV(30),NVB(30)
```

```

INTEGER PASSOMAX,PASSOS(100),PASSOSV(30,100),PASSOSVB(30),PSSM
INTEGER pzin,pzinME,pzinMEV(30),pzinMEVB(30),pzinV(30),pzinVB(30)
INTEGER QTESTI,QTESTIV(30),QNTS,QTPASSOS,QTPASSOSV(30)
INTEGER QTPASSOSVB(30),qzin,qzinME,qzinMEV(30),qzinMEVB(30)
INTEGER qzinV(30),qzinVB(30)
INTEGER SIMU,SPRPV(30),SPRPVB(30),SUP
INTEGER TER,TERK,TERV(30),TERVB(30),TetaVB(30),TIPEST
INTEGER VetOrdAR(1),VetOrdMA(1)
EXTERNAL DNORIN
REAL(8) DNORIN
TESTEFT=0
MAXBC=10
TOLSS=0.0
TOLBC=0.0
WRITE(*,*) 'Quantas simulacoes voce pretende fazer?'
READ(*,*) SIMU
DO K1=1,SIMU
  NVB(K1)=0
  NBVB(K1)=0
  NTVB(K1)=0
  pzinVB(K1)=0
  qzinVB(K1)=0
  FiVB(K1)=0
  TetaVB(K1)=0
  DVB(K1)=0
  TERVB(K1)=0
  GPHPVB(K1)=0
  SPRPVB(K1)=0
  FTPVB(K1)=0
  MEVB(K1)=0
  pzinMEVB(K1)=0
  qzinMEVB(K1)=0
  INT1VB(K1)=0
  INT2VB(K1)=0
  INT3VB(K1)=0
  QTPASSOSVB(K1)=0
  PASSOSVB(K1)=0
  CIPVB(K1)=0
ENDDO
DO 1 K1=1,SIMU
  WRITE(*,*) 'Simulacao atual'
  WRITE(*,*) K1
  IF (NVB(K1).EQ.0) THEN
    WRITE(*,*) 'Gerar quantas series? (No maximo 1100)'
    CALL PERGUNTADIS(NVB,NV,SIMU,K1)
  ENDIF
  IF (NBVB(K1).EQ.0) THEN
    WRITE(*,*) 'Entre com o numero de replicacoes bootstrap:'
    CALL PERGUNTADIS(NBVB,NBV,SIMU,K1)
  ENDIF
  IF (NTVB(K1).EQ.0) THEN

```



```

WRITE(*,*) 'Entre com o tamanho da serie [ NT > 40 ]'
CALL PERGUNTADIS(NTVB,NTV,SIMU,K1)
ENDIF
IF (FIVB(K1).EQ.0) THEN
WRITE(*,*) 'Entre com o valor de Fi1'
WRITE(*,*) '(Se deseja p=0, basta Fi1=0)'
CALL PERGUNTACON(FIVB,FiV,SIMU,K1)
ENDIF
IF (FiV(K1).EQ.0) THEN
pzinV(K1)=0
ELSE
pzinV(K1)=1
ENDIF
IF (TetaVB(K1).EQ.0) THEN
WRITE(*,*) 'Entre com o valor de Teta1'
WRITE(*,*) '(Se deseja q=0, basta Teta1=0)'
CALL PERGUNTACON(TetaVB,TetaV,SIMU,K1)
ENDIF
IF (TetaV(K1).EQ.0) THEN
qzinV(K1)=0
ELSE
qzinV(K1)=1
ENDIF
IF (DVB(K1).EQ.0) THEN
WRITE(*,*) 'Entre com valor de d'
WRITE(*,*) '(Se deseja s,rie Arima, basta d=0)'
CALL PERGUNTACON(DVB,DV,SIMU,K1)
ENDIF
IF (DV(K1).NE.0) THEN
IF (GPHPVB(K1).EQ.0) THEN
WRITE(*,*) 'Deseja utilizar estimador GPH?'
WRITE(*,*) 'Sim=1, Nao=0'
CALL PERGUNTADIS(GPHPVB,GPHPV,SIMU,K1)
ENDIF
IF (SPRPVB(K1).EQ.0) THEN
WRITE(*,*) 'Deseja utilizar estimador SPR?'
WRITE(*,*) 'Sim=1, Nao=0'
CALL PERGUNTADIS(SPRPVB,SPRPV,SIMU,K1)
ENDIF
IF (FTPVB(K1).EQ.0) THEN
WRITE(*,*) 'Deseja utilizar estimador FT?'
WRITE(*,*) 'Sim=1, Nao=0'
CALL PERGUNTADIS(FTPVB,FTPV,SIMU,K1)
ENDIF
ELSE
GPHPV(K1)=0
SPRPV(K1)=0
FTPV(K1)=0
ENDIF
IF (QTPASSOSVB(K1).EQ.0) THEN
WRITE(*,*) 'Quantos passos a frente diferentes?'

```

```

CALL PERGUNTADIS(QTPASSOSVB,QTPASSOSV,SIMU,K1)
ENDIF
2  FORMAT('Numero de passos a frente da previsao: ',I2)
IF (PASSOSVB(K1).EQ.0) THEN
DO K2=1,QTPASSOSV(K1)
WRITE(*,2) K2
READ(*,*) PASSOSV(K1,K2)
ENDDO
IF (SIMU.GT.1.AND.K1.LT.SIMU) THEN
WRITE(*,*) 'Repetir estas condicoes por quantas simulacoes?'
READ(*,*) QNTS
DO K2=(K+1),(K+QNTS)
DO K3=1,QTPASSOSV(K1)
PASSOSV(K2,K3)=PASSOSV(K1,K3)
ENDDO
PASSOSVB(K2)=1
ENDDO
ENDIF
ENDIF
IF (CIPVB(K1).EQ.0) THEN
WRITE(*,*) 'Entre com a cobertura nominal do intervalo de'
WRITE(*,*) 'previsao (entre 0 e 1):'
CALL PERGUNTACON(CIPVB,CIPV,SIMU,K1)
ENDIF
IF (TERVB(K1).EQ.0) THEN
WRITE(*,*) 'Entre com o tipo de erro:'
WRITE(*,*) '1 = Normal'
WRITE(*,*) '2 = Exponencial'
WRITE(*,*) '3 = Bimodal'
CALL PERGUNTADIS(TERVB,TERV,SIMU,K1)
ENDIF
IF (INT1VB(K1).EQ.0) THEN
! Este intervalo não foi usado na dissertação
WRITE(*,*) 'Deseja calcular intervalo 1?'
WRITE(*,*) 'Sim=1, Nao=0'
CALL PERGUNTADIS(INT1VB,INT1V,SIMU,K1)
ENDIF
IF (INT2VB(K1).EQ.0) THEN
WRITE(*,*) 'Deseja calcular intervalo EPB?'
WRITE(*,*) 'Sim=1, Nao=0'
CALL PERGUNTADIS(INT2VB,INT2V,SIMU,K1)
ENDIF
IF (INT3VB(K1).EQ.0) THEN
WRITE(*,*) 'Deseja calcular intervalo PPR?'
WRITE(*,*) 'Sim=1, Nao=0'
CALL PERGUNTADIS(INT3VB,INT3V,SIMU,K1)
ENDIF
IF (MEVB(K1).EQ.0) THEN
WRITE(*,*) 'Deseja ma especificacao?'
WRITE(*,*) 'Sim=1, Nao=0'
CALL PERGUNTADIS(MEVB,MEV,SIMU,K1)

```

```

IF (MEV(K1).EQ.1) THEN
  IF (pzinMEVB(K1).EQ.0) THEN
    WRITE(*,*) 'Valor do p mal especificado:'
    CALL PERGUNTADIS(pzinMEVB,pzinMEV,SIMU,K1)
  ENDIF
  IF (qzinMEVB(K1).EQ.0) THEN
    WRITE(*,*) 'Valor do q mal especificado:'
    CALL PERGUNTADIS(qzinMEVB,qzinMEV,SIMU,K1)
  ENDIF
ENDIF
ENDIF
ENDIF
IF (MEV(K1).EQ.0) THEN
  pzinMEV(K1)=pzinV(K1)
  qzinMEV(K1)=qzinV(K1)
ENDIF
1 CONTINUE
T1=CPSEC()
C ISEED = 2134569
ISEED = 2134569
CALL RNSET(ISEED)
KT=499
DO 3 K1=1,SIMU
C SIMULACAO
N=Nv(K1)
NB=NBv(K1)
NT=NTv(K1)
Fi=FiV(K1)
pzin=pzinV(K1)
Teta=TetaV(K1)
qzin=qzinV(K1)
D=DV(K1)
TER=TERV(K1)
QTESTI=0
IF (GPHPV(K1).EQ.1) THEN
  QTESTI=QTESTI+1
  ESTIMP(QTESTI)=1
ENDIF
IF (SPRPV(K1).EQ.1) THEN
  QTESTI=QTESTI+1
  ESTIMP(QTESTI)=2
ENDIF
IF (FTPV(K1).EQ.1) THEN
  QTESTI=QTESTI+1
  ESTIMP(QTESTI)=3
ENDIF
IF (D.EQ.0) THEN
  QTESTI=QTESTI+1
  ESTIMP(QTESTI)=4
ENDIF
QTESTIV(K1)=QTESTI
DO K2=1,QTESTI

```

```

ESTIMPV(K1,K2)=ESTIMP(K2)
ENDDO
QTPASSOS=QTPASSOSV(K1)
DO K2=1,QTPASSOS
  PASSOS(K2)=PASSOSV(K1,K2)
ENDDO
PASSOMAX=PASSOS(QTPASSOS)
CIP=CIPV(K1)
ME=MEV(K1)
pzinME=pzinMEV(K1)
IF (pzinME.EQ.1) THEN
  VetOrdAR(1)=1
ELSE
  VetOrdAR(1)=0
ENDIF
qzinME=qzinMEV(K1)
IF (qzinME.EQ.1) THEN
  VetOrdMA(1)=1
ELSE
  VetOrdMA(1)=0
ENDIF
INT1=INT1V(K1)
INT2=INT2V(K1)
INT3=INT3V(K1)
INT1E2(1)=1
INT1E2(2)=INT1
INT1E2(3)=INT2
INT1E2(4)=INT3
INT1E2V(K1,1)=1
INT1E2V(K1,2)=INT1
INT1E2V(K1,3)=INT2
INT1E2V(K1,4)=INT3
IF(pzinME.EQ.0.AND.qzinME.EQ.0) THEN
  ALFA=0.80D0
ELSE
  ALFA=0.50D0
ENDIF
IALFA2=NT**ALFA
IALFA=IFIX(IALFA2)
NN=NT+PASSOMAX+KT+pzin+qzin
NN2=NT+KT+pzin+qzin
NN3=KT+pzin+qzin
ZN(1)=DNORIN(0.5-CIP/2)
ZN(2)=DNORIN(0.5+CIP/2)
CALL FATD(1,D,NN-1,DF)
CALL ENCAR(1,DF,Fi,ARF,NN-1)
CALL ENCPSI(1,ARF,Teta,NN-1,PSIS)
CALL DPPREV(PSIS,PASSOS,QTPASSOS,DPPSI)
K2=0
CALERT(K1)=0
DO 4 WHILE (K2.LT.N)

```

```

C   MONTE CARLO
K2=K2+1
CALL IID(NN2,PASSOMAX,TER,R,GREB)
DO K3=1,PASSOMAX
  R(NN2+K3)=0
ENDDO
CALL SIMUNEST(1,R,ARF,NN,NN3,Teta,X1)
DO K3=1,QTPASSOS
  VESP(K3)=X1(1,NT+PASSOS(K3))
ENDDO
IF (TER.GT.1) THEN
  CALL SIMUNESTFUT(1,R,GREB,PSIS,QTPASSOS,PASSOS,NN2,GEB)
ENDIF
IF (ESTIMP(1).LT.4) THEN
  CALL SIMUCOV2(1,X1,NT,SCOV)
ENDIF
DO 5 K3=1,QTESTI
  TIPEST=ESTIMP(K3)
  IF (TIPEST.LT.3) THEN
    CALL SIMUSPE(1,SCOV,NT,IALFA,TIPEST,EST(K3))
  ENDIF
  IF (TIPEST.EQ.3) THEN
    CALL FTQU4(1,NT,SCOV,pzinME,qzinME,EST(K3),FIEST(K3),
+   TETAEST(K3))
    TETAEST(K3)=-TETAEST(K3)
  ENDIF
  IF (TIPEST.LT.4) THEN
    CALL FATD(K3,EST(K3),NT+PASSOMAX-1,DFMC)
    CALL SerieDif(1,K3,X1,NT,DFMC,DIF)
  ELSE
    DO K4=1,NT
      DIF(K3,K4)=X1(1,K4)
    ENDDO
  ENDIF
  IF(pzinME+qzinME.GT.0) THEN
    IF (TIPEST.LT.3.OR.TIPEST.EQ.4) THEN
      DO K4=1,NT
        W(K4)=DIF(K3,K4)
      ENDDO
      FIEST(K3)=0.0
      TETAEST(K3)=0.0
      CALL DNSLSE(NT,W,0,0,0,pzinME,FIEST(K3),VetOrdAR,qzinME,
+   TETAEST(K3),VetOrdMA,MAXBC,TOLBC,TOLSS,CNST,COV,2,NA,A,AVAR)
      TETAEST(K3)=-TETAEST(K3)
      IF (ABS(FIEST(K3)).GT.0.99) THEN
        FIEST(K3)=FIEST(K3)/ABS(FIEST(K3))*0.99
      ENDIF
      IF (ABS(TETAEST(K3)).GT.0.99) THEN
        TETAEST(K3)=TETAEST(K3)/ABS(TETAEST(K3))*0.99
      ENDIF
    ENDIF
  ENDIF

```

```

ELSE
  FIEST(K3)=0
  TETAEST(K3)=0
ENDIF
CALL SIMURESPAD(K3,DIF,NT,FIEST(K3),TETAEST(K3),RES,RESNA)
CALL DESVPADRES(K3,RES,NT,DP(K3))
IF (TIPEST.LT.4) THEN
  CALL ENCAR(K3,DFMC,FIEST(K3),ARFMC,NT+PASSOMAX-1)
ELSE
  ARFMC(K3,1)=FIEST(K3)
  DO K4=2,NT+PASSOMAX-1
    ARFMC(K3,K4)=0
  ENDDO
ENDIF
CALL ENCPSI(K3,ARFMC,TETAEST(K3),NT+PASSOMAX-1,PSISMC)
CALL ENCPI(K3,ARFMC,TETAEST(K3),NT+PASSOMAX-1,PI)
CALL PREVISAO(K3,X1,NT,PI,PSISMC,PASSOS,DP(K3),QTPASSOS,ZN,XF,
+ XFUL,K2,TESTE2)
5 CONTINUE
C BOOTSTRAP
DO 8 K4=1,QTESTI
7 FORMAT('Simulacao:',I4,' Monte Carlo:',I4,' Bootstrap:',I4)
K3=0
DO 6 WHILE (K3.LT.NB)
  K3=K3+1
  WRITE(*,7) K1,K2,K3
  CALL BOOTSTRAP(K4,RES,NN,NT,NN3,ARFMC,TETAEST(K4),YBOOT)
  TIPEST=ESTIMP(K4)
  IF (ESTIMP(1).LT.4) THEN
    CALL SIMUCOV2(K4,YBOOT,NT,SCOVNP)
  ENDIF
  CALL SIMUSPE(K4,SCOVNP,NT,IALFA,TIPEST,ESTNP(K4))
  MEB(K4,1)=(MEB(K4,1)*(K3-1)+ESTNP(K4))/K3
  IF (TIPEST.EQ.3) THEN
    CALL FTQQU4(K4,NT,SCOVNP,pzinME,qzinME,ESTNP(K4),
+ FIESTNP(K4),TETAESTNP(K4))
    TETAESTNP(K4)=-TETAESTNP(K4)
    MEB(K4,1)=(MEB(K4,1)*(K3-1)+ESTNP(K4))/K3
    MEB(K4,2)=(MEB(K4,2)*(K3-1)+FIESTNP(K4))/K3
    MEB(K4,3)=(MEB(K4,3)*(K3-1)+TETAESTNP(K4))/K3
  ENDIF
  IF (TIPEST.LT.4) THEN
    CALL FATD(K4,ESTNP(K4),NT+PASSOMAX-1,DFNP)
    CALL SerieDif(K4,K4,YBOOT,NT,DFNP,DIFNP)
  ELSE
    DO K5=1,NT
      DIFNP(K4,K5)=YBOOT(K4,K5)
    ENDDO
  ENDIF
  IF (pzinME+qzinME.GT.0) THEN
    IF (TIPEST.LT.3.OR.TIPEST.EQ.4) THEN

```

```

DO K5=1,NT
  W(K5)=DIFNP(K4,K5)
ENDDO
FIESTNP(K4)=0
TETAESTNP(K4)=0
CALL DNSLSE(NT,W,0,0,0,pzinME,FIESTNP(K4),VetOrdAR,qzinME,
+ TETAESTNP(K4),VetOrdMA,MAXBC,TOLBC,TOLSS,CNST,COV,2,NA,A,
+ AVAR)
TETAESTNP(K4)=-TETAESTNP(K4)
MEB(K4,2)=(MEB(K4,2)*(K3-1)+FIESTNP(K4))/K3
MEB(K4,3)=(MEB(K4,3)*(K3-1)+TETAESTNP(K4))/K3
IF (ABS(FIESTNP(K4)).GT.0.99) THEN
  FIESTNP(K4)=FIESTNP(K4)/ABS(FIESTNP(K4))*0.99
ENDIF
IF (ABS(TETAESTNP(K4)).GT.0.99) THEN
  TETAESTNP(K4)=TETAESTNP(K4)/ABS(TETAESTNP(K4))*0.99
ENDIF
ENDIF
ELSE
  FIESTNP(K4)=0
  TETAESTNP(K4)=0
ENDIF
CALL SIMURESPAD(K4,DIFNP,NT,FIESTNP(K4),TETAESTNP(K4),RESNP,
+ RESNPNA)
CALL DESVPADRES(K4,RESNP,NT,DPNP(K4))
MEB(K4,4)=(MEB(K4,4)*(K3-1)+DPNP(K4))/K3
IF (TIPEST.LT.4) THEN
  CALL ENCAR(K4,DFNP,FIESTNP(K4),ARFNP,NT+PASSOMAX-1)
ELSE
  ARFNP(K4,1)=FIESTNP(K4)
  DO K5=2,NT+PASSOMAX-1
    ARFNP(K4,K5)=0
  ENDDO
ENDIF
CALL ENCPI(K4,ARFNP,TETAESTNP(K4),NT+PASSOMAX-1,PINP)
PSSM=PASSOS(QTPASSOS)
CALL RNUND(PSSM,NT,KVET)
DO K5=1,PSSM
  RES2(K5)=RES(K4,KVET(K5))
ENDDO
IF (INT1.EQ.1) THEN
  CALL PREVISABOOT1(K4,K3,X1,NT,PINP,PASSOS,QTPASSOS,RES2,
+ XFNP)
ENDIF
IF (INT2.EQ.1) THEN
  CALL PREVISABOOT2(K4,K3,YBOOT,NT,PINP,PASSOS,QTPASSOS,XF,
+ XFNP)
ENDIF
IF (INT3.EQ.1) THEN
  !PRR
  CALL PREVISABOOT3(K4,K3,X1,NT,ARFNP,TETAESTNP(K4),PASSOS,

```

```

+ QTPASSOS,RESNA,RES2,XFNP)
  ENDIF
6  CONTINUE
8  CONTINUE
!WRITE(*,*) 'CORRECAO DE VICIO'
DO K3=1,3
DO K4=1,4
  VICIO(K3,K4)=0
  ESTBC(K3)=0
  FIBC(K3)=0
  TETABC(K3)=0
  DPBC(K3)=0
ENDDO
ENDDO
! Parte de Correcao de Vicio
DO K3=1,QTESTI
  TIPEST=ESTIMP(K3)
  IF (ABS(D).GT.0) THEN
    VICIO(K3,1)=MEB(K3,1)-EST(K3)
    ESTBC(K3)=EST(K3)-VICIO(K3,1)
    IF (ABS(ESTBC(K3)).GT.0.49) THEN
      ESTBC(K3)=ESTBC(K3)/ABS(ESTBC(K3))*0.49
    ENDIF
  ENDIF
  IF (ABS(Fi).GT.0) THEN
    VICIO(K3,2)=MEB(K3,2)-FIEST(K3)
    FIBC(K3)=FIEST(K3)-VICIO(K3,2)
    IF (ABS(FIBC(K3)).GT.0.99) THEN
      FIBC(K3)=FIBC(K3)/ABS(FIBC(K3))*0.99
    ENDIF
  ENDIF
  IF (ABS(Teta).GT.0) THEN
    VICIO(K3,3)=MEB(K3,3)-TETAEST(K3)
    TETABC(K3)=TETAEST(K3)-VICIO(K3,3)
    IF (ABS(TETABC(K3)).GT.0.99) THEN
      TETABC(K3)=TETABC(K3)/ABS(TETABC(K3))*0.99
    ENDIF
  ENDIF
  VICIO(K3,4)=MEB(K3,4)-DP(K3)
  DPBC(K3)=DP(K3)-VICIO(K3,4)
!Novas Estimativas - Fim
!Previsao com Novas Estimativas - Monte Carlo
IF (TIPEST.LT.4) THEN
  CALL FATD(K3,ESTBC(K3),NT+PASSOMAX-1,DFMC)
  CALL SerieDif(1,K3,X1,NT,DFMC,DIF)
ELSE
  DO K4=1,NT
    DIF(K3,K4)=X1(1,K4)
  ENDDO
ENDIF
CALL SIMURESPAD(K3,DIF,NT,FIBC(K3),TETABC(K3),RES,RESNA)

```



```

CALL DESVPADRES(K3,RES,NT,DP(K3))
DO K4=1,NT
  RES(K3,K4)=RES(K3,K4)*DPBC(K3)/DP(K3)
ENDDO
IF (TIPEST.LT.4) THEN
  CALL ENCAR(K3,DFMC,FIBC(K3),ARFMC,NT+PASSOMAX-1)
ELSE
  ARFMC(K3,1)=FIBC(K3)
  DO K4=2,NT+PASSOMAX-1
    ARFMC(K3,K4)=0
  ENDDO
ENDIF
CALL ENCPSI(K3,ARFMC,TETABC(K3),NT+PASSOMAX-1,PSISMC)
CALL ENCPI(K3,ARFMC,TETABC(K3),NT+PASSOMAX-1,PI)
CALL PREVISAO(K3,X1,NT,PI,PSISMC,PASSOS,DPBC(K3),QTPASSOS,ZN,
+ XFBC,XFULBC,K2,TESTE2)
!Previsao com Novas Estimativas - Monte Carlo – Fim
K4=0
DO 10 WHILE (K4.LT.NB)
  K4=K4+1
  WRITE(*,7) K1,K2,K4
  ALERTB=0
  CALL BOOTSTRAP(K3,RES,NN,NT,NN3,ARFMC,TETABC(K3),YBOOT)
  TIPEST=ESTIMP(K3)
  IF (ESTIMP(1).LT.4) THEN
    CALL SIMUCOV2(K3,YBOOT,NT,SCOVNP)
  ENDIF
  IF (TIPEST.LT.3) THEN
    CALL SIMUSPE(K3,SCOVNP,NT,IALFA,TIPEST,ESTNP(K3))
    ESTNP(K3)=ESTNP(K3)-VICIO(K3,1)
    IF (ABS(ESTNP(K3)).GT.0.49) THEN
      ESTNP(K3)=ESTNP(K3)/ABS(ESTNP(K3))*0.49
    ENDIF
  ENDIF
  IF (TIPEST.EQ.3) THEN
    CALL FTQQU4(K3,NT,SCOVNP,pzinME,qzinME,ESTNP(K3),
+ FIESTNP(K3),TETAESTNP(K3))
    TETAESTNP(K3)=-TETAESTNP(K3)
    ESTNP(K3)=ESTNP(K3)-VICIO(K3,1)
    IF (ABS(ESTNP(K3)).GT.0.49) THEN
      ESTNP(K3)=ESTNP(K3)/ABS(ESTNP(K3))*0.49
    ENDIF
    FIESTNP(K3)=FIESTNP(K3)-VICIO(K3,2)
    IF (ABS(FIESTNP(K3)).GT.0.99) THEN
      FIESTNP(K3)=FIESTNP(K3)/ABS(FIESTNP(K3))*0.99
    ENDIF
    TETAESTNP(K3)=TETAESTNP(K3)-VICIO(K3,3)
    IF (ABS(TETAESTNP(K3)).GT.0.99) THEN
      TETAESTNP(K3)=TETAESTNP(K3)/ABS(TETAESTNP(K3))*0.99
    ENDIF
  ENDIF
ENDIF

```

```

IF (TIPEST.LT.4) THEN
CALL FATD(K3,ESTNP(K3),NT+PASSOMAX-1,DFNP)
CALL SerieDif(K3,K3,YBOOT,NT,DFNP,DIFNP)
ELSE
DO K5=1,NT
DIFNP(K3,K5)=YBOOT(K3,K5)
ENDDO
ENDIF
IF(pzinME+qzinME.GT.0) THEN
IF (TIPEST.LT.3.OR.TIPEST.EQ.4) THEN
DO K5=1,NT
W(K5)=DIFNP(K3,K5)
ENDDO
FIESTNP(K3)=0
TETAESTNP(K3)=0
CALL DNSLSE(NT,W,0,0,0,pzinME,FIESTNP(K3),VetOrdAR,qzinME,
+ TETAESTNP(K3),VetOrdMA,MAXBC,TOLBC,TOLSS,CNST,COV,2,NA,A,
+ AVAR)
TETAESTNP(K3)=-TETAESTNP(K3)
FIESTNP(K3)=FIESTNP(K3)-VICIO(K3,2)
IF (ABS(FIESTNP(K3)).GT.0.99) THEN
FIESTNP(K3)=FIESTNP(K3)/ABS(FIESTNP(K3))*0.99
ENDIF
TETAESTNP(K3)=TETAESTNP(K3)-VICIO(K3,3)
IF (ABS(TETAESTNP(K3)).GT.0.99) THEN
TETAESTNP(K3)=TETAESTNP(K3)/ABS(TETAESTNP(K3))*0.99
ENDIF
ENDIF
ELSE
FIESTNP(K3)=0
TETAESTNP(K3)=0
ENDIF
CALL SIMURESPAD(K3,DIFNP,NT,FIESTNP(K3),TETAESTNP(K3),RESNP,
+ RESNPNA)
IF (TIPEST.LT.4) THEN
CALL ENCAR(K3,DFNP,FIESTNP(K3),ARFNP,NT+PASSOMAX-1)
ELSE
ARFNP(K3,1)=FIESTNP(K3)
DO K5=2,NT+PASSOMAX-1
ARFNP(K3,K5)=0
ENDDO
ENDIF
CALL ENCPI(K3,ARFNP,TETAESTNP(K3),NT+PASSOMAX-1,PINP)
PSSM=PASSOS(QTPASSOS)
CALL RNUND(PSSM,NT,KVET)
DO K5=1,PSSM
RES2(K5)=RES(K3,KVET(K5))
ENDDO
IF (INT1.EQ.1) THEN
CALL PREVISABOOT1(K3,K4,X1,NT,PINP,PASSOS,QTPASSOS,RES2,
+ XFNPPBC)

```

```

ENDIF
IF (INT2.EQ.1) THEN
  CALL PREVISABOOT2(K3,K4,YBOOT,NT,PINP,PASSOS,QTPASSOS,XFBC,
+  XFPNBC)
ENDIF
IF (INT3.EQ.1) THEN
  CALL PREVISABOOT3(K3,K4,X1,NT,ARFNP,TETAESTNP(K3),PASSOS,
+  QTPASSOS,RESNA,RES2,XFPNBC)
ENDIF
10  CONTINUE
ENDDO
! Parte de Correcao de Vicio - Fim
INF=INT((NB+1)*(0.5-CIP/2))
IF (INF.EQ.0) THEN
  INF=INF+1
ENDIF
SUP=INT((NB+1)*(0.5+CIP/2))
IF ((INT1+INT2+INT3).GT.0) THEN
  CALL ORDENACAO2(QTESTI,INT1E2,QTPASSOS,NB,INF,SUP,XFP,XFUL)
  CALL ORDENACAO2(QTESTI,INT1E2,QTPASSOS,NB,INF,SUP,XFPNBC,
+  XFULBC)
ENDIF
CALL INTERVALO(K1,TER,QTESTI,INT1E2,QTPASSOS,VESP,DPPI,N,GEB,
+  XFUL,NAOCOB,LIMITE,EQM,CIP,TESTE,K2)
CALL INTERVALO(K1,TER,QTESTI,INT1E2,QTPASSOS,VESP,DPPI,N,GEB,
+  XFULBC,NAOCOBBC,LIMITEBC,EQMBC,CIP,TESTE,K2)
4  CONTINUE
3  CONTINUE
CALL EXPORTACAO(NAOCOB,NAOCOBBC,LIMITE,LIMITEBC,EQM,EQMBC,SIMU,
+pzinV,qzinV,DV,FiV,TetaV,TERV,PASSOSV,QTPASSOSV,INT1E2V,ESTIMPV,
+QTESTIV,CIPV,CALERT,CALERTBM,CONTMC,NV,NBV,NTV)
T2=CPSEC()
TEMPOTOTAL=(T2-T1)/60.0D0
STOP
END

```

! Gera a série bootstrap

```

SUBROUTINE BOOTSTRAP(T,RES,NN,NT,NN3,AR,TETA,Y)
IMPLICIT NONE
REAL(8) RES(3,1100),AR(3,1100),TETA,Y(3,1100),RES2(1100)
REAL(8) Y2(3,1100)
INTEGER T,NN,NN3,KVET(1100),NT,K
CALL RNUND(NN,NT,KVET)
DO K=1,NN
  RES2(K)=RES(T,KVET(K))
ENDDO
CALL SIMUNEST(T,RES2,AR,NN,NN3,TETA,Y2)
DO K=1,NN-NN3
  Y(T,K)=Y2(T,K)
ENDDO
END

```

! Cálcula o desvio padrão dos resíduos

```
SUBROUTINE DESVPADRES(T,RES,N,DP)
IMPLICIT NONE
REAL(8) RES(3,1100),DP,GUS
INTEGER T,N,K
DP=0
DO K=1,N
  DP=(DP*(K-1)+RES(T,K)**2)/K
ENDDO
DP=SQRT(DP)
END
```

! Calcula o desvio padrão do valor futuro

```
SUBROUTINE DPPREV(P SIS,PSS,QTPSS,DP2)
IMPLICIT NONE
REAL(8) DP(100),DP2(100),P SIS(3,1100),GUS
INTEGER K1,PSSM,QTPSS,PSS(100)
PSSM=PSS(QTPSS)
DP(1)=1
DO K1=2,PSSM
  DP(K1)=DP(K1-1)+P SIS(1,K1-1)**2
ENDDO
DO K1=2,PSSM
  DP(K1)=SQRT(DP(K1))
ENDDO
DO K1=1,QTPSS
  DP2(K1)=DP(PSS(K1))
ENDDO
END
```

! Encontra os termos de  $\varphi(B)$

```
SUBROUTINE ENCAR(T,DF,FI,AR,N)
IMPLICIT NONE
REAL(8) DF(3,1100),FI,AR(3,1100)
INTEGER K,N,T
DO K=1,1100
  AR(T,K)=0
ENDDO
AR(T,1)=DF(T,1)+FI
DO K=2,N
  AR(T,K)=DF(T,K)-FI*DF(T,K-1)
ENDDO
END
```

! Encontra a forma autorregressiva pura

```
SUBROUTINE ENCPI(T,AR,TETA,N,PI)
IMPLICIT NONE
REAL(8) AR(3,1100),TETA,PI(3,1100)
INTEGER T,N,K1
DO K1=1,1100
```

```

PI(T,K1)=0
ENDDO
PI(T,1)=TETA+AR(T,1)
DO K1=2,N
  PI(T,K1)=AR(T,K1)-PI(T,K1-1)*TETA
ENDDO
END

```

! Encontra a forma de médias móveis pura

```

SUBROUTINE ENCPSI(T,ARF,TETA,N,PSIS)
IMPLICIT NONE
REAL(8) TETA,ARF(3,1100),PSIS(3,1100),GUS
INTEGER K1,K2,N,T
DO K1=1,1100
  PSIS(T,K1)=0
ENDDO
PSIS(T,1)=TETA+ARF(T,1)
DO K1=2,N
  PSIS(T,K1)=ARF(T,K1)
  DO K2=1,K1-1
    PSIS(T,K1)=PSIS(T,K1)+ARF(T,K2)*PSIS(T,K1-K2)
  ENDDO
ENDDO
END

```

! Imprime os resultados

```

SUBROUTINE EXPORTACAO(NAOCOB,NAOCOBBC,LIMITE,LIMITEBC,EQM,EQMBC,
+SIMU,pzinV,qzinV,DV,FiV,TetaV,TERV,PSSV,QTPSSV,INT1E2V,ESTIMPV,
+QTESTIV,CIPV,CALERT,CALERTBM,CONTMC,N,NB,NT)
IMPLICIT NONE
REAL(8) NAOCOB(30,2,3,4,100),LIMITE(30,2,3,4,100),FiV(30)
REAL(8) TetaV(30),GUS,CIPV(30),CALERTBM(30,3),DV(30)
REAL(8) NAOCOBBC(30,2,3,4,100),LIMITEBC(30,2,3,4,100)
REAL(8) EQM(2,30,3,4,100),EQMBC(2,30,3,4,100)
INTEGER SIMU,pzinV(30),qzinV(30),PSSV(30,100),QTPSSV(30)
INTEGER INT1E2V(30,4),ESTIMPV(30,3),K1,K2,K3,K4,TERV(30)
INTEGER QTESTIV(30),CALERT(30),CALERTB(30,3,1100),N(30),NB(30)
INTEGER NT(30),CONTMC(30,4,3)
DO K1=1,SIMU
  WRITE(*,*)
  WRITE(*,*) 'Simulacao:'
  WRITE(*,*) K1
  WRITE(*,*)
  WRITE(*,*) 'Modelo:'
1  FORMAT('Arfima(',I1,',',d,',',I2,')')
  WRITE(*,1) pzinV(K1),qzinV(K1)
2  FORMAT(F8.4)
  WRITE(*,*) 'D, Fi e Teta'
  WRITE(*,2) DV(K1)
  WRITE(*,2) FiV(K1)
  WRITE(*,2) TetaV(K1)

```

```

WRITE(*,*)
WRITE(*,*) 'Numero de replicacoes Monte Carlo:'
WRITE(*,*) N(K1)
WRITE(*,*) 'Numero de Replicacoes Bootstrap:'
WRITE(*,*) NB(K1)
WRITE(*,*) 'Tamanho da Serie:'
WRITE(*,*) NT(K1)
WRITE(*,*)
WRITE(*,*) 'Tipo de erro:'
WRITE(*,*) '1=Normal, 2=Exponencial, 3=Bimodal'
WRITE(*,*) TERV(K1)
WRITE(*,*)
WRITE(*,*) 'Taxa de cobertura do intervalo de previsao:'
WRITE(*,2) CIPV(K1)
3  FORMAT(' ',F6.4,' ',F6.4,' ',F6.4,' ',F7.4)
WRITE(*,*) 'Sem correcao bootstrap de vicio:'
DO K2=1,QTPSSV(K1)
WRITE(*,*) 'Passos:',PSSV(K1,K2)
DO K3=1,4
WRITE(*,*) 'Intervalo:',K3
IF (INT1E2V(K1,K3).EQ.1) THEN
DO K4=1,QTESTIV(K1)
WRITE(*,*) 'Estimador:',ESTIMPV(K1,K4)
WRITE(*,3) NAOCOB(K1,2,K4,K3,K2)-NAOCOB(K1,1,K4,K3,K2),
+ NAOCOB(K1,1,K4,K3,K2),1-NAOCOB(K1,2,K4,K3,K2),
+ LIMITE(K1,2,K4,K3,K2)-LIMITE(K1,1,K4,K3,K2)
ENDDO
ENDIF
ENDDO
ENDDO
WRITE(*,*)
WRITE(*,*) 'Copie o que for de interesse antes de entrar com'
WRITE(*,*) 'qualquer valor'
READ(*,*) GUS
WRITE(*,*) 'Com correcao bootstrap de vicio:'
DO K2=1,QTPSSV(K1)
WRITE(*,*) 'Passos:',PSSV(K1,K2)
DO K3=1,4
WRITE(*,*) 'Intervalo:',K3
IF (INT1E2V(K1,K3).EQ.1) THEN
DO K4=1,QTESTIV(K1)
WRITE(*,*) 'Estimador:',ESTIMPV(K1,K4)
WRITE(*,3) NAOCOBBC(K1,2,K4,K3,K2)-NAOCOBBC(K1,1,K4,K3,K2),
+ NAOCOBBC(K1,1,K4,K3,K2),1-NAOCOBBC(K1,2,K4,K3,K2),
+ LIMITEBC(K1,2,K4,K3,K2)-LIMITEBC(K1,1,K4,K3,K2)
4  FORMAT(F10.4)
ENDDO
ENDIF
ENDDO
ENDDO
WRITE(*,*)

```

```

WRITE(*,*) 'Copie o que for de interesse antes de entrar com'
WRITE(*,*) 'qualquer valor'
READ(*,*) GUS
WRITE(*,*)
ENDDO
5 FORMAT(' D: ',I4,' Fi: ',I4,' Teta: ',I4)
END

```

! Encontra os fatores de  $(1 - B)^d$

```

SUBROUTINE FATD(T,D,N,DF)
IMPLICIT NONE
REAL(8) D,DF(3,1100),GUS
INTEGER N,K,T
DO K=1,1100
DF(T,K)=0
ENDDO
DF(T,1)=D
DO K=2,N
DF(T,K)=DF(T,K-1)*((K-1-D)/K)
ENDDO
END

```

! Estimação de Fox-Taqqu

```

SUBROUTINE FTQQU4(T,TT,SCOV,P,Q,DW,AR,MA)
IMPLICIT NONE
REAL(8) SCOV(3,0:1100),SCOV1(0:1000),XP(5),XGUESS(5),XLB(5)
REAL(8) XUB(5),XSCALE(5), LIKE, RPARAM(7),FSCALE,DW
REAL(8) AR,MA,PERI(1000),PI,W(1000),PERI1(1000),GUS
INTEGER TT,TT1,P,P1,Q,Q1,RE,IPARAM(7),I,REA,ITB,CONT,NPQ,NP,J,K,V
INTEGER T
EXTERNAL DBCONF,WHITTLE4,ERSET
COMMON /BLOC1/SCOV1
COMMON /BLOC2/PERI1
COMMON /SIZE/TT1,P1,Q1,REA
PI=3.141592D0
NPQ=P+Q
CONT = 1
NP = P+Q+1
TT1=TT
P1=P
Q1=Q
c COVARIANCES SAMPLES FROM SECOV2
DO 5 I=0,TT-1
SCOV1(I)=SCOV(T,I)
5 CONTINUE !DO 5 I
XGUESS(1)= 0.0
XSCALE(1)=1.0
XLB(1)=-0.49
XUB(1)=0.49
FSCALE=1.0
IPARAM(1)=0

```

```

ITB = 0
IF(NPQ.GE.1) THEN
DO 10 I=2,NP
XGUESS(I)=0.0
XLB(I)=-0.99
XUB(I)=0.99
XSCALE(I)=1.0
XP(I)= 0.0
10 CONTINUE !DO 10 I
END IF ! NPQ =1
CALL ERSET(0,1,0)
DO K=1, INT((TT-1)/2 )
PERI(K)=0.0
DO 22 J=1,TT-1
W(K)=(2*PI*K)/TT
PERI(K)=PERI(K)+SCOV1(J)*COS(J*W(K))
22 CONTINUE !DO 22J
PERI(K)=(SCOV1(0)+2*PERI(K))/(2*PI)
ENDDO
DO K=1, INT((TT-1)/2 )
PERI1(K)=PERI(K)
ENDDO
CALL DBCONF(WHITTLE4,NP,XGUESS,ITB,XLB,XUB,XSCALE,FSCALE,
& IPARAM,RPARAM,XP,LIKE)
DW=XP(1)
1 FORMAT(F8.4)
IF (P.EQ.0) THEN
AR=0
MA=XP(2)
ELSE IF (Q.EQ.0) THEN
AR=XP(2)
MA=0
ELSE
AR=XP(2)
MA=XP(3)
ENDIF
RETURN
END
C SUBROUTINE WHITTLE4
SUBROUTINE WHITTLE4(NP,XP,LIKE)
REAL(8) SCOV(0:1000),SCOV1(0:1000),XP(NP),LIKE,PI,LIKE1
REAL(8) AR(2),MA(2),LIKE2,SPEC3
REAL(8) W(1000),SPEC1,SPEC2,SPEC(1000),PERI1(1000)
INTEGER TT,PW,QW,NP,J,K,TT1,P1,Q1,RE,REA,NP1,NPQ
COMMON/BLOC1/SCOV1
COMMON/BLOC2/PERI1
COMMON/SIZE/TT1,P1,Q1,REA
c COMMON VARIABLES BETWEEN PERIODO AND WHITTLE4
c COMMON/BLOCO2/LPERIO
PI=3.1416
RE = REA

```



```

TT=TT1
PW=P1
QW=Q1
NP1 = PW + QW + 1
NPQ = PW+QW
DO 6 J=0, TT-1
  SCOV(J)=SCOV1(J)
6  CONTINUE
DO 10 J=1,2
  AR(J)=0.0
  MA(J)=0.0
10 CONTINUE
IF(NPQ.GE.1) THEN
  IF (PW.EQ.0) THEN
    DO 11 J=2, NP1
      MA(J-1)=XP(J)
11  CONTINUE
    ENDF
    IF (PW.EQ.1) THEN
      AR(1)=XP(2)
      DO 12 J=3, NP1
        MA(J-2)=XP(J)
12  CONTINUE
      ENDF
      IF (PW.EQ.2) THEN
        AR(1)=XP(2)
        AR(2)=XP(3)
        DO 13 J=4, NP1
          MA(J-3)=XP(J)
13  CONTINUE
        ENDF
      END IF
      LIKE1 = 0.0
      LIKE2 = 0.0
      DO 20 K=1, INT((TT-1)/2 )
        W(K)=(2*PI*K)/TT
        SPEC1=1+MA(1)**2+MA(2)**2-2*MA(1)*(1-MA(2))*COS(W(K))
        SPEC1=SPEC1-2*MA(2)*COS(2*W(K))
        SPEC2=1+AR(1)**2+AR(2)**2-2*AR(1)*(1-AR(2))*COS(W(K))
        SPEC2=SPEC2-2*AR(2)*COS(2*W(K))
        SPEC3=(SPEC1)/(2*PI*SPEC2)
        SPEC(K)=SPEC3*((2*SIN(W(K)/2))**((-2)*XP(1)))
        LIKE1=LIKE1 +( PERI1(K)/SPEC(K))
        LIKE2=LIKE2 + LOG(SPEC(K))
20 CONTINUE
      LIKE=2*( LIKE1 + LIKE2)/TT
      RETURN
    END

```

! Gerar erros aleatórios para os valores futuros  
 SUBROUTINE IID(NN2,PM,TER,R,GREB)

```

IMPLICIT NONE
REAL(8) R(1100),GREB(1000,30),R2(1100)
INTEGER NN2,PM,TER,K1,K2,KVET(1100)
IF (TER.EQ.1) THEN
  CALL DRNNOR(NN2,R)
ELSE IF (TER.EQ.2) THEN
  CALL DRNEXP(NN2,R)
  DO K1=1,NN2
    R(K1)=R(K1)-1
  ENDDO
  DO K1=1,1000
    CALL DRNEXP(PM,R2)
    DO K2=1,PM
      GREB(K1,K2)=R2(K2)-1
    ENDDO
  ENDDO
ELSE
  CALL DRNNOR(NN2,R)
  CALL RNUND(NN2,10,KVET)
  DO K1=1,NN2
    IF (KVET(K1).EQ.10) THEN
      R(K1)=(R(K1)+9)/SQRT(10.0D0)
    ELSE
      R(K1)=(R(K1)-1)/SQRT(10.0D0)
    ENDIF
  ENDDO
  DO K1=1,1000
    CALL DRNNOR(PM,R2)
    CALL RNUND(PM,10,KVET)
    DO K2=1,PM
      IF (KVET(K2).EQ.10) THEN
        GREB(K1,K2)=(R2(K2)+9)/SQRT(10.0D0)
      ELSE
        GREB(K1,K2)=(R2(K2)-1)/SQRT(10.0D0)
      ENDIF
    ENDDO
  ENDDO
ENDIF
END

```

! Calcula cobertura e tamanho médios dos intervalos

```

SUBROUTINE INTERVALO(SM,TER,QTESTI,INT1E2,QTPSS,VESP,DP,N,GEB,
+XFUL,NAOCOB,LIMITE,EQM,CIP,TESTE,KMC)
IMPLICIT NONE
REAL(8) LIMITE(30,2,3,4,100),XFUL(2,3,4,100),VAL,ZN(2),XF(3,100)
REAL(8) DP(100),VESP(100),NAOCOB(30,2,3,4,100),GEB(1000,100)
REAL(8) DNORDF,TESTE(2,1100),GUS,EQM(2,30,3,4,100),CIP,COMP(3)
REAL(8) NCT(2)
INTEGER TER,K1,K2,K3,K4,K5,QTESTI,INT1E2(4),QTPSS,N,SM,KMC
EXTERNAL DNORDF
COMP(1)=0.5-CIP/2

```

```

COMP(2)=0.5+CIP/2
IF (TER.EQ.1) THEN
DO K2=1,QTESTI ! Estimador
DO K3=1,4 ! Intervalo
IF (INT1E2(K3).EQ.1) THEN
DO K4=1,QTPSS ! Passos
DO K1=1,2 ! Superior e inferior
LIMITE(SM,K1,K2,K3,K4)=LIMITE(SM,K1,K2,K3,K4)+XFUL(K1,K2,K3,K4)/N
VAL=(XFUL(K1,K2,K3,K4)-VESP(K4))/DP(K4)
NAOCOB(SM,K1,K2,K3,K4)=NAOCOB(SM,K1,K2,K3,K4)+DNORDF(VAL)/N
ENDDO
ENDDO
ENDIF
ENDDO
ENDDO
ELSE IF (TER.GT.1) THEN
DO K2=1,QTESTI
DO K3=1,4
IF (INT1E2(K3).EQ.1) THEN
DO K4=1,QTPSS
DO K1=1,2
LIMITE(SM,K1,K2,K3,K4)=LIMITE(SM,K1,K2,K3,K4)+XFUL(K1,K2,K3,K4)/N
DO K5=1,1000
IF (XFUL(K1,K2,K3,K4).GT.GEB(K5,K4)) THEN
NAOCOB(SM,K1,K2,K3,K4)=NAOCOB(SM,K1,K2,K3,K4)+1.0/(1000.0*N)
ENDIF
ENDDO
ENDDO
ENDDO
ENDIF
ENDDO
ENDDO
ENDIF
END

```

! Ordena os resultados do bootstrap

```

SUBROUTINE ORDENACAO2(QTESTI,INT1E2,QTPSS,NB,INF,SUP,XFNP,XFUL)
IMPLICIT NONE
REAL(8) XFNP(3,2:4,100,1100),XFUL(2,3,4,100),ALT,A
INTEGER QTESTI,INT1E2(4),QTPSS,NB,INF,SUP,K1,K2,K3,K4,K5,K6
DO K1=1,QTESTI
DO K2=2,4
IF (INT1E2(K2).EQ.1) THEN
DO K3=1,QTPSS
DO K4=2,NB
K5=0
K6=0
DO WHILE (K6.EQ.0)
IF (XFNP(K1,K2,K3,K4-K5).LT.XFNP(K1,K2,K3,K4-K5-1)) THEN
A=XFNP(K1,K2,K3,K4-K5-1)
XFNP(K1,K2,K3,K4-K5-1)=XFNP(K1,K2,K3,K4-K5)

```

```

XFNP(K1,K2,K3,K4-K5)=A
K5=K5+1
IF (K4-K5-1.EQ.0) THEN
  K6=1
ENDIF
ELSE
  K6=1
ENDIF
ENDDO
ENDDO
XFUL(1,K1,K2,K3)=XFNP(K1,K2,K3,INF)
XFUL(2,K1,K2,K3)=XFNP(K1,K2,K3,SUP)
ENDDO
ENDIF
ENDDO
ENDDO
END

```

! Coleta as respostas de valor contínuo das perguntas

```

SUBROUTINE PERGUNTACON(VB,V,SIMU,K)
INTEGER VB(30),SIMU,K,QNTS,K2
REAL(8) V(30),GUS
READ(*,*) V(K)
IF (SIMU.GT.1.AND.K.LT.SIMU) THEN
  WRITE(*,*) 'Repetir esta condicao por quantas simulacoes?'
  READ(*,*) QNTS
  DO K2=(K+1),(K+QNTS-1)
    V(K2)=V(K)
    VB(K2)=1
  ENDDO
ENDIF
END

```

! Coleta as respostas de valor discreto das perguntas

```

SUBROUTINE PERGUNTADIS(VB,V,SIMU,K)
IMPLICIT NONE
INTEGER VB(30),V(30),SIMU,K,QNTS,K2
READ(*,*) V(K)
IF (SIMU.GT.1.AND.K.LT.SIMU) THEN
  WRITE(*,*) 'Repetir esta condicao por quantas simulacoes?'
  READ(*,*) QNTS
  DO K2=(K+1),(K+QNTS-1)
    V(K2)=V(K)
    VB(K2)=1
  ENDDO
ENDIF
END

```

! Previsao pontual e de Box-Jenkins

```

SUBROUTINE PREVISAO(T,X,N,PI,PSI,PSS,DP,QTPSS,ZN,XF2,XFUL,KMC,
+      TESTE2)

```

```

IMPLICIT NONE
REAL(8) XF(100),PI(3,1100),X(3,1100),PSI(3,1100),XF2(3,100),A1,DP
REAL(8) XFUL(2,3,4,100),CIP,ZN(2),GUS,TESTE2(1100)
INTEGER T,N,K1,K2,PSSM,PSS(100),QTPSS,KMC
XF(1)=PI(T,1)*X(1,N)
PSSM=PSS(QTPSS)
DO K1=2,N
  XF(1)=XF(1)+PI(T,K1)*X(1,N-K1+1)
ENDDO
DO K1=2,PSSM
  XF(K1)=PI(T,1)*XF(K1-1)
  DO K2=2,(K1-1)
    XF(K1)=XF(K1)+PI(T,K2)*XF(K1-K2)
  ENDDO
  DO K2=1,N
    XF(K1)=XF(K1)+PI(T,K1+K2-1)*X(1,N-K2+1)
  ENDDO
ENDDO
DO K1=1,QTPSS
  XF2(T,K1)=XF(PSS(K1))
ENDDO
DO K1=1,QTPSS
  A1=1
  DO K2=1,PSS(K1)-1
    A1=A1+PSI(T,K2)**2
  ENDDO
  XFUL(1,T,1,K1)=XF2(T,K1)+ZN(1)*DP*SQRT(A1)
  XFUL(2,T,1,K1)=XF2(T,K1)+ZN(2)*DP*SQRT(A1)
ENDDO
END

```

! Esta rotina é para construir um intervalo que não foi usado na dissertação  
 SUBROUTINE PREVISAOBOOT1(T,BT,X,N,PI,PSS,QTPSS,RES2,XFNP)

! Rotina para construir o intervalo EPB  
 SUBROUTINE PREVISAOBOOT2(T,BT,X,N,PI,PSS,QTPSS,XFOR,XFNP)  
 IMPLICIT NONE  
 REAL(8) XF(100),PI(3,1100),X(3,1100),XFNP(3,2:4,100,1100)  
 REAL(8) XFOR(3,100),GUS  
 INTEGER T,N,K1,K2,PSSM,PSS(100),QTPSS,BT  
 PSSM=PSS(QTPSS)  
 XF(1)=PI(T,1)\*X(T,N)  
 DO K1=2,N  
 XF(1)=XF(1)+PI(T,K1)\*X(T,N-K1+1)  
 ENDDO  
 DO K1=2,PSSM  
 XF(K1)=PI(T,1)\*XF(K1-1)  
 DO K2=2,(K1-1)  
 XF(K1)=XF(K1)+PI(T,K2)\*XF(K1-K2)  
 ENDDO  
 DO K2=1,N

```

XF(K1)=XF(K1)+PI(T,K1+K2-1)*X(T,N-K2+1)
ENDDO
ENDDO
DO K1=1,QTPSS
XFNP(T,3,K1,BT)=XFOR(T,K1)-(XF(PSS(K1))-X(T,N+PSS(K1)))
ENDDO
END

```

! Rotina para construir o intervalo PRR

```

SUBROUTINE PREVISAOBOOT3(T,BT,X,N,AR,TETA,PSS,QTPSS,RES,RES2,XFNP)
IMPLICIT NONE
REAL(8) XF(100),PI(3,1100),X(3,1100),XFNP(3,2:4,100,1100)
REAL(8) RES(3,1100),RES2(1100),TETA,AR(3,1100),GUS
INTEGER T,N,K1,K2,PSSM,PSS(100),QTPSS,BT,KVET(1100)
PSSM=PSS(QTPSS)
XF(1)=AR(T,1)*X(1,N)+TETA*RES(T,N)+RES2(1)
DO K1=2,N
XF(1)=XF(1)+AR(T,K1)*X(1,N-K1+1)
ENDDO
DO K1=2,PSSM
XF(K1)=AR(T,1)*XF(K1-1)+TETA*RES2(K1-1)+RES2(K1)
DO K2=2,(K1-1)
XF(K1)=XF(K1)+AR(T,K2)*XF(K1-K2)
ENDDO
DO K2=1,N
XF(K1)=XF(K1)+AR(T,K1+K2-1)*X(1,N-K2+1)
ENDDO
ENDDO
DO K1=1,QTPSS
XFNP(T,4,K1,BT)=XF(PSS(K1))
ENDDO
1 FORMAT(F8.4)
END

```

! Diferencia a série

```

SUBROUTINE SERIEDIF(T1,T2,X,N,DF,DX)
IMPLICIT NONE
REAL(8) X(3,1100),DX(3,1100),DF(3,1100),GUS
INTEGER N,K1,K2,T,T1,T2
DX(T2,1)=X(T1,1)
DO K1=2,N
DX(T2,K1)=X(T1,K1)
DO K2=1,(K1-1)
DX(T2,K1)=DX(T2,K1)-DF(T2,K2)*X(T1,K1-K2)
ENDDO
ENDDO
END

```

! Cálculo da função de autocorrelação

```

SUBROUTINE SIMUCOV2(T,X,N,SCOV)
IMPLICIT NONE

```

```

REAL(8) SCOV(3,0:1100),X(3,1100),GUS,GUS2
INTEGER N,K1,K2,T,MEDIA
DO K1=0,1100
  SCOV(T,K1)=0
ENDDO
MEDIA=0
DO K1=1,N
  MEDIA=MEDIA+X(T,K1)/N
ENDDO
DO K1=0,(N-1)
  DO K2=1,(N-K1)
    SCOV(T,K1)=SCOV(T,K1)+(X(T,K2)-MEDIA)*(X(T,K2+K1)-MEDIA)
  ENDDO
  SCOV(T,K1)=SCOV(T,K1)/N
ENDDO
END

```

! Geração das séries

```

SUBROUTINE SIMUNEST(T,X,AR,NN,NN3,TETA,Y2)
IMPLICIT NONE
REAL(8) X(1100),Y2(3,1100),Y1(1100),AR(3,1100),TETA,GUS,GUS2
INTEGER K1,K2,NN,NN3,T
Y1(1)=X(1)
DO K1=2,NN
  Y1(K1)=X(K1-1)*TETA+X(K1)
  DO K2=1,(K1-1)
    Y1(K1)=Y1(K1)+AR(T,K2)*Y1(K1-K2)
  ENDDO
ENDDO
DO K1=1,NN-NN3
  Y2(T,K1)=Y1(K1+NN3)
ENDDO
1  FORMAT(F10.4)
END

```

! Geração de valores futuros da séries

```

SUBROUTINE SIMUNESTFUT(T,R,GREB,PSI,QTP,PASSOS,NN2,GEB)
IMPLICIT NONE
REAL(8) R(1100),GREB(1000,100),PSI(3,1100),GEB(1000,100)
INTEGER K2,K3,QTP,PASSOS(100),NN2,T,K1
DO K1=1,1000
  DO K2=1,QTP
    GEB(K1,K2)=GREB(K1,PASSOS(K2))
    DO K3=1,(PASSOS(K2)-1)
      GEB(K1,K2)=GEB(K1,K2)+PSI(T,K3)*GREB(K1,PASSOS(K2)-K3)
    ENDDO
  DO K3=1,NN2
    GEB(K1,K2)=GEB(K1,K2)+PSI(T,K3+PASSOS(K2)-1)*R(NN2-K3+1)
  ENDDO
ENDDO
ENDDO
ENDDO

```

END

! Cálculo dos resíduos

```
SUBROUTINE SIMURESPAD(T,XD,N,FI,TETA,EA,E)
IMPLICIT NONE
REAL(8) XD(3,1100),FI,TETA,E(3,1100),EA(3,1100),TOT,GUS
INTEGER T,N,K
E(T,1)=XD(T,1)
TOT=E(T,1)
DO K=2,N
E(T,K)=XD(T,K)-XD(T,K-1)*FI-E(T,K-1)*TETA
TOT=TOT+E(T,K)
ENDDO
TOT=TOT/N
DO K=1,N
EA(T,K)=E(T,K)-TOT
ENDDO
1 FORMAT(F8.4)
END
```

! Estimação de d por GPH e SPR

```
SUBROUTINE SIMUSPE(T,SCOV,N,IALFA,TIPEST,EST)
IMPLICIT NONE
REAL(8) SCOV(3,0:1100),A1,PW(1100),W(1100),PERIO(1100),TX,TY
REAL(8) Y(1100),GUS,GUS2
REAL(8) X(1100),SXX,SXY,EST,FALDAD
INTEGER N,MBETA,IALFA,TIPEST,N2,MBETA2,K1,K2,LH,T,C,K3
FALDAD=0
N2=N/2
MBETA=IFIX(N**0.9)
MBETA2=MBETA/2
IF (TIPEST.EQ.2) THEN
DO K1=1,(N-1)
A1=REAL(K1)/REAL(MBETA)
IF (K1.LE.MBETA2) THEN
PW(K1)=1.0D0-6.0D0*(A1**2.0D0)+6.0D0*(A1**3.0D0)
ELSE IF (K1.LE.MBETA) THEN
PW(K1)=2.0D0*((1.0D0-A1)**3.0D0)
ELSE
PW(K1)=0.0D0
ENDIF
ENDDO
ELSE
DO K1=1,(N-1)
PW(K1)=1.0D0
ENDDO
ENDIF
DO K1=1,N2
W(K1)=(2.0D0*3.14160D0*K1)/N
PERIO(K1)=0
DO K2=2,N-1
```



```

PERIO(K1)=PERIO(K1)+2.0D0*PW(K2)*SCOV(T,K2)*DCOS(K2*W(K1))
ENDDO
PERIO(K1)=(PERIO(K1)+SCOV(T,0))/6.28320D0
ENDDO
LH=0
TX=0.0D0
TY=0.0D0
DO K1=1,IALFA
IF (PERIO(K1).GT.0) THEN
LH=LH+1
Y(LH)=DLOG(PERIO(K1))
X(LH)=DLOG(4*DSIN(W(K1)/2)**2)
TX=TX+X(LH)
TY=TY+Y(LH)
ENDIF
ENDDO
TX=TX/LH
TY=TY/LH
SXX=0
SXY=0
DO K1=1,LH
SXX=SXX+(X(K1)-TX)**2
SXY=SXY+(X(K1)-TX)*(Y(K1)-TY)
ENDDO
EST=-SXY/SXX
IF (ABS(EST).GT.0.49) THEN
EST=EST/ABS(EST)*0.49
ENDIF
END

```