

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

**Inferência Bayesiana para modelos de volatilidade
estocástica baseados em mistura de escala da distribuição
normal assimétrica**

Ritha Rubi Huaysara Condori

Dissertação de Mestrado do Programa Interinstitucional de
Pós-Graduação em Estatística (PIPGes)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Ritha Rubi Huaysara Condori

Inferência Bayesiana para modelos de volatilidade estocástica baseados em mistura de escala da distribuição normal assimétrica

Dissertação apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP e
ao Departamento de Estatística – DEs-UFSCar,
como parte dos requisitos para obtenção do título
de Mestra em Estatística – Programa Interinstitucional
de Pós-Graduação em Estatística. *VERSÃO
REVISADA*

Área de Concentração: Estatística

Orientador: Prof. Dr. Ricardo Sandes Ehlers

USP – São Carlos
Maio de 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C746i Condori, Ritha Rubi Huaysara
 Inferência Bayesiana para modelos de
 volatilidade estocástica baseados em mistura de
 escala da distribuição normal assimétrica / Ritha
 Rubi Huaysara Condori; orientador Ricardo Sandes
 Ehlers. -- São Carlos, 2023.
 94 p.

 Dissertação (Mestrado - Programa
 Interinstitucional de Pós-graduação em Estatística) --
 Instituto de Ciências Matemáticas e de Computação,
 Universidade de São Paulo, 2023.

 1. Modelos de volatilidade estocástica. 2.
 Mistura de escala da distribuição normal
 assimétrica. 3. Algoritmo No-U-Turn Sampler. 4.
 Leave-Future-Out Cross-Validation. 5. Efeito de
 Alavancagem. I. Ehlers, Ricardo Sandes, orient. II.
 Título.

Ritha Rubi Huaysara Condori

**Bayesian inference for stochastic volatility models based on
scale mixtures of the skew-normal distribution**

Dissertation submitted to the Institute of Mathematics
and Computer Science – ICMC-USP and to the
Department of Statistics – DEs-UFSCar – in
accordance with the requirements of the Statistics
Interagency Graduate Program, for the degree of
Master in Statistics. *FINAL VERSION*

Concentration Area: Statistics

Advisor: Prof. Dr. Ricardo Sandes Ehlers

USP – São Carlos
May 2023

*Este trabalho é dedicado com amor ao meu querido pai,
Lino Huaysara Anampa.*

AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus pela vida e pelas oportunidades que Ele me tem dado. Gostaria de expressar minha gratidão à minha mãe, Santusa Condori, por todo o apoio que me deu ao longo da minha vida, e aos meus irmãos Jack e Jackeline.

Meus agradecimentos ao meu amigo Juan Pablo, que me ajudou e aconselhou durante o mestrado, aos amigos Alex Flores e Filomen por me ajudarem com o cluster, e a Danillo e Mariella pela amizade e ajuda na correção do português da minha dissertação. Também agradeço às pessoas que conheci durante o mestrado, como Osafu, Fabiano, Asrat, Laura e ao meu bom amigo William. Muito obrigada a todos.

Gostaria de agradecer aos meus tios Leo e Flavia, que foram parte importante desta jornada, e ao meu primo Eddy.

Ao casal Lorena e Miguel, que me acolheu com gentileza quando cheguei ao Brasil, minha sincera gratidão.

Quero expressar minha profunda gratidão ao meu orientador Ricardo Sandes Ehlers pela orientação, disponibilidade, paciência e conselhos durante o mestrado. Também gostaria de agradecer aos professores Mauricio Zevallos, Carlos Abanto e Marcio Poletti pela avaliação e correções que ajudaram a melhorar este trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

RESUMO

RITHA, R. H. CONDORI. **Inferência Bayesiana para modelos de volatilidade estocástica baseados em mistura de escala da distribuição normal assimétrica**. 2023. 94 p. Dissertação (Mestrado em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Esta dissertação tem como objetivo avaliar e comparar o desempenho do algoritmo *No-U-Turn Sampler* (NUTS), implementado no software Stan, na estimação dos parâmetros de modelos de volatilidade estocástica com alavancagem baseados em mistura de escala da distribuição normal assimétrica. Esses modelos SV conseguem capturar simultaneamente características importantes das séries de retornos financeiros, como efeito de alavancagem, caudas pesadas e assimetria. Os resultados dos estudos de simulação mostram que, de acordo com as medidas de viés e raiz quadrada do erro quadrático médio (RMSE), o algoritmo NUTS apresenta um bom desempenho. Ao comparar a abordagem de amostragem do NUTS com a do pacote `stochvol`, observamos que o `stochvol` apresenta tempos de execução menores, mas o NUTS supera essa abordagem em termos de tamanho efetivo da amostra. Além disso, propomos o uso da técnica de Validação Cruzada *Leave-Future-Out* (LFO-CV) para seleção de modelos de volatilidade estocástica e avaliamos o desempenho de critérios de informação e técnicas de validação cruzada para seleção de modelos. Finalmente, aplicamos a metodologia desenvolvida a séries de retornos reais.

Palavras-chave: Modelos de volatilidade estocástica, Mistura de escala da distribuição normal assimétrica, Algoritmo No-U-Turn Sampler, Leave-Future-Out Cross-Validation, Efeito de Alavancagem.

ABSTRACT

RITHA, R. H. CONDORI. **Bayesian inference for stochastic volatility models based on scale mixtures of the skew-normal distribution**. 2023. 94 p. Dissertação (Mestrado em Estatística – Programa Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

This dissertation aims to evaluate and compare the performance of the No-U-Turn Sampler (NUTS) algorithm, implemented in the Stan software, in estimating the parameters of stochastic volatility models with leverage based on scale mixtures of the skew-normal distribution. These SV models can simultaneously capture important features of financial return series, such as leverage effect, heavy tails, and asymmetry. The results of simulation studies show that, according to bias and root mean squared error (RMSE) measures, the NUTS algorithm performs well. When comparing the NUTS sampling approach with that of the stochvol package, we observe that stochvol has faster execution times, but NUTS outperforms it in terms of effective sample size. Additionally, we propose the use of the Leave-Future-Out Cross-Validation (LFO-CV) technique for selecting stochastic volatility models and evaluate the performance of information criteria and cross-validation techniques for model selection. Finally, we apply the developed methodology to real return series.

Keywords: Scale mixtures of the skew-normal distribution, Stochastic volatility models, Algorithm No-U-Turn Sampler, Leave-Future-Out Cross-Validation, Leverage Effect.

LISTA DE ILUSTRAÇÕES

Figura 1 – Gráficos de convergência dos parâmetros do modelo SV-N ajustado a uma série simulada, usando o algoritmo NUTS e o pacote <code>stochvol</code>	49
Figura 2 – Gráficos de convergência dos parâmetros do modelo SV-NL ajustado a uma série simulada, usando o algoritmo NUTS e o pacote <code>stochvol</code>	50
Figura 3 – Gráficos de convergência dos parâmetros do modelo SV-T ajustado a uma série simulada, usando o algoritmo NUTS e o pacote <code>stochvol</code>	51
Figura 4 – Gráficos de convergência dos parâmetros do modelo SV-TL ajustado a uma série simulada, usando o algoritmo NUTS e o pacote <code>stochvol</code>	52
Figura 5 – Gráficos de convergência dos parâmetros do modelo SV-N.	56
Figura 6 – Gráficos de convergência dos parâmetros do modelo SV-NL.	56
Figura 7 – Gráficos de convergência dos parâmetros do modelo SV-T.	57
Figura 8 – Gráficos de convergência dos parâmetros do modelo SV-TL.	57
Figura 9 – Gráficos de convergência dos parâmetros do modelo SV-STL.	58
Figura 10 – Gráficos de convergência dos parâmetros do modelo SV-SSL.	58
Figura 11 – Série temporais de preços (esquerda) e retornos (direita) de S&P500, MSFT, EURUSD e BTC.	62
Figura 12 – Função de autocorrelação dos retornos (lado esquerdo) e dos quadrados dos retornos (lado direito) de S&P500, MSFT, EURUSD e BTC.	63
Figura 13 – Resultados da estimação dos retornos diários do índice S&P500 (modelo SV-STL).	68
Figura 14 – Resultados da estimação dos retornos diários MSFT (modelo SV-STL).	68
Figura 15 – Resultados da estimação dos retornos diários EURUSD (modelo SV-STL).	69
Figura 16 – Resultados da estimação dos retornos diários BTC (modelo SV-T).	69
Figura 17 – Volatilidades estimadas pelo modelo SV escolhido para as séries de retornos S&P500, MSFT, EURUSD e BTC.	70

LISTA DE TABELAS

Tabela 1 – Tempos médios em segundos para ajustar os modelos SV, utilizando o algoritmo NUTS e o pacote <code>stochvol</code>	46
Tabela 2 – Resultados da comparação do algoritmo NUTS e o pacote <code>stochvol</code> para os modelos SV-N e SV-T em termos de ESS e ESS/s.	47
Tabela 3 – Comparação dos amostradores NUTS e <code>stochvol</code> em termos de tamanho efetivo de amostra (ESS) e tamanho efetivo de amostra por segundo (ESS/s).	48
Tabela 4 – Viés e RMSE para os parâmetros na classe de modelos SV-SMSN.	54
Tabela 5 – Tempos médios em minutos para ajustar os modelos SV às séries simuladas utilizando o algoritmo NUTS.	55
Tabela 6 – Frequências de seleção de modelo para as 50 séries simuladas a partir do modelo SV-STL.	60
Tabela 7 – Frequências de seleção de modelo para as 50 séries simuladas a partir do modelo SV-TL.	60
Tabela 8 – Número médio de reajustes necessários para calcular a medida $elpdl_{fo}$	60
Tabela 9 – Estatísticas descritivas das séries de retornos diários S&P500, MSFT, EURUSD e BTC.	62
Tabela 10 – Resultados da estimação das séries de retornos diários S&P500.	64
Tabela 11 – Resultados da estimação das séries de retornos diários MSFT e EURUSD.	65
Tabela 12 – Resultados da estimação da série de retornos diários BTC.	66
Tabela 13 – Resultados de comparação de modelos SV para as séries temporais.	67
Tabela 14 – Tempo de ajuste dos modelos SV (em minutos) para as séries S&P500, MSFT, EURUSD e BTC.	70

SUMÁRIO

1	INTRODUÇÃO	19
2	PRELIMINARES	23
2.1	Séries de retornos financeiros	23
2.2	Distribuição normal assimétrica	24
2.3	Misturas de escala da distribuição normal assimétrica	24
2.3.1	<i>Distribuição t-Student assimétrica</i>	25
2.3.2	<i>Distribuição slash assimétrica</i>	26
2.4	Métodos de Monte Carlo via cadeias de Markov	26
2.4.1	<i>Algoritmo Metropolis-Hastings</i>	27
2.4.2	<i>Algoritmo Monte Carlo Hamiltoniano</i>	27
2.4.3	<i>Algoritmo No-U-Turn Sampler</i>	30
3	MODELOS DE VOLATILIDADE ESTOCÁSTICA	31
3.1	Modelos SV com alavancagem baseados em SMSN	31
3.2	Estimação Bayesiana	35
4	MÉTODOS DE SELEÇÃO DE MODELOS	39
4.1	Critério de informação de desvio	39
4.2	Validação cruzada <i>Leave-One-Out</i>	40
4.3	Critério de informação de Watanabe	42
4.4	Validação cruzada <i>Leave-Future-Out</i>	43
5	ESTUDOS DE SIMULAÇÃO E APLICAÇÕES	45
5.1	Estudo de simulação 1	45
5.2	Estudo de simulação 2	53
5.3	Estudo de simulação 3	59
5.4	Aplicação a dados reais	61
6	CONCLUSÕES	71
	REFERÊNCIAS	73
	APÊNDICE A DEMONSTRAÇÕES	79

A.1	Distribuição normal assimétrica	79
A.2	Esperança e variância da distribuição t-Student assimétrica	80
A.3	Esperança e variância da distribuição slash assimétrica	82
APÊNDICE B	CÓDIGOS	83

INTRODUÇÃO

As séries de retornos financeiros são uma ferramenta comumente utilizada na gestão de risco em finanças. Essas séries apresentam volatilidade, que é uma medida de risco que os gestores estão interessados em medir e prever para a tomada de decisão em investimentos. Isso motiva a necessidade de desenvolver modelos estadísticos para estimar a volatilidade das séries de retornos financeiros.

Entre os modelos estadísticos mais utilizados para modelar a volatilidade, temos os modelos ARCH (*Autoregressive Conditional Heterocedasticity*) proposto por (ENGLE, 1982), GARCH (*Generalized Autoregressive Conditional Heterocedasticity*) proposto por (BOLLERS-LEV, 1986), e o modelo de volatilidade estocástica (SV), do inglês *Stochastic volatility models*; proposto por Taylor (1982). Este último assume que a distribuição dos retornos, condicionada a volatilidade latente, é normal, mas pode ser modificado para incluir outras características dos retornos financeiros, como caudas pesadas, assimetria e o efeito de alavancagem.

Por exemplo, para capturar a característica dos retornos financeiros de caudas pesadas, Meyer e Yu (2000), Chib, Nardari e Shephard (2002), Omori *et al.* (2007) e Abanto-Valle *et al.* (2010) estudaram o modelo SV com distribuição t-Student. Harvey e Shephard (1996) propõem o modelo SV com distribuição normal e efeito de alavancagem. O efeito de alavancagem também foi incluído no modelo SV com distribuição t-Student, visto em, por exemplo, Meyer e Yu (2000), Jacquier, Polson e Rossi (2004), e Omori *et al.* (2007)

Nesta dissertação, estudamos os modelos de volatilidade estocástica com alavancagem baseados em misturas de escala da distribuição normal assimétrica (SV-SMSN) propostos por Huang e Xu (2014). Uma vantagem desta classe de modelos SV é que contém como casos especiais os modelo SV com distribuição normal, t-Student, t de Student assimétrica (ABANTO-VALLE; LACHOS; DEY, 2015) e Slash assimétrica.

A estimação de parâmetros em modelos SV é difícil devido à forma intratável da função de verossimilhança. Para superar essa dificuldade, diversos métodos de estimação foram

desenvolvidos, incluindo a abordagem Bayesiana por meio de Métodos de Monte Carlo via cadeias de Markov (MCMC). Por exemplo, para estimar o modelo SV com distribuição normal [Jacquier, Polson e Rossi \(1994\)](#) adotaram o algoritmo de amostragem Gibbs e [Shephard e Pitt \(1997\)](#) empregaram o algoritmo de Metropolis-Hastings. [Meyer e Yu \(2000\)](#) e [Yu \(2005\)](#) mostraram como implementar alguns modelos SV usando o software BUGS (análise Bayesiana usando o algoritmo Gibbs) ([SPIEGELHALTER et al., 1996](#)). Além disso, métodos numéricos conhecidos como aproximações integradas de Laplace (INLA) foram utilizados por [Ehlers e Zevallos \(2015\)](#) para estimar os parâmetros no modelo SV com distribuição normal e t-Student. Recentemente, [Darjus e Gregor \(2021\)](#) desenvolveram o pacote `stochvol`, que permite ajustar modelos SV com ou sem alavancagem e distribuição normal e t-Student.

Outro método MCMC, chamado Monte Carlo Hamiltoniano (HMC), introduzido por [Duane et al. \(1987\)](#), tem sido utilizado para estimar modelos SV por [Takaishi \(2009\)](#), [Abanto-Valle, Rodríguez e Garrafa-Aragón \(2021\)](#) e [Nugroho e Morimoto \(2015\)](#), entre outros. O uso do algoritmo HMC tem aumentado, especialmente, com o desenvolvimento da linguagem de programação probabilística Stan ([CARPENTER et al., 2017](#)). O Stan implementa o algoritmo *No-U-Turn Sampler* (NUTS), proposto por [Hoffman, Gelman et al. \(2014\)](#), que é uma extensão do HMC que seleciona automaticamente os parâmetros de tamanho de passo e comprimento de trajetória do algoritmo HMC.

Nesta dissertação, utilizamos o algoritmo *No-U-Turn Sampler* (NUTS) para estimar os parâmetros e as volatilidades em modelos de volatilidade estocástica com alavancagem baseados em misturas de escala da distribuição normal assimétrica (SV-SMSN). [Dias e Ehlers \(2017\)](#) também utilizaram o algoritmo NUTS para estimar modelos SV com distribuições normal, t-Student, normal assimétrica e distribuição generalizada do erro (GED). Um estudo comparativo da eficiência de diversos algoritmos MCMC para estimar o modelo SV com alavancagem e distribuição normal foi realizado por [Hosszejni e Kastner \(2018\)](#). Eles demonstraram que, com base no tamanho efetivo de amostra, o algoritmo NUTS superou os outros métodos MCMC.

Dada a ampla variedade de modelos SV, a seleção do modelo mais adequado é crucial e pode ser feita por meio de diferentes métodos, como o critério de informação de desvio-DIC ([SPIEGELHALTER et al., 2002](#)), o critério de informação de Watanabe-WAIC ([WATANABE; OPPER, 2010](#)) e a validação cruzada *Leave-One-Out*-LOO, ([VEHTARI; GELMAN; GABRY, 2017](#)). No entanto, quando os dados são ordenados sequencialmente no tempo, o LOO-CV pode não ser a melhor opção, segundo [Bürkner, Gabry e Vehtari \(2020\)](#). Eles propõem a Validação Cruzada *Leave-Future-Out* (LFO-CV). Esse método de seleção de modelos foi implementado por [Rast, Martin e Rodriguez \(2020\)](#) e [Li, Clements e Drovandi \(2021\)](#) em modelos GARCH. Neste estudo, propomos o uso do LFO-CV para selecionar modelos SV.

Este trabalho apresenta três contribuições principais. Em primeiro lugar, avaliamos o desempenho do algoritmo NUTS na estimação dos modelos SV-SMSN. Em segundo lugar, comparamos a eficiência do algoritmo NUTS com a abordagem de amostragem implementada

no pacote R `stochvol`. Por fim, avaliamos o desempenho dos métodos de seleção de modelos DIC, WAIC, LOO-CV e LFO-CV para seleção de modelos de SV.

A dissertação está organizada da seguinte maneira: no [Capítulo 2](#), são apresentados os conceitos fundamentais necessários para a compreensão da dissertação, tais como a distribuição normal assimétrica e os métodos de Monte Carlo via cadeias de Markov, incluindo o algoritmo Hamiltoniano Monte Carlo utilizado na estimação dos modelos SV-SMSN. No [Capítulo 3](#), são apresentados os modelos SV-SMSN. No [Capítulo 4](#), são descritos os métodos de seleção de modelos. No [Capítulo 5](#), são apresentados os estudos de simulação para avaliar e comparar o desempenho do algoritmo NUTS e dos métodos de seleção, bem como uma aplicação dos modelos SV-SMSN a séries de retornos reais. Finalmente, no [Capítulo 6](#), são apresentadas as conclusões e trabalhos futuros da dissertação.

PRELIMINARES

Neste capítulo, apresentamos conceitos fundamentais para o desenvolvimento do trabalho. Por exemplo, introduzimos as séries de retornos financeiros, a distribuição normal assimétrica e a família de misturas de escala da distribuição normal assimétrica. Também realizamos uma breve descrição sobre os métodos de Monte Carlo via cadeias de Markov, em particular, o método Monte Carlo Hamiltoniano.

2.1 Séries de retornos financeiros

As séries de retornos financeiros são uma série temporal de variações no valor de um ativo financeiro, que representam o lucro ou prejuízo gerado por um investimento. Nesta dissertação, utilizamos as séries de retornos em vez de séries de preços de ativos, devido às propriedades estatísticas mais interessantes dos retornos financeiros. O log retorno de um ativo financeiro, também conhecido como retorno, é calculado utilizando a seguinte fórmula:

$$y_t = \log(P_t/P_{t-1}),$$

onde P_t é o preço do ativo no tempo t . Segundo [Morettin \(2017\)](#), as séries de retornos financeiros diários apresentam os seguintes fatos estilizados:

1. Volatilidade heterocedástica: a variância dos retornos varia ao longo do tempo, apresentando momentos de alta e baixa volatilidade.
2. Em geral, os retornos não apresentam autocorrelação.
3. Os quadrados dos retornos apresentam autocorrelação.
4. Sua média amostral é próxima de zero.
5. A distribuição dos retornos financeiros diários apresenta assimetria.

6. A distribuição dos retornos financeiros diários apresenta caudas pesadas.
7. Os retornos de ativos financeiros estão negativamente correlacionados com as mudanças nas volatilidades dos retornos, um fenômeno conhecido como "efeito de alavancagem". Esse fenômeno financeiro pode ocorrer quando os preços dos ativos da empresa caem, o que pode levar a um aumento na volatilidade dos retornos (BLACK, 1976).

2.2 Distribuição normal assimétrica

A distribuição normal assimétrica foi introduzida por O'hagan e Leonard (1976) e Azzalini (1985). Se uma variável aleatória X segue uma distribuição normal assimétrica, é denotada por $X \sim \text{SN}(\eta, \omega^2, \alpha)$, com parâmetros de locação, escala e assimetria dados por η , ω^2 e α , respectivamente. Sua função densidade de probabilidade é dada por:

$$f(x) = \frac{2}{\omega} \phi\left(\frac{x - \eta}{\omega}\right) \Phi\left(\frac{\alpha(x - \eta)}{\omega}\right), \quad x \in \mathbb{R},$$

onde $\phi(\cdot)$ e $\Phi(\cdot)$ são, respectivamente, a função densidade de probabilidade (fdp) e a função de distribuição acumulada (fda) da distribuição normal padrão. Valores positivos (negativos) de α implicam assimetria positiva (negativa); quando $\alpha = 0$, a distribuição corresponde a uma normal $N(\eta, \omega^2)$.

A média e a variância de uma variável aleatória X com distribuição normal assimétrica, $X \sim \text{SN}(\eta, \omega^2, \alpha)$, são, respectivamente, dadas por:

$$\begin{aligned} E(X) &= \eta + \sqrt{\frac{2}{\pi}} \delta, \\ V(X) &= \omega^2 \left(1 - \frac{2}{\pi} \delta^2\right), \end{aligned}$$

onde $\delta = \alpha / \sqrt{1 + \alpha^2}$. As demonstrações desses resultados podem ser encontradas no Apêndice A. Na dissertação, utilizamos a seguinte representação estocástica da distribuição normal assimétrica:

$$X = \eta + \omega \delta \left| \varepsilon^0 \right| + \omega \sqrt{1 - \delta^2} \varepsilon^1, \quad (2.1)$$

onde $\alpha = \delta / \sqrt{1 - \delta^2}$ com $\delta \in (-1, 1)$ e as variáveis ε^0 e ε^1 são independentes e identicamente distribuídos com uma normal padrão (AZZALINI, 1986; HENZE, 1986).

2.3 Misturas de escala da distribuição normal assimétrica

Sabe-se que a distribuição dos retornos financeiros apresenta assimetria e caudas pesadas, o que torna a classe de distribuições mistura de escala da normal assimétrica (SMSN) uma ferramenta útil para a modelagem desses dados. Nesta seção, apresentaremos alguns casos particulares da distribuição SMSN.

Uma variável aleatória U segue a distribuição SMSN, $U \sim \text{SMSN}(\eta, \omega^2, \alpha, \nu)$, se tiver a seguinte representação:

$$U = \eta + Z^{-1/2}X, \quad (2.2)$$

onde η é um parâmetro de locação, $X \sim \text{SN}(0, \omega^2, \alpha)$ e Z é uma variável aleatória positiva com função densidade de probabilidade $p(z | \nu)$, onde ν é um parâmetro positivo associado a distribuição de mistura Z .

Como pode ser observado na equação (2.2), a distribuição de U depende das especificações de Z . Por exemplo, quando $Z = 1$, a variável aleatória U tem distribuição normal assimétrica. Neste trabalho, assim como em [Huang e Xu \(2014\)](#), focamos em dois casos especiais da distribuição SMSN, a distribuição t-Student assimétrica e a distribuição slash assimétrica, que serão apresentadas a seguir.

2.3.1 Distribuição t-Student assimétrica

A distribuição t-Student assimétrica (ST) foi proposta por [Branco e Dey \(2001\)](#). Essa distribuição é obtida ao assumir que $Z \sim \text{Gama}(\nu/2, \nu/2)$, em que $\nu > 0$, na Equação 2.2. A notação $\text{Gama}(a, b)$ indica uma distribuição gama com média a/b . A função densidade de probabilidade de uma variável U com distribuição t-Student assimétrica é dada por:

$$f(u) = \frac{2}{\omega} t_\nu \left(\frac{u - \eta}{\omega} \right) T_{\nu+1} \left(\alpha \frac{u - \eta}{\omega} \sqrt{\frac{\nu + 1}{\nu + \omega^{-2}(u - \eta)^2}} \right),$$

em que t_ν e T_ν denotam, respectivamente, a função densidade de probabilidade e a função de distribuição acumulada da distribuição t-Student padrão com ν graus de liberdade. Quando $\nu \rightarrow \infty$, a distribuição t-Student assimétrica converge para a distribuição normal assimétrica. À medida que ν se torna maior, a densidade tem caudas leves. A distribuição t-Student assimétrica pode capturar assimetria negativa quando $\alpha < 0$ e assimetria positiva quando $\alpha > 0$, e inclui a distribuição t de Student simétrica como um caso especial quando $\alpha = 0$. A partir da Equação (2.2), obtém-se que a média e variância de $U \sim ST(\eta, \omega^2, \nu, \alpha)$ são dadas, respectivamente, por:

$$\begin{aligned} E(U) &= \eta + \frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \sqrt{\frac{\nu}{\pi}} \sigma \delta, \quad \nu > 1 \\ V(U) &= \frac{\nu}{\nu-2} \sigma^2 - \left(\frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \right)^2 \frac{\nu}{\pi} \sigma^2 \delta^2, \quad \nu > 2, \end{aligned} \quad (2.3)$$

onde $E(\cdot)$ e $V(\cdot)$ denotam a esperança e variância, respectivamente. A demonstração desses resultados pode ser encontrada no [Apêndice A](#). Na literatura de modelos SV, o uso da distribuição t-Student assimétrica como uma distribuição alternativa à distribuição normal tem sido frequentemente sugerido. Ver, por exemplo, [Nakajima e Omori \(2012\)](#) e [Abanto-Valle, Lachos e Dey \(2015\)](#).

2.3.2 Distribuição slash assimétrica

A distribuição slash assimétrica (SL) foi proposta por [Wang e Genton \(2006\)](#). Essa distribuição possui caudas pesadas e assimetria, o que pode ser útil para modelar séries de retornos financeiros. A distribuição SL é obtida fazendo $Z \sim \text{Beta}(\nu, 1)$, com $\nu > 0$, na Equação (2.2). Usamos $\text{Beta}(a, b)$ para denotar a distribuição Beta com parâmetros $a > 0$ e $b > 0$. A função densidade de probabilidade de uma variável U com distribuição slash assimétrica é dada por:

$$f(u) = 2\nu \int_0^1 s^{\nu-1} \phi(u; \eta, s^{-1}\omega^2) \Phi\left(\frac{\alpha s^{1/2}(u - \eta)}{\omega}\right) ds,$$

em que $\phi(u; \eta, s^{-1}\omega^2)$ denota a função densidade de probabilidade da distribuição normal com média η e variância $s^{-1}\omega^2$. Os parâmetros α e ν controlam a assimetria e a cauda pesada da distribuição slash assimétrica, respectivamente. Quando $\nu \rightarrow \infty$, a distribuição slash assimétrica converge para a distribuição normal assimétrica. A partir da Equação (2.2), obtém-se que a média e variância de $U \sim SL(\eta, \omega^2, \nu, \alpha)$ são dadas, respectivamente, por:

$$\begin{aligned} E(U) &= \eta + \frac{2\nu}{2\nu-1} \sqrt{\frac{2}{\pi}} \delta \omega, \quad \nu > 1/2, \\ V(U) &= \frac{\nu}{\nu-1} \omega^2 - \left(\frac{2\nu}{2\nu-1}\right)^2 \frac{2}{\pi} \delta^2 \omega^2, \quad \nu > 1. \end{aligned} \tag{2.4}$$

A demonstração desses resultados pode ser encontrada no [Apêndice A](#).

2.4 Métodos de Monte Carlo via cadeias de Markov

Nesta seção, abordaremos os métodos de Monte Carlo via cadeias de Markov (MCMC), que são amplamente utilizados para gerar amostras da distribuição a posteriori dos parâmetros do modelo. Essa distribuição é a base para a inferência estatística Bayesiana e é obtida por meio do teorema de Bayes:

$$p(\theta | y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y | \theta)p(\theta)}{\int p(y | \theta)p(\theta)d\theta}, \tag{2.5}$$

onde y é o vetor de observações e θ é o vetor de parâmetros do modelo. $p(\theta)$ é a distribuição a priori dos parâmetros, que representa o conhecimento prévio sobre a distribuição dos parâmetros antes de observar os dados. $p(y | \theta)$ é a função de verossimilhança dos dados observados.

Para aproximar a distribuição a posteriori da Equação (2.5), utilizamos os métodos MCMC, estes métodos usam o fato de que o denominador da Equação (2.5) é uma constante de normalização que não depende de θ . Portanto, a distribuição a posteriori pode ser expressa como:

$$p(\theta | y) \propto p(y | \theta)p(\theta).$$

Os métodos MCMC envolvem duas etapas: na primeira etapa, construímos uma cadeia de Markov, que consiste em uma sequência de variáveis aleatórias $\{\theta^{(i)}\}_{i=1}^S$ que converge para a

distribuição a posteriori $p(\theta | y)$. Na segunda etapa, empregamos métodos de Monte Carlo para resumir a distribuição a posteriori do parâmetro θ como saída do MCMC. Após um período de aquecimento suficientemente longo (o número de amostras descartadas desde o início da cadeia), os valores simulados da cadeia de Markov são usados para fazer inferências Bayesianas, como a média posterior, a variância e a estimativa de intervalos de credibilidade.

Para implementar a primeira etapa do MCMC, existem vários algoritmos disponíveis para produzir amostras da distribuição a posteriori. A seguir, apresentamos o algoritmo Metropolis-Hastings e uma extensão dele, o algoritmo Monte Carlo Hamiltoniano.

2.4.1 Algoritmo Metropolis-Hastings

O algoritmo Metropolis, proposto por [Metropolis et al. \(1953\)](#), foi o primeiro algoritmo MCMC publicado e posteriormente foi generalizado por [Hastings \(1970\)](#), resultando no algoritmo Metropolis-Hastings (MH). O algoritmo MH é utilizado para simular valores da distribuição a posteriori $p(\theta | y)$ e envolve o uso de uma distribuição proposta $q(\cdot)$. O algoritmo MH pode ser descrito da seguinte maneira:

1. Especifique um valor inicial $\theta^{(0)}$.
2. Para $i = 1, \dots, S$:
 - Gere θ^* a partir da distribuição proposta $q(\theta^* | \theta^{(i-1)})$, onde q é uma distribuição condicional arbitrária.
 - Calcule a probabilidade de aceitação

$$\alpha(\theta^{(i-1)}, \theta^*) = \min \left\{ \frac{p(\theta^* | y)q(\theta^{(i-1)} | \theta^*)}{p(\theta^{(i-1)} | y)q(\theta^* | \theta^{(i-1)})}, 1 \right\}.$$
 - Gere $u \sim U(0, 1)$. Se $u < \alpha(\theta^{(i-1)}, \theta^*)$, defina $\theta^{(i)} = \theta^*$; caso contrário, $\theta^{(i)} = \theta^{(i-1)}$.

Na prática, a escolha da distribuição proposta $q(\cdot)$ é crucial para a convergência da cadeia. Uma desvantagem do algoritmo MH é a alta autocorrelação entre as amostras geradas, o que leva a uma convergência lenta e, conseqüentemente, à necessidade de gerar grandes tamanhos de amostra.

A seguir, apresentaremos a método MCMC utilizado na dissertação para estimar os parâmetros em modelos de volatilidade estocástica com alavancagem baseados em mistura de escala da distribuição normal assimétrica (SV-SMSN).

2.4.2 Algoritmo Monte Carlo Hamiltoniano

O algoritmo Hamiltoniano Monte Carlo (HMC, do inglês *Hamiltonian Monte Carlo*) é um caso especial do algoritmo Metropolis-Hastings que não requer a especificação da distribuição

proposta $q(\cdot)$. Esse algoritmo foi inicialmente proposto na literatura de física por [Duane et al. \(1987\)](#) sob o nome de algoritmo Monte Carlo híbrido, mas logo passou a ser conhecido como Monte Carlo Hamiltoniano. Uma descrição mais detalhada do HMC pode ser encontrada em [Neal et al. \(2011\)](#).

O algoritmo HMC usa a dinâmica Hamiltoniana para simular amostras da distribuição a posteriori. A dinâmica Hamiltoniana depende de dois elementos: um vetor de posição (parâmetros) $\theta \in \mathbb{R}^d$ e um vetor auxiliar de momento $\omega \in \mathbb{R}^d$ com distribuição $N_d(0, M)$ (o HMC introduz uma variável auxiliar de momento ω_j para cada parâmetro θ_j).

A energia total do sistema de dinâmica Hamiltoniana é dada pela função Hamiltoniana, $H(\theta, \omega)$, que é a soma da energia potencial $U(\theta)$ e da energia cinética $K(\omega)$,

$$\begin{aligned} H(\theta, \omega) &= U(\theta) + K(\omega) \\ &= -\log p(\theta | y) + \frac{1}{2} \omega' M^{-1} \omega, \end{aligned}$$

onde $U(\theta)$ é definida como o negativo do logaritmo da distribuição a posteriori, e M é uma matriz diagonal que geralmente é definida como a matriz identidade. A dinâmica Hamiltoniana é definida pelo seguinte sistema de equações diferenciais:

$$\begin{aligned} \frac{d\theta}{d\tau} &= \frac{\partial H}{\partial \omega} = \nabla_{\omega} K(\omega) = M^{-1} \omega, \\ \frac{d\omega}{d\tau} &= -\frac{\partial H}{\partial \theta} = -\nabla_{\theta} U(\theta), \end{aligned} \tag{2.6}$$

em que ∇_x indica o vetor gradiente em relação à variável x . Estas equações determinam como θ e ω mudam ao longo do tempo τ , e sua solução fornece o novo valor proposto (θ^*, ω^*) . Da Equação (2.6) observamos que o HMC faz uso do gradiente do logaritmo da distribuição a posteriori e, portanto, aponta para regiões de maior probabilidade, tornando o algoritmo capaz de alcançar de forma mais rápida a distribuição de equilíbrio da cadeia de Markov.

Na prática, soluções analíticas das equações Hamiltonianas são dificilmente obtidas. Por meio do integrador numérico *Leapfrog*, descrito por [Leimkuhler e Reich \(2004\)](#), podemos aproximar a solução do sistema de equações diferenciais. Para valores iniciais $(\theta^{(\tau)}, \omega^{(\tau)})$, em um dado tempo inicial τ , o integrador *Leapfrog* calcula as atualizações por meio das seguintes expressões:

$$\begin{aligned} \omega^{(\tau+\varepsilon/2)} &= \omega^{(\tau)} + \frac{\varepsilon}{2} \frac{d\omega}{d\tau} = \omega^{(\tau)} - \frac{\varepsilon}{2} \nabla_{\theta} U(\theta^{(\tau)}), \\ \theta^{(\tau+\varepsilon)} &= \theta^{(\tau)} + \varepsilon \frac{d\theta}{d\tau} = \theta^{(\tau)} + \varepsilon \nabla_{\omega} K(\omega^{(\tau+\varepsilon/2)}), \\ \omega^{(\tau+\varepsilon)} &= \omega^{(\tau+\varepsilon/2)} + \frac{\varepsilon}{2} \frac{d\omega}{d\tau} = \omega^{(\tau+\varepsilon/2)} - \frac{\varepsilon}{2} \nabla_{\theta} U(\theta^{(\tau+\varepsilon)}), \end{aligned}$$

para algum tamanho de passo $\varepsilon > 0$. Começando com o estado atual (θ, ω) , o integrador *Leapfrog* é executado K vezes com tamanho de passo ε para gerar uma proposta (θ^*, ω^*) . Devido ao erro introduzido ao discretizar o sistema, avaliamos a aceitação ou não da proposta (θ^*, ω^*) mediante

a probabilidade de aceitação de Metropolis, definida como:

$$\alpha[(\theta, \omega), (\theta^*, \omega^*)] = \min \left(1, \frac{\exp \{-H(\theta^*, \omega^*)\}}{\exp \{-H(\theta, \omega)\}} \right) = \min(1, \exp\{-H(\theta^*, \omega^*) + H(\theta, \omega)\}).$$

Se não aceitarmos o novo estado proposto, o próximo estado será igual ao atual. O algoritmo HMC em sua forma simples ($M = I$), é apresentado a continuação:

1. Fornecer uma posição inicial $\theta^{(0)}$, tamanho do passo ε , número de passos K e o número de iterações S .
2. Para $i = 1, \dots, S$:
 - Gerar um vetor de momento $\omega^0 \sim N_d(0, I_d)$.
 - Definir $\omega^* \leftarrow \omega^0$, $\theta^* \leftarrow \theta^{(i-1)}$.
 - Começando com o estado atual $(\theta^{(i-1)}, \omega^0)$, executar o algoritmo *Leapfrog* em K passos com tamanho de passo ε para obter a proposta (θ^*, ω^*)

$$\begin{aligned}\omega^* &= \omega^* - \frac{\varepsilon}{2} \nabla_{\theta} U(\theta^*), \\ \theta^* &= \theta^* + \varepsilon \nabla_{\omega} K(\omega^*), \\ \omega^* &= \omega^* - \frac{\varepsilon}{2} \nabla_{\theta} U(\theta^*).\end{aligned}$$

- Gerar $u \sim U(0, 1)$. Se $u < \alpha[(\theta^{(i-1)}, \omega^0), (\theta^*, \omega^*)]$, fazer $\theta^{(i)} = \theta^*$; caso contrario $\theta^{(i)} = \theta^{(i-1)}$.

Muitos especialistas relataram que a definição dos parâmetros K e ε é uma tarefa desafiadora, o que pode dificultar o uso do algoritmo HMC. A escolha inadequada do tamanho do passo ε e do número de passos K pode resultar em mistura lenta ou em um alto custo computacional no algoritmo HMC, e não há uma orientação geral sobre como esses parâmetros devem ser escolhidos. Por exemplo:

- Se K for muito pequeno, o HMC exibe um comportamento de passeio aleatório, enquanto que se K for muito grande, menor será a autocorrelação entre as amostras, mas o custo computacional será maior.
- Se ε for muito grande, o algoritmo leva muito tempo para atingir a convergência e produzirá baixas taxas de aceitação. Enquanto se ε for muito pequeno, será difícil explorar todo o espaço paramétrico, a menos que um K grande seja usado.

Para contornar o problema de ajustar o tamanho do passo ε e o número de passos K , [Hoffman, Gelman et al. \(2014\)](#) propuseram o algoritmo *No-U-Turn Sampler*, que será descrito a seguir.

2.4.3 Algoritmo No-U-Turn Sampler

O algoritmo *No-U-Turn Sampler* (NUTS) é uma extensão do algoritmo HMC que ajusta automaticamente os parâmetros ϵ e K . O algoritmo NUTS começa com a posição inicial $\theta^{(0)}$ e usa o algoritmo HMC para gerar uma nova proposta θ^* . Em seguida, o algoritmo NUTS verifica se a trajetória gerada pelo algoritmo HMC não dá meia-volta (textitNo-U-Turn). Se a trajetória não der meia-volta, o algoritmo NUTS continua gerando novas propostas usando o algoritmo HMC. Caso contrário, o algoritmo NUTS para e retorna a última amostra gerada. Para ajustar o tamanho de passo, o algoritmo NUTS usa o método "dual averaging" e para ajustar o número de passos, o algoritmo NUTS usa o método "binary doubling". Para obter amostras MCMC da distribuição a posteriori, o algoritmo NUTS foi implementado em C++ como parte do software de inferência Bayesiana Stan ([CARPENTER et al., 2017](#)), uma linguagem de programação de alto nível.

Nos últimos anos, o uso do algoritmo Hamiltoniano Monte Carlo tem aumentado com o desenvolvimento do software Stan. Neste trabalho, utilizamos o Stan para estimar os parâmetros da família de modelos SV com alavancagem baseados em misturas de escala da distribuição normal assimétrica. No entanto, a avaliação do desempenho do algoritmo NUTS em modelos SV ainda é limitada, conforme evidenciado por trabalhos como [Wahl \(2018\)](#) e [Dias e Ehlers \(2017\)](#). Assim, este trabalho visa contribuir para preencher essa laguna de pesquisa.

MODELOS DE VOLATILIDADE ESTOCÁSTICA

Em aplicações financeiras, os modelos de volatilidade estocástica (SV) são utilizados para modelar a volatilidade presente nas séries de retornos financeiros. Neste capítulo, apresentamos os modelos de volatilidade estocástica com alavancagem baseados em mistura de escala da distribuição normal assimétrica (SMSN).

3.1 Modelos SV com alavancagem baseados em SMSN

A classe de modelos SV com alavancagem baseados na mistura de escala da distribuição normal assimétrica (SV-SMSN) foi proposta por [Huang e Xu \(2014\)](#) com o objetivo de capturar os principais fatos estilizados dos retornos financeiros diários, como o efeito de alavancagem, caudas pesadas e assimetria. Diferentemente do modelo SV básico proposto por [Taylor \(1982\)](#), que assume a distribuição normal para os erros dos retornos, os modelos SV-SMSN utilizam a mistura de escala da distribuição normal assimétrica (SMSN), apresentada na [Seção 2.3](#). Os modelos SV-SMSN são definidos como:

$$\begin{aligned}
 y_t &= \exp(h_t/2) u_t, & t &= 1, \dots, n, \\
 h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma \eta_t, & t &= 1, \dots, (n-1), \\
 u_t &\sim \text{SMSN}(0, 1, \alpha, v), \\
 \eta_t &\sim N(0, 1), \\
 \rho &= \text{corr}(u_t, \eta_t),
 \end{aligned} \tag{3.1}$$

onde y_t representa os retornos diários no tempo t e h_t é a log-volatilidade não observada de y_t . Os parâmetros μ , ϕ e σ são, respectivamente, a média, o parâmetro de persistência e o desvio padrão das log-volatilidades. Assume-se que $|\phi| < 1$, ou seja, o processo de log-volatilidade

é estacionário e o valor inicial $h_1 \sim N(\mu, \sigma^2/(1 - \phi^2))$. Quando o parâmetro $\rho < 0$, temos o efeito é negativo de alavancagem (YU, 2005).

Dado que $u_t \sim \text{SMSN}(0, 1, \alpha, \nu)$, podemos reescrever o erro u_t do modelo SV-SMSN da Equação (3.1), usando a Equação (2.2) e a representação estocástica da distribuição normal assimétrica dada na Equação (2.1). Assim, o modelo SV-SMSN pode ser reescrito da seguinte forma:

$$\begin{aligned} y_t &= \exp(h_t/2) \left(\frac{1}{\sqrt{z_t}} \left(\delta |\varepsilon_t^0| + \sqrt{1 - \delta^2} \varepsilon_t^1 \right) - E(u_t) \right), \quad t = 1, \dots, n, \\ h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma \eta_t, \quad t = 1, \dots, (n-1), \\ \varepsilon_t^0 &\sim N(0, 1), \\ \varepsilon_t^1 &\sim N(0, 1), \\ \eta_t &\sim N(0, 1), \\ \rho &= \text{corr}(\varepsilon_t^1, \eta_t), \end{aligned} \quad (3.2)$$

onde $\delta = \alpha/\sqrt{1 + \alpha^2}$. Observe que subtraímos $E(u_t)$ para garantir que $E(y_t | h_t) = 0$, já que $E(u_t) \neq 0$, como visto na Seção 2.3. Para que os erros ε_t^1 e η_t sejam não correlacionados, introduzimos a seguinte transformação:

$$\eta_t = \rho \varepsilon_t^1 + \sqrt{1 - \rho^2} k_t, \quad t = 1, \dots, n, \quad (3.3)$$

em que $k_t \sim N(0, 1)$ e $\text{corr}(\varepsilon_t^1, k_t) = 0$. Assim, substituindo (3.3) em (3.2), obtemos a seguinte reparametrização para a classe de modelos SV-SMSN:

$$\begin{aligned} y_t &= \exp(h_t/2) \left(\frac{1}{\sqrt{z_t}} \left(\delta |\varepsilon_t^0| + \sqrt{1 - \delta^2} \varepsilon_t^1 \right) - E(u_t) \right), \quad t = 1, \dots, n, \\ h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma \left(\rho \varepsilon_t^1 + \sqrt{1 - \rho^2} k_t \right), \quad t = 1, \dots, (n-1), \\ \varepsilon_t^0 &\sim N(0, 1), \\ \varepsilon_t^1 &\sim N(0, 1), \\ k_t &\sim N(0, 1), \end{aligned} \quad (3.4)$$

Na Seção 2.3, foi apresentada a família de distribuições SMSN, a qual inclui como casos especiais as distribuições normal, t-Student, t-Student assimétrica e slash assimétrica. A principal vantagem dos modelos SV-SMSN é que, alterando a distribuição da variável de mistura z_t , é possível obter como casos especiais outros modelos SV já conhecidos na literatura.

Por exemplo, ao assumir que $z_t \sim \text{Beta}(\nu, 1)$ na Equação (3.4), obtemos o modelo SV com alavancagem e distribuição slash assimétrica, que será denotado por SV-SSL neste trabalho

e é dado por:

$$\begin{aligned}
 y_t &= \exp(h_t/2) \left(\frac{1}{\sqrt{z_t}} \left(\delta |\varepsilon_t^0| + \sqrt{1 - \delta^2} \varepsilon_t^1 \right) - \frac{2v}{2v-1} \sqrt{\frac{2}{\pi}} \delta \right), \quad t = 1, \dots, n, \\
 h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma \left(\rho \varepsilon_t^1 + \sqrt{1 - \rho^2} k_t \right), \quad t = 1, \dots, n-1, \\
 z_t &\sim \text{Beta}(v, 1), \\
 \varepsilon_t^0 &\sim N(0, 1), \\
 \varepsilon_t^1 &\sim N(0, 1), \\
 k_t &\sim N(0, 1).
 \end{aligned} \tag{3.5}$$

É importante destacar que $E(u_t)$ foi substituído pela média da distribuição slash assimétrica apresentada na Equação (2.4) com $\eta = 0$ e $\omega = 1$. Na literatura de modelos GARCH, a distribuição slash assimétrica foi proposta por [Fuente, Miguel e Wiper \(2012\)](#) para modelar as inovações.

Por outro lado, se fizermos $z_t \sim \text{Gama}(v/2, v/2)$, para todo $t = 1, \dots, n$ na Equação (3.4), obtemos o modelo SV com alavancagem e distribuição t-Student assimétrica. Esse modelo será referido como SV-STL e é definido por:

$$\begin{aligned}
 y_t &= \exp(h_t/2) \left(\frac{1}{\sqrt{z_t}} \left(\delta |\varepsilon_t^0| + \sqrt{1 - \delta^2} \varepsilon_t^1 \right) - \frac{\Gamma\left(\frac{v-1}{2}\right)}{\Gamma\left(\frac{v}{2}\right)} \sqrt{\frac{v}{\pi}} \delta \right), \quad t = 1, \dots, n, \\
 h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma \left(\rho \varepsilon_t^1 + \sqrt{1 - \rho^2} k_t \right), \quad t = 1, \dots, n-1, \\
 z_t &\sim \text{Gama}(v/2, v/2), \\
 \varepsilon_t^0 &\sim N(0, 1), \\
 \varepsilon_t^1 &\sim N(0, 1), \\
 k_t &\sim N(0, 1).
 \end{aligned} \tag{3.6}$$

Note que na Equação (3.4), subtraímos a média da distribuição t-Student assimétrica dada na Equação (2.3) com $\eta = 0$ e $\omega = 1$. [Abanto-Valle, Lachos e Dey \(2015\)](#) também assumiram a distribuição t-Student assimétrica para a distribuição condicional dos retornos, no entanto, sem considerar o efeito de alavancagem, ou seja, $\rho = 0$.

As equações em (3.6) com $\alpha = 0$ (ou equivalentemente $\delta = 0$) definem o modelo SV com alavancagem e distribuição t-Student. Neste trabalho, usaremos a notação SV-TL para nos

referirmos a esse modelo, que assume a seguinte representação para os retornos y_t :

$$\begin{aligned}
 y_t &= \exp(h_t/2) \frac{1}{\sqrt{z_t}} \varepsilon_t^1, & t = 1, \dots, n, \\
 h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma \left(\rho \varepsilon_t^1 + \sqrt{1 - \rho^2} k_t \right), & t = 1, \dots, n-1, \\
 z_t &\sim \text{Gama}(v/2, v/2), \\
 \varepsilon_t^1 &\sim N(0, 1), \\
 k_t &\sim N(0, 1).
 \end{aligned} \tag{3.7}$$

O modelo SV-TL, proposto por [Jacquier, Polson e Rossi \(2004\)](#), considera duas características dos retornos financeiros: o efeito de alavancagem e caudas pesadas. Quando $\rho = 0$ na Equação (3.7), obtemos o modelo SV com distribuição t-Student simétrica (SV-T) como um caso particular, definido como:

$$\begin{aligned}
 y_t &= \exp(h_t/2) \frac{1}{\sqrt{z_t}} \varepsilon_t^1, & t = 1, \dots, n, \\
 h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma k_t, & t = 1, \dots, n-1, \\
 z_t &\sim \text{Gama}(v/2, v/2), \\
 \varepsilon_t^1 &\sim N(0, 1), \\
 k_t &\sim N(0, 1).
 \end{aligned} \tag{3.8}$$

Alguns estudos que utilizam a distribuição t-Student incluem [Meyer e Yu \(2000\)](#), [Chib, Nardari e Shephard \(2002\)](#), [Jacquier, Polson e Rossi \(2004\)](#), [Omori *et al.* \(2007\)](#) e [Choy, Wan e Chan \(2009\)](#), entre muitos outros. Além disso, [Abanto-Valle *et al.* \(2010\)](#) expressaram a função densidade t-Student usando mistura de escala de distribuições normais.

As equações em (3.6) com $\delta = 0$ e $z_t = 1$ para $t = 1, \dots, n$ definem o modelo SV com distribuição normal e efeito de alavancagem, que será denotado por SV-NL, dado por:

$$\begin{aligned}
 y_t &= \exp(h_t/2) \varepsilon_t^1, & t = 1, \dots, n, \\
 h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma \left(\rho \varepsilon_t^1 + \sqrt{1 - \rho^2} k_t \right), & t = 1, \dots, n-1, \\
 \varepsilon_t^1 &\sim N(0, 1), \\
 k_t &\sim N(0, 1).
 \end{aligned} \tag{3.9}$$

O modelo SV-NL foi proposto por [Harvey e Shephard \(1996\)](#) e investigado em artigos como [Yu \(2005\)](#), [Omori *et al.* \(2007\)](#) e [Nakajima e Omori \(2009\)](#), entre outros. Quando fazemos $\rho = 0$ na Equação (3.9), obtemos o modelo SV com distribuição normal (SV-N), definido como:

$$\begin{aligned}
 y_t &= \exp(h_t/2) \varepsilon_t^1, & t = 1, \dots, n, \\
 h_{t+1} &= \mu + \phi(h_t - \mu) + \sigma k_t, & t = 1, \dots, n-1, \\
 \varepsilon_t^1 &\sim N(0, 1), \\
 k_t &\sim N(0, 1),
 \end{aligned} \tag{3.10}$$

em que ε_t^1 e k_t são variáveis independentes. Esse foi o primeiro modelo SV proposto e tem sido amplamente utilizado para modelar a volatilidade de séries de retornos financeiros, ver por exemplo Taylor (1982), Kim, Shephard e Chib (1998), Ehlers e Zavallos (2015) e muitos outros.

3.2 Estimação Bayesiana

A estimação dos parâmetros dos modelos SV é uma tarefa desafiadora, pois a função de verossimilhança do modelo SV envolve uma integral de alta dimensão. Denotando por $\vartheta = (\mu, \phi, \sigma, \rho, \nu, \alpha)$ o vetor de parâmetros dos modelos SV-SMSN, por $h = (h_1, \dots, h_n)$ o vetor de log-volatilidades e por $y = (y_1, \dots, y_n)$ os retornos até o tempo n . Na abordagem Bayesiana utilizada neste trabalho, empregamos o princípio de aumento de dados, que considera h como variáveis latentes. A distribuição posterior conjunta dos parâmetros e variáveis latentes é dada por:

$$p(h, \vartheta | y) \propto p(y | \vartheta, z, h) p(h | \vartheta) p(\vartheta),$$

Para estimar a classe de modelos SV-SMSN descritos na seção anterior, utilizamos o algoritmo NUTS, disponível no pacote R `rstan` (Interface R para Stan) (Stan Development Team, 2022). O algoritmo NUTS requer o cálculo do gradiente da distribuição a posteriori, o qual é avaliado automaticamente pelo Stan por meio de diferenciação automática (CARPENTER *et al.*, 2015). Uma vantagem de utilizar o Stan é a facilidade de implementação de modelos Bayesianos. A seguir, apresentaremos um exemplo de como implementar o modelo SV-TL no Stan.

A partir das equações em (3.7), é possível obter a densidade condicional dos retornos e das log-volatilidades. Portanto, o modelo SV-TL pode ser especificado de forma alternativamente como:

$$\begin{aligned} y_t | h_t, z_t &\sim N\left(0, \frac{\exp(h_t)}{z_t}\right), \\ h_{t+1} | h_t, z_t, y_t &\sim N\left(\mu + \phi(h_t - \mu) + \sigma\rho\sqrt{z_t}\exp(-0,5h_t)y_t, \sigma^2(1 - \rho^2)\right), \\ h_1 &\sim N\left(\mu, \frac{\sigma^2}{1 - \phi^2}\right). \end{aligned}$$

O primeiro passo é definir nossos dados no bloco "data":

```
1: data {
2:   int<lower=0> n;      // número de observações
3:   vector[n] y;        // log-retornos
4: }
```

Para o modelo de volatilidade estocástica com distribuição normal (SV-N), a guia do usuário do Stan (<<https://mc-stan.org/docs/stan-users-guide/stochastic-volatility-models.html>>) mostra que o tamanho efetivo da amostra aumenta quando usamos a log-volatilidade padronizada h_std ,

dada por:

$$h_{std} = \frac{h_{t+1} - \mu - \phi(h_t - \mu) - \sigma\rho\sqrt{z_t}\exp(-0,5h_t)y_t}{\sigma\sqrt{1-\rho^2}} \sim N(0, 1).$$

Na prática, para a classe de modelos SV estudados nesta dissertação, confirmamos que o tamanho efetivo da amostra aumenta ao declararmos h_{std} no bloco de parâmetros e o valor original de h no bloco de parâmetros transformados. Em seguida, definimos os parâmetros do modelo SV-TL no bloco "parameters":

```

1: parameters {
2:   real mu;
3:   real<lower=0,upper=1> phistar;
4:   real<lower=0> sigma2;
5:   real<lower=0,upper=1> rhostar;
6:   real<lower=2> nu;
7:   vector<lower=0> [n] z;
8:   vector[n] h_std;
9: }
```

Definimos explicitamente o espaço de parâmetros válidos para nossos parâmetros, por exemplo, $0 < \phi^* < 1$. O Stan transforma automaticamente todos os parâmetros restritos para a reta real, permitindo que sejam amostrados sem restrições, mas essa transformação fica oculta para o usuário.

No bloco "transformed parameters" podemos definir novos parâmetros como funções dos parâmetros originais, os parâmetros transformados não são amostrados. Stan amostra os parâmetros declarados no bloco de parâmetros e, em seguida, aplica as fórmulas do bloco de parâmetros transformados após a amostragem.

```

1: transformed parameters {
2:   real<lower=-1, upper=1> phi;
3:   real<lower=0> sigma;
4:   real<lower=-1, upper=1> rho;
5:   vector[n] h;
6:
7:   phi = fma(phistar, 2, - 1);
8:   sigma = sqrt(sigma2);
9:   rho = 2*rhostar - 1;
10:  h = h_std * sigma*sqrt(1-square(rho));
11:  h[1] /= (sqrt(1 - square(phi))*sqrt(1-square(rho)));
12:  h += mu;
13:  for (t in 2:n)
```

```

14:     h[t] += phi*(h[t-1]-mu)+sigma*rho*exp(-0.5*h[t-1])*y[t-1]*
      sqrt(z[t-1]);
15: }

```

É também observado na guia do usuário do Stan que a vetorização dos retornos pode reduzir o tempo de execução do modelos SV-N. Portanto, para a classe de modelos SV-SMSN, seguiremos essa ideia. No bloco "model", definimos a distribuição a posteriori e especificamos as distribuições a priori dos parâmetros do modelo SV-TL:

```

1: model {
2:   mu ~ normal(0,10);
3:   phistar ~ beta(20, 1.5);
4:   sigma2 ~ inv_gamma(2.5,0.025);
5:   rhostar ~ beta(3,6);
6:   nu ~ exponential(0.3333333);
7:   z ~ gamma(0.5*nu,0.5*nu);
8:   h_std ~ std_normal();
9:   y ~ normal(0, exp(0.5*h) ./ sqrt(z));
10: }

```

Note-se que na linguagem Stan, para a distribuição normal, especificamos a média e o desvio padrão. A partir do código, podemos ver que foram utilizadas as seguintes distribuições a priori para os parâmetros do modelo SV-TL:

$$\begin{aligned}
 \mu &\sim N(0, 10^2), \\
 (\phi + 1)/2 &\sim \text{Beta}(20; 1, 5), \\
 \sigma^2 &\sim \text{IG}(2, 5; 0, 025), \\
 (\rho + 1)/2 &\sim \text{Beta}(3, 6), \\
 v &\sim \text{Exp}(1/3), \quad 2 < v.
 \end{aligned}$$

No bloco "generated quantities" definimos as quantidades para a obtenção dos critérios de informação, utilizados para seleção de modelos.

```

1: generated quantities {
2:   vector[n] log_lik;
3:
4:   for (t in 1:n)
5:     log_lik[t] = normal_lpdf(y[t]|0, exp(0.5*h[t])/sqrt(z[t]));
6: }

```

Agora podemos ajustar o modelo SV-TL no R, através do pacote `rstan`. Outra vantagem de usar o Stan é que quando o algoritmo NUTS encontra problemas, ele retorna diversos avisos,

os quais não devem ser ignorados. No [Apêndice B](#), apresentamos o código em Stan dos modelos SV-SMSN implementados na linguagem de programação probabilística Stan.

MÉTODOS DE SELEÇÃO DE MODELOS

Neste capítulo, descrevemos os métodos de seleção de modelos que serão utilizados na dissertação para comparar modelos SV. Devido a ampla gama de modelos SV disponíveis para analisar as séries temporais financeiras, é importante selecionar o modelo SV que melhor represente a dinâmica da série de retornos. No contexto dos modelos SV, θ é o vetor que contém os parâmetros do modelo e as log-volatilidades latentes, $h = (h_1, \dots, h_n)$.

4.1 Critério de informação de desvio

Um dos critérios mais utilizados para comparar modelos Bayesianos é o critério de informação de desvio (DIC), proposto por Spiegelhalter *et al.* (2002). O DIC é definido como:

$$\text{DIC} = -2\log p(y | \bar{\theta}) + 2p_{\text{DIC}}, \quad (4.1)$$

onde o primeiro termo é uma medida Bayesiana de quão bem o modelo se ajusta aos dados, ou seja, a log-verossimilhança avaliada na estimativa de Bayes. O segundo termo é uma medida de complexidade do modelo, p_{DIC} , que é definido como duas vezes a diferença entre o desvio avaliado na média posterior dos parâmetros e a média posterior do desvio:

$$p_{\text{DIC}} = 2 \left(\log p(y | \bar{\theta}) - E_{\theta|y}(\log p(y | \theta)) \right). \quad (4.2)$$

O critério DIC é uma métrica popular para comparar modelos de volatilidade estocástica (ver, por exemplo, Berg, Meyer e Yu (2004)), e um valor menor indica um modelo melhor. O DIC é fácil de calcular a partir da saída do MCMC. No entanto, um estudo de Chan e Grant (2014) demonstra que o DIC tende a favorecer modelos SV mais complexos, o que pode levar a seleção de modelos com maior complexidade do que o necessário. Portanto, é importante ter cautela ao utilizar o DIC como única medida de comparação de modelos.

Outra maneira de comparar modelos é através da medição de sua precisão preditiva, para o qual definimos a distribuição preditiva posterior.

Definição 1. A distribuição preditiva posterior é dada por:

$$p(\tilde{y} | y) = \int p(\tilde{y}_t | \theta) p(\theta | y) d\theta,$$

onde $p(\theta | y)$ é a distribuição a posteriori. Quanto maior a distribuição preditiva posterior melhor será a precisão preditiva do modelo.

Calculamos a esperança dessa medida para cada uma das n observações do conjunto de dados

$$\text{elpd} = \sum_{t=1}^n \int f(\tilde{y}_t) \log p(\tilde{y}_t | y) d\tilde{y}_t,$$

onde elpd é a esperança do log da densidade preditiva pontual e $f(\tilde{y}_t)$ é a distribuição que representa o verdadeiro processo de geração de dados para \tilde{y}_t . A medida de precisão preditiva elpd, avalia a distribuição preditiva posterior dos modelos para quantificar a incerteza em suas previsões. Grandes valores de elpd indicam que o modelo prevê novas observações \tilde{y}_t bem, enquanto pequenos valores de elpd mostram que o modelo não generaliza bem para novos dados.

No entanto, na prática, a verdadeira densidade de probabilidade $f(\tilde{y}_t)$ é desconhecida, e é por isso que usamos métodos de validação cruzada (CV) como: *Leave-One-Out* (LOO-CV) e *Leave-Future-Out* (LFO-CV) ou o critério de informação de Watanabe (WAIC), para estimar o elpd para um modelo [Vehtari, Gelman e Gabry \(2017\)](#).

4.2 Validação cruzada *Leave-One-Out*

Na validação cruzada *Leave-One-Out* (LOO-CV), para um conjunto de dados de n observações, estimaremos $n - 1$ modelos. Ou seja, deixamos uma observação de fora e estimamos o modelo usando os dados sem essa observação e repetimos para todas as observações do conjunto de dados. Segundo [Vehtari, Gelman e Gabry \(2017\)](#), podemos calcular a medida elpd usando LOO-CV, da seguinte maneira:

$$\text{elpd}_{\text{loo}} = \sum_{t=1}^n \log p(y_t | y_{-t}),$$

onde

$$p(y_t | y_{-t}) = \int p(y_t | \theta) p(\theta | y_{-t}) d\theta,$$

em que $p(y_t | y_{-t})$ é a densidade preditiva condicionada nos dados sem a observação t . Implementar LOO-CV implica reajustar o modelo n vezes, o que se torna computacionalmente caro. Para resolver esse problema, [Gelfand, Dey e Chang \(1992\)](#) propuseram estimar LOO por meio do uso de amostragem por importância com os seguintes pesos de importância:

$$r_t^s = \frac{1}{p(y_t | \theta^s)} \propto \frac{p(\theta^s | y_{-t})}{p(\theta^s | y)}.$$

A distribuição preditiva da amostragem por importância sem a t -ésima observação dos dados é estimada por:

$$p(\tilde{y}_t | y_{-t}) \approx \frac{\sum_{s=1}^S r_t^s p(\tilde{y}_t | \theta^s)}{\sum_{s=1}^S r_t^s}.$$

Avaliando a log densidade preditiva LOO no ponto de dados retido y_t , obtemos

$$p(y_t | y_{-t}) \approx \frac{1}{\frac{1}{S} \sum_{s=1}^S \frac{1}{p(y_t | \theta^s)}}. \quad (4.3)$$

No entanto, de acordo com [Vehtari, Gelman e Gabry \(2017\)](#), é provável que a posteriori $p(\theta | y)$ tenha uma variância menor e caudas mais finas do que as distribuições sem a t -ésima observação $p(\theta | y_{-t})$. Portanto, o uso direto de (4.3) pode induzir instabilidade, uma vez que os pesos de importância r_t^s podem ter alta variância. Como resultado, a estimativa LOO usando amostragem por importância pode ser imprecisa.

[Vehtari, Gelman e Gabry \(2017\)](#) propuseram uma abordagem para aproximar o LOO-CV usando amostragem de importância com suavização de Pareto (*Pareto smoothed importance sampling* – PSIS) ([VEHTARI et al., 2015](#)). O PSIS nos permite calcular o LOO usando pesos de importância que, de outra forma, seriam instáveis. Em resumo, o PSIS executa as seguintes etapas para cada observação do conjunto de dados $t = 1, \dots, n$.

1. Ajustar a distribuição de Pareto generalizada para os 20% maiores pesos de importância r_t^s , onde $s = 1, \dots, S$. Assim, para cada t , existe um vetor distinto de S pesos.
2. Substituir os maiores pesos M (por exemplo, $0, 2S$) pelos valores esperados das estatísticas de ordem da distribuição de Pareto generalizada ajustada

$$F^{-1} \left(\frac{z - 1/2}{M} \right), \quad z = 1, \dots, M,$$

onde F^{-1} é a inversa da distribuição acumulada da distribuição de Pareto generalizada. Rotular esses novos pesos suavizados como \tilde{w}_t^s .

3. Para garantir a variância finita da estimativa, truncar e rotular cada peso suavizado da seguinte forma:

$$w_t^s = \min(\tilde{w}_t^s, S^{3/4} \bar{w}_t),$$

onde $\bar{w}_t = \frac{1}{S} \sum_{s=1}^S \tilde{w}_t^s$. O resultado é um vetor de pesos suavizados w_t^s , onde $s = 1, \dots, S$.

Assim, a estimativa PSIS-LOO do elpd é dada por:

$$\widehat{\text{elpd}}_{\text{psis-loo}} = \sum_{t=1}^n \log \left(\frac{\sum_{s=1}^S w_t^s p(y_t | \theta^s)}{\sum_{s=1}^S w_t^s} \right).$$

Cada observação fornece um parâmetro de forma estimado \hat{k} na distribuição generalizada de Pareto, que pode ser usado para avaliar a confiabilidade da estimativa PSIS-LOO. Essa avaliação pode ser feita da seguinte forma:

- Se $k < \frac{1}{2}$, a variância dos r_t^s é finita e a estimativa converge rapidamente.
- Se $\frac{1}{2} < k < 1$, a variância dos r_t^s é infinita, mas a média existe e a convergência da estimativa é mais lenta. A variância da estimativa PSIS-LOO é finita, mas pode ser grande.
- Se $k > 1$, a variância da estimativa PSIS-LOO é finita, mas pode ser grande.

Quando $\hat{k} > 0,7$, [Vehtari, Gelman e Gabry \(2017\)](#) recomendam amostrar diretamente de $p(\theta^s | y_{-t})$ para as observações problemáticas. Se houver muitas observações problemáticas, é recomendável utilizar métodos computacionais como *K-fold cross-validation*, ou ajustar um modelo mais robusto ao conjunto de dados.

Para estar na escala do desvio, a estimativa PSIS-LOO é dada por:

$$\text{LOO} = -2\widehat{\text{elpd}}_{\text{psis-loo}}$$

O modelo que melhor se ajustou ao conjunto de dados é aquele que obteve o menor valor do critério LOO. [Vehtari, Gelman e Gabry \(2017\)](#) implementaram os cálculos em um pacote R chamado `loo` ([VEHTARI et al., 2022](#)) e demonstraram o uso ajustando modelos por meio do software de inferência Bayesiana Stan ([CARPENTER et al., 2017](#)). O pacote R `loo` fornece a função `loo()` para calcular eficientemente a estimativa PSIS-LOO para modelos Bayesianos, usando os métodos descritos acima. A função `loo()` retorna $\widehat{\text{elpd}}_{\text{loo}}$, $\text{looic} = -2\widehat{\text{elpd}}_{\text{loo}}$ e as estimativas para o parâmetro de forma \hat{k} com uma mensagem informando ao usuário se algum \hat{k} é problemático.

4.3 Critério de informação de Watanabe

O critério de informação de Watanabe (WAIC), proposto por [Watanabe e Oppen \(2010\)](#), é um critério totalmente Bayesiano, no sentido de que usa toda a distribuição a posteriori e é assintoticamente igual ao LOO-CV. Assim como o LOO-CV, o critério WAIC também é uma abordagem para calcular a medida de precisão preditiva elpd , da seguinte maneira:

$$\widehat{\text{elpd}}_{\text{waic}} = \widehat{\text{lpd}} - \hat{p}_{\text{waic}},$$

onde $\widehat{\text{lpd}}$ é a estimativa do log da densidade preditiva pontual, dada por:

$$\widehat{\text{lpd}} = \sum_{t=1}^n \log p(y_t | y) = \sum_{t=1}^n \log \int p(y_t | \theta) p(\theta | y) d\theta,$$

e \hat{p}_{waic} é o número efetivo de parâmetros, que podemos calcular usando a variância posterior do log da densidade preditiva para cada observação y_t :

$$\hat{p}_{\text{waic}} = \sum_{t=1}^n V_{s=1}^S (\log p(y_t | \theta^s)),$$

onde $V_{s=1}^S$ representa a variância amostral, definida por $V_{s=1}^S a_s = \frac{1}{S-1} \sum_{s=1}^S (a_s - \bar{a})^2$. Logo, para estar na escala do desvio, o critério WAIC é estimado por:

$$\text{WAIC} = -2\widehat{\text{elpd}}_{\text{waic}}.$$

O menor valor do WAIC indica melhor modelo.

Um dos objetivos ao analisar séries temporais é fazer previsões. Neste contexto, [Bürkner, Gabry e Vehtari \(2020\)](#) afirmam que o critério LOO-CV não seria o mais adequado para selecionar modelos, uma vez que as informações de observações futuras estão disponíveis para influenciar as previsões do passado. Para contornar esse problema [Bürkner, Gabry e Vehtari \(2020\)](#) propuseram a validação cruzada *Leave-Future-Out* (LFO-CV), que será utilizada nesta dissertação para avaliar a precisão preditiva dos modelos SV.

4.4 Validação cruzada *Leave-Future-Out*

Nesta seção, descreveremos a validação cruzada *Leave-Future-Out* (LFO-CV). A implementação deste método consiste em duas etapas: a) definir o número mínimo de observações da série (L) necessárias antes de fazer previsões, e b) estimar o parâmetro θ com base nas primeiras t observações da série temporal (y_1, \dots, y_n) e calcular as densidades preditivas $p(y_{t+1} | y_{1:t})$, para cada $t \in \{L, \dots, n-1\}$.

Usando o conjunto de dados disponíveis, podemos obter a precisão preditiva para previsões 1 passo à frente (*1-step-ahead predictions*, 1-SAP). A medida de precisão preditiva elpd é definida como:

$$\text{elpd} = \sum_{t=L}^{n-1} \int f(\tilde{y}_{t+1}) \log p(\tilde{y}_{t+1} | y_{1:t}) d\tilde{y}_{t+1},$$

onde $y_{1:t} = (y_1, \dots, y_t)$. O problema é que não conhecemos a verdadeira distribuição $f(\tilde{y}_{t+1})$ do processo de geração de novos dados \tilde{y}_{t+1} . Em vez disso, [Bürkner, Gabry e Vehtari \(2020\)](#) aproximaram o elpd substituindo a realização y_{t+1} por \tilde{y}_{t+1} , e assim

$$\text{elpd}_{\text{lfo}} = \sum_{t=L}^{n-1} \log p(y_{t+1} | y_{1:t}).$$

O modelo com o maior valor de elpd_{lfo} é aquele que possui o melhor desempenho preditivo. Podemos calcular a densidade preditiva posterior $p(y_{t+1} | y_{1:t})$ com a ajuda da distribuição a posteriori $p(\theta | y_{1:t})$, mas com base apenas nas primeiras t observações:

$$p(y_{t+1} | y_{1:t}) = \int p(y_{t+1} | y_{1:t}, \theta) p(\theta | y_{1:t}) d\theta. \quad (4.4)$$

A integral em (4.4) não pode ser resolvida analiticamente, mas pode ser aproximada usando métodos de Monte-Carlo. Para obter S amostras aleatórias $(\theta_{1:t}^{(1)}, \dots, \theta_{1:t}^{(S)})$ da distribuição

a posteriori $p(\theta | y_{1:t})$, nós usamos o algoritmo *No-U-Turn Sampler* (NUTS) (HOFFMAN; GELMAN *et al.*, 2014). Assim, estimamos a densidade preditiva da seguinte forma:

$$p(y_{t+1} | y_{1:t}) \approx \frac{1}{S} \sum_{s=1}^S p(y_{t+1} | y_{1:t}, \theta_{1:t}^{(s)}).$$

Embora o LFO-CV forneça o ELPD exato, este critério é computacionalmente caro, especialmente para grandes conjuntos de dados, pois seria necessário reajustar o modelo $n - (L + 1)$ vezes.

Bürkner, Gabry e Vehtari (2020) propõem um método que utiliza a amostragem de importância com suavização de Pareto (PSIS) para aproximar o LFO-CV exato, reduzindo drasticamente os custos computacionais. Essa aproximação é denominada PSIS-LFO-CV. Nesta dissertação, será utilizada a versão *backward* PSIS-LFO-CV. Começando com $t = n - 1$, a densidade preditiva é aproximada por:

$$p(y_{t+1} | y_{1:t}) \approx \frac{\sum_{s=1}^S w_t^{(s)} p(y_{t+1} | y_{1:t}, \theta^{(s)})}{\sum_{s=1}^S w_t^{(s)}},$$

onde $\theta^{(s)}$ são amostras da distribuição a posteriori com base em todas as observações e $w_t^{(s)}$ são os pesos estabilizados mediante PSIS, calculados a partir de

$$\begin{aligned} r_t^{(s)} = r_t(\theta^{(s)}) &= \frac{f_{1:t}(\theta^{(s)})}{f_{1:t^*}(\theta^{(s)})} \propto \frac{p(\theta^{(s)}) \prod_{j \in 1:t} p(y_j | y_{1:(j-1)}, \theta^{(s)})}{p(\theta^{(s)}) \prod_{j \in 1:t^*} p(y_j | y_{1:(j-1)}, \theta^{(s)})} \\ &= \frac{1}{\prod_{j \in (t+1):t^*} p(y_j | y_{1:(j-1)}, \theta^{(s)})}, \end{aligned}$$

PSIS é usado para estabilizar os pesos $r_t^{(s)}$, conforme descrito anteriormente na Seção 4.2. A função $f_{1:t}$ denota a distribuição a posteriori baseada nas primeiras observações t , ou seja, $f_{1:t} = p(\theta | y_{1:t})$, com $f_{1:t^*}$ definida de forma análoga. O conjunto de índices $(t + 1) : t^*$ indica todas as observações que fazem parte dos dados para o modelo realmente ajustado $f_{1:t^*}$, mas não para o modelo cujo desempenho preditivo estamos tentando aproximar, $f_{1:t}$. A afirmação proporcional surge do fato de que ignoramos as constantes normalizadoras $p(y_{1:t})$ e $p(y_{1:t^*})$ das posteriores comparadas.

A partir de $t = n - 1$, diminuimos gradualmente o valor de t em 1 (ou seja, retrocedemos no tempo) e repetimos o processo. Em alguma observação t , a variabilidade dos pesos de importância $r_t^{(s)}$ se tornará muito grande e a amostragem de importância falhará, e esse valor particular de t será rotulado como t_1^* . Para identificar o valor de t_1^* , verificamos em qual valor de t o parâmetro de forma estimado k da distribuição generalizada de Pareto é maior que 0,6. Somente então reajustamos o modelo usando apenas as observações até t_1^* e recomeçamos o processo, definindo $\theta^{(s)} = \theta_{1:t_1^*}^{(s)}$ assim como $i^* = i_1^*$ até o próximo reajuste. Essa abordagem é repetida até chegarmos a $t = L$.

ESTUDOS DE SIMULAÇÃO E APLICAÇÕES

Neste capítulo, apresentaremos estudos de simulação com o intuito de avaliar e comparar o desempenho do algoritmo NUTS para estimar os parâmetros da classe de modelos SV-SMSN apresentados no [Capítulo 3](#). Além disso, avaliaremos o desempenho dos critérios de informação DIC e WAIC, bem como das técnicas de validação cruzada LOO-CV e LFO-CV, apresentados no [Capítulo 4](#), para seleção de modelos SV. Também ajustamos e comparamos os modelos SV-SMSN a séries de retornos diários reais.

5.1 Estudo de simulação 1

O objetivo deste estudo de simulação é comparar a eficiência do algoritmo NUTS e do algoritmo MCMC implementado no pacote R `stochvol` ([DARJUS; GREGOR, 2021](#)). Para mais informações sobre o método MCMC implementado no pacote `stochvol`, consulte [Kastner e Frühwirth-Schnatter \(2014\)](#). A eficiência de ambos métodos de amostragem para estimar os parâmetros dos modelos SV-N, SV-NL, SV-T e SV-TL é calculada com base no tamanho efetivo de amostra (ESS), em inglês *Effective Sample Size*, definido por:

$$ESS = \frac{S}{1 + 2 \sum_{j=1}^{\infty} \rho_j},$$

em que S é o número de amostras MCMC e ρ_j é a autocorrelação na defasagem j . O ESS pode ser interpretado como o número de amostras efetivamente independentes. Quando o valor de ESS está próximo de S , significa que mais amostras independentes estão sendo fornecidas pelo algoritmo MCMC. Neste estudo de simulação, o ESS foi estimado mediante o pacote `coda` ([PLUMMER et al., 2006](#)). Na prática, a eficiência computacional é importante, e por isso, para comparar os dois algoritmos, também calculamos a razão entre o ESS e o tempo requerido do algoritmo para ajustar os modelos SV a cada série simulada.

Para a geração das séries, consideramos os modelos SV-N, SV-NL, SV-T e SV-TL. Para cada modelo, foram simuladas 100 séries de retornos de tamanho 1000, a partir das equações

(3.7), (3.8), (3.9) e (3.10). Os valores dos parâmetros foram escolhidos da seguinte forma:

$$(\mu, \phi, \sigma, \rho, \nu) = (-9; 0, 95; 0, 15; -0, 5; 5).$$

Essa escolha de parâmetros foi investigada previamente por Nakajima e Omori (2012) e Wang, Chan e Choy (2011). Valores similares de ν foram obtidos em aplicações empíricas em Zevallos, Gasco e Ehlers (2017). Como o objetivo é a comparação dos algoritmos de amostragem NUTS e `stochvol`, utilizamos as distribuições a priori implementadas no pacote `stochvol` (DARJUS; GREGOR, 2021):

$$\begin{aligned}\mu &\sim N(0, 10^2), \\ (\phi + 1)/2 &\sim \text{Beta}(20; 1, 5), \\ \sigma^2 &\sim \text{Gama}(0, 5; 0, 5), \\ (\rho + 1)/2 &\sim \text{Beta}(3, 6), \\ \nu &\sim \text{Exp}(1/3), \quad 2 < \nu.\end{aligned}$$

Os hiperparâmetros a priori de μ , ϕ e σ^2 foram escolhidos de acordo com Kim, Shephard e Chib (1998), enquanto uma priori levemente informativa sobre ρ foi proposta por Hosszejni e Kastner (2018). Para o parâmetro ν , utilizamos a distribuição a priori proposta por Zevallos, Gasco e Ehlers (2017).

Uma vez definidas as distribuições a priori, a cada uma das séries simuladas foram ajustados os respectivos modelos SV, usando o pacote `rstan` (Stan Development Team, 2022) versão 2.21.7 e o pacote R `stochvol` (DARJUS; GREGOR, 2021). Para ambos algoritmos de amostragem, foram executadas 42000 iterações e descartadas as primeiras 2000 iterações como período de aquecimento. A fim de reduzir a autocorrelação entre as amostras vizinhas, as amostras foram salvas a cada 8 iterações, resultando em uma amostra final de tamanho 5000.

Para cada modelo SV e algoritmo de amostragem, as Tabelas 2 e 3 apresentam a média a posteriori dos parâmetros, o tamanho efetivo de amostra (ESS) e o valor do ESS/s , onde s é o tempo médio de execução. A Tabela 1 mostra os tempos médios em segundos para ajustar os modelos SV utilizando os dois algoritmos de amostragem. Podemos observar que o pacote `stochvol` produz o menor tempo de execução em todos os modelos SV e é menor a um minuto.

Tabela 1 – Tempos médios em segundos para ajustar os modelos SV, utilizando o algoritmo NUTS e o pacote `stochvol`.

Modelo	NUTS	Stochvol
SV-N	217,511	11,961
SV-NL	311,169	20,327
SV-T	430,694	20,181
SV-TL	605,163	37,187

Conforme a Tabela 2, observamos que, no caso dos modelos SV-N e SV-T, as médias a posteriori dos parâmetros para ambos algoritmos são semelhantes e próximos dos valores

reais. No entanto, para todos os parâmetros, os valores de ESS são maiores quando utilizamos o algoritmo NUTS. Apesar disso, devido à eficiência computacional do pacote `stochvol`, o ESS/s é maior quando utilizamos esse pacote. Portanto, para estimar os parâmetros dos modelos SV-N e SV-T sugerimos usar o pacote `stochvol`, especialmente se o conjunto de dados for grande.

Tabela 2 – Resultados da comparação do algoritmo NUTS e o pacote `stochvol` para os modelos SV-N e SV-T em termos de ESS e ESS/s.

Modelo	Parâmetro	Método	Média	ESS	ESS/s
SV-N	μ	NUTS	-9,007	4810,010	22,114
		Stochvol	-9,008	4320,183	361,186
	ϕ	NUTS	0,917	4762,946	21,897
		Stochvol	0,918	721,306	60,304
	σ	NUTS	0,187	4697,016	21,594
		Stochvol	0,188	639,973	53,505
SV-T	μ	NUTS	-8,991	2529,192	5,872
		Stochvol	-8,500	870,128	43,117
	ϕ	NUTS	0,894	2974,873	6,907
		Stochvol	0,894	361,475	17,912
	σ	NUTS	0,210	2314,318	5,373
		Stochvol	0,210	300,791	14,905
	ν	NUTS	5,697	916,416	2,128
		Stochvol	5,698	258,128	12,791

Nota – Parâmetros verdadeiros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$ e $\nu = 5$.

A [Tabela 3](#) apresenta os resultados do estudo de simulação para os modelos SV-NL e SV-TL. Observamos que as médias a posteriori dos parâmetros, com exceção da estimativa do parâmetro ρ , são similares e próximas dos valores reais para ambos algoritmos. É importante notar que as médias a posteriori do parâmetro ρ estão mais próximas do valor real quando utilizamos o algoritmo NUTS. Em relação aos valores de ESS, o algoritmo NUTS é mais eficiente que o pacote `stochvol`. Também observamos que, para a maioria dos parâmetros, o valor do ESS/s é maior quando utilizamos o algoritmo NUTS, com exceção dos parâmetros μ e ν . Com base nesses resultados, sugerimos utilizar o algoritmo NUTS para estimar os parâmetros dos modelos SV-NL e SV-TL.

Em resumo, o algoritmo NUTS é mais eficiente em termos de tamanho efetivo de amostra, porém seu custo computacional é maior em comparação com o pacote `stochvol`. É importante notar que Stan é uma estrutura de modelagem probabilística de uso geral e tem um bom desempenho em comparação com o pacote `stochvol`, que foi desenvolvido especificamente para os modelos SV em questão. Outra vantagem do Stan é que permite a escolha de diferentes distribuições a priori para os parâmetros dos modelos SV, ao contrário do pacote R `stochvol`, que já tem distribuições a priori definidas.

Tabela 3 – Comparação dos amostradores NUTS e `stochvol` em termos de tamanho efetivo de amostra (ESS) e tamanho efetivo de amostra por segundo (ESS/s).

Modelo	Parâmetro	Método	Média	ESS	ESS/s
SV-NL	μ	NUTS	-9,003	4946,456	15,896
		Stochvol	-9,004	2717,088	133,668
	ϕ	NUTS	0,927	4853,599	15,598
		Stochvol	0,923	121,571	5,981
	σ	NUTS	0,178	4773,051	15,339
		Stochvol	0,185	100,095	4,924
	ρ	NUTS	-0,460	4977,328	15,996
		Stochvol	-0,406	126,696	6,233
SV-TL	μ	NUTS	-8,996	3061,499	5,059
		Stochvol	-8,476	700,847	18,847
	ϕ	NUTS	0,914	4067,206	6,721
		Stochvol	0,909	94,774	2,549
	σ	NUTS	0,188	3320,861	5,488
		Stochvol	0,197	79,833	2,147
	ρ	NUTS	-0,446	4422,661	7,308
		Stochvol	-0,379	108,179	2,909
	ν	NUTS	5,263	1262,553	2,086
		Stochvol	5,311	288,710	7,764

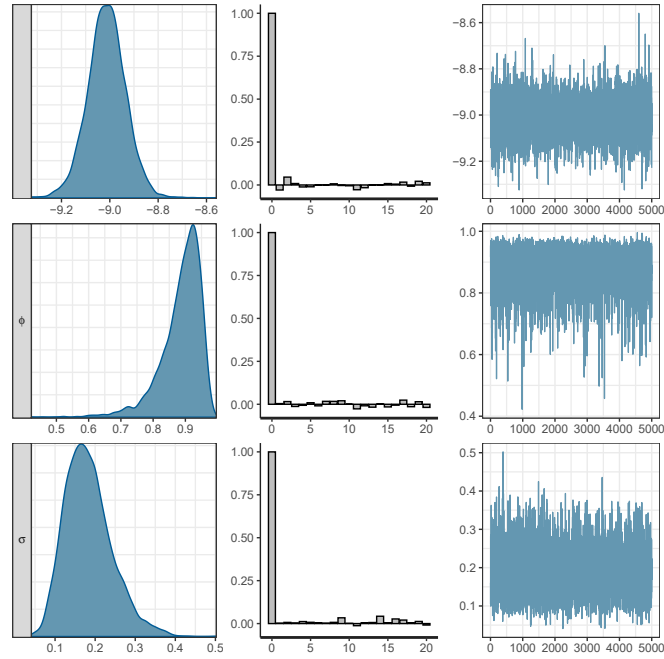
Nota – Parâmetros verdadeiros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$, $\rho = -0,5$ e $\nu = 5$.

Em ambas abordagens de amostragem utilizadas, verificamos a convergência das cadeias utilizando os gráficos de convergência. Para uma das séries simuladas a partir dos modelos SV-N, SV-NL, SV-T e SV-TL, as Figuras 1, 2, 3 e 4 mostram as densidades a posteriori dos parâmetros dos modelos SV, gráficos de autocorrelação e as saídas do MCMC, quando é utilizado o algoritmo NUTS e o pacote `stochvol`.

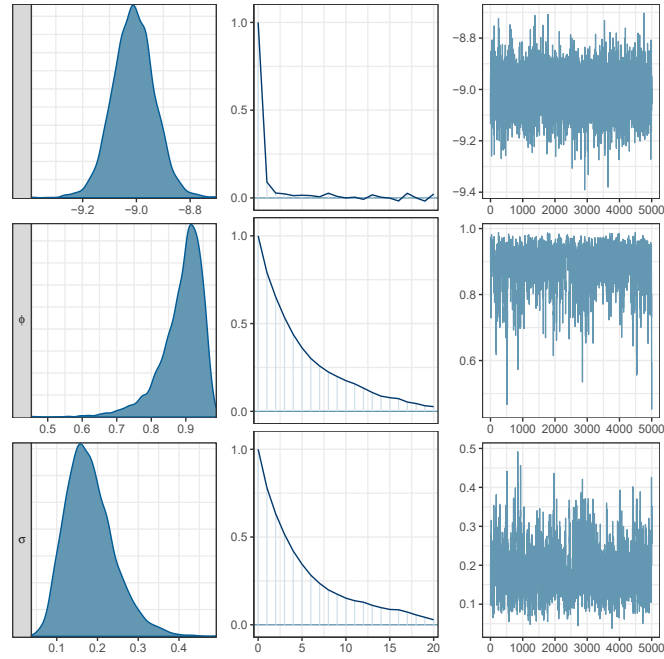
Para os modelos SV-N e SV-T, ambos algoritmos de amostragem NUTS e `stochvol` produzem densidades a posteriori que se concentram em torno dos valores verdadeiros. Além disso, os gráficos de autocorrelação mostram um declínio rápido da autocorrelação à medida que o intervalo de tempo entre as amostras aumenta.

Por outro lado, no caso dos modelos SV-NL e SV-TL, podemos observar que as cadeias MCMC resultantes usando o algoritmo NUTS apresentam menor valor de autocorrelação em comparação com as cadeias resultantes de usar o pacote `stochvol`. Além disso, no modelo SV-TL, a mistura de cadeias é melhor ao usar o algoritmo NUTS. Finalmente, ao comparar os gráficos de autocorrelação resultantes de ambos algoritmos de amostragem, observamos que o algoritmo NUTS produz as autocorrelações mais baixas em todos os modelos SV.

Figura 1 – Gráficos de convergência dos parâmetros do modelo SV-N ajustado a uma série simulada, usando o algoritmo NUTS e o pacote `stochvol`.



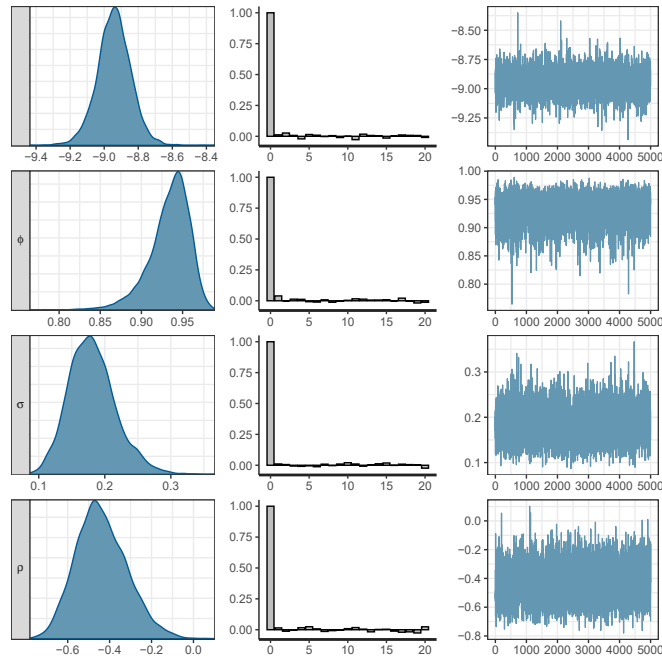
(a) NUTS



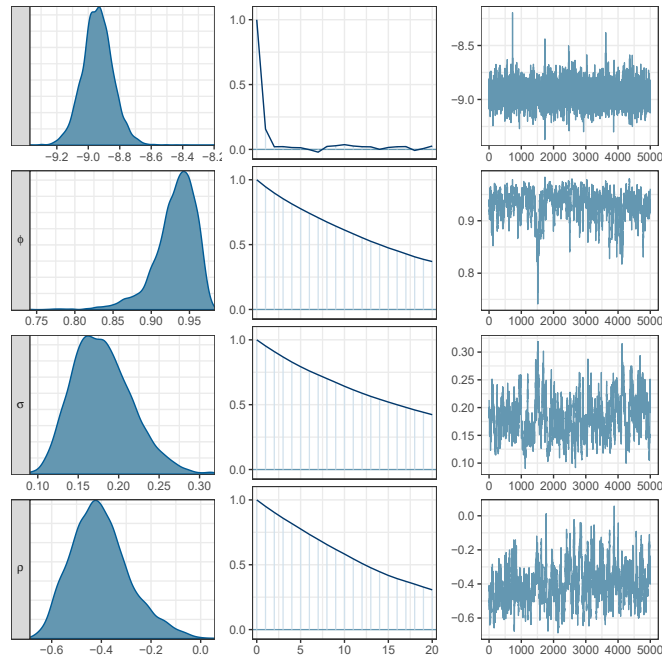
(b) `stochvol`

Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$ e $\sigma = 0,15$.

Figura 2 – Gráficos de convergência dos parâmetros do modelo SV-NL ajustado a uma série simulada, usando o algoritmo NUTS e o pacote `stochvol`.



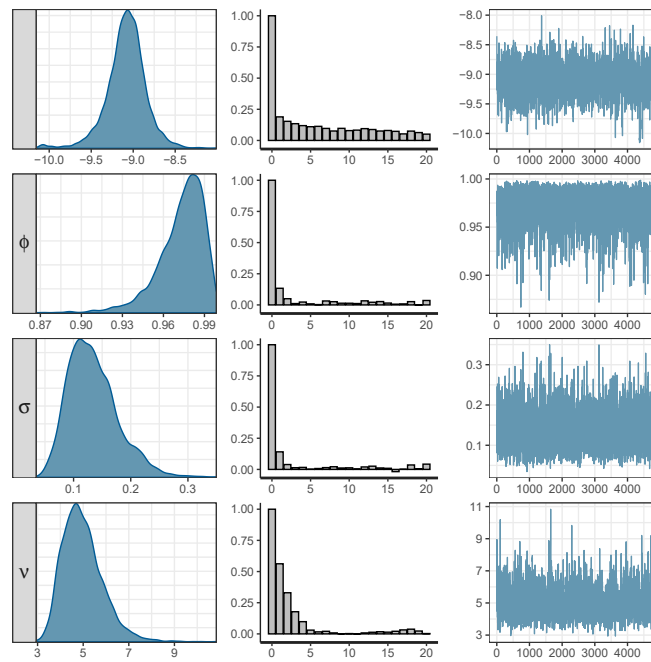
(a) NUTS



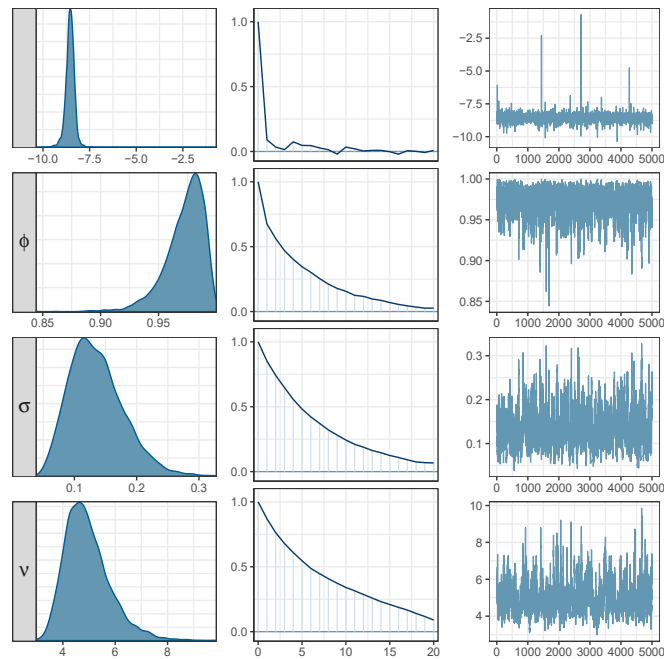
(b) `stochvol`

Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$ e $\rho = -0,5$.

Figura 3 – Gráficos de convergência dos parâmetros do modelo SV-T ajustado a uma série simulada, usando o algoritmo NUTS e o pacote `stochvol`.



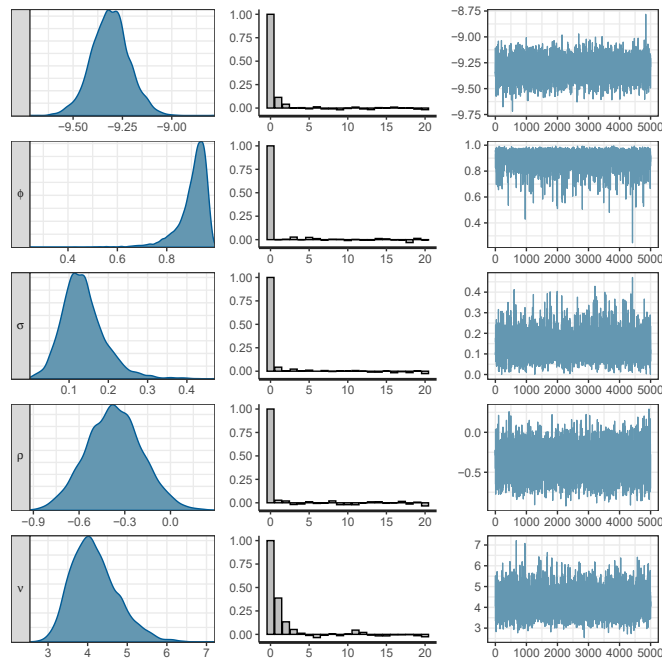
(a) NUTS



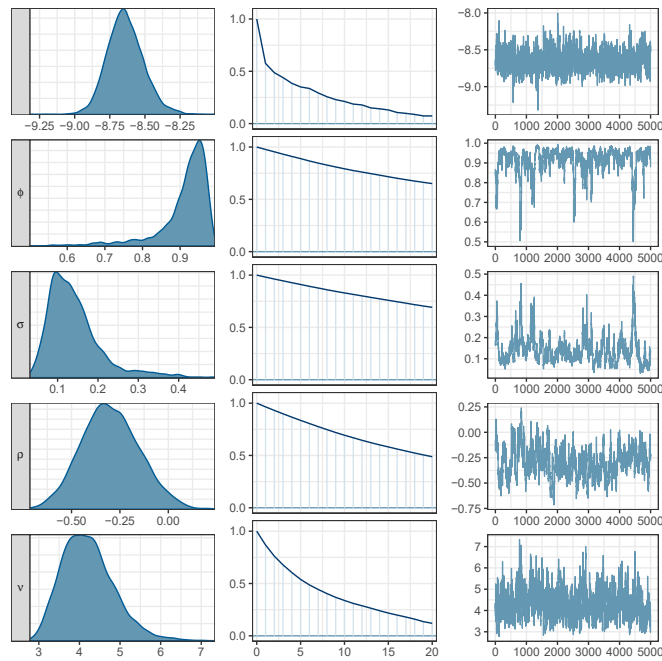
(b) `stochvol`

Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$ e $v = 5$.

Figura 4 – Gráficos de convergência dos parâmetros do modelo SV-TL ajustado a uma série simulada, usando o algoritmo NUTS e o pacote `stochvol`.



(a) NUTS



(b) `stochvol`

Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saída do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$, $\rho = -0,5$ e $v = 5$.

5.2 Estudo de simulação 2

Este estudo de simulação busca avaliar o desempenho do algoritmo NUTS para estimar os parâmetros dos modelos SV-SMSN. Primeiramente, geramos 100 conjuntos de séries de 1000 observações a partir dos modelos SV-STL, SV-TL, SV-T, SV-NL, SV-N e SV-SSL, utilizando as equações (3.6), (3.7), (3.8), (3.9), (3.10) e (3.5), respectivamente (ver no [Apêndice B](#) o [Código-fonte 1](#)). As séries de retornos foram simuladas definindo os seguintes valores para os parâmetros em comum dos modelos SV:

$$(\mu, \phi, \sigma) = (-9; 0, 95; 0, 15).$$

Assumimos o valor de -0,5 para os parâmetros de alavancagem (ρ) e assimetria (α). O parâmetro que controla as caudas pesadas da distribuição t-Student assimétrica foi fixado em $\nu = 5$, seguindo [Wang, Chan e Choy \(2011\)](#). Enquanto para a distribuição slash assimétrica assumimos $\nu = 2$. Estes valores de parâmetros, exceto pelo valor de ν , foram utilizados por [Nakajima e Omori \(2012\)](#). A escolha de ν da distribuição slash assimétrica foi baseada em resultados empíricos obtidos em [Abanto-Valle *et al.* \(2010\)](#) e [Abanto-Valle e Garrafa-Aragón \(2019\)](#). Assumimos as seguintes priors para os parâmetros:

$$\begin{aligned} \mu &\sim N(0, 10^2), \\ (\phi + 1)/2 &\sim \text{Beta}(20; 1, 5), \\ \sigma^2 &\sim \text{IG}(2, 5; 0, 025), \\ (\rho + 1)/2 &\sim \text{Beta}(3, 6), \\ \nu &\sim \text{Exp}(1/3), \quad 2 < \nu, \quad \text{para os modelos SV-STL, SV-TL e SV-T}, \\ \nu &\sim \text{Gama}(0, 08; 0, 04), \quad \nu > 1, \quad \text{para o modelo SV-SSL}, \\ \delta &\sim U(-1, 1). \end{aligned}$$

Adotamos as mesmas distribuições a priori de [Kim, Shephard e Chib \(1998\)](#) para os parâmetros (μ, ϕ, σ) . A priori do parâmetro ρ foi definida por [Hosszejni e Kastner \(2018\)](#). A escolha da distribuição a priori do parâmetro ν da distribuição t-Student foi baseada em [Zevallos, Gasco e Ehlers \(2017\)](#), enquanto a distribuição a priori do parâmetro de forma ν da distribuição slash seguiu [Abanto-Valle e Garrafa-Aragón \(2019\)](#). Para o parâmetro δ , assumimos a priori sugerida por [Huang e Xu \(2014\)](#).

Ajustamos os respectivos modelos SV a cada série simulada, utilizando o algoritmo NUTS por meio do pacote R `rstan` ([Stan Development Team, 2022](#)). Executamos simultaneamente cinco cadeias, cada uma com 10000 iterações, das quais descartamos as 2000 primeiras amostras como período de aquecimento. Além disso, para reduzir as autocorrelações entre os valores sucessivos simulados, salvamos somente uma a cada oito amostras dentre as 8000 amostras restantes. Assim, obtivemos uma amostra final com tamanho 5000.

A fim de avaliar o desempenho do algoritmo NUTS, calculamos as seguintes medidas: o viés e a raiz quadrada do erro quadrático médio (RMSE), definidas a continuação:

$$\text{viés} = \frac{1}{m} \sum_{i=1}^m \left(\hat{\theta}^{(i)} - \theta \right),$$

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m \left(\hat{\theta}^{(i)} - \theta \right)^2},$$

em que m é o número de séries simuladas (no nosso caso $m = 100$), θ é o verdadeiro valor do parâmetro e $\hat{\theta}^{(i)}$ é a média a posteriori do parâmetro θ na i -ésima série de retornos simulada. Valores menores da medida RMSE indicam estimativas mais precisas.

Os resultados deste estudo de simulação são apresentados na [Tabela 4](#). Para cada modelo SV, na primeira e segunda linha, respectivamente, apresentamos as medidas viés e RMSE de cada parâmetro. Observamos que os valores de viés são próximos de zero para todos os parâmetros dos modelos SV, o que indica que as estimativas obtidas dos parâmetros estão próximas dos valores verdadeiros. Assim, segundo os valores do viés, foram obtidos bons resultados em todos os modelos SV considerados. Além disso, o algoritmo NUTS tende a subestimar o verdadeiro valor dos parâmetros ϕ e σ em todos os modelos SV.

Com relação aos valores de RMSE, obtivemos bons resultados para os parâmetros μ , ϕ e σ em todos os modelos SV. Com relação ao parâmetro ρ , foram obtidos resultados razoáveis. No entanto, as estimativas dos parâmetros ν e α apresentaram valores de RMSE relativamente altos.

Tabela 4 – Viés e RMSE para os parâmetros na classe de modelos SV-SMSN.

Modelo	Medidas	μ	ϕ	σ	ρ	ν	α
SV-N	Viés	0,003	-0,008	-0,003	—	—	—
	RMSE	0,104	0,023	0,028	—	—	—
SV-NL	Viés	0,005	-0,006	-0,004	0,019	—	—
	RMSE	0,106	0,023	0,024	0,104	—	—
SV-T	Viés	-0,002	-0,016	-0,012	—	0,216	—
	RMSE	0,126	0,036	0,028	—	0,906	—
SV-TL	Viés	-0,002	-0,012	-0,010	0,023	0,029	—
	RMSE	0,125	0,030	0,028	0,114	0,698	—
SV-STL	Viés	-0,029	-0,013	-0,013	0,010	0,082	0,079
	RMSE	0,173	0,031	0,028	0,117	0,810	0,195
SV-SSL	Viés	0,013	-0,009	-0,010	0,012	0,079	0,083
	RMSE	0,159	0,029	0,026	0,124	0,349	0,223

Nota – Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$, $\rho = -0,5$, $\nu = 5$, $\nu = 2$ (SV-SSL) e $\alpha = -0,5$.

Após executar o algoritmo NUTS, é necessário avaliar a convergência das amostras MCMC obtidas. Nesta dissertação, a convergência das cadeias foi verificada usando a estatística \hat{R} de Gelman e Rubin ([GELMAN; RUBIN, 1992](#)). Verificamos que os valores de \hat{R} foram

próximos de 1, o que indica convergência. Além disso, a convergência foi diagnosticada usando análises gráficas.

As Figuras 5 a 10 apresentam as densidades a posteriori, gráficos de autocorrelação e traços dos parâmetros nos modelos SV, para uma das séries simuladas. Verifica-se que a densidade a posteriori dos parâmetros se concentra em torno do valor fixado. Também observamos que as funções de autocorrelação decaem rapidamente, segundo as saídas do MCMC, as estimativas de cada parâmetro mostraram-se bem próximas dos verdadeiros valores dos parâmetros. Esses resultados confirmam que as cadeias dos parâmetros convergiram.

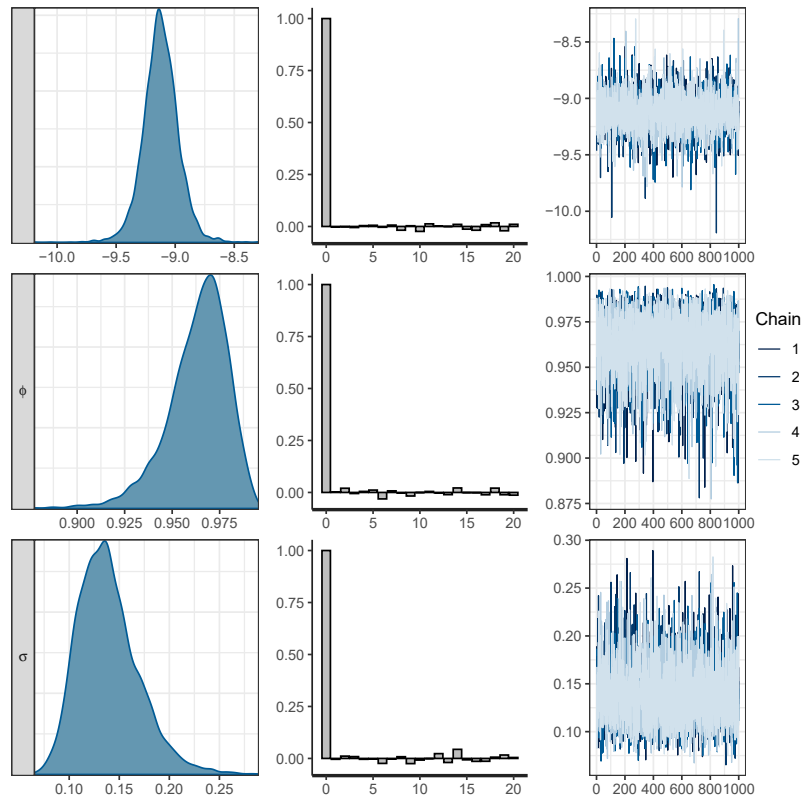
Finalmente, a Tabela 5 apresenta os tempos médios em minutos para ajustar os respectivos modelos SV a cada série simulada. Em relação à eficiência computacional, observamos que o ajuste dos modelos SV-STL e SV-SSL são os mais demorados. Por exemplo, para ajustar os modelos SV-STL e SV-SSL a uma série de retornos de tamanho 1000, precisamos, respectivamente, de 7 e 9 minutos.

Tabela 5 – Tempos médios em minutos para ajustar os modelos SV às séries simuladas utilizando o algoritmo NUTS.

Modelo	Tempo (min)
SV-N	0,89
SV-NL	1,42
SV-T	1,75
SV-TL	2,62
SV-STL	6,6
SV-SSL	9,17

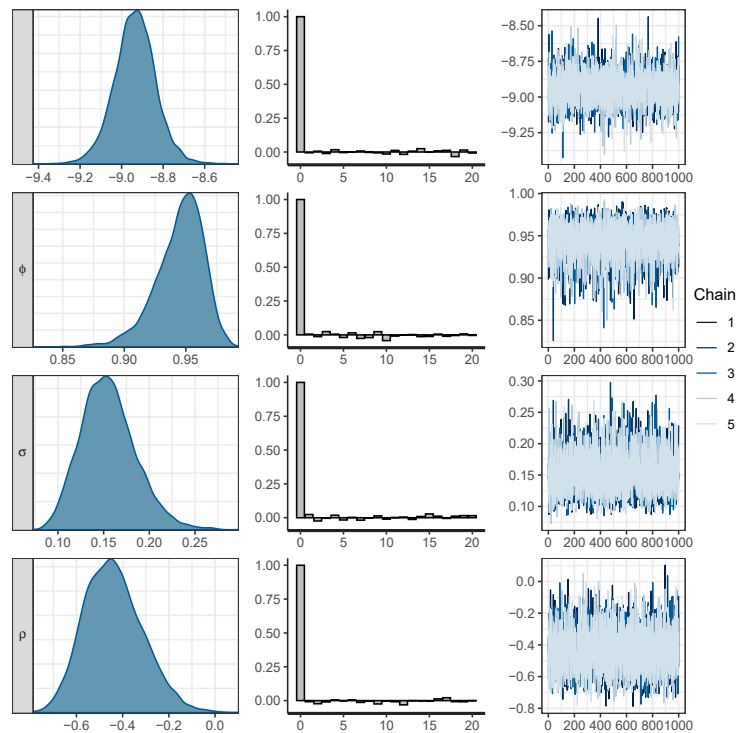
Em resumo, os resultados deste estudo de simulação mostram que o algoritmo NUTS consegue estimar bem os parâmetros na classe de modelos SV-SMSN. Além disso, pode-se observar na Tabela 5 que o tempo computacional de ajuste dos modelos SV foi razoavelmente bom. Em suma, para os modelos SV-SMSN, recomendamos o uso do algoritmo NUTS para o usuário aplicado lidando com séries de retornos financeiros diários.

Figura 5 – Gráficos de convergência dos parâmetros do modelo SV-N.



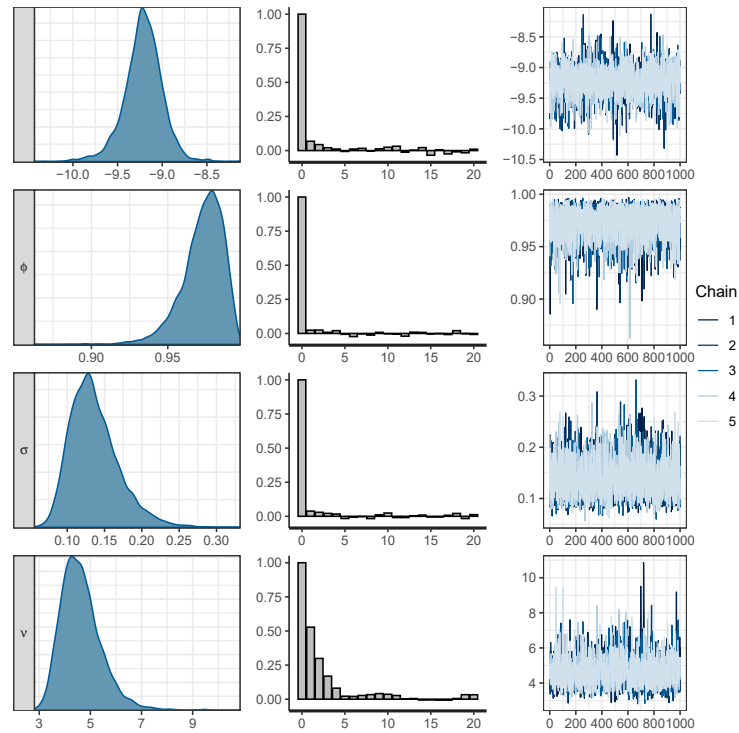
Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$ e $\sigma = 0,15$.

Figura 6 – Gráficos de convergência dos parâmetros do modelo SV-NL.



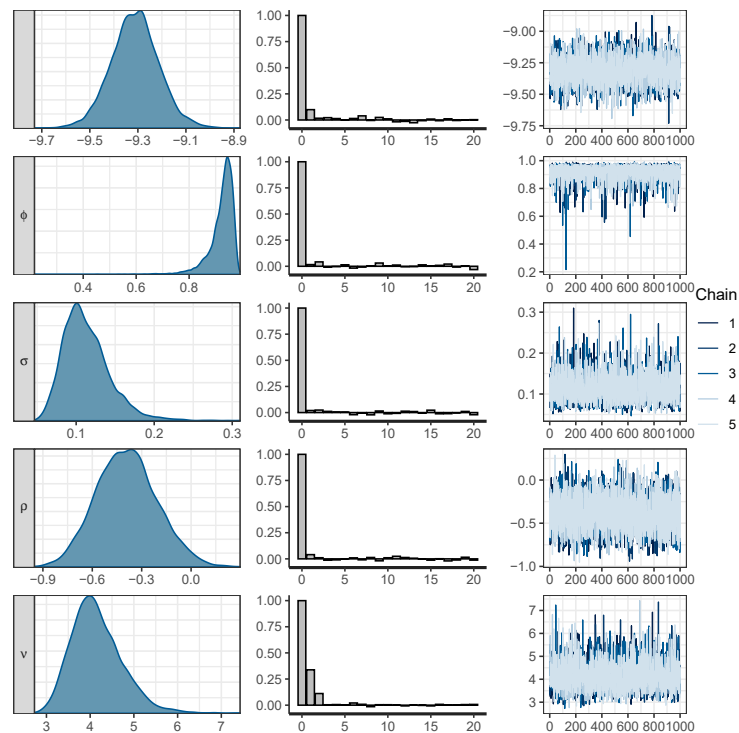
Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$ e $\rho = -0,5$.

Figura 7 – Gráficos de convergência dos parâmetros do modelo SV-T.



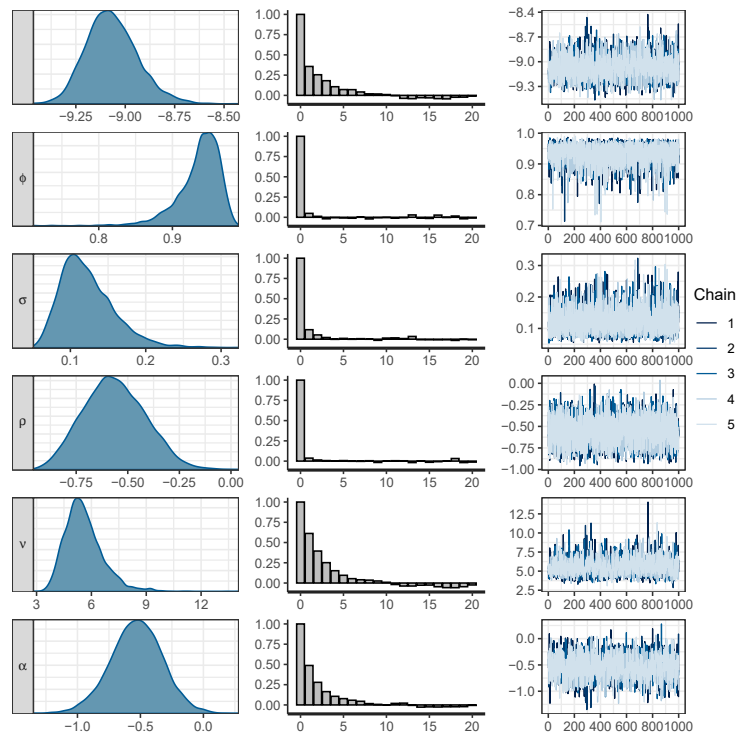
Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$ e $\nu = 5$.

Figura 8 – Gráficos de convergência dos parâmetros do modelo SV-TL.



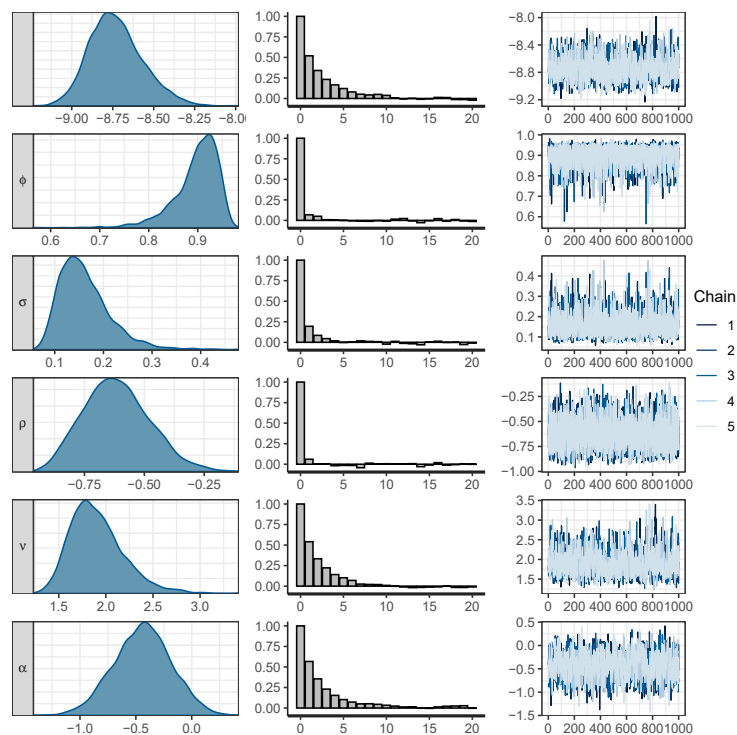
Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$, $\rho = -0,5$ e $\nu = 5$.

Figura 9 – Gráficos de convergência dos parâmetros do modelo SV-STL.



Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$, $\rho = -0,5$, $\nu = 5$ e $\alpha = -0,5$.

Figura 10 – Gráficos de convergência dos parâmetros do modelo SV-SSL.



Nota – Densidades a posteriori (esquerda), gráficos de autocorrelação (meio) e saídas do MCMC (direita).
Parâmetros: $\mu = -9$, $\phi = 0,95$, $\sigma = 0,15$, $\rho = -0,5$, $\nu = 2$ e $\alpha = -0,5$.

5.3 Estudo de simulação 3

Neste estudo de simulação, foi avaliado o desempenho dos métodos de seleção de modelos descritos no [Capítulo 4](#). Os códigos para implementar a técnica de validação cruzada LFO-CV podem ser encontrados no [Apêndice B](#).

Foram geradas 50 séries de retornos de 1000 observações cada uma, a partir dos seguintes modelos: a) modelo SV-STL (mais complexo) b) modelo SV-TL (mais simples). Os valores fixados para os parâmetros foram:

$$(\mu; \phi; \sigma; \rho; \nu; \alpha) = (-9; 0,95; 0,15; -0,5; 5; -0,5).$$

Para os parâmetros dos modelos, as distribuições a priori são as definidas na [Seção 5.2](#). Posteriormente foram ajustados todos os modelos SV candidatos a cada uma das séries simuladas, por meio do pacote `rstan` ([Stan Development Team, 2022](#)). Executamos 5 cadeias simultaneamente cada uma com 7000 iterações, das quais as 2000 primeiras foram descartadas como amostras de aquecimento. Das amostras restantes foram salvas somente a cada quinta amostra, resultando assim em uma amostra final de tamanho 5000.

Para selecionar o modelo mais adequado para cada série simulada, calculamos os métodos de seleção de modelos descritos no [Capítulo 4](#). Para o cálculo do WAIC e LOO-CV, utilizamos o pacote `Rloo` ([VEHTARI et al., 2022](#)), que calcula LOO por meio de amostragem de importância suavizada de Pareto. Para o cálculo do DIC, escrevemos um programa R para extrair as amostras posteriores produzidos pelo `rstan` e calculamos o DIC com base nas equações (4.1) e (4.2). O código para calcular o DIC e LFO-CV está disponível no [Apêndice B](#). Para o LFO-CV, o número mínimo de observações antes de fazer previsões foi definido como $L = 700$.

Os resultados do número de vezes que cada critério selecionou o verdadeiro modelo quando as séries foram simuladas a partir do modelo SV-STL e SV-TL são apresentados na [Tabela 6](#) e [Tabela 7](#), respectivamente. Como os modelos usados para gerar os dados são os modelos SV-STL e SV-TL, um bom critério de informação deve conseguir escolher esses modelos na maioria das vezes.

É possível notar pelas [Tabelas 6 e 7](#) que o critério DIC não conseguiu identificar o modelo SV do qual foram gerados os dados, tendo selecionado erroneamente o modelo SV-SSL. O critério WAIC teve um bom desempenho quando o verdadeiro modelo foi o SV-STL. Porém, quando os dados foram gerados usando o modelo SV-TL, ele só conseguiu identificar o verdadeiro modelo sete vezes.

A técnica de validação cruzada LOO-CV teve um bom desempenho em ambos cenários considerados. Quando o verdadeiro modelo foi o SV-STL, o LFO-CV teve um desempenho inferior ao LOO-CV, enquanto que, quando os dados foram gerados a partir do modelo SV-TL, o desempenho do LFO-CV foi semelhante ao LOO-CV.

Tabela 6 – Frequências de seleção de modelo para as 50 séries simuladas a partir do modelo SV-STL.

Modelo	Métodos de seleção de modelo			
	DIC	WAIC	LOO-CV	LFO-CV
SV-N	0	0	0	3
SV-NL	0	0	0	0
SV-T	0	0	1	2
SV-TL	0	0	2	6
SV-STL	0	49	45	29
SV-SSL	50	1	2	10

Tabela 7 – Frequências de seleção de modelo para as 50 séries simuladas a partir do modelo SV-TL.

Modelo	Métodos de seleção de modelo			
	DIC	WAIC	LOO-CV	LFO-CV
SV-N	0	0	0	0
SV-NL	0	0	0	4
SV-T	0	4	8	8
SV-TL	0	7	15	14
SV-STL	0	39	27	14
SV-SSL	50	0	0	10

A [Tabela 8](#) apresenta o número de reajustes necessários para os dois cenários considerados. Por exemplo, quando o verdadeiro modelo é o SV-STL, para calcular a medida $elpd_{lfo}$ para o modelo SV-SSL foi necessário ajustar o modelo SV-SSL cerca de 44 vezes em cada uma das 50 séries simuladas.

Tabela 8 – Número médio de reajustes necessários para calcular a medida $elpd_{lfo}$.

Modelo	SV-STL	SV-TL
SV-N	45	47
SV-NL	42	45
SV-T	41	42
SV-TL	39	42
SV-STL	52	46
SV-SSL	44	34

Nota – Número mínimo de observações para fazer previsões $L=700$.

Com base nos resultados obtidos, podemos afirmar que a validação cruzada LOO-CV apresentou o melhor desempenho e baixo custo computacional na seleção de modelos SV. Por outro lado, o critério DIC apresentou o pior desempenho, o que é uma descoberta importante, já que o DIC é amplamente utilizado em aplicações empíricas.

No entanto, a comparação de modelos SV usando LFO-CV envolve procedimentos computacionalmente pesados, o que limita sua viabilidade em grandes conjuntos de dados. Esta é uma pesquisa ainda em andamento, pois o artigo de [Bürkner, Gabry e Vehtari \(2020\)](#) não propõe um critério de escolha do valor mínimo de observações L . Portanto, para este estudo de simulação, alguns valores de L foram testados, o que dificulta o uso do LFO-CV em séries de retornos reais.

Em resumo, LOO-CV e LFO-CV são dois métodos de seleção de modelos totalmente Bayesianos que demonstraram ter um desempenho melhor do que outros critérios comuns, como WAIC e DIC, no contexto da seleção de modelos de volatilidade estocástica.

5.4 Aplicação a dados reais

Nesta seção, ajustamos os modelos SV apresentados no [Capítulo 3](#) às seguintes quatro séries de retornos diários: o índice Standard & Poor's 500 (S&P500) utilizado por [Huang e Xu \(2014\)](#) no período de 2 de janeiro de 1987 a 29 de dezembro de 1995, da empresa Microsoft (MSFT) utilizada por [Han, Lee et al. \(2020\)](#) de 3 de janeiro de 2006 a 30 de junho de 2015, a taxa de câmbio EUR/USD (EURUSD) e o preço da criptomoeda Bitcoin em USD (BTC) no período de 1 de janeiro de 2015 a 30 de junho de 2018, utilizado por [Hosszejni e Kastner \(2018\)](#). Os dados de preços de fechamento diários foram coletados no site de finanças do Yahoo, utilizando o pacote `quantmod` do R ([RYAN; ULRICH, 2022](#)).

Os retornos diários corrigidos pela média são calculados como:

$$y_t = (\log P_t - \log P_{t-1}) - \frac{1}{n} \sum_{j=2}^n (\log P_j - \log P_{j-1}), \quad t = 1, 2, \dots, n,$$

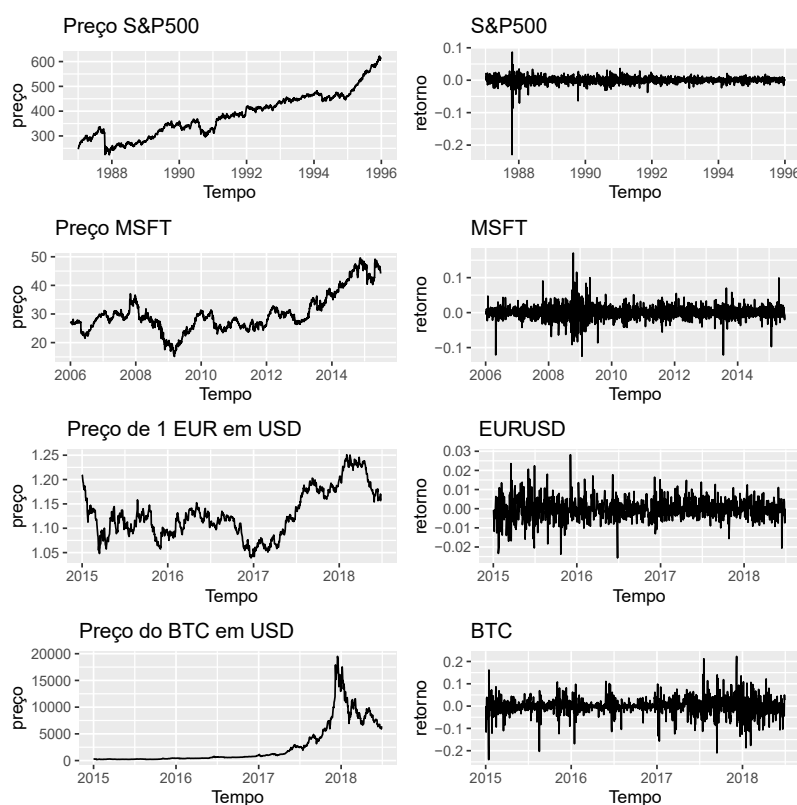
onde P_t é o preço de fechamento no dia t . A [Tabela 9](#) apresenta as estatísticas descritivas das séries de retornos analisadas. O desvio padrão da série de retornos BTC é maior do que das outras séries. Podemos observar que a série de retornos BTC é muito mais volátil do que as outras. É possível notar que a curtose de todas as séries de retornos é maior do que 3, o que pode ser considerado um indicativo da presença de retornos extremos. Também indicam que a distribuição dos retornos tem picos mais altos e caudas mais pesadas do que uma distribuição normal. As séries de retornos S&P500 e BTC são assimétricas à esquerda. Por outro lado, as séries de retornos MSFT e EURUSD mostram uma ligeira assimetria positiva.

Tabela 9 – Estatísticas descritivas das séries de retornos diários S&P500, MSFT, EURUSD e BTC.

	S&P500	MSFT	EURUSD	BTC
Duração	1987-01-02 1995-12-29	2006-01-03 2015-06-30	2015-01-01 2018-06-30	2015-01-01 2018-06-30
Tamanho da amostra	2274	2388	909	1276
Mínimo	-0,23	-0,12	-0,03	-0,24
Máximo	0,09	0,17	0,03	0,22
SD	0,01	0,02	0,01	0,04
Assimetria	-5,21	0,05	0,02	-0,42
Curtose	119,39	13,40	5,26	8,52

A [Figura 11](#) mostra as séries temporais dos preços de fechamento e dos retornos diários. Observamos que a série de preços do índice S&P500 apresentou uma tendência crescente e os retornos parecem variar mais na primeira parte da série, que inclui a segunda-feira negra correspondente ao dia 19 de outubro de 1987. Os retornos MSFT parecem oscilar mais durante a crise financeira, no período de 2008 a meados de 2009, enquanto no "período normal" os retornos exibem pouca variação. Pode-se ver que o preço da criptomoeda Bitcoin aumentou significativamente em 2017 e atingiu o máximo no final deste mesmo ano, para depois cair.

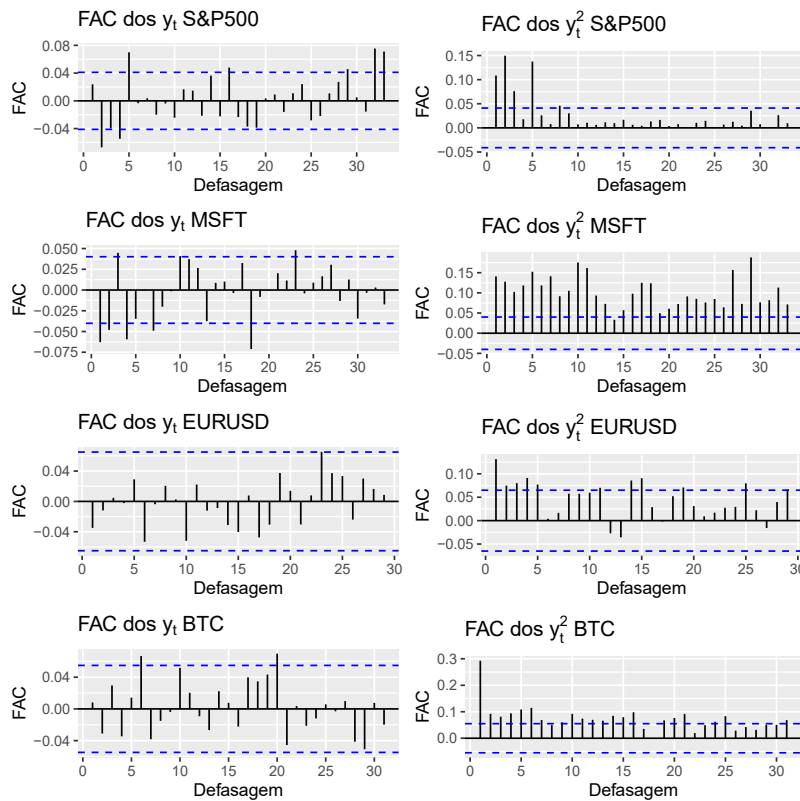
Figura 11 – Série temporais de preços (esquerda) e retornos (direita) de S&P500, MSFT, EURUSD e BTC.



Em continuidade com a inspeção visual das séries de retornos, utilizou-se a análise da

função de autocorrelação (FAC), que pode ser visualizada na [Figura 12](#). Para todas as séries, as FACs das séries de retornos apresentam baixa estrutura de correlação serial. Entretanto, a FAC dos quadrados dos retornos declina lentamente, o que é característico em séries temporais financeiras. A autocorrelação de declínio lento nos quadrados dos retornos indica um comportamento de memória longa, ou seja, as observações distantes passadas ainda impactam as recentes.

Figura 12 – Função de autocorrelação dos retornos (lado esquerdo) e dos quadrados dos retornos (lado direito) de S&P500, MSFT, EURUSD e BTC.



Nota – Os limites azuis indicam o intervalo de confiança de 95% para a autocorrelação estimada.

As distribuições a priori dos parâmetros são as mesmas da [Seção 5.2](#). Para ajustar os modelos SV às séries de retornos, usamos o pacote R *rstan*. O número de iterações MCMC e amostras iniciais descartadas são os mesmos descritos na [Seção 5.2](#). Com as 5000 amostras resultantes, calculamos as médias a posteriori, os desvios padrões e intervalos de credibilidade de 95% para todos os parâmetros dos modelos SV considerados.

As Tabelas [10-12](#) mostram os resultados das estimativas posteriores para as séries de retornos S&P500, MSFT, USDJPY e BTC. Para todas as séries de retornos consideradas, observamos que a média posterior de ϕ ficou entre 0,93 e 0,99, indicando alta persistência da volatilidade nos retornos financeiros. Além disso, em todas as séries de retornos, as médias a posteriori de ϕ nos modelos SV-N e SV-NL são menores do que nos outros quatro modelos SV. Como esperado, as médias a posteriori de σ nos modelos SV-N e SV-NL são maiores do que

nos outros quatro modelos SV, indicando que o processo de log-volatilidade dos outros quatro modelos é menos variável do que os modelos SV-N e SV-NL.

Nas séries S&P500, MSFT e EURUSD os valores de ρ são estimados como negativos, implicando a presença de efeitos de alavancagem. No entanto, na série EURUSD, a média posterior do parâmetro ρ não foi considerada significativa, pois os intervalos de credibilidade de 95% contêm o zero. A partir da Tabela 12, observamos que, na série BTC, a média do parâmetro de alavancagem ρ é positiva, ou seja, não encontramos nenhuma evidência do efeito de alavancagem no mercado de Bitcoin.

As médias das estimativas do parâmetro de grau de liberdade ν , estão entre 1,31 e 7,97, indicando caudas pesadas nas distribuições dos retornos financeiros, especialmente para a série de retornos BTC.

Em relação ao parâmetro de assimetria α , as médias a posteriori de α são negativos nas séries S&P500 e BTC. Para a série S&P500 e modelo SV-STL, o intervalo de credibilidade de 95% de α contêm o zero. No entanto, sua distribuição posterior está principalmente localizada em domínio negativo, conforme mostrado na Figura 13. A Tabela 11 mostra que as estimativas de α são positivas nas séries MSFT e EURUSD. Embora o intervalo de credibilidade de α contenha o zero, a distribuição posterior está principalmente localizada em domínio positivo, conforme mostrado nas Figuras 15 e 14.

Tabela 10 – Resultados da estimação das séries de retornos diários S&P500.

	μ	ϕ	σ	ρ	ν	α
S&P500						
SV-N	-9,84	0,97	0,20	—	—	—
	0,17	0,01	0,03	—	—	—
	(-10,19; -9,51)	(0,96; 0,99)	(0,15; 0,25)	—	—	—
SV-NL	-9,83	0,96	0,23	-0,37	—	—
	0,13	0,01	0,03	0,07	—	—
	(-10,08; -9,55)	(0,94; 0,98)	(0,18; 0,30)	(-0,50; -0,23)	—	—
SV-T	-10,16	0,99	0,09	—	5,65	—
	0,32	0,00	0,01	—	0,71	—
	(-10,77; -9,44)	(0,98; 1,00)	(0,07; 0,12)	—	(4,48; 7,24)	—
SV-TL	-10,09	0,99	0,11	-0,50	5,86	—
	0,24	0,00	0,02	0,10	0,78	—
	(-10,55; -9,57)	(0,98; 1,00)	(0,08; 0,15)	(-0,68; -0,28)	(4,60; 7,64)	—
SV-STL	-10,08	0,99	0,11	-0,52	6,06	-0,20
	0,24	0,00	0,02	0,10	0,84	0,15
	(-10,55; -9,57)	(0,98; 1,00)	(0,08; 0,15)	(-0,69; -0,30)	(4,69; 8,00)	(-0,49; 0,09)
SV-SSL	-10,34	0,99	0,12	-0,52	2,01	-0,27
	0,24	0,00	0,02	0,09	0,22	0,19
	(-10,78; -9,86)	(0,98; 1,00)	(0,09; 0,16)	(-0,69; -0,32)	(1,66; 2,50)	(-0,65; 0,12)

Nota – Primeira linha: média posterior. Segunda linha: desvios padrão. Terceira linha: intervalo de credibilidade de 95%.

Tabela 11 – Resultados da estimação das séries de retornos diários MSFT e EURUSD.

	μ	ϕ	σ	ρ	ν	α
MSFT						
SV-N	-8,61	0,94	0,33	—	—	—
	0,11	0,01	0,04	—	—	—
	(-8,83; -8,39)	(0,90; 0,96)	(0,26; 0,41)	—	—	—
SV-NL	-8,61	0,93	0,34	-0,25	—	—
	0,11	0,01	0,04	0,06	—	—
	(-8,82; -8,39)	(0,90; 0,96)	(0,27; 0,42)	(-0,37; -0,13)	—	—
SV-T	-8,96	0,99	0,11	—	5,26	—
	0,27	0,00	0,02	—	0,56	—
	(-9,52; -8,41)	(0,98; 1,00)	(0,08; 0,15)	—	(4,24; 6,45)	—
SV-TL	-8,93	0,99	0,12	-0,48	5,20	—
	0,19	0,00	0,02	0,10	0,55	—
	(-9,31; -8,56)	(0,98; 0,99)	(0,08; 0,15)	(-0,68; -0,27)	(4,24; 6,37)	—
SV-STL	-8,88	0,99	0,11	-0,46	5,22	0,15
	0,21	0,00	0,02	0,11	0,55	0,14
	(-9,28; -8,46)	(0,98; 0,99)	(0,09; 0,15)	(-0,66; -0,24)	(4,28; 6,44)	(-0,12; 0,42)
SV-SSL	-9,23	0,99	0,12	-0,47	1,72	0,14
	0,21	0,00	0,02	0,10	0,14	0,16
	(-9,64; -8,81)	(0,98; 0,99)	(0,09; 0,16)	(-0,66; -0,26)	(1,48; 2,00)	(-0,17; 0,47)
EURUSD						
SV-N	-10,42	0,96	0,15	—	—	—
	0,18	0,03	0,06	—	—	—
	(-10,75; -10,04)	(0,88; 0,99)	(0,08; 0,29)	—	—	—
SV-NL	-10,41	0,97	0,12	-0,20	—	—
	0,21	0,02	0,05	0,18	—	—
	(-10,81; -9,96)	(0,91; 1,00)	(0,07; 0,25)	(-0,55; 0,13)	—	—
SV-T	-10,65	0,99	0,08	—	7,88	—
	0,27	0,01	0,02	—	1,77	—
	(-11,17; -10,05)	(0,96; 1,00)	(0,05; 0,13)	—	(5,23; 12,16)	—
SV-TL	-10,67	0,99	0,08	-0,35	7,69	—
	0,27	0,01	0,02	0,21	1,76	—
	(-11,18; -10,10)	(0,97; 1,00)	(0,05; 0,12)	(-0,73; 0,06)	(5,08; 11,96)	—
SV-STL	-10,59	0,99	0,08	-0,34	7,97	0,23
	0,29	0,01	0,02	0,22	1,83	0,28
	(-11,15; -9,98)	(0,97; 1,00)	(0,05; 0,12)	(-0,73; 0,10)	(5,27; 12,38)	(-0,33; 0,78)
SV-SSL	-10,88	0,99	0,08	-0,32	2,27	0,23
	0,29	0,01	0,02	0,21	0,39	0,31
	(-11,42; -10,27)	(0,97; 1,00)	(0,05; 0,12)	(-0,70; 0,11)	(1,67; 3,19)	(-0,37; 0,83)

Nota – Primeira linha: média posterior. Segunda linha: desvios padrão. Terceira linha: intervalo de credibilidade de 95%.

Tabela 12 – Resultados da estimação da série de retornos diários BTC.

	μ	ϕ	σ	ρ	ν	α
BTC						
SV-N	-7,21	0,94	0,50	—	—	—
	0,24	0,02	0,06	—	—	—
	(-7,68; -6,73)	(0,90; 0,97)	(0,39; 0,62)	—	—	—
SV-NL	-7,22	0,94	0,48	0,09	—	—
	0,25	0,02	0,06	0,07	—	—
	(-7,69; -6,72)	(0,91; 0,97)	(0,38; 0,61)	(-0,04; 0,22)	—	—
SV-T	-7,62	0,98	0,27	—	3,68	—
	0,44	0,01	0,04	—	0,51	—
	(-8,46; -6,70)	(0,96; 0,99)	(0,20; 0,35)	—	(2,85; 4,84)	—
SV-TL	-7,66	0,98	0,25	0,20	3,58	—
	0,48	0,01	0,04	0,11	0,48	—
	(-8,50; -6,71)	(0,97; 0,99)	(0,18; 0,33)	(0,00; 0,42)	(2,78; 4,68)	—
SV-STL	-7,63	0,98	0,25	0,19	3,68	-0,12
	0,43	0,01	0,04	0,11	0,51	0,14
	(-8,49; -6,73)	(0,97; 0,99)	(0,19; 0,33)	(-0,02; 0,40)	(2,83; 4,85)	(-0,40; 0,14)
SV-SSL	-8,05	0,98	0,27	0,16	1,31	-0,19
	0,43	0,01	0,04	0,10	0,14	0,16
	(-8,88; -7,18)	(0,96; 0,99)	(0,20; 0,35)	(-0,04; 0,37)	(1,07; 1,64)	(-0,51; 0,11)

Nota – Primeira linha: média posterior. Segunda linha: desvios padrão. Terceira linha: intervalo de credibilidade de 95%.

Para selecionar o melhor modelo SV para cada série, foram calculados os métodos de seleção de modelos apresentados no [Capítulo 4](#). Os resultados da comparação dos modelos SV são apresentados na [Tabela 13](#), onde valores menores para esses critérios indicam o melhor modelo. Para todas as séries analisadas, os valores DIC indicam que o modelo SV-SSL é o melhor modelo entre todos os modelos SV ajustados. Segundo os critérios WAIC e LOO-CV o melhor modelo para as séries de retornos S&P500, MSFT e EURUSD é o modelo SV-STL. Por outro lado, para a série de retornos BTC, o critério WAIC seleciona ao modelo SV-TL como o melhor, enquanto o LOO-CV seleciona o modelo SV-T.

Vale notar que, na série BTC, o LOO-CV favoreceu o modelo SV-N em relação ao modelo SV-NL e favoreceu o modelo SV-T em relação ao modelo SV-TL, indicando que os retornos BTC podem não apresentar o efeito de alavancagem. Isso apoia os resultados da [Tabela 12](#), onde o parâmetro de alavancagem ρ foi encontrado positivo e não significativo para os retornos BTC.

Os resultados desta seção podem ser resumidos da seguinte forma: o modelo de volatilidade estocástica com cauda pesada SV-STL fornece o melhor ajuste, indicando a ocorrência de retornos extremos nas séries S&P500, MSFT e EURUSD. Além disso, incorporar componentes de alavancagem e assimetria na volatilidade estocástica não apresentou melhorias significativas na estimativa da volatilidade da criptomoeda Bitcoin.

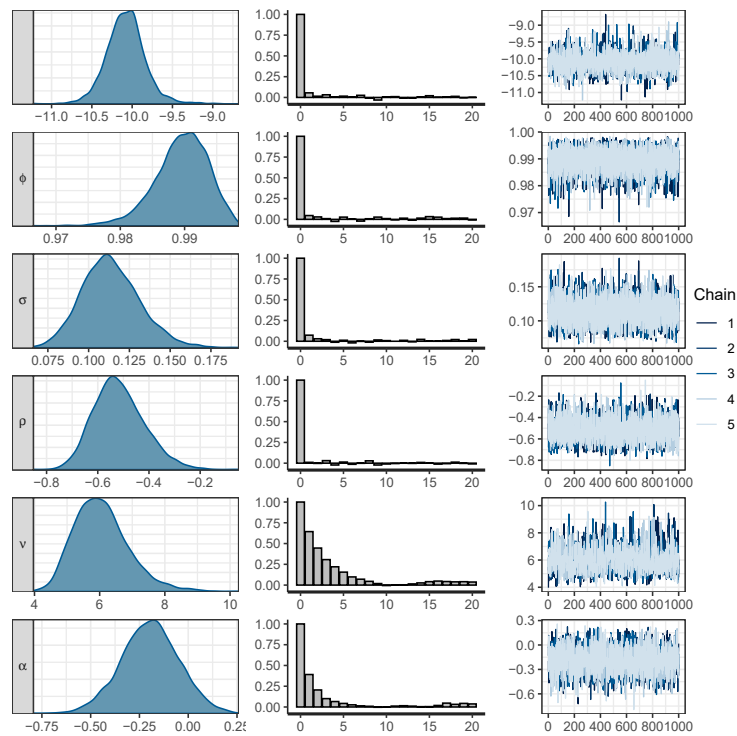
Tabela 13 – Resultados de comparação de modelos SV para as séries temporais.

	Métodos de seleção de modelos		
	DIC	WAIC	LOO-CV
S&P500			
SV-N	-15799,50	-15756,78	-15732,05
SV-NL	-15820,73	-15779,76	-15752,66
SV-T	-15884,47	-15939,00	-15857,65
SV-TL	-15895,52	-15947,04	-15868,79
SV-STL	-15938,60	-15953,86	-15871,24
SV-SSL	-16014,42	-15910,79	-15850,29
MSFT			
SV-N	-13530,58	-13492,78	-13435,84
SV-NL	-13541,01	-13506,32	-13454,00
SV-T	-13598,31	-13676,38	-13576,30
SV-TL	-13602,58	-13683,53	-13589,74
SV-STL	-13645,52	-13693,70	-13598,28
SV-SSL	-13738,80	-13666,92	-13586,19
EURUSD			
SV-N	-6871,71	-6856,55	-6851,00
SV-NL	-6867,43	-6852,60	-6848,61
SV-T	-6896,52	-6913,32	-6890,88
SV-TL	-6895,85	-6912,18	-6890,64
SV-STL	-6952,00	-6918,42	-6891,02
SV-SSL	-6991,93	-6911,07	-6888,82
BTC			
SV-N	-5446,32	-5417,25	-5357,41
SV-NL	-5443,45	-5412,88	-5353,06
SV-T	-5513,29	-5568,61	-5481,97
SV-TL	-5513,06	-5569,84	-5477,62
SV-STL	-5530,12	-5565,11	-5470,57
SV-SSL	-5593,74	-5549,31	-5455,55

Durante nossa análise, utilizamos a estatística \hat{R} de Gelman-Rubin (GELMAN; RUBIN, 1992), que analisa a variância entre as cadeias geradas. Verificamos que o valor de \hat{R} para todos os parâmetros dos modelos SV estivesse próximo de 1, o que indica que todas as cadeias convergiram. Além disso, analisamos a convergência usando análises gráficas, mediante o pacote R bayesplot (GABRY; MAHR, 2022).

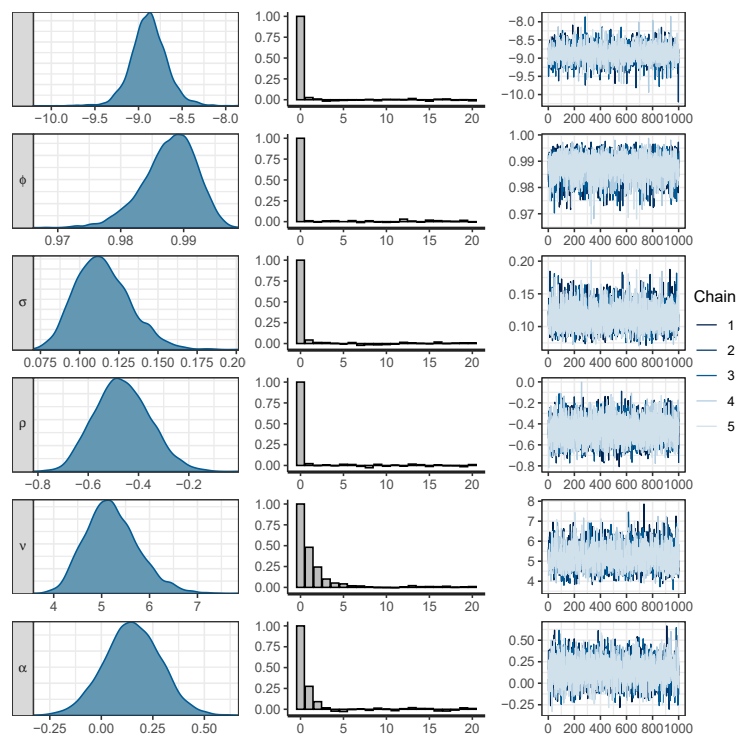
As Figuras 13-16 apresentam as densidades a posteriori, os gráficos de autocorrelação e as saídas do MCMC dos parâmetros dos modelos SV selecionados pelo LOO-CV para cada série de retorno analisada. A partir da saída do MCMC e as funções de autocorrelação, observamos a alta velocidade de convergência. As autocorrelações das iterações decaíram de forma relativamente rápida em todos os parâmetros.

Figura 13 – Resultados da estimação dos retornos diários do índice S&P500 (modelo SV-STL).



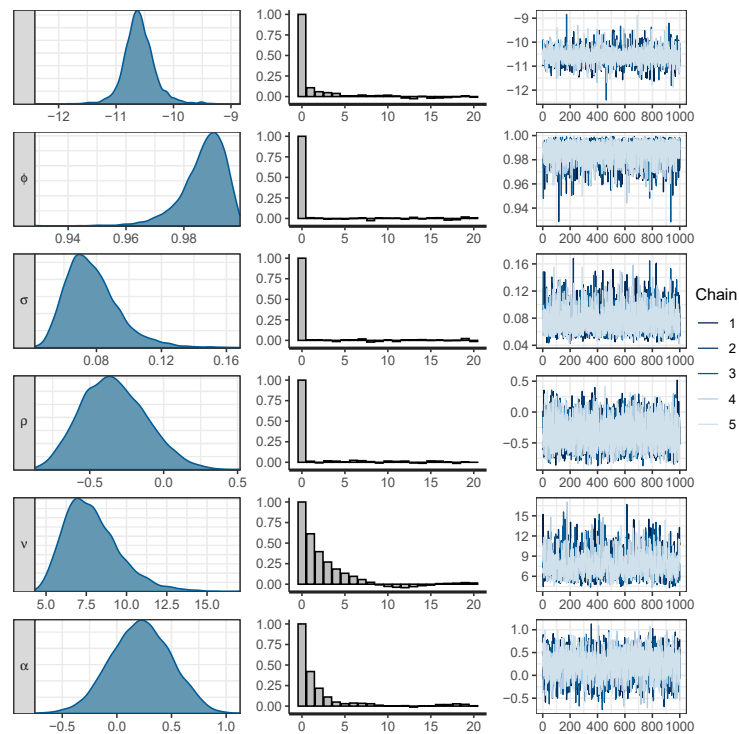
Nota – Densidades posteriores (esquerda), funções de autocorrelação (ACF) (médio), saída do MCMC (direita) para os parâmetros μ , ϕ , σ , ρ , v e α .

Figura 14 – Resultados da estimação dos retornos diários MSFT (modelo SV-STL).



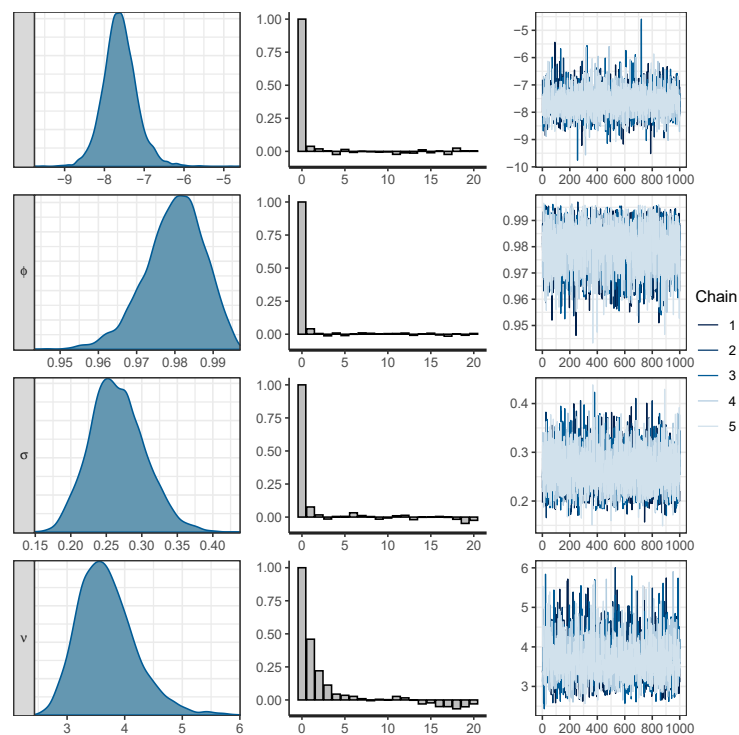
Nota – Densidades posteriores (esquerda), funções de autocorrelação (ACF) (médio), saída do MCMC (direita) para os parâmetros μ , ϕ , σ , ρ , v e α .

Figura 15 – Resultados da estimação dos retornos diários EURUSD (modelo SV-STL).



Nota – Densidades posteriores (esquerda), funções de autocorrelação (ACF) (médio), saída do MCMC (direita) para os parâmetros μ , ϕ , σ , ρ , v e α .

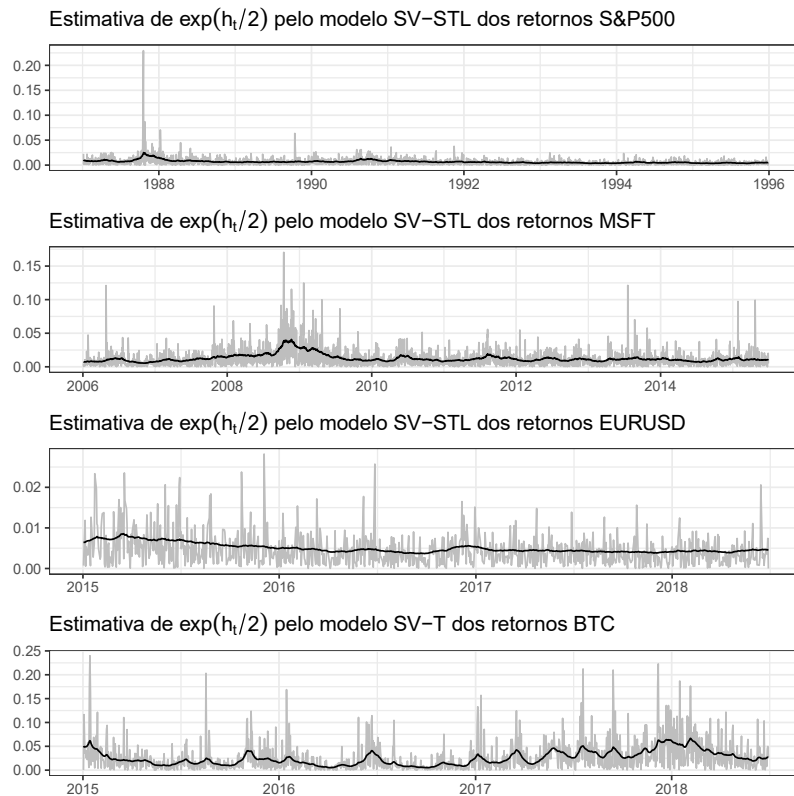
Figura 16 – Resultados da estimação dos retornos diários BTC (modelo SV-T).



Nota – Densidades posteriores (esquerda), funções de autocorrelação (ACF) (médio), saída do MCMC (direita) para os parâmetros μ , ϕ , σ , e v .

A Figura 17 mostra os retornos absolutos das séries (em cinza) e a média de $\exp(h_t/2)$ (em preto) para todas as séries consideradas. A partir dos gráficos, podemos ver que as volatilidades estimadas acompanham o comportamento das séries de retornos absolutos e capturam os períodos de grandes oscilações. Pode-se observar que a crise financeira nos EUA (2008) teve sérias repercussões no comportamento da volatilidade da série MSFT.

Figura 17 – Volatilidades estimadas pelo modelo SV escolhido para as séries de retornos S&P500, MSFT, EURUSD e BTC.



Nota – Média posterior de $\exp(h_t/2)$ (preto) e retornos absolutos (cinza).

Finalmente, a Tabela 14 apresenta os tempos computacionais em minutos que levaram ajustar cada um dos modelos SV às séries de retornos reais.

Tabela 14 – Tempo de ajuste dos modelos SV (em minutos) para as séries S&P500, MSFT, EURUSD e BTC.

Modelo	S&P500	MSFT	EURUSD	BTC
SV-N	3,83	2,30	1,85	3,53
SV-NL	6,61	3,73	2,05	5,61
SV-T	7,83	8,39	4,08	10,17
SV-TL	10,57	11,19	3,54	12,46
SV-STL	19,84	21,65	9,87	15,91
SV-SSL	23,52	25,36	12,63	17,44

CONCLUSÕES

Nesta dissertação, avaliamos e comparamos o desempenho do algoritmo NUTS na estimação dos parâmetros em modelos de volatilidade estocástica com alavancagem baseados em mistura de escala da distribuição normal assimétrica (SV-SMSN). Utilizando a representação estocástica da distribuição normal assimétrica, foi possível implementar eficientemente a classe de modelos SV-SMSN na linguagem Stan.

Com base nas medidas de viés e RMSE, o algoritmo NUTS apresentou um bom desempenho na estimação dos parâmetros na classe de modelos SV-SMSN. A comparação entre o algoritmo NUTS e o algoritmo MCMC implementado no pacote `stochvol` para estimar os parâmetros dos modelos SV considerados no `stochvol` mostrou que o algoritmo NUTS superou o `stochvol` em termos de tamanho efetivo da amostra. No entanto, quando se considerou o tamanho efetivo da amostra em relação ao tempo de execução, nenhum dos dois algoritmos foi superior ao outro para todos os modelos SV.

O estudo de simulação para avaliar o desempenho dos critérios de informação DIC e WAIC, assim como das técnicas de validação cruzada LOO-CV e LFO-CV, na seleção de modelos SV-SMSN, mostrou que o critério DIC teve o pior desempenho, enquanto o LOO-CV teve o melhor desempenho na seleção do verdadeiro modelo SV do qual as séries de retornos foram geradas. Para a implementação do LFO-CV nas séries de retornos reais, o trabalho de determinar o número mínimo de observações para fazer previsões (L) está atualmente em andamento.

Ao aplicar os modelos SV aos dados reais, as estimativas indicaram fortes evidências de cauda pesada. Em geral, para as séries de retornos analisadas, o modelo SV-STL teve um bom desempenho ao modelar a volatilidade das séries.

Como trabalho futuro, planejamos estender a classe de modelos SV-SMSN para uma versão multivariada. Além disso, seria interessante incorporar saltos permanentes à volatilidade. Está prevista a disponibilização de um pacote R para fornecer uma ferramenta para estimar a volatilidade de séries de retornos diários.

REFERÊNCIAS

ABANTO-VALLE, C.; LACHOS, V.; DEY, D. K. Bayesian estimation of a skew-student-t stochastic volatility model. **Methodology and Computing in Applied Probability**, Springer, v. 17, n. 3, p. 721–738, 2015. Citado nas páginas [19](#), [25](#) e [33](#).

ABANTO-VALLE, C. A.; BANDYOPADHYAY, D.; LACHOS, V. H.; ENRIQUEZ, I. Robust bayesian analysis of heavy-tailed stochastic volatility models using scale mixtures of normal distributions. **Computational Statistics & Data Analysis**, Elsevier, v. 54, n. 12, p. 2883–2898, 2010. Citado nas páginas [19](#), [34](#) e [53](#).

ABANTO-VALLE, C. A.; GARRAFA-ARAGÓN, H. B. Threshold stochastic volatility models with heavy tails: A bayesian approach. **Economía**, v. 42, n. 83, p. 32–53, 2019. Citado na página [53](#).

ABANTO-VALLE, C. A.; RODRÍGUEZ, G.; GARRAFA-ARAGÓN, H. B. Stochastic volatility in mean: Empirical evidence from latin-american stock markets using hamiltonian monte carlo and riemann manifold hmc methods. **The Quarterly Review of Economics and Finance**, Elsevier, v. 80, p. 272–286, 2021. Citado na página [20](#).

AZZALINI. Further results on a class of distributions which includes the normal ones. **Statistica**, v. 46, n. 2, p. 199–208, 1986. Citado na página [24](#).

AZZALINI, A. A class of distributions which includes the normal ones. **Scandinavian journal of statistics**, JSTOR, p. 171–178, 1985. Citado na página [24](#).

BERG, A.; MEYER, R.; YU, J. Dic as a model comparison criterion for stochastic volatility models. **Journal of Business and Economic Statistics**, Citeseer, v. 22, p. 107–120, 2004. Citado na página [39](#).

BLACK, F. Studies of stock market volatility changes. **1976 Proceedings of the American statistical association business and economic statistics section**, American Statistical Association, 1976. Citado na página [24](#).

BOLLERSLEV, T. Generalized autoregressive conditional heteroskedasticity. **Journal of econometrics**, Elsevier, v. 31, n. 3, p. 307–327, 1986. Citado na página [19](#).

BRANCO, M. D.; DEY, D. K. A general class of multivariate skew-elliptical distributions. **Journal of Multivariate Analysis**, Elsevier, v. 79, n. 1, p. 99–113, 2001. Citado na página [25](#).

BÜRKNER, P.-C.; GABRY, J.; VEHTARI, A. Approximate leave-future-out cross-validation for bayesian time series models. **Journal of Statistical Computation and Simulation**, Taylor & Francis, v. 90, n. 14, p. 2499–2523, 2020. Citado nas páginas [20](#), [43](#), [44](#) e [61](#).

CARPENTER, B.; GELMAN, A.; HOFFMAN, M. D.; LEE, D.; GOODRICH, B.; BETANCOURT, M.; BRUBAKER, M.; GUO, J.; LI, P.; RIDDELL, A. Stan: A probabilistic programming language. **Journal of statistical software**, Columbia Univ., New York, NY (United States); Harvard Univ., Cambridge, MA . . . , v. 76, n. 1, 2017. Citado nas páginas [20](#), [30](#) e [42](#).

- CARPENTER, B.; HOFFMAN, M. D.; BRUBAKER, M.; LEE, D.; LI, P.; BETANCOURT, M. The stan math library: Reverse-mode automatic differentiation in c++. **arXiv preprint arXiv:1509.07164**, 2015. Citado na página 35.
- CHAN, J. C.; GRANT, A. Issues in comparing stochastic volatility models using the deviance information criterion. CAMA Working Paper, 2014. Citado na página 39.
- CHIB, S.; NARDARI, F.; SHEPHARD, N. Markov chain monte carlo methods for stochastic volatility models. **Journal of Econometrics**, Elsevier, v. 108, n. 2, p. 281–316, 2002. Citado nas páginas 19 e 34.
- CHOY, B.; WAN, W. Y.; CHAN, C. M. Bayesian student-t stochastic volatility models via scale mixtures. **Wai Yin and Chan, Chun Man, Bayesian Student-T Stochastic Volatility Models Via Scale Mixtures (August 18, 2009)**, 2009. Citado na página 34.
- DARJUS, H.; GREGOR, K. Modeling univariate and multivariate stochastic volatility in R with stochvol and factorstochvol. **Journal of Statistical Software**, v. 100, n. 12, p. 1–34, 2021. Citado nas páginas 20, 45 e 46.
- DIAS, D. S.; EHLERS, R. S. Stochastic volatility models using hamiltonian monte carlo methods and stan. **arXiv preprint arXiv:1712.02326**, 2017. Citado nas páginas 20 e 30.
- DUANE, S.; KENNEDY, A. D.; PENDLETON, B. J.; ROWETH, D. Hybrid monte carlo. **Physics letters B**, Elsevier, v. 195, n. 2, p. 216–222, 1987. Citado nas páginas 20 e 28.
- EHLERS, R.; ZEVALLOS, M. Bayesian estimation and prediction of stochastic volatility models via inla. **Communications in Statistics-Simulation and Computation**, Taylor & Francis, v. 44, n. 3, p. 683–693, 2015. Citado nas páginas 20 e 35.
- ENGLE, R. F. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. **Econometrica: Journal of the econometric society**, JSTOR, p. 987–1007, 1982. Citado na página 19.
- FUENTE, C. García de la; MIGUEL, P. G. S.; WIPER, M. P. Modeling financial time series with the skew slash distribution. 2012. Citado na página 33.
- GABRY, J.; MAHR, T. **bayesplot: Plotting for Bayesian Models**. 2022. R package version 1.9.0. Disponível em: <<https://mc-stan.org/bayesplot/>>. Citado na página 67.
- GELFAND, A. E.; DEY, D. K.; CHANG, H. **Model determination using predictive distributions with implementation via sampling-based methods**. [S.l.], 1992. Citado na página 40.
- GELMAN, A.; RUBIN, D. B. Inference from iterative simulation using multiple sequences. **Statistical science**, JSTOR, p. 457–472, 1992. Citado nas páginas 54 e 67.
- HAN, H.; LEE, E. *et al.* Triple regime stochastic volatility model with threshold and leverage effects. **The Korean Economic Review**, v. 36, n. 2, p. 481–509, 2020. Citado na página 61.
- HARVEY, A. C.; SHEPHARD, N. Estimation of an asymmetric stochastic volatility model for asset returns. **Journal of Business & Economic Statistics**, Taylor & Francis, v. 14, n. 4, p. 429–434, 1996. Citado nas páginas 19 e 34.

HASTINGS, W. K. Monte carlo sampling methods using markov chains and their applications. Oxford University Press, 1970. Citado na página 27.

HENZE, N. A probabilistic representation of the 'skew-normal' distribution. **Scandinavian journal of statistics**, JSTOR, p. 271–275, 1986. Citado na página 24.

HOFFMAN, M. D.; GELMAN, A. *et al.* The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. **J. Mach. Learn. Res.**, v. 15, n. 1, p. 1593–1623, 2014. Citado nas páginas 20, 29 e 44.

HOSSZEJNI, D.; KASTNER, G. Approaches toward the bayesian estimation of the stochastic volatility model with leverage. In: SPRINGER. **International Conference on Bayesian Statistics in Action**. [S.l.], 2018. p. 75–83. Citado nas páginas 20, 46, 53 e 61.

HUANG, J.-Z.; XU, L. Stochastic volatility models for asset returns with leverage, skewness and heavy-tails via scale mixture. **The Quarterly Journal of Finance**, World Scientific, v. 4, n. 03, p. 1450011, 2014. Citado nas páginas 19, 25, 31, 53 e 61.

JACQUIER, E.; POLSON, N.; ROSSI, P. E. Bayesian analysis of stochastic volatility models with fat-tails and correlated errors. **Journal of Econometrics**, Elsevier, v. 122, n. 1, p. 185–212, 2004. Citado nas páginas 19 e 34.

JACQUIER, E.; POLSON, N. G.; ROSSI, P. E. Bayesian analysis of stochastic volatility models. **Journal of Business & Economic Statistics**, [American Statistical Association, Taylor & Francis, Ltd.], v. 12, n. 4, p. 371–389, 1994. Citado na página 20.

KASTNER, G.; FRÜHWIRTH-SCHNATTER, S. Ancillarity-sufficiency interweaving strategy (asis) for boosting mcmc estimation of stochastic volatility models. **Computational Statistics & Data Analysis**, Elsevier, v. 76, p. 408–423, 2014. Citado na página 45.

KIM, S.; SHEPHARD, N.; CHIB, S. Stochastic volatility: likelihood inference and comparison with arch models. **The review of economic studies**, Wiley-Blackwell, v. 65, n. 3, p. 361–393, 1998. Citado nas páginas 35, 46 e 53.

LEIMKUHLER, B.; REICH, S. **Simulating hamiltonian dynamics**. [S.l.]: Cambridge university press, 2004. Citado na página 28.

LI, D.; CLEMENTS, A.; DROVANDI, C. Efficient bayesian estimation for garch-type models via sequential monte carlo. **Econometrics and Statistics**, Elsevier, v. 19, p. 22–46, 2021. Citado na página 20.

METROPOLIS, N.; ROSENBLUTH, A. W.; ROSENBLUTH, M. N.; TELLER, A. H.; TELLER, E. Equation of state calculations by fast computing machines. **The journal of chemical physics**, American Institute of Physics, v. 21, n. 6, p. 1087–1092, 1953. Citado na página 27.

MEYER, R.; YU, J. Bugs for a bayesian analysis of stochastic volatility models. **The Econometrics Journal**, Wiley Online Library, v. 3, n. 2, p. 198–215, 2000. Citado nas páginas 19, 20 e 34.

MORETTIN, P. A. **Econometria financeira: um curso em séries temporais financeiras**. [S.l.]: Editora Blucher, 2017. Citado na página 23.

- NAKAJIMA, J.; OMORI. Leverage, heavy-tails and correlated jumps in stochastic volatility models. **Computational Statistics & Data Analysis**, Elsevier, v. 53, n. 6, p. 2335–2353, 2009. Citado na página 34.
- NAKAJIMA, J.; OMORI, Y. Stochastic volatility model with leverage and asymmetrically heavy-tailed error using GH skew student's t-distribution. **Computational Statistics & Data Analysis**, Elsevier, v. 56, n. 11, p. 3690–3704, 2012. Citado nas páginas 25, 46 e 53.
- NEAL, R. M. *et al.* Mcmc using hamiltonian dynamics. **Handbook of markov chain monte carlo**, Chapman and Hall/CRC, v. 2, n. 11, p. 2, 2011. Citado na página 28.
- NUGROHO, D. B.; MORIMOTO, T. Estimation of realized stochastic volatility models using hamiltonian monte carlo-based methods. **Computational Statistics**, Springer, v. 30, n. 2, p. 491–516, 2015. Citado na página 20.
- O'HAGAN, A.; LEONARD, T. Bayes estimation subject to uncertainty about parameter constraints. **Biometrika**, Oxford University Press, v. 63, n. 1, p. 201–203, 1976. Citado na página 24.
- OMORI, Y.; CHIB, S.; SHEPHARD, N.; NAKAJIMA, J. Stochastic volatility with leverage: Fast and efficient likelihood inference. **Journal of Econometrics**, Elsevier, v. 140, n. 2, p. 425–449, 2007. Citado nas páginas 19 e 34.
- PLUMMER, M.; BEST, N.; COWLES, K.; VINES, K. Coda: Convergence diagnosis and output analysis for mcmc. **R News**, v. 6, n. 1, p. 7–11, 2006. Disponível em: <<https://journal.r-project.org/archive/>>. Citado na página 45.
- RAST, P.; MARTIN, S. R.; RODRIGUEZ, J. E. Bayesian multivariate garch models for within-person variance forecasting. **PsyArXiv**, 2020. Citado na página 20.
- RYAN, J. A.; ULRICH, J. M. **quantmod: Quantitative Financial Modelling Framework**. [S.l.], 2022. R package version 0.4.20. Disponível em: <<https://CRAN.R-project.org/package=quantmod>>. Citado na página 61.
- SHEPHARD, N.; PITT, M. K. Likelihood analysis of non-gaussian measurement time series. **Biometrika**, Oxford University Press, v. 84, n. 3, p. 653–667, 1997. Citado na página 20.
- SPIEGELHALTER, D.; THOMAS, A.; BEST, N.; GILKS, W. Bugs 0.5: Bayesian inference using gibbs sampling manual (version ii). **MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK**, p. 1–59, 1996. Citado na página 20.
- SPIEGELHALTER, D. J.; BEST, N. G.; CARLIN, B. P.; LINDE, A. V. D. Bayesian measures of model complexity and fit. **Journal of the royal statistical society: Series b (statistical methodology)**, Wiley Online Library, v. 64, n. 4, p. 583–639, 2002. Citado nas páginas 20 e 39.
- Stan Development Team. **RStan: the R interface to Stan**. 2022. R package version 2.21.7. Disponível em: <<https://mc-stan.org/>>. Citado nas páginas 35, 46, 53 e 59.
- TAKAISHI, T. Bayesian inference of stochastic volatility model by hybrid monte carlo. **Journal of Circuits, Systems, and Computers**, World Scientific, v. 18, n. 08, p. 1381–1396, 2009. Citado na página 20.

- TAYLOR, S. J. Financial returns modelled by the product of two stochastic processes—a study of the daily sugar prices 1961–75. **Time series analysis: theory and practice**, North-Holland, v. 1, p. 203–226, 1982. Citado nas páginas 19, 31 e 35.
- VEHTARI, A.; GABRY, J.; MAGNUSSON, M.; YAO, Y.; BÜRKNER, P.-C.; PAANANEN, T.; GELMAN, A. **loo: Efficient leave-one-out cross-validation and WAIC for Bayesian models**. 2022. R package version 2.5.1. Disponível em: <<https://mc-stan.org/loo/>>. Citado nas páginas 42 e 59.
- VEHTARI, A.; GELMAN, A.; GABRY, J. Practical bayesian model evaluation using leave-one-out cross-validation and waic. **Statistics and computing**, Springer, v. 27, n. 5, p. 1413–1432, 2017. Citado nas páginas 20, 40, 41 e 42.
- VEHTARI, A.; SIMPSON, D.; GELMAN, A.; YAO, Y.; GABRY, J. Pareto smoothed importance sampling. **arXiv preprint arXiv:1507.02646**, 2015. Citado na página 41.
- WAHL, J. C. **Parameter Estimation of Multivariate Factor Stochastic Volatility Models**. Dissertação (Mestrado) — The University of Bergen, 2018. Citado na página 30.
- WANG, J.; GENTON, M. G. The multivariate skew-slash distribution. **Journal of Statistical Planning and Inference**, Elsevier, v. 136, n. 1, p. 209–220, 2006. Citado na página 26.
- WANG, J. J.; CHAN, J. S.; CHOY, S. B. Stochastic volatility models with leverage and heavy-tailed distributions: A bayesian approach using scale mixtures. **Computational Statistics & Data Analysis**, Elsevier, v. 55, n. 1, p. 852–862, 2011. Citado nas páginas 46 e 53.
- WATANABE, S.; OPPER, M. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. **Journal of machine learning research**, v. 11, n. 12, 2010. Citado nas páginas 20 e 42.
- YU, J. On leverage in a stochastic volatility model. **Journal of Econometrics**, Elsevier, v. 127, n. 2, p. 165–178, 2005. Citado nas páginas 20, 32 e 34.
- ZEVALLOS, M.; GASCO, L.; EHLERS, R. Riemann manifold langevin methods on stochastic volatility estimation. **Communications in Statistics-Simulation and Computation**, Taylor & Francis, v. 46, n. 10, p. 7942–7956, 2017. Citado nas páginas 46 e 53.

DEMONSTRAÇÕES

A.1 Distribuição normal assimétrica

Lema 1. Se $W \sim N(0, 1)$ e seja Φ a fda de W , então $E(\Phi(hW + k)) = \Phi\left(\frac{k}{\sqrt{1+h^2}}\right)$, \forall real h, k .

Teorema 1. Seja $Y \sim \text{SN}(\eta, \sigma, \alpha)$, então sua função geradora de momentos é dada por:

$$M(t) = 2e^{t\eta + \frac{1}{2}t^2\sigma^2} \Phi(\delta\sigma t).$$

Demonstração.

$$\begin{aligned} M_Y(t) &= E(e^{ty}) = \int_{-\infty}^{\infty} e^{ty} f(y) dy \\ &= \frac{2}{\sigma} \int_{-\infty}^{\infty} e^{ty} \phi\left(\frac{y-\eta}{\sigma}\right) \Phi\left(\frac{\alpha(y-\eta)}{\sigma}\right) dy \\ &= \frac{2}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} e^{ty} e^{-\frac{1}{2}\left(\frac{y-\eta}{\sigma}\right)^2} \Phi\left(\frac{\alpha(y-\eta)}{\sigma}\right) dy, \quad \text{fazendo } v = \frac{y-\eta}{\sigma} \\ &= \frac{2e^{t\eta}}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{tv\sigma - \frac{1}{2}v^2} \Phi(\alpha v) dv \\ &= \frac{2e^{t\eta}}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(v^2 - 2tv\sigma + t^2\sigma^2) + \frac{1}{2}t^2\sigma^2} \Phi(\alpha v) dv \\ &= \frac{2e^{t\eta}}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}(v-t\sigma)^2 + \frac{1}{2}t^2\sigma^2} \Phi(\alpha v) dv \\ &= 2e^{t\eta + \frac{1}{2}t^2\sigma^2} \int_{-\infty}^{\infty} \phi(v-t\sigma) \Phi(\alpha v) dv, \quad \text{fazendo } w = v-t\sigma \\ &= 2e^{t\eta + \frac{1}{2}t^2\sigma^2} \int_{-\infty}^{\infty} \phi(w) \Phi(\alpha w + t\alpha\sigma) dw \\ &= 2e^{t\eta + \frac{1}{2}t^2\sigma^2} E(\Phi(\alpha w + t\alpha\sigma)) \\ &= 2e^{t\eta + \frac{1}{2}t^2\sigma^2} \Phi\left(\frac{t\alpha\sigma}{\sqrt{1+\alpha^2}}\right) \\ &= 2e^{t\eta + \frac{1}{2}t^2\sigma^2} \Phi(\delta t\sigma), \quad \text{onde } \delta = \alpha/\sqrt{1+\alpha^2}. \end{aligned}$$

□

O lema a seguir é útil para calcular os momentos da variável aleatória normal assimétrica. Ele fornece a derivada da fdp e fda da distribuição normal padrão, respectivamente.

Lema 2. Seja c um número real, ϕ e Φ a fdp e fda da distribuição normal padrão, respectivamente. Então

$$\begin{aligned}\frac{d}{dt}[\phi(ct)] &= -c^2 t \phi(ct) \\ \frac{d}{dt}[\Phi(ct)] &= c \phi(ct).\end{aligned}$$

O primeiro momento da distribuição normal assimétrica é dado por:

$$\begin{aligned}E(Y) &= M'(0) = 2[e^{t\eta + \frac{1}{2}t^2\sigma^2}(\eta + t\sigma^2)\Phi(\delta t\sigma) + e^{t\eta + \frac{1}{2}t^2\sigma^2}\delta\sigma\phi(\delta t\sigma)]|_{t=0} \\ &= 2\eta\Phi(0) + 2\delta\sigma\phi(0) \\ &= \eta + \sqrt{\frac{2}{\pi}}\delta\sigma.\end{aligned}$$

O segundo momento da distribuição normal assimétrica é dado por:

$$\begin{aligned}E(Y^2) &= M''(0) \\ &= 2e^{t\eta + \frac{1}{2}t^2\sigma^2}(\eta + t\sigma^2)[(\eta + t\sigma^2)\Phi(\delta t\sigma) + \delta\sigma\phi(\delta t\sigma)] \\ &\quad + 2e^{t\eta + \frac{1}{2}t^2\sigma^2}[\sigma^2\Phi(\delta t\sigma) + (\eta + t\sigma^2)\delta\sigma\phi(\delta t\sigma) - \delta^3\sigma^3t\phi(\delta t\sigma)]|_{t=0} \\ &= \eta^2 + \sigma^2 + \frac{2\eta}{\sqrt{2\pi}}(\eta + \delta\sigma).\end{aligned}$$

Segue que, a esperança e variância da distribuição normal assimétrica $Y \sim \text{SN}(\eta, \sigma^2, \alpha)$, são dadas por:

$$\begin{aligned}E(Y) &= \eta + \sqrt{\frac{2}{\pi}}\delta\sigma, \\ V(Y) &= E(Y^2) - [E(Y)]^2 = \sigma^2 \left(1 - \frac{2}{\pi}\delta^2\right).\end{aligned}$$

Na [Equação 2.2](#) $Y \sim \text{SN}(0, \sigma^2, \alpha)$, então o primeiro e segundo momento de Y são dados por:

$$\begin{aligned}E(Y) &= \sqrt{\frac{2}{\pi}}\delta\sigma, \\ E(Y^2) &= \sigma^2.\end{aligned}\tag{A.1}$$

A.2 Esperança e variância da distribuição t-Student assimétrica

Lembrando que:

$$U = \eta + Z^{-1/2}Y,$$

Seja $Z \sim \text{Gama}(\nu/2, \nu/2)$, $\nu > 0$, então:

$$\begin{aligned}
 E\left(Z^{-k/2}\right) &= \int_0^\infty z^{-k/2} f(z) dz \\
 &= \int_0^\infty z^{-k/2} \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} z^{\frac{\nu}{2}-1} e^{-\frac{\nu}{2}z} dz \\
 &= \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \int_0^\infty z^{\frac{\nu}{2}-\frac{k}{2}-1} e^{-\frac{\nu}{2}z} dz, \quad \text{fazendo } x = \frac{\nu}{2}z \rightarrow dx = \frac{\nu}{2}dz \\
 &= \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{2}{\nu}\right)^{\frac{\nu}{2}-\frac{k}{2}} \int_0^\infty x^{\left(\frac{\nu}{2}-\frac{k}{2}\right)-1} e^{-x} dx \\
 &= \frac{\Gamma\left(\frac{\nu-k}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{\nu}{2}\right)^{\frac{k}{2}}.
 \end{aligned} \tag{A.2}$$

Utilizando a [Equação A.2](#) com $k = 1$ e a [Equação A.1](#), obtemos a esperança de uma variável aleatória U com distribuição t-Student assimétrica:

$$E(U) = E(\eta + Z^{-1/2}Y) = \eta + E(Z^{-1/2})E(Y) = \eta + \frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \sqrt{\frac{\nu}{\pi}} \sigma \delta, \quad \nu > 1. \tag{A.3}$$

Utilizando as equações (A.3), (A.1) e (A.2) com $k = 1$ e $k = 2$, a variância de uma variável U com distribuição t-Student assimétrica é dada por:

$$\begin{aligned}
 V(U) &= E(U^2) - (E(U))^2 \\
 &= E\left(\eta^2 + 2\eta Z^{-1/2}Y + Z^{-1}Y^2\right) - \left(\eta + \frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \sqrt{\frac{\nu}{\pi}} \sigma \delta\right)^2 \\
 &= \eta^2 + 2\eta E(Z^{-1/2})E(Y) + E(Z^{-1})E(Y^2) - \eta^2 - 2\eta \frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \sqrt{\frac{\nu}{\pi}} \sigma \delta - \left(\frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\right)^2 \frac{\nu}{\pi} \sigma^2 \delta^2 \\
 &= 2\eta \frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \left(\frac{\nu}{2}\right)^{\frac{1}{2}} \sqrt{\frac{2}{\pi}} \sigma \delta + \frac{\nu}{\nu-2} \sigma^2 - 2\eta \frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \sqrt{\frac{\nu}{\pi}} \sigma \delta - \left(\frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\right)^2 \frac{\nu}{\pi} \sigma^2 \delta^2 \\
 &= \frac{\nu}{\nu-2} \sigma^2 - \left(\frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\right)^2 \frac{\nu}{\pi} \sigma^2 \delta^2.
 \end{aligned}$$

Assim, a esperança e variância de uma variável U com distribuição t-Student assimétrica são dadas, respectivamente, por:

$$\begin{aligned}
 E(U) &= \eta + \frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \sqrt{\frac{\nu}{\pi}} \sigma \delta, & \nu > 1, \\
 V(U) &= \frac{\nu}{\nu-2} \sigma^2 - \left(\frac{\Gamma\left(\frac{\nu-1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)}\right)^2 \frac{\nu}{\pi} \sigma^2 \delta^2, & \nu > 2.
 \end{aligned}$$

A.3 Esperança e variância da distribuição slash assimétrica

Seja $Z \sim \text{Beta}(\nu, 1)$, então:

$$\begin{aligned}
 E\left(Z^{-k/2}\right) &= \int_{-\infty}^{\infty} z^{-k/2} f(z) dz \\
 &= \int_0^1 z^{-k/2} \nu z^{\nu-1} dz \\
 &= \nu \int_0^1 z^{\nu-\frac{k}{2}-1} dz \\
 &= \nu \frac{1}{\nu-\frac{k}{2}} z^{\nu-\frac{k}{2}} \Big|_0^1 \\
 &= \frac{2\nu}{2\nu-k}.
 \end{aligned} \tag{A.4}$$

Utilizando a [Equação A.4](#) com $k = 1$ e a [Equação A.1](#), obtemos a esperança de uma variável aleatória U com distribuição slash assimétrica:

$$E(U) = E(\eta + Z^{-1/2}Y) = \eta + E(Z^{-1/2})E(Y) = \eta + \frac{2\nu}{2\nu-1} \sqrt{\frac{2}{\pi}} \delta\sigma, \quad \text{onde } \nu > 1/2. \tag{A.5}$$

Utilizando as equações (A.5), (A.1) e (A.4) com $k = 1$ e $k = 2$, a variância de uma variável U com distribuição slash assimétrica é dada por:

$$\begin{aligned}
 V(U) &= E(U^2) - (E(U))^2 \\
 &= E((\eta + Z^{-1/2}Y)^2) - \left(\eta + \frac{2\nu}{2\nu-1} \sqrt{\frac{2}{\pi}} \delta\sigma\right)^2 \\
 &= E(\eta^2 + 2\eta Z^{-1/2}Y + Z^{-1}Y^2) - \eta^2 - 2\eta \frac{2\nu}{2\nu-1} \sqrt{\frac{2}{\pi}} \delta\sigma - \left(\frac{2\nu}{2\nu-1}\right)^2 \frac{2}{\pi} \delta^2 \sigma^2 \\
 &= \eta^2 + 2\eta E(Z^{-1/2})E(Y) + E(Z^{-1})E(Y^2) - \eta^2 - 2\eta \frac{2\nu}{2\nu-1} \sqrt{\frac{2}{\pi}} \delta\sigma - \left(\frac{2\nu}{2\nu-1}\right)^2 \frac{2}{\pi} \delta^2 \sigma^2 \\
 &= 2\eta \frac{2\nu}{2\nu-1} \sqrt{\frac{2}{\pi}} \delta\sigma + \frac{\nu}{\nu-1} \sigma^2 - 2\eta \frac{2\nu}{2\nu-1} \sqrt{\frac{2}{\pi}} \delta\sigma - \left(\frac{2\nu}{2\nu-1}\right)^2 \frac{2}{\pi} \delta^2 \sigma^2 \\
 &= \frac{\nu}{\nu-1} \sigma^2 - \left(\frac{2\nu}{2\nu-1}\right)^2 \frac{2}{\pi} \delta^2 \sigma^2.
 \end{aligned}$$

Assim, a esperança e variância de uma variável U com distribuição slash assimétrica são dadas, respectivamente, por:

$$\begin{aligned}
 E(U) &= \eta + \frac{2\nu}{2\nu-1} \sqrt{\frac{2}{\pi}} \delta\sigma, & \nu > 1/2, \\
 V(U) &= \frac{\nu}{\nu-1} \sigma^2 - \left(\frac{2\nu}{2\nu-1}\right)^2 \frac{2}{\pi} \delta^2 \sigma^2, & \nu > 1.
 \end{aligned}$$

CÓDIGOS

Código-fonte 1 – Gerando conjuntos de dados simulados a partir do modelo SV-STL

```
1: mu <- -9
2: phi <- 0.95
3: sigma <- 0.15
4: rho <- -0.5
5: nu <- 5
6: alpha <- -0.5
7: n <- 1000
8: delta <- alpha/sqrt(1+alpha^2)
9: z <- e0 <- e1 <- k <- h <- y <- rep(0,n)
10: for (t in 1:n){
11:   z[t] <- rgamma(1, shape = nu/2, rate = nu/2)
12:   e0[t] <- rnorm(1)
13:   e1[t] <- rnorm(1)
14:   k[t] <- rnorm(1)
15: }
16: h[1] <- rnorm(1, mean = mu, sd = sigma/sqrt(1-phi^2) )
17:
18: for (t in 1:(n-1)) h[t+1] <- mu + phi*(h[t]-mu) + sigma*(rho*
    e1[t]+sqrt(1-rho^2)*k[t])
19:
20: for (t in 1:n) y[t] <- exp(0.5*h[t])*(1/sqrt(z[t]))*(delta*abs(
    e0[t])+sqrt(1-delta^2)*e1[t])-(gamma(0.5*(nu-1))/gamma(0.5*
    nu))*sqrt(nu/pi)*delta)
21:
22: save(y, file = "sample_data.RData")
```

Código-fonte 2 – Código em Stan do modelo SV-N

```

1: stan_code_n = "
2: data {
3:   int<lower=0> n;
4:   vector[n] y;
5: }
6: parameters {
7:   real mu;
8:   real<lower=0,upper=1> phistar;
9:   real<lower=0> sigma2;
10:  vector[n] h_std;
11: }
12: transformed parameters {
13:   real<lower=-1, upper=1> phi;
14:   real<lower=0> sigma;
15:   vector[n] h;
16:   phi = fma(phistar,2,- 1);
17:   sigma = sqrt(sigma2);
18:   h = h_std*sigma;
19:   h[1] /= sqrt(1-square(phi));
20:   h += mu;
21:   for (t in 2:n) h[t] += phi*(h[t-1]-mu);
22: }
23: model {
24:   mu ~ normal(0,10);
25:   phistar ~ beta(20,1.5);
26:   sigma2 ~ inv_gamma(2.5,0.025);
27:   h_std ~ std_normal();
28:   y ~ normal(0,exp(0.5*h));
29: }
30: generated quantities{
31:   real loglike;
32:   vector[n] log_lik;
33:   loglike = 0;
34:   for (t in 1:n) {
35:     loglike=loglike+normal_lpdf(y[t]|0,exp(0.5*h[t])); //dic
36:     log_lik[t] = normal_lpdf(y[t]|0,exp(0.5*h[t]));
37:   }
38: }
39: "

```

Código-fonte 3 – Código em Stan do modelo SV-NL

```

1: stan_code_nl = "
2: data {
3:   int<lower=0> n;
4:   vector[n] y;
5: }
6: parameters {
7:   real mu;
8:   real<lower=0,upper=1> phistar;
9:   real<lower=0> sigma2;
10:  real<lower=0,upper=1> rhostar;
11:  vector[n] h_std;
12: }
13: transformed parameters {
14:  real<lower=-1, upper=1> phi;
15:  real<lower=0> sigma;
16:  real<lower=-1, upper=1> rho;
17:  vector[n] h;
18:  phi = fma(phistar,2,- 1);
19:  sigma = sqrt(sigma2);
20:  rho = 2*rhostar-1;
21:  h = h_std*sigma*sqrt(1-square(rho));
22:  h[1] /= (sqrt(1-square(phi))*sqrt(1-square(rho)));
23:  h += mu;
24:  for (t in 2:n) h[t]+=phi*(h[t-1]-mu)+sigma*rho*exp(-0.5*h[t
    -1])*y[t-1];
25: }
26: model {
27:  mu ~ normal(0,10);
28:  phistar ~ beta(20, 1.5);
29:  sigma2 ~ inv_gamma(2.5,0.025);
30:  rhostar ~ beta(3,6);
31:  h_std ~ std_normal();
32:  y ~ normal(0, exp(0.5*h));
33: }
34: generated quantities{
35:  real loglike;
36:  vector[n] log_lik;
37:  loglike = 0;
38:  for (t in 1:n) {
39:    loglike=loglike+normal_lpdf(y[t]|0,exp(0.5*h[t])); //dic

```

```

40:     log_lik[t] = normal_lpdf(y[t]| 0, exp(0.5*h[t]));
41:   }
42: }
43: "

```

Código-fonte 4 – Código em Stan do modelo SV-T

```

1: stan_code_t = "
2: data {
3:   int<lower=0> n;
4:   vector[n] y;
5: }
6: parameters {
7:   real mu;
8:   real<lower=0,upper=1> phistar;
9:   real<lower=0> sigma2;
10:  real<lower=2> nu;
11:  vector<lower=0> [n] z;
12:  vector[n] h_std;
13: }
14: transformed parameters {
15:   real<lower=-1, upper=1> phi;
16:   real<lower=0> sigma;
17:   vector[n] h;
18:   phi = fma(phistar,2, - 1);
19:   sigma = sqrt(sigma2);
20:   h = h_std*sigma;
21:   h[1] /= sqrt(1-square(phi));
22:   h += mu;
23:   for (t in 2:n) h[t] += phi*(h[t-1]-mu);
24: }
25: model {
26:   mu ~ normal(0,10);
27:   phistar ~ beta(20, 1.5);
28:   sigma2 ~ inv_gamma(2.5,0.025);
29:   nu ~ exponential(0.3333333);
30:   z ~ gamma(0.5*nu,0.5*nu);
31:   h_std ~ std_normal();
32:   y ~ normal(0, exp(0.5*h) ./sqrt(z));
33: }
34: generated quantities{
35:   real loglike;

```

```

36:   vector[n] log_lik;
37:   loglike = 0;
38:   for (t in 1:n) {
39:     loglike=loglike+normal_lpdf(y[t]|0,exp(0.5*h[t])/sqrt(z[t]))
      );      //dic
40:     log_lik[t] = normal_lpdf(y[t]|0,exp(0.5*h[t])/sqrt(z[t]));
41:   }
42: }
43: "

```

Código-fonte 5 – Código em Stan do modelo SV-TL

```

1: stan_code_t1 = "
2: data {
3:   int<lower=0> n;
4:   vector[n] y;
5: }
6: parameters {
7:   real mu;
8:   real<lower=0,upper=1> phistar;
9:   real<lower=0> sigma2;
10:  real<lower=0,upper=1> rhostar;
11:  real<lower=2> nu;
12:  vector<lower=0> [n] z;
13:  vector[n] h_std; // std log volatility time t
14: }
15: transformed parameters {
16:   real<lower=-1, upper=1> phi;
17:   real<lower=0> sigma;
18:   real<lower=-1, upper=1> rho;
19:   vector[n] h;
20:   phi = fma(phistar,2,- 1);
21:   sigma = sqrt(sigma2);
22:   rho = 2*rhostar-1;
23:   h = h_std*sigma*sqrt(1-square(rho));
24:   h[1] /= (sqrt(1-square(phi))*sqrt(1-square(rho)));
25:   h += mu;
26:   for (t in 2:n) h[t]+=phi*(h[t-1]-mu)+sigma*rho*exp(-0.5*h[t-1])*y[t-1]*sqrt(z[t-1]);
27: }
28: model {
29:   mu ~ normal(0,10);

```

```

30:  phistar ~ beta(20, 1.5);
31:  sigma2 ~ inv_gamma(2.5,0.025);
32:  rhostar ~ beta(3,6);
33:  nu ~ exponential(0.3333333);
34:  z ~ gamma(0.5*nu,0.5*nu);
35:  h_std ~ std_normal();
36:  y ~ normal(0, exp(0.5*h) ./sqrt(z));
37: }
38: generated quantities{
39:   real loglike;
40:   vector[n] log_lik;
41:   loglike = 0;
42:   for (t in 1:n) {
43:     loglike=loglike+normal_lpdf(y[t]|0,exp(0.5*h[t])/sqrt(z[t]))
      ); //dic
44:     log_lik[t] = normal_lpdf(y[t]|0,exp(0.5*h[t])/sqrt(z[t]));
45:   }
46: }
47: "

```

Código-fonte 6 – Código em Stan do modelo SV-STL

```

1: stan_code_stl = "
2: data {
3:   int<lower=0> n;
4:   vector[n] y;
5: }
6: parameters {
7:   real mu;
8:   real<lower=0,upper=1> phistar;
9:   real<lower=0> sigma2;
10:  real<lower=0,upper=1> rhostar;
11:  real<lower=2> nu;
12:  real<lower=-1, upper=1> delta;
13:  vector<lower=0> [n] z;
14:  vector<lower=0> [n] e0;
15:  vector[n] h_std;
16: }
17: transformed parameters {
18:   real<lower=-1, upper=1> phi;
19:   real<lower=0> sigma;
20:   real<lower=-1,upper=1> rho;

```

```

21:  real alpha;
22:  real meanST;
23:  vector[n] h;
24:  phi = fma(phistar,2,- 1);
25:  sigma = sqrt(sigma2);
26:  rho = 2*rhostar - 1;
27:  alpha = delta/sqrt(1-square(delta));
28:  meanST=tgamma(0.5*(nu-1))/tgamma(0.5*nu)*sqrt(nu/pi())*delta;
29:  h = h_std*sigma*sqrt(1-square(rho));
30:  h[1] /= (sqrt(1-square(phi))*sqrt(1-square(rho)));
31:  h += mu;
32:  for (t in 2:n) h[t]+=phi*(h[t-1]-mu)+sigma*rho*pow(1-square(
    delta),-0.5)*(sqrt(z[t-1])*(exp(-0.5*h[t-1])*y[t-1]+meanST)-
    delta*e0[t-1]));
33: }
34: model {
35:  mu ~ normal(0,10);
36:  phistar ~ beta(20, 1.5);
37:  sigma2 ~ inv_gamma(2.5,0.025);
38:  rhostar ~ beta(3,6);
39:  nu ~ exponential(0.3333333);
40:  delta ~ uniform(-1, 1);
41:  z ~ gamma(0.5*nu,0.5*nu);
42:  e0 ~ std_normal();
43:  h_std ~ std_normal();
44:  y ~ normal(exp(0.5*h) .*(delta*e0 ./sqrt(z)-meanST),exp(0.5*h
    ) ./sqrt(z)*sqrt(1-square(delta)));
45: }
46: generated quantities{
47:  real loglike;
48:  vector[n] log_lik;
49:  loglike = 0;
50:  for (t in 1:n) {
51:    loglik =loglike+normal_lpdf(y[t]|exp(0.5*h[t])*(delta*e0[t
    ]/sqrt(z[t])-meanST),exp(0.5*h[t])/sqrt(z[t])*sqrt(1-square(
    delta))); //dic
52:    log_lik[t]=normal_lpdf(y[t]|exp(0.5*h[t])*(delta*e0[t]/sqrt
    (z[t])-meanST),exp(0.5*h[t])/sqrt(z[t])*sqrt(1-square(delta)
    ));
53:  }
54: }
55: "

```

Código-fonte 7 – Código em Stan do modelo SV-SSL

```

1: stan_code_ssl = "
2: data {
3:   int<lower=0> n;
4:   vector[n] y;
5: }
6: parameters {
7:   real mu;
8:   real<lower=0,upper=1> phistar;
9:   real<lower=0> sigma2;
10:  real<lower=0,upper=1> rhostar;
11:  real<lower=1> nu;
12:  real<lower=-1, upper=1> delta;
13:  vector<lower=0,upper=1> [n] z;
14:  vector<lower=0> [n] e0;
15:  vector[n] h_std;
16: }
17: transformed parameters {
18:   real<lower=-1, upper=1> phi;
19:   real<lower=0> sigma;
20:   real<lower=-1,upper=1> rho;
21:   real alpha;
22:   real meanSS;
23:   vector[n] h;
24:   phi = fma(phistar,2,- 1);
25:   sigma = sqrt(sigma2);
26:   rho = 2*rhostar - 1;
27:   alpha = delta/sqrt(1 - square(delta));
28:   meanSS = 2*nu/(2*nu-1)*sqrt(2/pi())*delta;
29:   h = h_std*sigma*sqrt(1-square(rho));
30:   h[1] /= (sqrt(1-square(phi))*sqrt(1-square(rho)));
31:   h += mu;
32:   for (t in 2:n) h[t]+=phi*(h[t-1]-mu)+sigma*rho*pow(1-square(
      delta),-0.5)*(sqrt(z[t-1])*(exp(-0.5*h[t-1])*y[t-1]+meanSS)-
      delta*e0[t-1]));
33: }
34: model {
35:   mu ~ normal(0,10);
36:   phistar ~ beta(20, 1.5);

```

```

37:  sigma2 ~ inv_gamma(2.5,0.025);
38:  rhostar ~ beta(3,6);
39:  nu ~ gamma(0.08,0.04);
40:  delta ~ uniform(-1, 1);
41:  z ~ beta(nu,1);
42:  e0 ~ std_normal();
43:  h_std ~ std_normal();
44:  y ~ normal(exp(0.5*h) .*(delta*e0 ./sqrt(z)-meanSS),exp(0.5*h
    ) ./sqrt(z)*sqrt(1-square(delta)));
45: }
46: generated quantities{
47:   real loglike;
48:   vector[n] log_lik;
49:   loglike = 0;
50:   for (t in 1:n) {
51:     loglike=loglike+normal_lpdf(y[t]|exp(0.5*h[t])*(delta*e0[t]
      ]/sqrt(z[t])-meanSS),exp(0.5*h[t])/sqrt(z[t])*sqrt(1-square(
      delta))); //dic
52:     log_lik[t]=normal_lpdf(y[t]|exp(0.5*h[t])*(delta*e0[t]/sqrt
      (z[t])-meanSS),exp(0.5*h[t])/sqrt(z[t])*sqrt(1-square(delta)
      ));
53:   }
54: }
55: "

```

Código-fonte 8 – Funções para o cálculo do LFO-CV

```

1: # more stable than log(sum(exp(x)))
2: log_sum_exp <- function(x) {
3:   max_x <- max(x)
4:   max_x + log(sum(exp(x - max_x)))
5: }
6: # more stable than log(mean(exp(x)))
7: log_mean_exp <- function(x) {
8:   log_sum_exp(x) - log(length(x))
9: }
10: # compute log of raw importance ratios
11: # sums over observations *not* over posterior samples
12: sum_log_ratios <- function(ll, ids = NULL) {
13:   if (!is.null(ids)) ll <- ll[, ids, drop = FALSE]
14:   - rowSums(ll)
15: }

```

Código-fonte 9 – Código em Stan para fazer previsão um passo a frente

```

1: forecast_stl <- stan_model(model_code = "
2: data {
3:   int<lower=0> n;
4:   vector[n] y;
5:   real future_y; // future observation to obtain log_lik
6: }
7: parameters {
8:   real mu;
9:   real<lower=-1, upper=1> phi;
10:  real<lower=0> sigma;
11:  real<lower=-1, upper=1> rho;
12:  real<lower=2> nu;
13:  real<lower=-1, upper=1> delta;
14:  vector<lower=0> [n] z;
15:  vector<lower=0> [n] e0;
16:  vector[n] h;
17:  real meanST;
18: }
19: generated quantities{
20:   real h_fore;
21:   real<lower=0> z_fore;
22:   real e0_fore;
23:   real log_lik;
24:   h_fore = mu+phi*(h[n]-mu)+sigma*(rho/sqrt(1-square(delta))*(
      sqrt(z[n])*(y[n]*exp(-0.5*h[n])+meanST)-delta*e0[n])+sqrt(1-
      square(rho))*normal_rng(0,1));
25:   z_fore = gamma_rng(0.5*nu,0.5*nu);
26:   e0_fore = normal_rng(0,1);
27:   log_lik = normal_lpdf(future_y|exp(0.5*h_fore)*(delta*fabs(
      e0_fore)/sqrt(z_fore)-meanST), exp(0.5*h_fore)/sqrt(z_fore)*
      sqrt(1-square(delta)));
28: }"
29: )

```

Código-fonte 10 – Código para cálculo dos critérios de informação (modelo SV-STL)

```

1: library(rstan)
2: library(bayesplot)
3: library(loo)
4: rstan_options(auto_write = TRUE)
5: options(mc.cores = parallel::detectCores())
6:
7: load("sample_data.RData")
8: fit <- stan(model_code=stan_code_stl,
9:           data=list(y=y, n=length(y)),
10:          chains=5, thin=5, iter=7000, warmup=2000)
11: #Calculo DIC:
12: sfit<-summary(fit)
13: hrow <- grep("h\\[",rownames(sfit$summary))
14: hbar <- sfit$summary[hrow, "mean"]
15: deltabar <- sfit$summary["delta","mean"]
16: nubar <- sfit$summary["nu","mean"]
17: e0row <- grep("e0\\[",rownames(sfit$summary))
18: e0bar <- sfit$summary[e0row, "mean"]
19: zrow <- grep("z\\[",rownames(sfit$summary))
20: zbar <- sfit$summary[zrow, "mean"]
21:
22: loglike_bar = sum(dnorm(y,exp(0.5*hbar)*((deltabar*e0bar/sqrt(
      zbar)-(gamma(0.5*(nubar-1))/gamma(0.5*nubar))*sqrt(nubar/pi)
      *deltabar),sqrt((1-deltabar^2)/zbar)*exp(0.5*hbar),log=T))
23: loglike = extract_log_lik(fit, parameter_name="loglike")
24: p_DIC = 2*(loglike_bar - mean(loglike))
25: DIC = -2*loglike_bar + 2*p_DIC
26: print(DIC)
27: #Cálculo L00
28: LL <- extract_log_lik(fit, merge_chains = FALSE)
29: r_eff <- relative_eff(exp(LL))
30: Loo = loo(LL, r_eff = r_eff)
31: print(Loo)
32: #Cálculo WAIC
33: log_lik <- extract(fit, "log_lik")$log_lik
34: Waic = waic(log_lik)
35: print(Waic)
36: # Cálculo LFO-CV
37: N=length(y)
38: L <- 700

```

```

39: k_thres <- 0.6
40: nsamples <- dim(fit)[1]*dim(fit)[2]
41: loglik <- matrix(nrow = nsamples, ncol = N)
42: approx_elpds_1sap <- rep(NA, N)
43: ks <- NULL
44: refits <- NULL
45: fit_past <- fit
46: i_refit <- N # last observation included in the model fitting
47: for (i in (N - 1):L) {
48:   loglik[,i+1] = extract(fit_past, pars="log_lik")$log_lik[,i+1]
49:   logratio <- sum_log_ratios(loglik, (i + 1):i_refit)
50:   psis_obj <- suppressWarnings(psis(logratio))
51:   k <- pareto_k_values(psis_obj)
52:   ks <- c(ks, k)
53:   if (k > k_thres) {
54:     i_refit <- i
55:     refits <- c(refits, i)
56:     df_past <- y[1:i]
57:     df_oos <- y[i+1]
58:     fit_past <- stan(model_code=stan_code_stl,
59:                      data=list(y=df_past, n=length(df_past)),
60:                      chains=5, thin=5, iter=7000, warmup=2000)
61:     standat <- list(n=length(df_past),
62:                    y=df_past, future_y=df_oos)
63:     forecasted <- gqs(forecast_stl,
64:                       draws=as.matrix(fit_past), data=standat)
65:     loglik[,i+1] <- extract(forecasted, pars="log_lik")$log_lik
66:     approx_elpds_1sap[i+1] <- log_mean_exp(loglik[, i+1])
67:   } else {
68:     lw <- weights(psis_obj, normalize = TRUE)[, 1]
69:     approx_elpds_1sap[i+1] <- log_sum_exp(lw + loglik[, i+1])
70:   }
71: }
72: cat("Using threshold ", k_thres,
73:     ", model was refit ", length(refits),
74:     " times, at observations", refits)
75:
76: aprox_elpd_1sap <- sum(approx_elpds_1sap, na.rm = TRUE)

```
