



Instituto de  
MATEMÁTICA  
E ESTATÍSTICA

UFRGS



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

DEPARTAMENTO DE ESTATÍSTICA

**Modelagem de volatilidade utilizando modelos GAS semiparamétricos**

**GABRIEL MARTINS BROCK**

Porto Alegre  
2016

# Modelagem de volatilidade utilizando modelos GAS semiparamétricos

Autor: Gabriel Martins Brock

Trabalho de Conclusão de Curso  
apresentado para obtenção  
do grau de Bacharel em Estatística.

Banca Examinadora:

Professor Dr. Hudson Torrent (orientador) - UFRGS

Professor Dr. Flávio Ziegelmann – UFRGS

Professor Dr. Marcio Valk - UFRGS

Porto Alegre, 20 de dezembro de 2016.

*“Se você fizer algo errado, repita esse erro mais cinco ou seis vezes...  
Só para ter significância estatística.”*

- Desconhecido

## **Agradecimentos**

Agradeço ao meu orientador por toda a ajuda e paciência, aos familiares e amigos pelo apoio, tolerância e compreensão e a todos que me auxiliaram e ensinaram durante a graduação, principalmente neste último semestre.

## Resumo

Técnicas recentes propõe a utilização de modelos semiparamétricos para diminuir o viés e compensar a perda de eficiência no caso de se supor (erroneamente) distribuição normal para os erros de um GARCH. Este trabalho reproduziu em dados simulados as condições de má especificação do GARCH para comparar com a metodologia do GAS semiparamétrico. Em um contexto empírico, utilizaram-se dados da IBOVESPA para realizar uma comparação semelhante.

Foi observado que a abordagem semiparamétrica para estimar os parâmetros do modelo produz estimativas mais acuradas ou equivalentes às usuais. Em casos onde a mecânica da volatilidade não é especificada de maneira correta o desempenho do SP-GAS foi superior comparado ao modelo GARCH gaussiano tradicional.

Palavras-chave: Séries temporais, volatilidade, semiparamétrico.

## Abstract

Recent techniques propose the use of semiparamétrico models to shrink the bias and compensate the efficiency loss when supposing (possibly incorrectly) normal distribution for the errors in a GARCH model. This paper reproduced in simulated data conditions where GARCH is miss specified to compare with semiparametric GAS. In an empirical context, IBOVESPA data was used to make a similar comparison.

It was observed that to estimate the parameters of the model, the semiparametric approach produced more accurate or equivalent estimates comparing to the usual methods. SP-GAS was superior compared to the traditional GARCH when the volatility mechanics is not specified in the right way.

Keywords: Time series, volatility, semiparametric.

## Sumário

1- Introdução.....	6
2- Especificações e definições do modelo.....	8
3- Simulações de Monte Carlo .....	14
4- Aplicação a dados empíricos .....	17
5- Considerações finais .....	20
Referências Bibliográficas .....	21

# 1-Introdução

Aplicações mais comuns de séries temporais envolvem a modelagem da estrutura do valor médio condicional ao tempo de uma série. Porém, de maneira mais frequente em áreas como a estatística ou economia se torna necessário modelar a variância condicional. Medir a incerteza com base em suas observações anteriores definiu a base para os modelos ARCH (*Autoregressive Conditional Heteroskedasticity*). Alguns anos mais tarde, estes foram generalizados criando-se a classe de modelos GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*).

Recorrer à modelagem ARCH ou GARCH exige a suposição de uma distribuição para os erros do modelo. Em diversos estudos empíricos a distribuição normal é escolhida para tal por conveniência. Mesmo que essa suposição seja incorreta, os estimadores produzidos são consistentes. Porém, a perda de eficiência estatística pode ser considerável. Para evitar este tipo de situação, outras distribuições podem ser usadas, como a t de student em casos de caudas mais pesadas do que a gaussiana ou a gamma para casos de assimetria diferente de zero. Abordagens semiparamétricas dos modelos GARCH também podem ser empregadas como alternativa.

Creal *et al.* (2011) sugeriram em seu trabalho a classe de modelos GAS (*Generalized Autoregressive Scores*) que utiliza um escore baseado na derivada da densidade da verossimilhança dentro da mecânica da variância condicional, garantindo, por exemplo, que valores nas caudas mais curtas de distribuições assimétricas sejam mais informativos quanto a um acréscimo na volatilidade. Este escore depende da densidade completa da série e não apenas dos momentos de primeira ou segunda ordem, destacando o modelo GAS de várias abordagens existentes na literatura.

O trabalho de Blasques *et al.* (2016) estende a classe de modelos GAS para o contexto semiparamétrico. Utilizando simulações de quatro tipos diferentes de distribuições (gaussiana, duas t de student e mistura de normais) para a dinâmica de volatilidade, Blasques *et al* (2016) testaram o ajuste do modelo proposto contra três outras estimativas. São elas: o

estimador de máxima verossimilhança, uma modificação do modelo GARCH e o modelo GAS com uma distribuição t de student. Para aferir a qualidade do SP-GAS frente aos demais modelos quando nenhum deles é bem especificado os autores selecionaram padrões de volatilidade determinísticos e periódicos, tal como a função seno. Por último testou-se a qualidade do modelo a um contexto real. Utilizando valores do retorno da taxa de câmbio entre euro e dólar, novamente foram ajustados os quatro modelos e o erro das estimativas computado.

Ao final, os autores mostram que a nova metodologia possui resultados satisfatórios e consistentes na maioria dos experimentos. Foram apontadas vantagens sobre as abordagens convencionais quando há erros na especificação da distribuição para a volatilidade, como redução dos resíduos e estimadores menos viesados se comparados ao GARCH e ao GAS tradicional. Foi ressaltado que a primeira iteração do processo semiparamétrico traz um ganho considerável para a estimação, sem a necessidade de repetir todo o procedimento iterativo.

Este trabalho visa reproduzir a metodologia do estudo de Blasques *et al* (2016). Seguindo a mesma lógica para testar a qualidade do ajuste do modelo SP-GAS contra outras estimativas, inicialmente foi feito o estudo com dados simulados utilizando distribuições normal e t de Student para modelar a mecânica da volatilidade dos dados.

Para abordar o GAS semiparamétrico em um quadro empírico foi utilizado como modelo o trabalho de Santos e Ziegelmann (2011). Em seu trabalho, os autores visam comparar o desempenho de modelos GARCH e modelos aditivos semiparamétricos em épocas de crise fazendo uso da taxa de retorno de ações da IBOVESPA. A ideia é utilizar uma metodologia de comparação semelhante, porém utilizando a modelagem GARCH com o SP-GAS de Blasques *et al* (2016). Sendo assim, este trabalho se vale do método de modelagem da média condicional da série, um AR(2), nos valores de retorno e os resíduos resultantes são empregados na modelagem da volatilidade.

## 2-Especificações e definições do modelo

Como já citado, o foco deste trabalho é modelar a variância condicional de uma série. Essa variância condicional também é chamada de volatilidade, que embora não possa ser medida diretamente apresenta características notáveis. A volatilidade tende a aparecer em grupos, alguns de maior e outros de menor variabilidade e pode reagir de formas distintas a valores positivos ou negativos da série. Padrões deste tipo ocorrem em diversos campos da ciência. Notadamente, os campos de economia e finanças foram onde mais contribuições e aplicações práticas de modelagem da variância condicional da série surgiram.

Esta seção visa descrever os modelos utilizados e algumas de suas características, começando pelos modelos ARCH e GARCH, seguidos da modelagem GAS paramétrica e semiparamétrica.

### 2.1 ARCH

Os precursores da modelagem da volatilidade na literatura são os modelos ARCH (*Autoregressive Conditional Heteroskedasticity*) propostos por Engle (1982). Em seu artigo, a problemática central era definir um método para estimar a variância condicional da inflação do Reino Unido. A fórmula geral de um ARCH(p) pode ser escrita como:

$$y_t = \mu + \xi_t = \mu + \tilde{f}_t^{\frac{1}{2}} \varepsilon_t, \quad (1')$$

$$\tilde{f}_t = \alpha_0 + \alpha_1 y_{t-1}^2 + \dots + \alpha_p y_{t-p}^2, \quad (2')$$

onde  $\varepsilon_t \sim q(\varepsilon_t)$ .

O termo  $y_t \in \mathbb{R}$ ,  $t = 1, 2, \dots, T$  é um valor de uma série temporal de T observações com média condicional  $\mu \in \mathbb{R}$ .  $\varepsilon_t$  representa o termo de erro padronizado e segue uma distribuição  $q(\cdot)$ .  $\tilde{f}_t \in \mathbb{R}^+$  é a variância condicional, a volatilidade que se tem interesse em modelar. Os coeficientes devem seguir  $\alpha_0 > 0$ ,  $\alpha_i \geq 0$ ,  $i = 1, 2, \dots, p$ .

## 2.2 GARCH

Com base no ARCH, Bollerslev (1986) criou a classe de modelos GARCH (*Generalized Autoregressive Conditional Heteroskedasticity*). Atualmente é um dos modelos mais utilizados na ciência quando há necessidade de modelar a volatilidade da série temporal. Seu trabalho adicionou aos modelos ARCH termos da volatilidade de tempos passados da série (*lags*). Dado o modelo GARCH(p,q), o termo  $\tilde{f}_t$  da equação (1') é apresentado da seguinte forma:

$$\tilde{f}_t = \alpha_0 + \sum_{i=1}^p \alpha_i y_{t-i}^2 + \sum_{j=1}^q \beta_j \tilde{f}_{t-1}, \quad (3')$$

Os parâmetros devem satisfazer  $\beta_j \geq 0, \sum(\alpha_i + \beta_i) < 1, i = 1, \dots, \max(p, q)$ .

Em diversos casos, o GRACH reduz o número de parâmetros necessários para explicar uma série quando comparado ao modelo ARCH. Em aplicações práticas isto é preferível, pois a complexidade de um modelo também serve como critério de seleção da metodologia final.

Um dos casos mais simples deste modelo é o GARCH(1,1), ou seja

$$\tilde{f}_t = \alpha_0 + \alpha_1 y_{t-1}^2 + \beta_1 \tilde{f}_{t-1}. \quad (4')$$

Blasques *et al.* (2016) apresentam uma maneira diferente de escrever o GARCH(1,1). Segundo eles, a volatilidade de (1') pode ser escrita como:

$$\tilde{f}_{t+1} = \tilde{\omega}(1 - \tilde{\alpha} - \tilde{\beta}) + \tilde{\alpha}\xi_t^2 + \tilde{\beta}\tilde{f}_t, \quad (5')$$

onde  $\tilde{\omega}, \tilde{\alpha}, \tilde{\beta} > 0$  são parâmetros estáticos satisfazendo  $\tilde{\alpha} + \tilde{\beta} < 1$ .

A distribuição  $q(\cdot)$  dificilmente é conhecida, mas é possível assumir distribuição normal para  $q(\cdot)$  e obter estimativas consistentes. Porém, neste caso a perda de eficiência é significativa. Conforme já comentado, para diminuir esta perda algumas alternativas já foram propostas, tal como testar outras distribuições para  $q(\cdot)$  ou utilizar métodos semiparamétricos.

## 2.3 GAS

Outra possibilidade de substituto para casos de má especificação do GARCH gaussiano é o modelo GAS (*Generalized Autoregressive Scores*) de Creal *et al.* (2011). A sua construção é muito similar a dos modelos GARCH, mas possui diferenças notáveis. Seja o modelo GAS(1,1):

$$y_t = \mu + \xi_t = \mu + \exp(f_t/2)\varepsilon_t, \quad (1)$$

$$f_{t+1} = \omega(1 - \beta) + \alpha s_t + \beta f_t, \quad (2)$$

$$s_t = \frac{\partial \ln[p(y_t|f_t)]}{\partial f_t} = -0.5\{\exp(-f_t/2)\xi_t * h[q(\exp(-f_t/2)\xi_t)] - 1\}, \quad (3)$$

$$\text{onde } h[q(x)] = \frac{\partial \ln[q(x)]}{\partial x} \text{ e } \varepsilon_t \sim q(\varepsilon_t).$$

Pode-se ver que a incorporação da volatilidade é feita utilizando a exponencial de  $f_t/2$  em (1), diferente dos modelos ARCH e GARCH que aplicam a raiz de  $f_t$  conforme (1'). A principal característica deste modelo é o escore  $s_t$  que é definido como a derivada da função de densidade geradora de  $y_t$  condicional à volatilidade. Desta forma, incorporam-se aos valores de  $f_t$  a magnitude e a direção da distribuição da verossimilhança. Isto garante, por exemplo, que valores nas caudas mais curtas de distribuições assimétricas ou valores das caudas de distribuições de alta curtose sejam mais informativos quanto a um acréscimo na volatilidade.

Algumas escolhas específicas de distribuição para os erros são a gaussiana ou a t de student. Se tomarmos que  $\varepsilon_t \sim N(\mu, \sigma^2)$  teremos:

$$h[q(x)] = \frac{\partial \ln[q(x)]}{\partial x} = \frac{\mu - x}{\sigma^2}. \quad (4)$$

Ou no caso de  $\varepsilon_t \sim t(\nu)$ :

$$h[q(x)] = \frac{\partial \ln[q(x)]}{\partial x} = \frac{x(1-\nu)}{\nu+x^2}. \quad (5)$$

Outro ponto favorável ao modelo GAS é que há facilidade de estimação via máxima verossimilhança. É possível estimar os parâmetros do modelo maximizando o logaritmo da função de verossimilhança:

$$L_t(\underline{\theta}, q) = 1/T \sum_{t=1}^T -0.5f_t(\underline{\theta}) + \ln\{q[\exp(-f_t(\underline{\theta})/2) * (y_t - \mu)]\}, \quad (6)$$

onde  $\underline{\theta} = (\mu, \omega, \alpha, \beta)$ .

Em Creal *et al.* (2011) o modelo GAS foi estudado de forma minuciosa através de estudos simulados e empíricos. Concluiu-se que sua utilização reduz o erro de predição de um passo e que é uma alternativa para casos reais.

## 2.4 GAS semiparamétrico

Métodos semiparamétricos também são uma possibilidade para reduzir o efeito da má especificação para a densidade da volatilidade. Estes são designados desta maneira pelo fato de que são compostos de duas partes principais, uma paramétrica e outra não paramétrica. Cada vez mais esta alternativa tem sido proposta, dados os recentes avanços no campo da computação proporcionam mais facilidade para realizar algoritmos iterativos que repetidas vezes são inerentes ao método.

O trabalho de Blasques *et al.* (2016) propõe a extensão dos modelos GAS para o domínio semiparamétrico (SP-GAS) ao utilizar o estimador de densidade kernel em um processo iterativo para reestimar os parâmetros do modelo. O estimador kernel é um método de suavização da densidade baseado em uma amostra finita e no valor da banda  $b_t$ . Sendo assim, utiliza-se a estimativa de densidade kernel para encontrar uma aproximação para a distribuição dos erros  $q(\cdot)$ .

O desenvolvimento do GAS semiparamétrico é dado da seguinte forma: inicialmente, supõe-se uma distribuição para  $q(\cdot)$  para encontrar as estimativas  $\hat{\underline{\theta}}$ . Em seguida, estas estimativas servem para calcular os resíduos  $\hat{\varepsilon}_t = \exp(-f_t(\hat{\underline{\theta}})/2) * (y_t - \hat{\mu})$  que são empregados na estimativa de densidade kernel

$$\hat{q}_t(x) = \frac{1}{T * b_t} * \sum_{t=1}^T K\left(\frac{\hat{\varepsilon}_t - x}{b_t}\right), \quad (7)$$

onde  $K(\cdot)$  é a uma distribuição conhecida.

O valor da banda  $b_t \in \mathbb{R}^+$  é arbitrário, podendo ser usado conhecimento prévio sobre o problema (literatura da área) ou métodos de validação de banda (como *cross validation*). Existe um *trade-off* entre o viés e a variância do estimador no que diz respeito a banda. Quanto maior o valor da banda, maior será o viés e menor será a variância. Idealmente procura-se o valor da banda que minimize o erro quadrático médio integrado.

Os valores encontrados para  $\hat{q}_t(\cdot)$  são centralizados e padronizados para se obter média zero e variância unitária e são usados para reestimar os parâmetros do modelo. Este processo é chamado de SP-GAS(1), ou seja as estimativas encontradas pelo primeiro passo semiparamétrico. Uma vez calculado, é possível valer-se das estimativas do SP-GAS(1) para obter novos valores para os resíduos  $\hat{\varepsilon}_t$  e computar numa nova estimativa para a densidade kernel, criando assim o SP-GAS(2). Generalizando, SP-GAS(i) representa a i-ésima iteração deste procedimento.

Blasques *et al.* (2016) sugerem repetir as iterações até o ponto onde o valor da máxima verossimilhança de um passo seja menor que o valor do passo anterior. Sendo assim, tomam-se os valores estimados dos parâmetros do passo anterior como as estimativas finais do modelo. Este procedimento é ilustrado de forma mais concisa pela figura 1.

1. Estimar  $\underline{\theta}$  utilizando uma distribuição conhecida via máxima verossimilhança.
2. Com base em  $\underline{\hat{\theta}}$ , calcular  $\hat{\varepsilon}_t$  e aplicar na função  $\hat{q}_t$ .
3. Dado  $\hat{q}_t$ , reestimar  $\underline{\theta}$  via máxima verossimilhança, obtendo o SP-GAS(1).
4. Com base em  $\underline{\hat{\theta}}$  do passo anterior, calcular  $\hat{\varepsilon}_t$  e aplicar na função  $\hat{q}_t$ .
5. Dado  $\hat{q}_t$  do passo 4, reestimar  $\underline{\theta}$  via máxima verossimilhança.
6. Caso o ponto de máximo da verossimilhança seja maior do que o obtido pela iteração anterior, repetir desde o passo 4. Caso contrário, utilizar  $\underline{\hat{\theta}}$  da iteração anterior como estimativa final dos parâmetros do modelo.

**Figura1 – Esquema do processo iterativo para obter as estimativas dos parâmetros do modelo SP-GAS**

Conforme demonstrado por Blasques *et al.* (2016), este também será o modelo de menor AIC (*Akaike Information Criterion*, ou critério de informação de Akaike) dentre todos os modelos estimados pelo processo iterativo a partir do passo quatro. Vale lembrar que um AIC menor significa um modelo candidato melhor, conforme Akaike (1973).

No caso semiparamétrico a derivada da distribuição dos erros se torna mais complexa, os erros estão em função do estimador de densidade kernel que é uma soma de densidades. Blasques *et al.* (2016) utilizam uma t de student com 5 graus de liberdade, o que segundo eles estabiliza as simulações especialmente nos casos onde o processo gerador dos dados possui caudas pesadas. Sendo assim, supondo  $K \sim t(5)$ :

$$h[\hat{q}_t(x)] = \frac{\partial \ln[\hat{q}_t(x)]}{\partial x} = \frac{\sum_{t=1}^T (-v-1) * (x - \hat{\varepsilon}_t) * \left[1 + \frac{(\hat{\varepsilon}_t - x)^2}{vb_t^2}\right]^{-\left(\frac{v+1}{2}\right)-1}}{\sum_{t=1}^T \left[1 + \frac{(\hat{\varepsilon}_t - x)^2}{vb_t^2}\right]^{-\left(\frac{v+1}{2}\right)}}. \quad (8)$$

A utilização do SP-GAS neste trabalho será feita utilizando (8) como estimativa para a derivada do logaritmo dos erros.

### 3-Simulações de Monte Carlo

Para verificar a eficiência da metodologia nova (SP-GAS) frente às convencionais, inicialmente foi feito um estudo com dados simulados. Utilizando a mesma ideia de Blasques *et al.* (2016), foram criadas amostras onde o modelo GAS é o verdadeiro processo gerador (DGP). Sendo assim, definiram-se os parâmetros para simular as amostras:  $\mu = 0$ ,  $\omega = 2$ ,  $\alpha = 0.3$  e  $\beta = 0.9$ .

Três escolhas diferentes para a distribuição para  $q(\cdot)$  geraram três amostras distintas utilizadas para testar o modelo SP-GAS. As distribuições escolhidas foram: normal com média zero e desvio padrão unitário e duas t de student com três e cinco graus de liberdade. As amostras geradas possuem tamanho  $T=1500$  e o procedimento foi repetido 250 vezes para aumentar a confiabilidade.

Em seguida, quatro métodos de estimação dos parâmetros foram utilizados para cada uma das amostras. São eles, o estimador de máxima verossimilhança (MLE), o modelo GARCH supondo distribuição normal dos erros seguindo (4'), o modelo GAS apresentado nas equações (1), (2) e (3) com distribuição t de student com 3 graus de liberdade para o erro e por último o modelo GAS semiparamétrico, ilustrado pela figura 1, utilizando as estimativas do GAS paramétrico como ponto de partida. Para o estimador de densidade kernel, utilizou-se uma t de student com 5 graus de liberdade e a banda  $b_t$  foi escolhida através da função *bw.nrd* já implementada na linguagem de programação R. A escolha de todas as distribuições segue do artigo de Blasques *et al* (2016).

Para quantificar o desempenho de cada um desses modelos, foi calculada a raiz da média dos quadrados dos erros (RMSE) de predição dentro da amostra, também seguindo a metodologia de Blasques *et al.* (2016). Lembrando que quanto menor o RMSE do método melhor é seu ajuste global.

Outra maneira de se verificar a qualidade do ajuste de um modelo é o erro da estimativa dos parâmetros. Compararam-se os valores obtidos através do processo iterativo envolvendo a densidade kernel com o passo paramétrico inicial do modelo GAS e o MLE.

A tabela 1 apresenta os resultados do ajuste das estimativas dos parâmetros em função das estimativas do MLE. A tabela 2 apresenta os resultados da avaliação de resíduos em função das estimativas do GARCH.

**Tabela 1 – Raiz da média de erros ao quadrado das estimativas dos parâmetros**

DGP		$\mu$	$\omega$	$\alpha$	$\beta$
N(0,1)	MLE	1.00	1.00	1.00	1.00
	GAS	0.93	1.94	1.25	0.80
	SP-GAS	0.91	1.59	1.96	1.04
t(3)	MLE	1.00	1.00	1.00	1.00
	GAS	1.02	1.17	1.01	1.78
	SP-GAS	1.01	1.11	1.00	1.19
t(5)	MLE	1.00	1.00	1.00	1.00
	GAS	1.05	0.97	1.26	1.49
	SP-GAS	1.13	0.93	2.25	0.80

**Tabela 2 – Raiz da média de erros ao quadrado dos ajustes dentro da amostra**

DGP		RMSE
N(0,1)	MLE	0.86
	GARCH	1.00
	GAS	1.20
	SP-GAS	1.05
t(3)	MLE	0.91
	GARCH	1.00
	GAS	0.89
	SP-GAS	0.86
t(5)	MLE	0.44
	GARCH	1.00
	GAS	0.92
	SP-GAS	0.89

O modelo GARCH teve o melhor desempenho de previsão dentro da amostra entre os modelos quando a distribuição de  $q(\cdot)$  segue distribuição normal, ficando com RMSE muito próximo do estimador de máxima verossimilhança. Porém, como já se esperava, seu desempenho cai quando a verdadeira distribuição é uma t de student. Nestes casos as

estimativas do modelo GAS semiparamétrico foram as que mais se aproximaram do MLE, seguidas do modelo GAS. Quanto às estimativas dos parâmetros, o SP-GAS se aproxima mais dos parâmetros do DGP com exceção do parâmetro  $\alpha$ , onde seu desempenho é pior.

Notou-se que o SP-GAS(1) trouxe um ganho para as estimações dos parâmetros, conforme encontrado por Blasques *et al.* (2016).

## 4-Aplicação a dados empíricos

Em dados financeiros como retorno de ações é possível verificar várias sequências de valores extremos. Isso se deve ao fato de que esses dados refletem fatores exógenos ou muito voláteis, como a confiança de compradores e vendedores de ações. Em casos como estes é mais interessante modelar a volatilidade dos dados já que a média condicional pode ser descrita por um passeio aleatório. Modelos GAS e GARCH são alternativas que se ajustam muito bem para este tipo de desfecho.

Sendo assim, com o intuito de verificar o desempenho da modelagem GAS semiparamétrica em dados reais, utilizamos valores dos retornos da IBOVESPA no período de 3 de janeiro de 1995 até 19 de agosto de 2016, totalizando 5372 observações.

O valor de retorno de uma ação é calculado através da seguinte equação:

$$y_t = \ln(x_t) - \ln(x_{t-1}),$$

onde  $x_t$  é o valor da ação no tempo  $t$  e  $y_t$  é o retorno no tempo  $t$ .

Seguindo a metodologia proposta por Santos e Ziegelmann (2011), os dados foram modelados utilizando um AR(2), para então os resíduos serem modelados pelo GAS semiparamétrico.

Na figura 2 se encontra o gráfico dos resíduos do modelo AR(2) que serão usados na modelagem. Podem-se perceber alguns períodos de alta volatilidade representados pelos picos no gráfico dos resíduos do modelo, tornando estes dados ideais para a modelagem desejada.

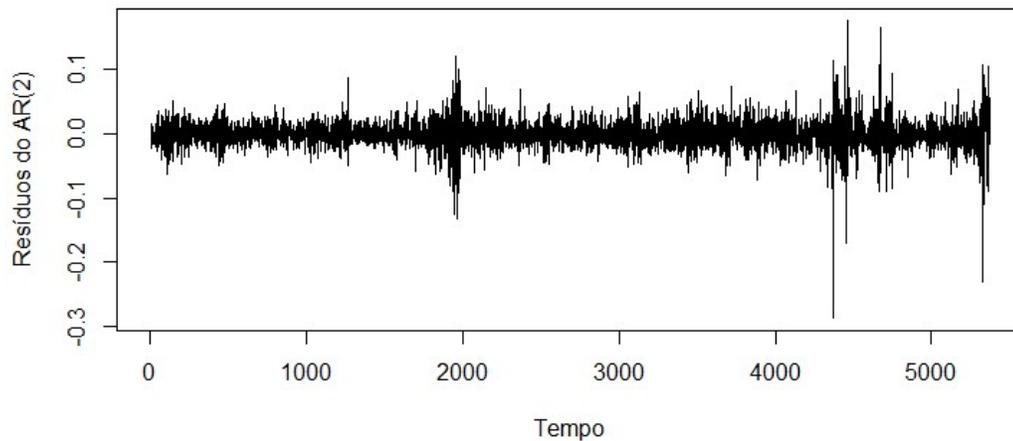
Outro aspecto importante é a distribuição dos resíduos, demonstrada na figura 3 através do histograma. Apesar de sua simetria, os resíduos padronizados não apresentam comportamento muito semelhante ao esperado de dados com distribuição normal padrão (linha sobreposta ao histograma), apontando que o GARCH gaussiano não tenha bom desempenho.

Sendo assim, foram ajustados aos resíduos do AR(2) um GARCH com distribuição normal de média 0 e variância 1 para os erros e o GAS semiparamétrico com distribuição t de student de 3 graus de liberdade para os erros e uma t de student com 5 graus de liberdade para o estimador de densidade kernel. Os valores dos RMSE da tabela 3 resultam das estimativas dos modelos ajustados para toda a amostra:

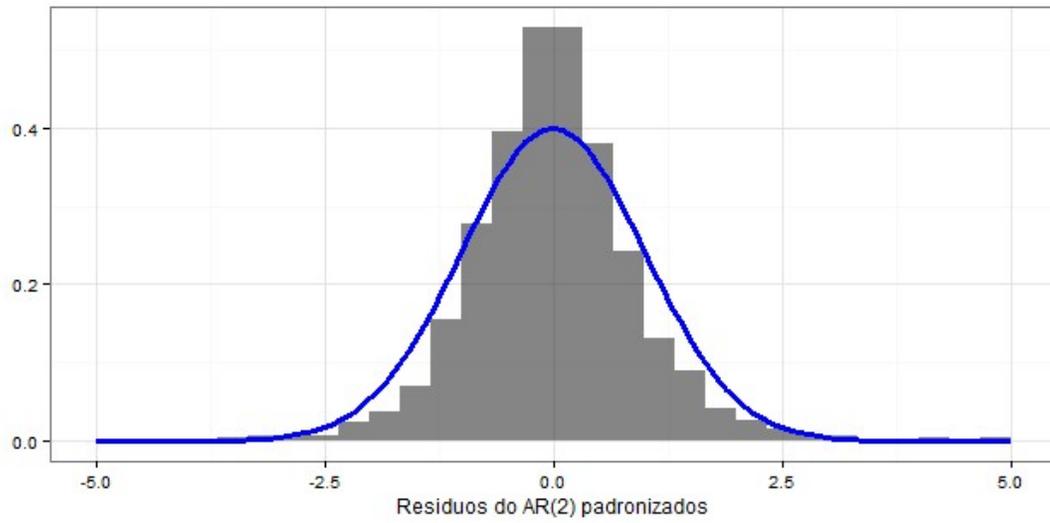
**Tabela 3 – Raiz da média de erros ao quadrado dentro da amostra dos modelos**

Modelo	RMSE
SP-GAS	0.81
GARCH	1

O SP-GAS apresentou resíduos menores, apontando para um melhor ajuste do modelo. Novamente, foi possível ver que o primeiro passo das iterações do modelo semiparamétrico já obtém estimativas muito próximas das obtidas pelo algoritmo completo. Isso torna mais evidente que é possível utilizá-lo para obter resultados satisfatórios caso realizar todo o procedimento seja muito custoso.



**Figura 2 – Resíduos do modelo AR(2)**



**Figura 3 – Histograma dos resíduos padronizados do modelo AR(2)**

## 5-Considerações finais

Na maioria dos casos testados a nova abordagem apresentou melhor ajuste dentro da amostra se comparada ao modelo GARCH gaussiano. Testar o ajuste do SP-GAS contra um modelo GARCH que utilize outras distribuições para os erros é uma sugestão pertinente.

Foi possível constatar que a primeira iteração do processo já obtém estimativas muito próximas dos parâmetros do modelo, ou no caso empírico próximas da estimativa final. Uma vez que o processo computacional para que se obtenham as estimativas do SP-GAS(i) é dispendioso, o primeiro passo pode ser utilizado como alternativa para reduzir este problema.

Para trabalhos futuros, seria possível aperfeiçoar a função que realiza as iterações para acelerar o processo, facilitando o uso desta técnica e consequentemente diminuindo seu *trade-off*.

## Referências Bibliográficas

AKAIKE H.. Information theory and an extension of the maximum likelihood principle. *2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR, September 2-8, 1971*, Budapeste: Akadémiai Kiadó, p. 267–281, 1973.

BLASQUES, F.;JIANGYU, J.; LUCAS, A..Semiparametric score driven volatility models. *Computational Statistics and Data Analysis*. n. 100, p. 58-69, 2016.

BOLLERSLEV, T.. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*. Vol 31, p. 307-327, 1986.

CREAL, D.; KOOPMAN, S. J.; LUCAS, A.. Generalized Autoregressive Score Models with Applications. *Journal of Applied Econometrics*. n. 28, p. 777-795, 2013.

CRYER, J.D.; CHAN, K. S.. *Time Series Analysis: with applications in R*. Segunda edição. New York: Springer, 2008. 491 p.

ENGLE, R. F.. Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation. *Econometrica*. Vol 50, p. 987-1008, 1982.

FRANSES, P. H.. *Time series models for business and economic forecasting*. Segunda edição. Cambridge: Cambridge Press, 1999. 280 p.

R DOCUMENTATION. Bandwidth Selectors for Kernel Density Estimation. Disponível em: < <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/bandwidth.html>>. Data de acesso: 20/10/2016.

SANTOS, D. G. dos; ZIEGELMANN, F. A.. Estimação e previsão de Volatilidade em períodos de crise: Um estudo comparando modelos GARCH e modelos aditivos semiparamétricos. *Revista Brasileira de Finanças*. Rio de Janeiro, vol. 10, No. 1, p. 49-70.

## APENDICE A

```
veross3 <- function(x, y, f0, etc, bt=NA, distq){
  #funcao que calcula as f's do modelo
  n <- length(y)
  mi <- x[1]
  omega <- x[2]
  a <- x[3]
  b <- x[4]
  parq <- as.numeric(distq[2:length(distq)])
  if(is.na(bt))
    bt=bw.nrd(y)

  f <- s <- numeric(n+1)
  f[1] <- f0

  for(i in 1:(n)){
    aux <- switch(distq[1],
      t={
        x.aux <- exp(-f[i]/2)*(y[i]-mi)
        r1 <- x.aux*(1-parq)
        r2 <- parq+x.aux^2

        r3 <- r1/r2
      },
      normal= (parq[1]-(exp(-f[i]/2)*(y[i]-mi)))/parq[2],
      kt={
        x.aux <- exp(-f[i]/2)*(y[i]-mi)
        r1 <- -parq-1
        r2 <- etc - x.aux
        r3 <- 1 + r2^2/(parq*bt^2)
        r4 <- r1*(-r2)/(parq*bt^2)* r3^(1/2*(-parq-1)-1)
        r5 <- r3^(-parq+1)/2

        r6 <- sum(r4)/sum(r5)
      }
    )

    s <- -0.5*(exp(-f[i]/2)*(y[i]-mi)*aux - 1)
    f[i+1] <- omega*(1-b) + a*s + b*f[i]
  }
  out <- list(f=f,s=s)
  return(out)
}
```

```
veross2 <- function(x, y, f0, etc, bt, distq){
  #funcao que calcula as f's do modelo
  n <- length(y)
  mi <- x[1]
  omega <- x[2]
  a <- x[3]
  b <- x[4]
  parq <- as.numeric(distq[2:length(distq)])

  f <- numeric(n)
  f[1] <- f0

  s <- numeric(1)

  for(i in 1:(n-1)){
    aux <- switch(distq[1],
      t={
        x.aux <- exp(-f[i]/2)*(y[i]-mi)
        r1 <- x.aux*(1-parq)
        r2 <- parq+x.aux^2
```

```

        r3 <- r1/r2
      },
      normal= (parq[1]-(exp(-f[i]/2)*(y[i]-mi)))/parq[2],
      kt={
        x.aux <- exp(-f[i]/2)*(y[i]-mi)
        r1 <- -parq-1
        r2 <- etc - x.aux
        r3 <- 1 + r2^2/(parq*bt^2)
        r4 <- r1*(-r2)/(parq*bt^2)* r3^(1/2*(-parq-1)-1)
        r5 <- r3^(-parq+1)/2

        r6 <- sum(r4)/sum(r5)
      }
    )

    s <- -0.5*(exp(-f[i]/2)*(y[i]-mi)*aux - 1)
    f[i+1] <- omega*(1-b) + a*s + b*f[i]
  }
  out <- f
  return(out)
}

veross <- function(x, y, f0, etc=NA, bt=NA, distq=c("normal",0,1)){
  if(any(is.na(y)==T)){
    warning("Missing observations are going to be removed!")
  }
  y <- na.omit(y)
  mi <- x[1]
  n <- length(y)
  parq <- as.numeric(distq[2:length(distq)])
  f <- veross2(x,y,f0,etc,bt,distq)

  q <- switch(distq[1],
    t = dt(exp(-f/2)*(y-mi),parq),
    normal = dnorm(exp(-f/2)*(y-mi), parq[1], parq[2]),
    kt = {
      bt <- bw.nrd(y)
      dif <- numeric(n)
      for(i in 1:n){
        dif[i] <- sum((etc - (exp(-f[i]/2)*(y[i]-mi)))/bt)
      }
      1/(n*bt)*sum(dt(dif, parq))
    }
  )

  ver <- -1/n*sum(-0.5*f + log(q))
  return(ver)
}

##### SPGAS #####
spgas <- function(data, x, f0=0, distqS=c("normal",0,1), start.par=NA,
  distqK=c("t", 5), bt=0.5){
  n <- length(data)
  distqK[1] <- paste0("k",distqK[1])

  if(is.na(start.par[1])){
    start <- optim(x, veross, y=data, f0=f0, distq=distqS)
  }else{
    start <- data.frame(par=as.numeric(start.par),value=NA,convergence=NA)
  }
  fc <- veross2(x=start$par, y=data, f0=f0, etc=NA, bt=NA, distq=distqS)
  etc <- exp(-fc/2)*(data-start$par[1])

  print(paste("Iteration n", 1))
}

```

```

max1 <- optim(par=start$par, fn=veross, y=data, f0=f0, etc=etc, distq=distqK,
bt=bt)
max2 <- max1

n_it <- 1 #numero de iteracoes
max <- data.frame(it=0,mi=start$par[1],omega=start$par[2],alpha=start$par[3],
beta=start$par[4], max=start$value[1],
conv=start$convergence[1])#guarda as maximizacoes
max <- rbind(max,c(1,max1$par,max1$value,max1$convergence))

comp <- 0
while(comp==0){
n_it <- n_it+1
print(paste("Iteration n", n_it))
test <- max
fc <- veross2(x=max1$par, y=data, f0=f0, etc=etc, bt=bt, distq=distqK)
etc <- exp(-fc/2)*(data-max1$par[1])
max2 <- optim(par=max1$par, fn=veross, y=data, f0=(mean(fc[1:2])), etc=etc,
bt=bt, distq=distqK)

if(max1$value>max2$value){
comp <- 1
etcFinal <- etc
} else{
max1$value <- max2$value
}
max <- rbind(max,c(n_it, max2$par, max2$value, max2$convergence))
}
return(list(etcCalc=etcFinal,max=max))
}

spgasPredict1 <- function(model,y,f0,distq=c("kt",5),bt=NA){
par <- as.numeric(model$max[nrow(model$max)-1,2:5])
etc <- model$etcCalc
aux <- veross3(x=par,y=y, f0=f0, etc=etc, bt=bt, distq=distq)
return(aux)
}

spgasPredict2 <- function(model,y,f0,distq=c("normal",5),bt=NA){
par <- as.numeric(model$par)
# etc <- model$etcCalc
aux <- veross3(x=par,y=y, f0=f0, etc=NA, bt=bt, distq=distq)
return(aux)
}

garchPredict <- function(model,y,f0){
par <- as.numeric(model@fit$matcoef[,1])
n <- length(y)
f <- numeric(n)
f[1] <- f0

for(i in 2:n){
f[i] <- par[2]*(1-par[3]-par[4]) + par[3]*(y[i-1]-par[1]) + par[4]*f[i-1]
}
return(f)
}

```

```

#Analise empirica
data <- read.csv("table.csv")
data$retorno <- NA

for(i in 2:nrow(data)){
  data$retorno[i] = log(data$Adj.Close[i]) - log(data$Adj.Close[i-1])
}

y_real <- data[-1,c(1,8)]

ar2 <- arima(y_real$retorno, order=c(2,0,0))
plot(ar2$residuals, ylab="Residuos do AR(2)")
hist(ar2$residuals, xlab="Histograma dos residuos do AR(2)", ylab="Frequencia",
main="")

ym_real <- (ar2$residuals - mean(ar2$residuals))/sd(ar2$residuals)

plot(ym_real, ylab="Residuos do AR(2) padronizados", xlab="Tempo")
dataFramePlot <- data.frame(dados=ym_real)
plot1 <- ggplot(dataFramePlot) + geom_histogram(aes(y=..density.., x=dados),
alpha=0.6) +
  stat_function(fun = dnorm, args=list(mean=0, sd=1), colour = "blue", lwd=1.2) +
  stat_function(fun = dnorm, args=list(mean=0, sd=0.8), colour = "red", lwd=1.2) +
  theme_bw() + xlim(-5,5) + ylab("") + xlab("Residuos do AR(2) padronizados")
show(plot1)
shapiro.test(dataFramePlot$dados[1:5000])
ks.test(x=dataFramePlot$dados, y=rnorm(5000, mean=0, sd=1))
sd(dataFramePlot$dados)

model1 <- garchFit(ym_real~garch(1,1),trace=FALSE)
model2 <- spgas(ym_real,c(0,2,.5,.5))

#### Analise simulada ####
if(!("fGarch") %in% installed.packages())install.packages("fGarch")
library(fGarch)

B <- 250 #Numero de repeticoes do estudo
N <- 1500 #Tamanho da amostra simulada
val <- 0 #Para validacao do modelo
burn <- 100 #burn inicial

N <- N + val + burn

#Parametros da funcao
mi <- 0
omega <- 2
a <- 0.3
b <- 0.9
f0 <- 0
parReais <- c(mi,omega,a,b)

# #### q <- N(0,1) ####
# sqErrPrev <- data.frame(m1e=NA,garch=NA,gas=NA,SPgas=NA)
# sqErrPar <- list(mi=sqErrPrev[,c(1,3,4)],
#                 omega=sqErrPrev[,c(1,3,4)],
#                 alpha=sqErrPrev[,c(1,3,4)],
#                 beta=sqErrPrev[,c(1,3,4)]) # recebera os parametros
#
#
# for(j in 1:B){
#   print(paste("it n", j))
#   ##Simulando dados
#   f_n <- numeric(N)

```

```

# y_n <- numeric(N)
# e_n <- rnorm(N)
# #Padronizando e centralizando
# e_n <- (e_n-mean(e_n))/sd(e_n)
#
# #Definindo as observacoes iniciais
# f_n[1] <- f0
# y_n[1] <- mi + exp(f_n[1]/2)*e_n[1]
#
# for(i in 1:(N-1)){
#   s_n <- 0.5*(exp(-f_n[i])*((y_n[i]-mi)^2) + 1)
#   f_n[i+1] <- omega*(1-b) + a*s_n + b*f_n[i]
#   y_n[i+1] <- mi + exp(f_n[i]/2)*e_n[i]
# }
#
# ##Separar as amostras p/ validacao
# y_n <- y_n[-(1:burn)]
# if(val!=0)
#   y2_n <- y_n[(N-val-burn+1):(N-burn)]
# y_n <- y_n[1:(N-val-burn)]
#
# f_n <- f_n[-(1:burn)]
# if(val!=0)
#   y2_n <- f_n[(N-val-burn+1):(N-burn)]
# f_n <- f_n[1:(N-val-burn)]
#
# ts.plot(y_n)
# ts.plot(f_n)
#
# ## Estimacao dos parametros
# mle_n <- optim(c(0,3,0.25,0.7), veross, y=y_n, f0=f0)
# garch_n <- garchFit(data=y_n, cond.dist="norm", trace=F)
# gast_n <- optim(c(0,3,0.25,0.7), veross, y=y_n, f0=0, distq=c("t",5))
# spgas_n <- spgas(data=y_n, x=c(0,2,0.3,0.9), f0=0,
#                   distq=c("t", 5), bt=0.5)
#
# ##Comparacao dos modelos
# mle_n_pred <- spgasPredict2(mle_n,y_n,f0,distq=c("normal",0,1))
# gast_n_pred <- spgasPredict2(gast_n,y_n,f0,distq=c("t",5))
# garch_n_pred <- garchPredict(garch_n,y_n,f0)
# spgas_n_pred <- spgasPredict1(spgas_n,y_n,f0,distq=c("kt",5))
#
# sqErrPrev <- rbind(sqErrPrev,
#                   c(
#                     mle=sum(((mle_n_pred$f[-length(mle_n_pred$f)])-f_n)^2),
#                     garch=sum(((garch_n_pred)-f_n)^2),
#                     gast=sum(((gast_n_pred$f[-length(gast_n_pred$f)])-f_n)^2),
#                     spgas=sum(((spgas_n_pred$f[-length(spgas_n_pred$f)])-
# f_n)^2)
#                   )
#                   )
#
# sqErrPar$mi <- rbind(sqErrPar$mi,
# c(mle_n$par[1],gast_n$par[1],spgas_n$max[nrow(spgas_n$max)-1,2]))
# sqErrPar$omega <- rbind(sqErrPar$omega,
# c(mle_n$par[2],gast_n$par[2],spgas_n$max[nrow(spgas_n$max)-1,3]))
# sqErrPar$alpha <- rbind(sqErrPar$alpha,
# c(mle_n$par[3],gast_n$par[3],spgas_n$max[nrow(spgas_n$max)-1,4]))
# sqErrPar$beta <- rbind(sqErrPar$beta,
# c(mle_n$par[4],gast_n$par[4],spgas_n$max[nrow(spgas_n$max)-1,5]))
# }
#
# RMSEPrev <- sqrt(sapply(sqErrPrev,mean,na.rm=T))
# RMSEPar <-
# data.frame(mi=rep(NA,3),omega=rep(NA,3),alpha=rep(NA,3),beta=rep(NA,3),
#           row.names=c("MLE","GAS","SP-GAS"))
# for(i in 1:4){
#   data <- (sqErrPar[[i]][-1,] - parReais[i])^2

```

```

# for(j in 1:3){
#   RMSEPar[j,i] <- sqrt(mean(data[,j]))
# }
# }

# #### q <- t(3) ####
# sqErrPrev <- data.frame(mle=NA,garch=NA,gas=NA,SPgas=NA)
# sqErrPar <- list(mi=sqErrPrev[,c(1,3,4)],
#                 omega=sqErrPrev[,c(1,3,4)],
#                 alpha=sqErrPrev[,c(1,3,4)],
#                 beta=sqErrPrev[,c(1,3,4)]) # recebera os parametros
#
# for(i in 1:B){
#   print(paste("it n", i))
#   ##Simulando dados
#   f_t3 <- numeric(N)
#   y_t3 <- numeric(N)
#   e_t3 <- rt(N,3)
#   #Padronizando e centralizando
#   e_t3 <- (e_t3-mean(e_t3))/sd(e_t3)
#
#   #Definindo as observacoes iniciais
#   f_t3[1] <- f0
#   y_t3[1] <- mi + exp(f_t3[1]/2)*e_t3[1]
#
#   for(i in 1:(N-1)){
#     x.aux <- exp(-f_t3[i]/2)*(y_t3[i]-mi)
#     s_t3 <- -0.5*((-2*x.aux/(3+x.aux^2)) * x.aux - 1)
#     f_t3[i+1] <- omega*(1-b) + a*s_t3 + b*f_t3[i]
#     y_t3[i+1] <- mi + exp(f_t3[i]/2)*e_t3[i]
#   }
#
#   ##Separar as amostras p/ validacao
#   y_t3 <- y_t3[-(1:burn)]
#   if(val!=0)
#     y2_t3 <- y_t3[(N-val-burn+1):(N-burn)]
#   y_t3 <- y_t3[1:(N-val-burn)]
#
#   f_t3 <- f_t3[-(1:burn)]
#   if(val!=0)
#     y2_t3 <- f_t3[(N-val-burn+1):(N-burn)]
#   f_t3 <- f_t3[1:(N-val-burn)]
#
#   ts.plot(y_t3)
#   ts.plot(f_t3)
#
#   ## Estimacao dos parametros
#   mle_t3 <- optim(c(0,3,0.25,0.7), veross, y=y_t3, f0=f0, distq=c("t",3))
#   garch_t3 <- garchFit(data=y_t3, cond.dist="norm", trace=F)
#   gast_t3 <- optim(c(0,3,0.25,0.7), veross, y=y_t3, f0=0, distq=c("t",5))
#   spgas_t3 <- spgas(data=y_t3, x=c(0,2,0.3,0.9), f0=0,
#                     distq=c("t", 5), bt=0.5)
#
#   ##Comparacao dos modelos
#   mle_t3_pred <- spgasPredict2(mle_t3,y_t3,f0,distq=c("normal",0,1))
#   gast_t3_pred <- spgasPredict2(gast_t3,y_t3,f0,distq=c("t",5))
#   garch_t3_pred <- garchPredict(garch_t3,y_t3,f0)
#   spgas_t3_pred <- spgasPredict1(spgas_t3,y_t3,f0,distq=c("kt",5))
#
#   sqErrPrev <- rbind(sqErrPrev,
#                     c(
#                       mle=sum(((mle_t3_pred$f[-length(mle_t3_pred$f)])-f_t3)^2),
#                       garch=sum(((garch_t3_pred)-f_t3)^2),
#                       gast=sum(((gast_t3_pred$f[-length(gast_t3_pred$f)])-
#f_t3)^2),

```

```

#           spgas=sum(((spgas_t3_pred$f[-length(spgas_t3_pred$f)])-
f_t3)^2)
#           )
#   )
#   sqErrPar$mi <- rbind(sqErrPar$mi,
c(mle_t3$par[1],gast_t3$par[1],spgas_t3$max[nrow(spgas_t3$max)-1,2]))
#   sqErrPar$omega <- rbind(sqErrPar$omega,
c(mle_t3$par[2],gast_t3$par[2],spgas_t3$max[nrow(spgas_t3$max)-1,3]))
#   sqErrPar$alpha <- rbind(sqErrPar$alpha,
c(mle_t3$par[3],gast_t3$par[3],spgas_t3$max[nrow(spgas_t3$max)-1,4]))
#   sqErrPar$beta <- rbind(sqErrPar$beta,
c(mle_t3$par[4],gast_t3$par[4],spgas_t3$max[nrow(spgas_t3$max)-1,5]))
# }
#
# RMSEPrev <- sqrt(sapply(sqErrPrev,mean,na.rm=T))
# RMSEPar <-
data.frame(mi=rep(NA,3),omega=rep(NA,3),alpha=rep(NA,3),beta=rep(NA,3),
#           row.names=c("MLE","GAS","SP-GAS"))
# for(i in 1:4){
#   data <- (sqErrPar[[i]][-1,] - parReais[i])^2
#   for(j in 1:3){
#     RMSEPar[j,i] <- sqrt(mean(data[,j]))
#   }
# }

#### q <- t(5) ####
sqErrPrev <- data.frame(mle=NA,garch=NA,gas=NA,SPgas=NA) #recebera a media dos
residuos ao quadrado
sqErrPar <- list(mi=sqErrPrev[,c(1,3,4)],
               omega=sqErrPrev[,c(1,3,4)],
               alpha=sqErrPrev[,c(1,3,4)],
               beta=sqErrPrev[,c(1,3,4)]) # recebera os parametros

for(i in 1:B){
  print(paste("it n", i))
  ##Simulando dados
  f_t5 <- numeric(N)
  y_t5 <- numeric(N)
  e_t5 <- rnorm(N)
  #Padronizando e centralizando
  e_t5 <- (e_t5-mean(e_t5))/sd(e_t5)

  #Definindo as observacoes iniciais
  f_t5[1] <- f0
  y_t5[1] <- mi + exp(f_t5[1]/2)*e_t5[1]

  for(i in 1:(N-1)){
    x.aux <- exp(-f_t5[i]/2)*(y_t5[i]-mi)
    s_t5 <- -0.5*((-4*x.aux/(5+x.aux^2)) * x.aux - 1)
    f_t5[i+1] <- omega*(1-b) + a*s_t5 + b*f_t5[i]
    y_t5[i+1] <- mi + exp(f_t5[i]/2)*e_t5[i]
  }

  ##Separar as amostras p/ validacao
  y_t5 <- y_t5[-(1:burn)]
  if(val!=0)
    y2_t5 <- y_t5[(N-val-burn+1):(N-burn)]
  y_t5 <- y_t5[1:(N-val-burn)]

  f_t5 <- f_t5[-(1:burn)]
  if(val!=0)
    y2_t5 <- f_t5[(N-val-burn+1):(N-burn)]
  f_t5 <- f_t5[1:(N-val-burn)]

  ts.plot(y_t5)
  ts.plot(f_t5)

```

```

## Estimacao dos parametros
mle_t5 <- optim(c(0,3,0.25,0.7), veross, y=y_t5, f0=f0)
garch_t5 <- garchFit(data=y_t5, cond.dist="norm", trace=F)
gast_t5 <- optim(c(0,3,0.25,0.7), veross, y=y_t5, f0=0, distq=c("t",5))
spgas_t5 <- spgas(data=y_t5, x=c(0,2,0.3,0.9), f0=0,
start.par=as.numeric(gast_t5$par),
distqK=c("t", 5), bt=0.5)

##Comparacao dos modelos
mle_t5_pred <- spgasPredict2(mle_t5,y_t5,f0,distq=c("normal",0,1))
gast_t5_pred <- spgasPredict2(gast_t5,y_t5,f0,distq=c("t",5))
garch_t5_pred <- garchPredict(garch_t5,y_t5,f0)
spgas_t5_pred <- spgasPredict1(spgas_t5,y_t5,f0,distq=c("kt",5))

sqErrPrev <- rbind(sqErrPrev,
c(
mle=sum(((mle_t5_pred$f[-length(mle_t5_pred$f)])-f_t5)^2),
garch=sum(((garch_t5_pred)-f_t5)^2),
gast=sum(((gast_t5_pred$f[-length(gast_t5_pred$f)])-
f_t5)^2),
spgas=sum(((spgas_t5_pred$f[-length(spgas_t5_pred$f)])-
f_t5)^2)
)
)
sqErrPar$mi <- rbind(sqErrPar$mi,
c(mle_t5$par[1],gast_t5$par[1],spgas_t5$max[nrow(spgas_t5$max)-1,2]))
sqErrPar$omega <- rbind(sqErrPar$omega,
c(mle_t5$par[2],gast_t5$par[2],spgas_t5$max[nrow(spgas_t5$max)-1,3]))
sqErrPar$alpha <- rbind(sqErrPar$alpha,
c(mle_t5$par[3],gast_t5$par[3],spgas_t5$max[nrow(spgas_t5$max)-1,4]))
sqErrPar$beta <- rbind(sqErrPar$beta,
c(mle_t5$par[4],gast_t5$par[4],spgas_t5$max[nrow(spgas_t5$max)-1,5]))
}

RMSEPrev <- sqrt(sapply(sqErrPrev,mean,na.rm=T))
RMSEPar <- data.frame(mi=rep(NA,3),omega=rep(NA,3),alpha=rep(NA,3),beta=rep(NA,3),
row.names=c("MLE","GAS","SP-GAS"))

for(i in 1:4){
data <- (sqErrPar[[i]][-1,] - parReais[i])^2
for(j in 1:3){
RMSEPar[j,i] <- sqrt(mean(data[,j]))
}
}
}

```