

Relatório do Trabalho de Estatística Bayesiana I

Mistura de Normais com Variância Contaminada

Caio Balieiro
Taiguara Melo Tupinambás
Walmir dos Reis Miranda Filho

Prof. Dani Gamerman
Prof^a Rosangela Helena Loschi

Programa de Pós-Graduação em Estatística
Instituto de Ciências Exatas
Universidade Federal de Minas Gerais

Belo Horizonte, 2 de dezembro de 2019

1 Introdução

O presente trabalho tem como objetivo obter, dada uma densidade *a posteriori* conjunta dos parâmetros de um modelo probabilístico para uma amostra previamente observada, as densidades *a posteriori* marginais de cada parâmetro, bem como as estatísticas de média; variância; assimetria e curtose associadas, a partir da implementação de três métodos numéricos, a saber: (i) integração via quadratura de Riemann; (ii) reamostragem por importância sequencial (em inglês, *Sequential Importance Resampling*, ou SIR); e (iii) integração via Monte Carlo em cadeias de Markov (em inglês, *Markov Chain Monte Carlo*, ou MCMC) com inovações dadas pelo algoritmo de Metropolis-Hastings (MH).

Para o modelo que gerou a amostra observada, será pressuposto que sua especificação é dada por uma mistura finita de normais. Sejam X_1, \dots, X_n amostras aleatórias independentes, condicionalmente a um vetor de parâmetros $\theta = (\mu, \sigma^2, \nu)$, e identicamente distribuídas com função densidade dada por

$$f(x|\mu, \sigma^2, \nu) = \nu\phi(x|\mu, 100\sigma^2) + (1 - \nu)\phi(x|\mu, \sigma^2), \quad x \in \mathbb{R}, \quad (1)$$

onde $\phi(x|\mu, \sigma^2) = (2\pi\sigma^2)^{-1} \exp[-(x - \mu)^2/(2\sigma^2)]$ denota a função densidade da distribuição normal com média μ e variância σ^2 avaliada no ponto x . Para o suporte de cada parâmetro, tem-se que $\mu \in \mathbb{R}$, $\sigma^2 \in \mathbb{R}_+$ e $\nu \in (0, 1)$.

A mistura finita descrita em (1) possui duas componentes normais que diferem nas respectivas variâncias: para a primeira componente, o valor é 100 vezes o da segunda. Desta forma, tem-se uma mistura finita de normais com variância *contaminada*.

Para os parâmetros μ , σ^2 e ν , será pressuposto que cada um segue uma distribuição *a priori* predeterminada: $\mu|\sigma^2 \sim N(m, V\sigma^2)$, onde $N(\cdot)$ denota a distribuição normal com média $m \in \mathbb{R}$ e variância $V\sigma^2$, $V > 0$; $\sigma^2 \sim GI(a, d)$, onde $GI(\cdot)$ denota a distribuição gama inversa com parâmetros de forma $a > 0$ e de taxa $d > 0$ (inverso da escala); e $\nu \sim U(0, 1)$, a distribuição uniforme contínua padrão.

Para gerar uma amostra aleatória do modelo em (1), foi utilizada uma representação hierárquica (Lachos *et al.*, 2013)[1] tal que

$$X_i|\mu, \sigma^2, U_i = u_i \sim N(\mu, \sigma^2 u_i^{-1}), \quad U_i|\mu \sim p_d(1, 100) : P(U_i = 100) = \nu. \quad (2)$$

onde $p_d(a, b)$ denota uma função de probabilidade (discreta) que atribui massa probabilística apenas aos pontos a e b .

A partir do produto entre a função de verossimilhança $f(\mathbf{x}|\mu, \sigma^2, \nu)$ para uma amostra $\mathbf{x} = (x_1, \dots, x_n)$ gerada através da representação hierárquica em (2) e das densidades *a priori* para cada parâmetro do modelo, obtém-se o núcleo (em inglês, *kernel*) da densidade *a posteriori* conjunta $p(\mu, \sigma^2, \nu|\mathbf{x})$. Como a expressão da densidade *a posteriori* possui uma constante de proporcionalidade $f(x_1, \dots, x_n) = f(\mathbf{x})$ não facilmente calculável, mas que não depende dos parâmetros, pode-se usar o núcleo daquela para inferir sobre cada um dos parâmetros. No modelo de mistura finita de duas componentes normais com variância contaminada, temos que

$$\begin{aligned} p(\mu, \sigma^2, \nu|\mathbf{x}) &= \frac{f(\mathbf{x}|\mu, \sigma^2, \nu) \times p(\mu, \sigma^2, \nu)}{f(\mathbf{x})} \propto \prod_{i=1}^n f(x_i) \times p(\mu|\sigma^2) \times p(\sigma^2) \times p(\nu) \\ &\propto \prod_{i=1}^n [\nu\phi(x_i|\mu, 100\sigma^2) + (1 - \nu)\phi(x_i|\mu, \sigma^2)] \times \\ &\times \phi(\mu|m, V\sigma^2) \times \frac{d^a}{\Gamma(a)} \left(\frac{1}{\sigma^2}\right)^{a+1} \exp\left(-\frac{d}{\sigma^2}\right) \\ &\propto \left(\frac{1}{\sigma^2}\right)^{(n+1)/2+a+1} \exp\left\{-\frac{[(\mu - m)^2/(2V) + d]}{\sigma^2}\right\} \times A(\mathbf{x}|\mu, \sigma^2, \nu), \end{aligned} \quad (3)$$

onde

$$A(\mathbf{x}|\mu, \sigma^2, \nu) = \prod_{i=1}^n \left\{ \frac{\nu}{10} \exp \left[-\frac{(x_i - \mu)^2}{200\sigma^2} \right] + (1 - \nu) \exp \left[-\frac{(x_i - \mu)^2}{2\sigma^2} \right] \right\}.$$

Evidentemente, trabalhar diretamente com o lado direito em (3) não é computacionalmente agradável, uma vez que esta é um produto de várias quantidades sempre menores do que 1, muitas delas próximas de 0, o que pode levar a problemas de aproximação numérica. Desta forma, sempre que for necessário calcular o núcleo da *posteriori*, dada uma amostra, inicialmente será tomada a soma do logaritmo de todas as quantidades que compõem este produto e só depois esta mesma soma será exponenciada. Dependendo do quão negativa for a soma (o logaritmo de termos entre 0 e 1 é sempre negativo), quando exponenciada ela pode ou não ser numericamente igual a 0.

Logo, na geração da amostra \mathbf{x} deve-se escolher valores de μ , σ^2 e ν tais que o núcleo de $p(\mu, \sigma^2, \nu|\mathbf{x})$ não seja numericamente nulo para os verdadeiros valores dos parâmetros. Considerando o uso da linguagem de programação *R* (R Core Team, 2019)[2], esta escolha não é difícil. Na versão 3.5.3, o menor valor inteiro que pode ser tomado como argumento da função `exp()` do *R* é -745 . Desta forma, basta tomar μ , σ^2 e ν tais que o logaritmo do lado direito em (3) seja maior do que -745 .

Para o presente trabalho, foram considerados uma amostra de tamanho $n = 500$ (Figura ??) da mistura finita de normais com variância contaminada parametrizada de tal forma que $\mu = 11$; $\sigma^2 = 0.64$ e $\nu = 0.2$, com hiperparâmetros $m = 11$; $V = 1$; $a = 7$ e $d = 4$ nas distribuições *a priori*. Fixada uma semente aleatória, para a amostra gerada a partir dos valores citados, o logaritmo do núcleo da densidade *a posteriori* é igual a -518.9061 , cuja exponencial é aproximadamente igual a 4.385×10^{-226} .

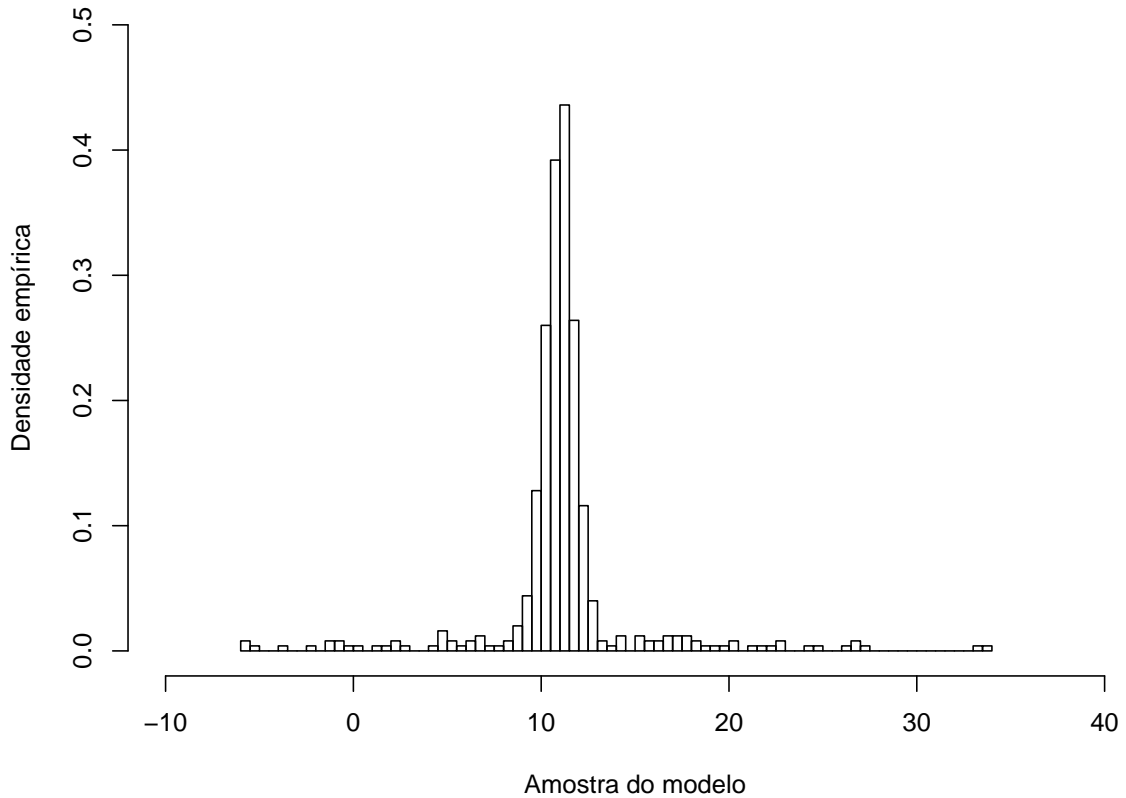
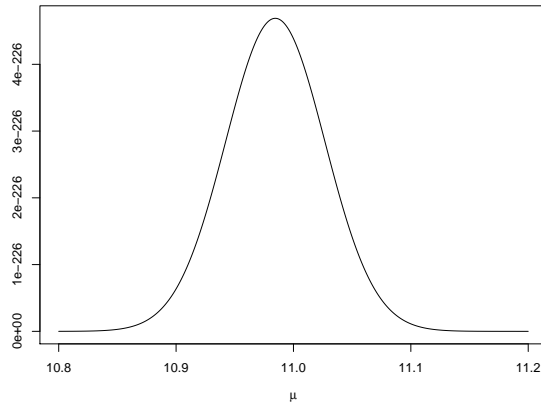


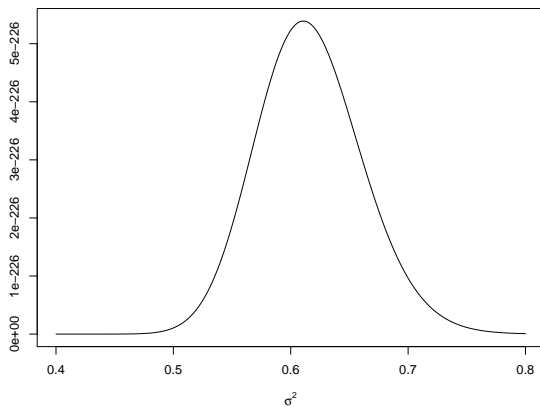
Figura 1: Histograma da amostra gerada do modelo

Como não se tem uma expressão fechada para $p(\mu, \sigma^2, \nu | \mathbf{x})$, mas apenas de seu núcleo, para obter as densidades *a posteriori* marginais de μ , σ^2 e ν dado \mathbf{x} , bem como as estatísticas associadas a cada uma delas, é necessário aproximá-las por algum método numérico. Na seção 2, é apresentado o primeiro dos três métodos considerados: a integração via quadratura de Riemann. Por sua vez, na seção 3 é descrito o método SIR. Por fim, na seção 4, é retratada a integração via MCMC com inovações MH. Todos os métodos serão testados e comparados no contexto do modelo exposto acima. Na seção 5, são resumidos os resultados obtidos para os três métodos, bem como é feita uma avaliação comparativa da qualidade de sua convergência.

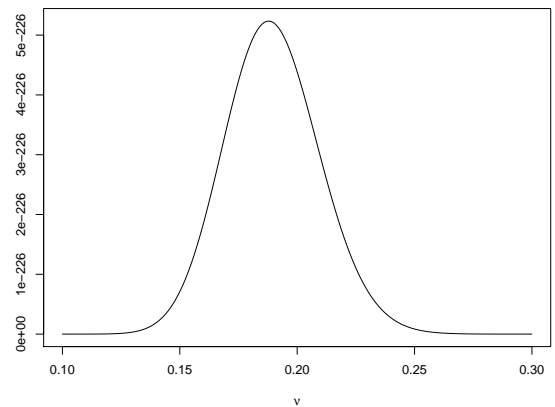
Antes de implementar cada um dos métodos numéricos, foram feitos três gráficos (Figuras 2a a 2c) do núcleo de $p(\mu, \sigma^2, \nu | \mathbf{x})$ quando se varia um dos três parâmetros e os outros dois são mantidos fixos nos respectivos valores verdadeiros pressupostos para o modelo que gerou a amostra. Tais gráficos permitem dizer, para cada parâmetro, quais são os intervalos do suporte correspondente que concentram praticamente toda a massa probabilística da densidade *a posteriori* marginal associada, ainda que as densidades calculadas sejam válidas apenas quando os outros dois parâmetros são fixados em valores arbitrários. Esta informação será útil para definir os intervalos de integração na quadratura de Riemann e a matriz de covariância da distribuição proposta nos métodos SIR e MCMC-MH para aproximação das densidades *a posteriori* marginais verdadeiras.



(a) $I_\mu = (10.85, 11.13)$, dados $\sigma^2 = 0.64, \nu = 0.2$



(b) $I_{\sigma^2} = (0.48, 0.78)$, dados $\mu = 11, \nu = 0.2$



(c) $I_\nu = (0.13, 0.26)$ dados $\mu = 11, \sigma^2 = 0.64$

Figura 2: Intervalos de massa probabilística para cada parâmetro (variável aleatória) do núcleo de $p(\mu, \sigma^2, \nu | \mathbf{x})$

2 Quadratura de Riemann

Em problemas multidimensionais, a densidade marginal correspondente a cada variável aleatória para a qual se deseja fazer inferência é obtida integrando a densidade conjunta sobre as demais. Frequentemente, o cálculo das integrais pode ser difícil ou mesmo impossível analiticamente. No caso de uma densidade *a posteriori* conjunta deve-se aproximar algumas das, ou todas as, integrais associadas a cada densidade *a posteriori* marginal de interesse por métodos numéricos, os quais podem ser determinísticos ou estocásticos. Neste presente trabalho, isto será feito para todos os 3 parâmetros do modelo de mistura finita de normais com variância contaminada.

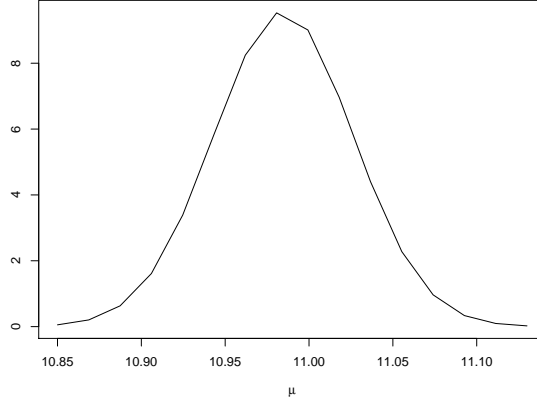
Dentre os métodos determinísticos mais usuais, para a aproximação das integrais correspondentes às densidades *a posteriori* marginais de μ , σ^2 e ν , será utilizado o método da quadratura de Riemann, um caso particular e o mais simples da família das *fórmulas de Newton-Cotes* (Chapra, 2015)[3], as quais substituem o verdadeiro integrando por uma aproximação polinomial dentro de cada subintervalo contido no intervalo de integração. Na quadratura de Riemann, o integrando avaliado em cada subintervalo é aproximado por uma função constante avaliada no limite inferior (ou superior) do mesmo subintervalo.

Aproximações polinomiais com nós igualmente espaçados a cada subintervalo, como as regras de quadratura do Trapézio (grau 1) ou de Simpson (grau 2), obtém resultados com menor erro do que a quadratura de Riemann. Quando o grau da aproximação polinomial tende ao infinito, a quadratura converge para o verdadeiro valor da integral. Porém, em problemas de dimensão igual a 2 ou superior, usar aproximações polinomiais com nós torna-se muito custoso computacionalmente, uma vez que temos três ou mais termos na soma que aproximará a integral para cada dimensão. Por esta razão, escolheu-se a quadratura de Riemann para obter densidades *a posteriori* marginais de μ , σ^2 e ν , pois esta regra possui um único termo na soma aproximadora, facilitando a programação das rotinas iterativas associadas. Em uma dada aproximação, o mesmo número de intervalos será usado para todas as dimensões.

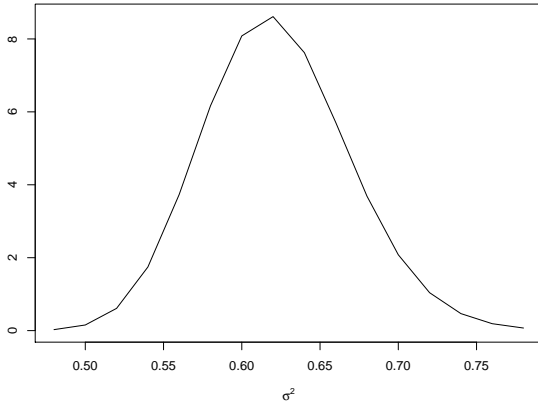
Antes de aproximar as densidades *a posteriori* marginais de cada parâmetro, é necessário aproximar o inverso da constante de proporcionalidade. Dados três parâmetros $(\alpha_1, \alpha_2, \alpha_3)$ e uma amostra dos dados \mathbf{y} quaisquer, suponha que se deseja aproximar a densidade *a posteriori* marginal de α_3 dados os pontos r_i, s_j, t_k da grade formada por todos os subintervalos de integração, $i, j, k \in \{1, \dots, L\}$. Temos pela quadratura de Riemann que

$$\begin{aligned} p(\alpha_3|\mathbf{y}) &= \iint p(\alpha_1, \alpha_2, \alpha_3|\mathbf{y}) d\alpha_1 d\alpha_2 \\ \Rightarrow p(t_k|\mathbf{y}) &= \iint p(\alpha_1, \alpha_2, t_k|\mathbf{y}) d\alpha_1 d\alpha_2 \approx \sum_{i=1}^L \sum_{j=1}^L p(r_i, s_j, t_k|\mathbf{y}) \Delta_i \Delta_j \\ &= \sum_{i=1}^L \sum_{j=1}^L c \cdot h(r_i, s_j, t_k|\mathbf{y}) \Delta_i \Delta_j. \end{aligned} \tag{4}$$

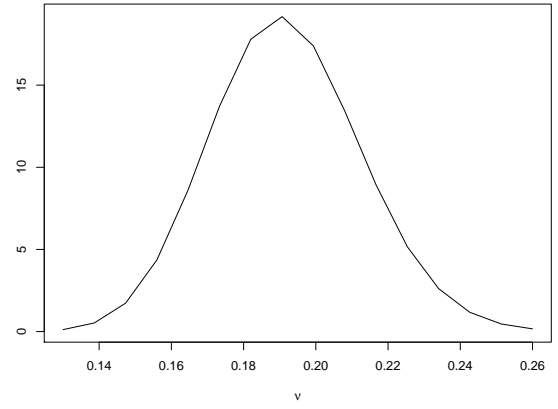
Como c , a constante de proporcionalidade, é dada pelo inverso da densidade *a priori* preditiva $f(\mathbf{y})$, a qual é obtida integrando-se em todo o espaço paramétrico o produto entre a função de verossimilhança $f(\mathbf{y}|\alpha_1, \alpha_2, \alpha_3)$ e as densidades (ou funções de probabilidade) *a priori* para α_1 , α_2 e α_3 , também é possível aproximar c pela quadratura de Riemann. Neste caso, $c^{-1} \approx \sum_{i=1}^L \sum_{j=1}^L \sum_{k=1}^L h(r_i, s_j, t_k|\mathbf{y}) \Delta_i \Delta_j \Delta_k$. Com o valor aproximado para c , é possível calcular (4) nos limites superior e inferior de todos os subintervalos de um dado parâmetro e enfim obter uma aproximação da densidade *a posteriori* marginal deste mesmo parâmetro através de uma curva gráfica que liga todos os valores calculados. Evidentemente, quanto menor o tamanho de cada subintervalo, melhor a curva traçada aproximará a verdadeira densidade.



(a) Densidade *a posteriori* de μ



(b) Densidade *a posteriori* de σ^2



(c) Densidade *a posteriori* de ν

Figura 3: Densidades *a posteriori* marginais pela quadratura de Riemann com $L = 15$

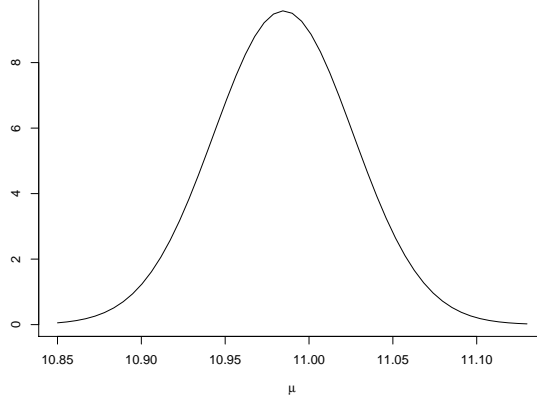
Para o modelo de mistura finita de normais com variância contaminada foram consideradas três quantidades distintas de subintervalos, correspondentes aos 3 cenários para a aproximação via quadratura de Riemann: $L = \{15, 50, 100\}$. Em cada cenário, dividiram-se os três intervalos de massa probabilística obtidos nas figuras 2a a 2c pelo respectivo valor de L . Nos três cenários, os valores calculados para o inverso da constante de proporcionalidade foram bem próximos: 3.9071×10^{-229} para $L = 15$; 3.9014×10^{-229} para $L = 50$ e 3.8999×10^{-229} para $L = 100$. Com tais valores de c^{-1} , foram estimadas as densidades *a posteriori* marginais para μ , σ^2 e ν , cujas curvas são apresentadas nas figuras 3a a 3c; 4a a 4c e 5a a 5c:

Para $L = 15$, a aproximação não é muito suave, mas começa a indicar como cada parâmetro se comporta com relação à locação e à dispersão de sua densidade *a posteriori* marginal. Entretanto, ainda não é possível dizer como é o comportamento com relação à simetria.

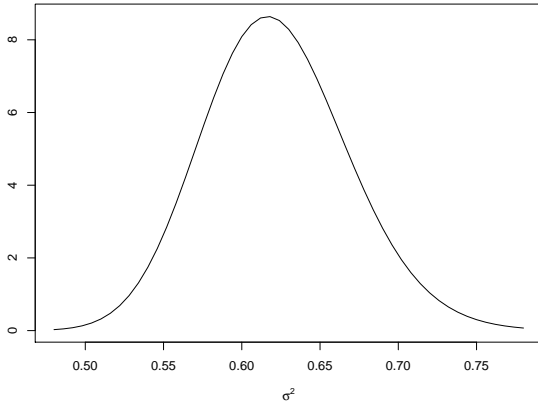
Para $L = 50$, a aproximação é bem suave. Além do comportamento com relação à locação e à dispersão, é possível dizer também como é o comportamento com relação à simetria da densidade *a posteriori* marginal de cada parâmetro. Aparentemente, as densidades para σ^2 e ν são fracamente assimétricas à direita.

Para $L = 100$, os gráficos confirmam a tendência apresentada pelos dois cenários anteriores, mas com uma suavização ainda melhor. Porém, computacionalmente a integração numérica é bem mais demorada, uma vez que o número de pontos para cálculo das densidades *a posteriori* marginais cresce a um fator de ordem $\mathcal{O}(L^3)$.

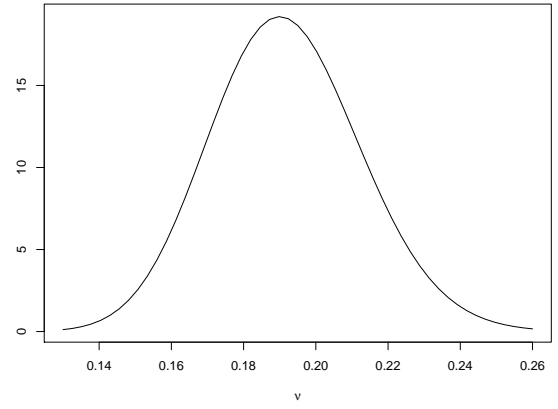
Obtidas as densidades *a posteriori* marginais, é possível ainda estimar pela quadratura



(a) Densidade *a posteriori* de μ



(b) Densidade *a posteriori* de σ^2



(c) Densidade *a posteriori* de ν

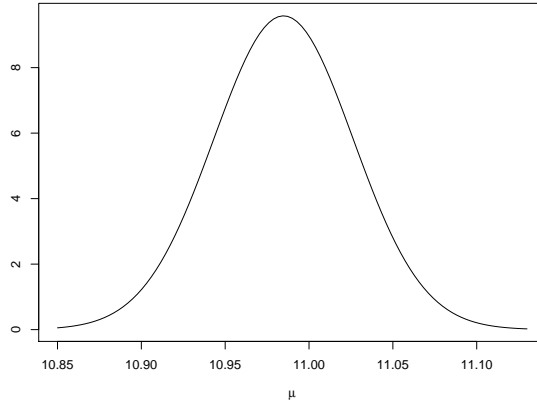
Figura 4: Densidades *a posteriori* marginais pela quadratura de Riemann com $L = 50$

de Riemann os momentos *a posteriori* de ordem m para cada parâmetro e com os mesmos aproximar as estatísticas de média, variância, assimetria e curtose. Sem perda de generalidade, suponha que se deseja obter as estatísticas *a posteriori* para α_1 . A aproximação pela quadratura de Riemann para um momento m é dada por

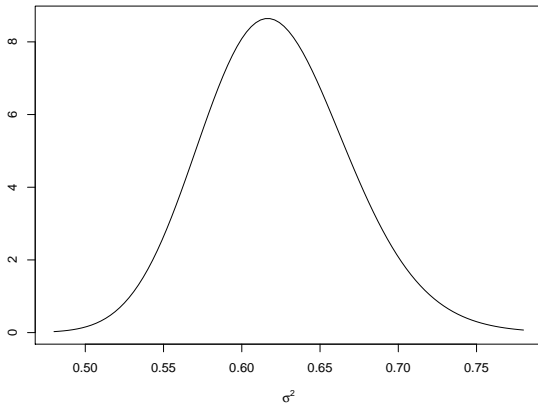
$$\begin{aligned}
 \mathbb{E}(\alpha_1^m | \mathbf{y}) &= \iint \alpha_1^m p(\alpha_1, \alpha_2, \alpha_3 | \mathbf{y}) d\alpha_1 d\alpha_2 d\alpha_3 \approx \sum_{i=1}^L \sum_{j=1}^L \sum_{k=1}^L r_i^m p(r_i, s_j, t_k | \mathbf{y}) \Delta_i \Delta_j \Delta_k \\
 &= \sum_{i=1}^L r_i^m \Delta_i \left[\sum_{j=1}^L \sum_{k=1}^L p(r_i, s_j, t_k | \mathbf{y}) \Delta_j \Delta_k \right] \\
 &= \sum_{i=1}^L r_i^m \Delta_i p(r_i | \mathbf{y}).
 \end{aligned} \tag{5}$$

Como todas as estatísticas de interesse são funções dos quatro primeiros momentos, basta substituir $m = \{1, 2, 3, 4\}$ em (5) para obter as respectivas aproximações. Na Tabela 1, são apresentados os valores aproximados para tais estatísticas em cada um dos três cenários.

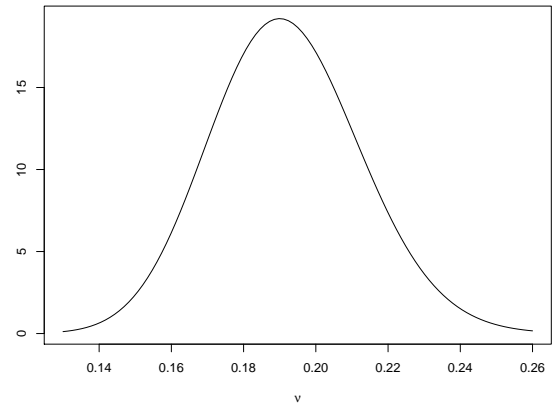
Pela Tabela 1, pode-se concluir que tanto a média quanto a variância *a posteriori* praticamente não variaram nos três cenários, independente do parâmetro considerado. Isto quer dizer que poucos subintervalos são necessários para se obter uma boa aproximação destas duas estatísticas. Por outro lado, o mesmo não pode ser dito para a assimetria e curtose *a posteriori*,



(a) Densidade *a posteriori* de μ



(b) Densidade *a posteriori* de σ^2



(c) Densidade *a posteriori* de ν

Figura 5: Densidades *a posteriori* marginais pela quadratura de Riemann com $L = 100$

para as quais há mudanças na terceira ou mesmo na segunda casa decimal, inclusive quando se passa de $L = 50$ para $L = 100$ subintervalos. Isto se explica pelo fato de que tanto a assimetria quanto a curtose são funções que dependem de vários momentos, até a terceira e quarta ordem respectivamente. Portanto, há maior dificuldade em aproximá-las. Apesar deste fato, para o problema considerado se pode dizer que houve uma estabilidade nas aproximações destas duas últimas estatísticas na medida em que a quantidade de subintervalos crescia.

Com relação aos valores em si, as médias *a posteriori* para os 3 parâmetros ficaram bem próximas, mas não necessariamente iguais, aos respectivos valores do modelo para a distribuição amostral, mesmo para uma amostra muito grande ($n = 500$). É interessante notar que isso ocorre mesmo para o parâmetro ν , cuja distribuição *a priori* é não-informativa no sentido de Bayes-Laplace. Apesar disto, sua distribuição *a posteriori* é influenciada pelas informações de μ e σ^2 contidas tanto nas distribuições *a priori* correspondentes quanto na função de verossimilhança.

Para as variâncias *a posteriori*, todas elas foram bem pequenas, em especial para ν . Isto se deve ao fato de que a amostra inicial dos dados era grande, portanto se poderia esperar uma redução na dispersão de cada parâmetro quando sua informação é atualizada com a distribuição amostral, por menos informativa que fosse a distribuição *a priori* (como no caso de ν). Com relação à assimetria *a posteriori*, esta foi praticamente nula para μ e positiva, mas fraca, para σ^2 e ν (um pouco mais forte para a primeira). Por fim, as aproximações para a curtose *a posteriori* foram todas bem próximas de 3, o valor encontrado para uma distribuição normal,

Tabela 1: Estatísticas *a posteriori* para (μ, σ^2, ν) pela quadratura de Riemann

Cenário	Parâmetro	Média	Variância	Assimetria	Curtose
$L = 15$	μ	10.9847	0.0017	0.0022	2.9603
	σ^2	0.6222	0.0022	0.2230	2.9984
	ν	0.1918	0.0004	0.1572	2.9348
$L = 50$	μ	10.9848	0.0017	0.0056	2.9346
	σ^2	0.6222	0.0021	0.2149	2.9625
	ν	0.1918	0.0004	0.1523	2.8995
$L = 100$	μ	10.9848	0.0017	0.0064	2.9284
	σ^2	0.6222	0.0021	0.2131	2.9542
	ν	0.1918	0.0004	0.1512	2.8913

com o parâmetro σ^2 mais próximo desse valor e ν o mais afastado.

3 Reamostragem Por Importância Sequencial (SIR)

Como uma das aproximações estocásticas a serem consideradas, o método da reamostragem por importância sequencial (SIR), ou reamostragem ponderada, foi uma das primeiras alternativas aos métodos de aproximação determinísticos previamente existentes. Se antes a densidade conjunta era aproximada por uma regra de quadratura, nas aproximações estocásticas o objetivo é simular amostras de uma densidade que aproxime a desejada com o menor erro possível. Este erro dependerá de k , o tamanho de uma amostra retirada da densidade aproximada, de ordem $\mathcal{O}(k^{-1})$, e não mais do conjunto de dados amostrais do modelo inicial.

Proposto por Gordon *et al.* (1993)[4], o método SIR utiliza uma *função de amostragem por importância* g para aproximar (sem perda de generalidade) uma densidade de interesse p . Neste caso, g é uma densidade conhecida e da qual se sabe gerar uma amostra aleatória. Cada ponto selecionado na amostra de g é ponderado para corrigir a probabilidade de amostragem de tal forma que a amostra ponderada aproxime uma outra que seria extraída da densidade de p caso se soubesse gerar da mesma. Os pesos usados para corrigir as probabilidades de amostragem são chamados *pesos de importância padronizados*. Sejam $\theta_1, \dots, \theta_k$ uma amostra aleatória de g e $\mathbf{y} = (y_1, \dots, y_n)$ uma amostra do modelo para os dados observados. Para cada ponto θ_j , $j = 1, \dots, k$, os pesos são dados por

$$w_j(\theta_j) = \frac{p(\theta_j|\mathbf{y})/q(\theta_j)}{\sum_{j=1}^k p(\theta_j|\mathbf{y})/q(\theta_j)}. \quad (6)$$

Após gerar cada valor $w_j(\theta_j)$, uma reamostragem é feita do suporte discreto $\theta_1, \dots, \theta_k$ com probabilidades $w_1(\theta_1), \dots, w_k(\theta_k)$. A nova amostra resultante terá uma distribuição aproximadamente igual à de p , convergindo para esta quando $k \rightarrow \infty$. Em outras palavras, diz-se que o método SIR gera uma amostra de p assintoticamente. Observe que a densidade g a ser escolhida deve ter o mesmo suporte de p para que o método funcione.

No presente trabalho, tem-se 3 parâmetros (μ, σ^2, ν) a serem gerados dada uma densidade *a posteriori* $p(\mu, \sigma^2, \nu|\mathbf{x})$. Como esta densidade possui uma constante de proporcionalidade c associada, pela razão do lado direito de (6) esta é cancelada, sendo possível trabalhar apenas com o núcleo de $p(\mu, \sigma^2, \nu|\mathbf{x})$ calculado no lado direito em (3). Isto por si só é uma vantagem do método SIR em relação à quadratura de Riemann, para a qual se viu que era necessário aproximar c . Como se deseja aproximar uma função conjunta de 3 parâmetros, a densidade

g a ser escolhida deve não apenas ter a mesma dimensão, mas também cada uma de suas componentes marginais deve ter o mesmo suporte do parâmetro correspondente no núcleo de $p(\mu, \sigma^2, \nu|\mathbf{x})$.

Como feito em muitos trabalhos, para g será escolhida uma densidade normal trivariada $N_3(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, cujas componentes têm, cada uma, suporte em toda a reta real. Note que para 2 parâmetros, σ^2 e ν , o respectivo espaço paramétrico não é a reta real ($\Theta_{\sigma^2} = \mathbb{R}_+$ e $\Theta_{\nu} = [0, 1]$, respectivamente). Desta forma, antes de usar o método é necessário reparametrizar (3) de modo que todos os novos parâmetros tenham suporte em \mathcal{R} . Para a reparametrização, consideram-se as transformações $\theta_1 = \mu$, $\theta_2 = \log(\sigma^2)$ e $\theta_3 = \log[\nu/(1 - \nu)]$. Logo, a expressão do núcleo reparametrizado é dada por

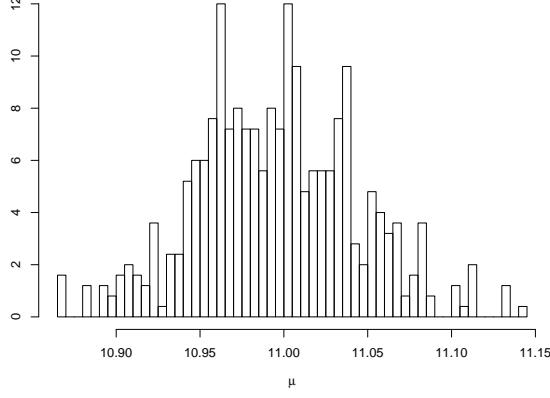
$$\begin{aligned}
p(\theta_1, \theta_2, \theta_3|\mathbf{x}) &= p(\theta_1 = \mu, \theta_2 = \log(\sigma^2), \theta_3 = \log[\nu/(1 - \nu)]|\mathbf{x}) \\
&= p(\mu = \theta_1, \sigma^2 = \exp(\theta_2), \nu = 1/[1 + \exp(-\theta_3)]|\mathbf{x}) \times |J(\theta_1, \theta_2, \theta_3)| \\
&\propto [\exp(\theta_2)]^{-[(n+1)/2+a+1]} \times \exp\left\{-\frac{[(\theta_1 - m)^2/(2V) + d]}{\exp(\theta_2)}\right\} \\
&\times A^*(\mathbf{x}|\theta_1, \theta_2, \theta_3) \times \frac{\exp(\theta_2) \exp(\theta_3)}{[1 + \exp(-\theta_3)]^{-2}} \\
&\propto \left(\frac{1}{\sigma^2}\right)^{(n+1)/2+a+1} \times \exp\left\{-\frac{[(\mu - m)^2/(2V) + d]}{\sigma^2}\right\} \times A(\mathbf{x}|\mu, \sigma^2, \nu) \\
&\times \sigma^2 \nu^3 (1 - \nu)^{-1},
\end{aligned} \tag{7}$$

onde $|J(\theta_1, \theta_2, \theta_3)|$ é o determinante da matriz jacobiana das derivadas parciais de (μ, σ^2, ν) com respeito a $(\theta_1, \theta_2, \theta_3)$. No cálculo deste determinante, todos os elementos fora da diagonal principal da matriz jacobiana serão nulos, já que na reparametrização utilizada não se assumiu nenhuma função dependente de mais de um parâmetro.

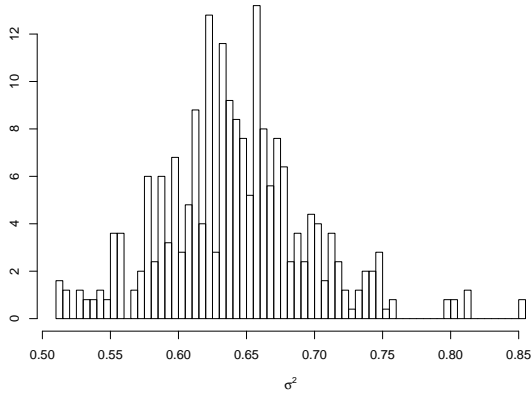
Novamente, para a versão reparametrizada do núcleo da densidade em (7) deve-se primeiro tomar o logaritmo de cada termo em seu produto para só depois exponenciá-la, de modo a evitar problemas numéricos. Para a amostra de tamanho $n = 500$ da mistura finita de normais com variância contaminada tal que $\mu = 11$; $\sigma^2 = 0.64$; $\nu = 0.2$; $m = 11$; $V = 1$; $a = 7$ e $d = 4$ (Figura 1), o logaritmo do núcleo da densidade *a posteriori* reparametrizada é igual a -523.9575 (bem próximo do valor na parametrização original), cuja exponencial é aproximadamente igual a 2.806×10^{-228} .

Os gráficos das densidades *a posteriori* marginais para cada parâmetro quando os demais são fixados nos seus valores verdadeiros (Figuras 2a a 2c), os quais fornecem os respectivos intervalos de massa probabilística, novamente serão úteis, mas agora para definir a parametrização da distribuição normal trivariada a ser usada no método SIR. Para as médias das componentes desta distribuição, será fixado $\boldsymbol{\mu} = (\theta_1, \theta_2, \theta_3) = (\mu, \log(\sigma^2), \log[\nu/(1 - \nu)]) = (11, \log(0.64), \log[0.2/0.8])$. Para a matriz de covariância $\boldsymbol{\Sigma}$, cada elemento da diagonal principal será dado pelo quadrado de $1/6$ do intervalo de massa probabilística do parâmetro correspondente. Esta escolha se justifica pelo fato de que as distribuições mostradas de 2a a 2c têm comportamento próximo à normalidade. Para uma distribuição normal, 99.7% de sua massa probabilística está concentrada a até uma distância de 3 vezes o seu desvio padrão em relação ao ponto médio. Para os elementos fora da diagonal principal de $\boldsymbol{\Sigma}$, todos foram assumidos iguais a zero. Desta forma, $\boldsymbol{\Sigma} = \text{diag}\{0.0022, 0.0065, 0.0203\}$.

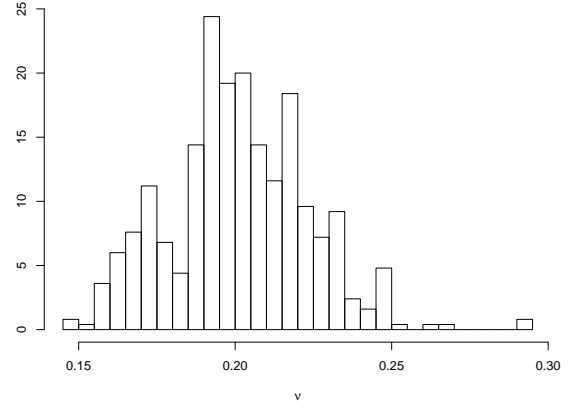
Escolhida a densidade g , resta aplicar o método SIR para obter amostras *a posteriori* de μ , σ^2 e ν . Para este trabalho, foram considerados 3 cenários: $k = \{500, 5000, 50000\}$. Em cada cenário, foram geradas amostras de cada componente de g . Cada ponto amostral foi reparametrizado (quando fosse o caso) de volta para obter os valores estimados de μ , σ^2 e ν . Por fim, para todos os pontos amostrais de g também foi calculado o valor da densidade conjunta, necessária



(a) Histograma de μ



(b) Histograma de σ^2



(c) Histograma de ν

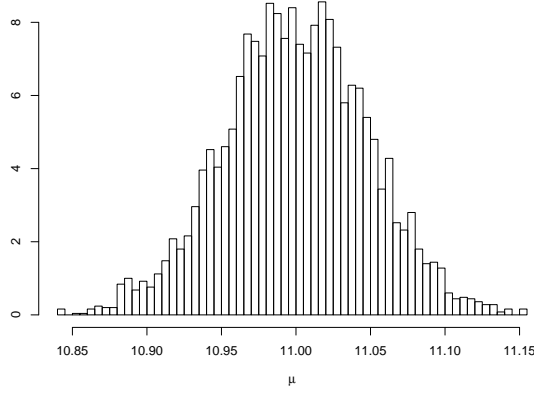
Figura 6: Histograma das densidades *a posteriori* marginais pela método SIR com $k = 500$

para obter os pesos de importância padronizados do método SIR. Construído o suporte discreto para cada parâmetro a partir dos pontos amostrais da componente de g correspondente, amostras *a posteriori* são obtidas associando cada peso ao ponto amostral correspondente de g . Nas figuras 6a a 6c; 7a a 7c e 8a a 8c são apresentados os histogramas para as amostras *a posteriori* de cada parâmetro por cenário.

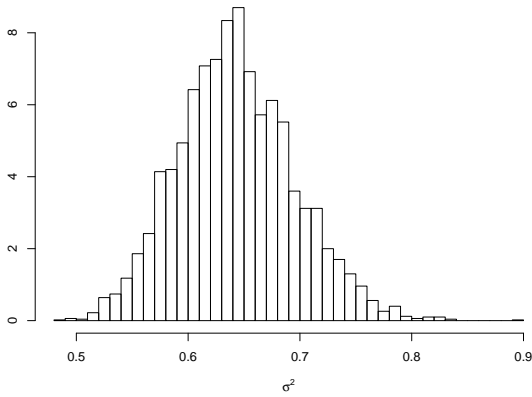
Para $k = 500$, a aproximação da densidade não é boa para nenhum dos 3 parâmetros: há alguns pontos extremos (em inglês, *outliers*) distantes do que seria a massa principal, especialmente para σ^2 . Além disso, há vários pontos de máximo locais (modas), indicando uma curva longe de ser suave. Entretanto, é possível dizer já neste cenário como é o comportamento de cada parâmetro com relação à locação.

Para $k = 5000$, a aproximação é muito mais suave e há bem menos *outliers*. O número de modas locais também é severamente reduzido em comparação ao cenário anterior, embora esta quantidade ainda seja moderada para μ . Além do comportamento com relação à locação, é possível dizer também como é o comportamento com relação à dispersão e à simetria da densidade *a posteriori* marginal de cada parâmetro. Aparentemente, as densidades para σ^2 e ν são fracamente assimétricas à direita.

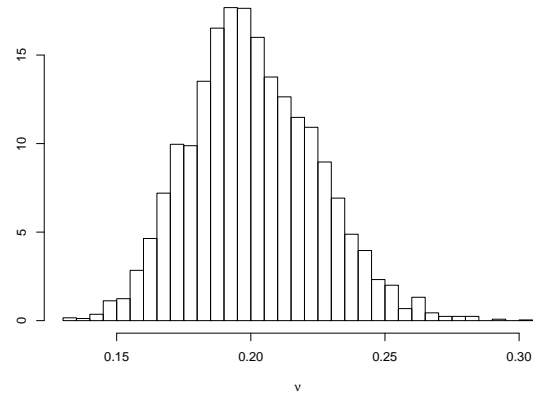
Para $k = 50000$, os gráficos confirmam a tendência apresentada pelo cenário anterior, mas com uma suavização ainda melhor. Agora, todas as distribuições são unimodais e não há mais *outliers*. Computacionalmente, a reamostragem é um pouco mais demorada, mas não na mesma taxa da quadratura de Riemann, já que agora o número de pontos amostrais cresce a um fator



(a) Histograma de μ



(b) Histograma de σ^2



(c) Histograma de ν

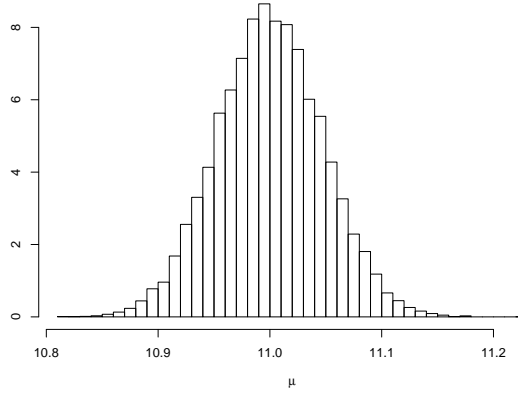
Figura 7: Histograma das densidades *a posteriori* marginais pela método SIR com $k = 5000$

de ordem $\mathcal{O}(k)$, logo de modo linear.

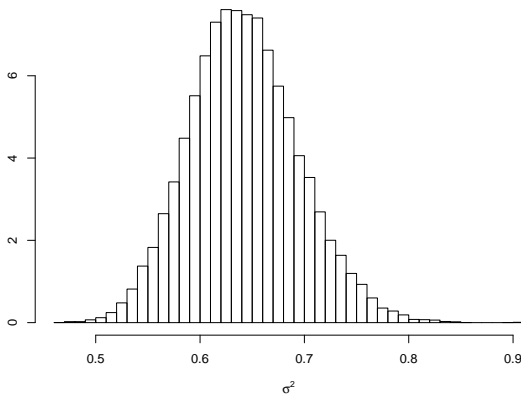
Obtidas as amostras *a posteriori* marginais de cada parâmetro, para aproximar as estatísticas de média, variância, assimetria e curtose é suficiente que se tomem as respectivas estimativas amostrais. Na Tabela 2, são apresentados os valores aproximados para tais estatísticas em cada um dos três cenários.

Pela Tabela 2, pode-se concluir que tanto a média quanto a variância *a posteriori* variaram muito pouco nos três cenários, independente do parâmetro considerado. Isto quer dizer que amostras *a posteriori* de tamanho moderado são o suficiente para se obter uma boa aproximação destas duas estatísticas. Por outro lado, o mesmo não pode ser dito para a assimetria e curtose *a posteriori*, para as quais há grandes mudanças inclusive na primeira casa decimal, mesmo de $k = 5000$ para $k = 50000$. Assim como na quadratura de Riemann, isto se explica pelo fato de que tanto a assimetria quanto a curtose são funções que dependem de vários momentos, até a terceira e quarta ordem respectivamente. Apesar deste fato, também no método SIR se pode dizer que houve uma estabilidade nas aproximações destas duas últimas estatísticas na medida em que k crescia.

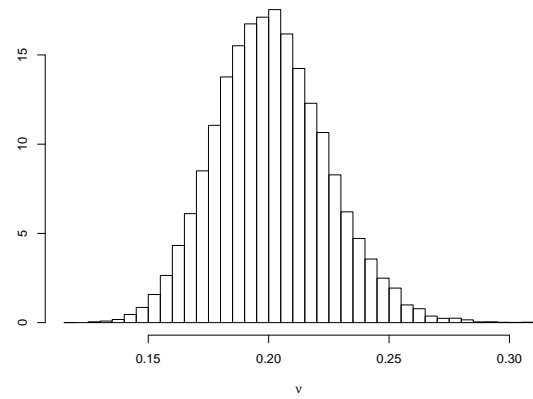
Com relação aos valores em si, as médias *a posteriori* para os 3 parâmetros coincidiram, até a terceira casa decimal, com os respectivos valores do modelo para a distribuição amostral. Desta forma, as médias *a posteriori* aproximadas pelo método SIR estão mais próximas dos valores no modelo do que as obtidas na Tabela 1 pela quadratura de Riemann. Para as variâncias *a posteriori*, todas elas foram bem pequenas, em especial para ν , assim como na quadratura.



(a) Histograma de μ



(b) Histograma de σ^2



(c) Histograma de ν

Figura 8: Histograma das densidades *a posteriori* marginais pela método SIR com $k = 50000$

Com relação à assimetria *a posteriori*, esta foi praticamente nula para μ e positiva, mas fraca, para σ^2 e ν (um pouco mais forte para a segunda, ao contrário da conclusão para a quadratura). Por fim, as aproximações para a curtose *a posteriori* foram todas bem próximas de 3 (desta vez, acima deste valor), o valor encontrado para uma distribuição normal, com o parâmetro μ mais próximo desse valor e ν o mais afastado, um pouco diferente do que foi encontrado para a quadratura.

4 Monte Carlo via Cadeias de Markov (MCMC)

5 Considerações Finais

Referências

- [1] Lachos, V. H.; Castro, L. M. & Dey, D. K. Bayesian inference in nonlinear mixed-effects models using normal independent distributions. *Computational Statistics and Data Analysis*, 64(16):237–252, 2013.
- [2] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

Tabela 2: Estatísticas *a posteriori* para (μ, σ^2, ν) pelo método SIR

Cenário	Parâmetro	Média	Variância	Assimetria	Curtose
$k = 500$	μ	10.9956	0.0024	0.1394	3.0639
	σ^2	0.6423	0.0030	0.4046	4.0116
	ν	0.2010	0.0005	0.3050	3.5598
$k = 5000$	μ	11.0001	0.0023	-0.0244	2.9656
	σ^2	0.6428	0.0027	0.2869	3.1554
	ν	0.2010	0.0006	0.3157	3.0594
$k = 50000$	μ	11.0001	0.0022	0.0266	3.0106
	σ^2	0.6422	0.0027	0.2529	3.1077
	ν	0.2012	0.0005	0.2654	3.1680

- [3] Chapra, S. C. & Canale, R. P. *Numerical Methods for Engineers*. McGraw-Hill Education, New York, NY, 7th edition, 2015.
- [4] Gordon, N.J.; Salmond, D.J. & Smith, A.F.M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2):107–113, 1993.

Apêndice: Código R para os Resultados Apresentados

Introdução

```
# Tamanho amostral "n"; parâmetros "mu", "s2" ( $\sigma^2$ ) e "nu" do modelo para  
# a distribuição amostral e hiperparâmetros m, V, a e d das distribuições "a  
# priori" (somente para "mu" e "s2"):
```

```
n = 500; mu = 11; s2 = 0.64; nu = 0.2; m = 11; V = 1; a = 7; d = 4
```

```
# Geração hierárquica da amostra de tamanho "n" do modelo:
```

```
U_pdf = function(n, nu) {  
  u = sample(c(100, 1), prob=c(nu, 1-nu), size=n, replace=T)  
}
```

```
set.seed(122019)
```

```
u = U_pdf(n, nu)  
sam = rnorm(n, mu, sqrt(s2*u))
```

```
# Histograma da amostra gerada do modelo (figura 1):
```

```
par(mar = c(5,5,3,2))  
hist(sam, breaks=100, prob=T, main="", xlim = c(-10, 40), ylim = c(0, 0.5),  
     xlab = "Amostra do modelo", ylab = "Densidade empírica")
```

```
# Cálculo do logaritmo do núcleo da distribuição "a posteriori" original:
```

```
logA = function(X, mu, sigma2, nu) {  
  n = length(X)  
  k1 = (nu/10)*exp(-(X-mu)^2/(2*100*sigma2))  
  k2 = (1-nu)*exp(-(X-mu)^2/(2*sigma2))  
  k = log(k1 + k2)  
  return(k)  
}
```

```
logh = function(n, mu, sigma2, nu, m, V, a, d) {  
  k1 = -((n + 1)/2 + a + 1)*log(sigma2)  
  k2 = -((mu - m)^2/(2*V) + d)/sigma2  
  k = k1 + k2  
  return(k)  
}
```

```
logkpost = function(X, mu, sigma2, nu, m, V, a, d) {  
  n = length(X)  
  lA = logA(X, mu, sigma2, nu)  
  lh = logh(n, mu, sigma2, nu, m, V, a, d)  
  lkp = sum(lA) + lh  
  return(lkp)  
}
```

```
# Gráficos dos intervalos de massa probabilística para cada parâmetro do
# núcleo da distribuição "a posteriori", fixados os demais, após algumas
# tentativas anteriores para redução dos limites do gráfico e melhor vi-
# sualização da curva (figuras 2a a 2c):
```

```
mu_sup=seq(10.8, 11.2, 0.001); t_mu=length(mu_sup); kp_mu=numeric(t_mu)
```

```
for(i in 1:t_mu) {
kp_mu[i] = exp(logkpost(X=sam,
mu=mu_sup[i], sigma2=s2, nu=nu, m=m, V=V, a=a, d=d))
}
plot(mu_sup, kp_mu, type="l", main="",
      xlab=expression(paste(mu)), ylab="")
```

```
s2_sup=seq(0.4, 0.8, 0.001); t_s2=length(s2_sup); kp_s2=numeric(t_s2)
```

```
for(i in 1:t_s2) {
kp_s2[i] = exp(logkpost(X=sam,
mu=mu, sigma2=s2_sup[i], nu=nu, m=m, V=V, a=a, d=d))
}
plot(s2_sup, kp_s2, type="l", main="",
      xlab=expression(paste(sigma^2)), ylab="")
```

```
nu_sup=seq(0.1, 0.3, 0.001); t_nu=length(nu_sup); kp_nu=numeric(t_nu)
```

```
for(i in 1:t_nu) {
kp_nu[i] = exp(logkpost(X=sam,
mu=mu, sigma2=s2, nu=nu_sup[i], m=m, V=V, a=a, d=d))
}
plot(nu_sup, kp_nu, type="l", main="",
      xlab=expression(paste(nu)), ylab="")
```

O Método da Quadratura de Riemann

```
# Cenários considerados, variando no número de subintervalos utilizados:
```

```
L1 = 15; L2 = 50; L3 = 100
```

```
# Tamanhos dos subintervalos, dada a combinação de cenário e parâmetro, e
# grades definidas pelos mesmos:
```

```
mu_step1 = (11.13 - 10.85)/L1; mu_gr1 = seq(10.85, 11.13, mu_step1)
mu_step2 = (11.13 - 10.85)/L2; mu_gr2 = seq(10.85, 11.13, mu_step2)
mu_step3 = (11.13 - 10.85)/L3; mu_gr3 = seq(10.85, 11.13, mu_step3)
```

```
s2_step1 = (0.78 - 0.48)/L1; s2_gr1 = seq(0.48, 0.78, s2_step1)
s2_step2 = (0.78 - 0.48)/L2; s2_gr2 = seq(0.48, 0.78, s2_step2)
s2_step3 = (0.78 - 0.48)/L3; s2_gr3 = seq(0.48, 0.78, s2_step3)
```



```

nu_step1 = (0.26 - 0.13)/L1; nu_gr1 = seq(0.13, 0.26, nu_step1)
nu_step2 = (0.26 - 0.13)/L2; nu_gr2 = seq(0.13, 0.26, nu_step2)
nu_step3 = (0.26 - 0.13)/L3; nu_gr3 = seq(0.13, 0.26, nu_step3)

# Combinando as três grades unidimensionais em cada cenário, todas com
# número igual de subintervalos e de pontos (nos limites dos interval-
# os) para integração numérica:

grid_tri1 = cbind(mu_gr1, s2_gr1, nu_gr1); l1 = nrow(grid_tri1)
grid_tri2 = cbind(mu_gr2, s2_gr2, nu_gr2); l2 = nrow(grid_tri2)
grid_tri3 = cbind(mu_gr3, s2_gr3, nu_gr3); l3 = nrow(grid_tri3)

# Produto triplo dos tamanhos dos subintervalos em cada cenário:

prod_step1 = mu_step1*s2_step1*nu_step1
prod_step2 = mu_step2*s2_step2*nu_step2
prod_step3 = mu_step3*s2_step3*nu_step3

# Cálculo do inverso da constante de proporcionalidade em cada cenário:

cprop = function(l, X, mgr, s2gr, ngr, prst, m, V, a, d) {
  c = 0
  for (i in 1:l) {
    for (j in 1:l) {
      for (k in 1:l) {
        aux = logkpost(X, mgr[i], s2gr[j], ngr[k], m, V, a, d)
        c = c + exp(aux)*prst
      }
    }
  }
  return(c)
}

c1 = cprop(l=l1, X=sam, mgr=mu_gr1, s2gr=s2_gr1, ngr=nu_gr1,
  prst=prod_step1, m=m, V=V, a=a, d=d)
c2 = cprop(l=l2, X=sam, mgr=mu_gr2, s2gr=s2_gr2, ngr=nu_gr2,
  prst=prod_step2, m=m, V=V, a=a, d=d)
c3 = cprop(l=l3, X=sam, mgr=mu_gr3, s2gr=s2_gr3, ngr=nu_gr3,
  prst=prod_step3, m=m, V=V, a=a, d=d)
c1; c2; c3

# Cálculo das densidades "a posteriori" marginais em cada cenário, com
# uso do produto duplo dos tamanhos dos subintervalos em cada cenário:

pr1_dup12 = mu_step1*s2_step1
pr1_dup13 = mu_step1*nu_step1
pr1_dup23 = s2_step1*nu_step1

```

```

pr2_dup12 = mu_step2*s2_step2
pr2_dup13 = mu_step2*nu_step2
pr2_dup23 = s2_step2*nu_step2

pr3_dup12 = mu_step3*s2_step3
pr3_dup13 = mu_step3*nu_step3
pr3_dup23 = s2_step3*nu_step3

postmu_quarie = function(l, X, mgr, s2gr, ngr, prst, m, V, a, d, c) {
  postmu = numeric(l)
  for (i in 1:l) {
    postconj = 0
    for (j in 1:l) {
      for(k in 1:l) {
        aux = logkpost(X, mgr[i], s2gr[j], ngr[k], m, V, a, d)
        postconj = postconj + exp(aux)*prst
      }
    }
    postmu[i] = postconj/c
  }
  return(postmu)
}

pmq1 = postmu_quarie(l=l1, X=sam, mgr=mu_gr1, s2gr=s2_gr1, ngr=nu_gr1,
  prst=pr1_dup23, m=m, V=V, a=a, d=d, c=c1)
pmq2 = postmu_quarie(l=l2, X=sam, mgr=mu_gr2, s2gr=s2_gr2, ngr=nu_gr2,
  prst=pr2_dup23, m=m, V=V, a=a, d=d, c=c2)
pmq3 = postmu_quarie(l=l3, X=sam, mgr=mu_gr3, s2gr=s2_gr3, ngr=nu_gr3,
  prst=pr3_dup23, m=m, V=V, a=a, d=d, c=c3)

plot(mu_gr1,pmq1,type="l",main="",xlab=expression(paste(mu)),ylab="")
plot(mu_gr2,pmq2,type="l",main="",xlab=expression(paste(mu)),ylab="")
plot(mu_gr3,pmq3,type="l",main="",xlab=expression(paste(mu)),ylab="")

posts2_quarie = function(l, X, mgr, s2gr, ngr, prst, m, V, a, d, c) {
  posts2 = numeric(l)
  for (i in 1:l) {
    postconj = 0
    for (j in 1:l) {
      for(k in 1:l) {
        aux = logkpost(X, mgr[j], s2gr[i], ngr[k], m, V, a, d)
        postconj = postconj + exp(aux)*prst
      }
    }
    posts2[i] = postconj/c
  }
  return(posts2)
}

```

```

psq1 = posts2_quarie(l=l1, X=sam, mgr=mu_gr1, s2gr=s2_gr1, ngr=nu_gr1,
  prst=pr1_dup13, m=m, V=V, a=a, d=d, c=c1)
psq2 = posts2_quarie(l=l2, X=sam, mgr=mu_gr2, s2gr=s2_gr2, ngr=nu_gr2,
  prst=pr2_dup13, m=m, V=V, a=a, d=d, c=c2)
psq3 = posts2_quarie(l=l3, X=sam, mgr=mu_gr3, s2gr=s2_gr3, ngr=nu_gr3,
  prst=pr3_dup13, m=m, V=V, a=a, d=d, c=c3)

plot(s2_gr1, psq1, type="l", main="", xlab=expression(paste(sigma^2)),
  ylab="")
plot(s2_gr2, psq2, type="l", main="", xlab=expression(paste(sigma^2)),
  ylab="")
plot(s2_gr3, psq3, type="l", main="", xlab=expression(paste(sigma^2)),
  ylab="")

postnu_quarie = function(l, X, mgr, s2gr, ngr, prst, m, V, a, d, c) {
  postnu = numeric(l)
  for (i in 1:l) {
    postconj = 0
    for (j in 1:l) {
      for(k in 1:l) {
        aux = logkpost(X, mgr[j], s2gr[k], ngr[i], m, V, a, d)
        postconj = postconj + exp(aux)*prst
      }
    }
    postnu[i] = postconj/c
  }
  return(postnu)
}

pnq1 = postnu_quarie(l=l1, X=sam, mgr=mu_gr1, s2gr=s2_gr1, ngr=nu_gr1,
  prst=pr1_dup12, m=m, V=V, a=a, d=d, c=c1)
pnq2 = postnu_quarie(l=l2, X=sam, mgr=mu_gr2, s2gr=s2_gr2, ngr=nu_gr2,
  prst=pr2_dup12, m=m, V=V, a=a, d=d, c=c2)
pnq3 = postnu_quarie(l=l3, X=sam, mgr=mu_gr3, s2gr=s2_gr3, ngr=nu_gr3,
  prst=pr3_dup12, m=m, V=V, a=a, d=d, c=c3)

plot(nu_gr1,pnq1,type="l",main="",xlab=expression(paste(nu)),ylab="")
plot(nu_gr2,pnq2,type="l",main="",xlab=expression(paste(nu)),ylab="")
plot(nu_gr3,pnq3,type="l",main="",xlab=expression(paste(nu)),ylab="")

```

```
# Cálculo da média, variância, assimetria e curtose "a posteriori" para
# cada parâmetro:
```

```
stat_post = function(gr, marg, prst) {
  media = 0
  var = 0
  assim = 0
  cur = 0
  l = length(gr)

  aux1 = sum(gr*marg*prst)      # Aproxima 1º momento.
  aux2 = sum((gr^2)*marg*prst) # Aproxima 2º momento.
  aux3 = sum((gr^3)*marg*prst) # Aproxima 3º momento.
  aux4 = sum((gr^4)*marg*prst) # Aproxima 4º momento.

  media = aux1
  var = aux2 - (media)^2
  assim = (aux3 - 3*media*var - media^3)/(var^(3/2))
  cur = (aux4 - 4*media*aux3 + 6*media^2*aux2 - 3*(media^4))/(var^2)
  return(list(media, var, assim, cur))
}

stat_post(mu_gr1, pmq1, mu_step1)
stat_post(mu_gr2, pmq2, mu_step2)
stat_post(mu_gr3, pmq3, mu_step3)

stat_post(s2_gr1, psq1, s2_step1)
stat_post(s2_gr2, psq2, s2_step2)
stat_post(s2_gr3, psq3, s2_step3)

stat_post(nu_gr1, pnq1, nu_step1)
stat_post(nu_gr2, pnq2, nu_step2)
stat_post(nu_gr3, pnq3, nu_step3)
```

O Método da Reamostragem Por Importância Sequencial (SIR)

```
# Cálculo do logaritmo do núcleo da distribuição "a posteriori" reparametrizada:
```

```
logr = function(sigma2, nu) {
  k = log(sigma2) + 3*log(nu) - log(1-nu)
  return(k)
}
```

```

logkpost_re = function(X, mu, sigma2, nu, m, V, a, d) {
  n = length(X)
  lA = logA(X, mu, sigma2, nu)
  lh = logh(n, mu, sigma2, nu, m, V, a, d)
  lr = logr(sigma2, nu)
  lkrp = sum(lA) + lh + lr
  return(lkrp)
}

logkpost_re(sam, mu, s2, nu, m, V, a, d)

# Cenários de tamanhos para amostragem da "posteriori" via SIR:

am1 = 500; am2 = 5000; am3 = 50000

# Valores reparametrizados da diagonal principal da matriz de covariâncias
# da distribuição normal trivariada:

((11.13-10.85)/6)^2
((log(0.78)-log(0.48))/6)^2
((log(0.26/0.74)-log(0.13/0.87))/6)^2

# Geração de amostras da distribuição normal trivariada:

library(mvtnorm)

set.seed(122019)
r1_q = rmvnorm(am1, mean=c(11, log(0.64), log(0.2/0.8)),
  sigma=diag(c(0.0022, 0.0065, 0.0203)))
set.seed(122019)
r2_q = rmvnorm(am2, mean=c(11, log(0.64), log(0.2/0.8)),
  sigma=diag(c(0.0022, 0.0065, 0.0203)))
set.seed(122019)
r3_q = rmvnorm(am3, mean=c(11, log(0.64), log(0.2/0.8)),
  sigma=diag(c(0.0022, 0.0065, 0.0203)))

# Cálculo da densidade para cada ponto das amostras geradas:

d1_q = dmvnorm(r1_q, mean=c(11, log(0.64), log(0.2/0.8)), sigma=diag(3))
d2_q = dmvnorm(r2_q, mean=c(11, log(0.64), log(0.2/0.8)), sigma=diag(3))
d3_q = dmvnorm(r3_q, mean=c(11, log(0.64), log(0.2/0.8)), sigma=diag(3))

```

```

# Cálculo dos pesos de reamostragem:

weiSIR = function(X, r_q, m, V, a, d, d_q) {
  k = nrow(r_q)
  aux1 = aux2 = w = numeric(k)
  for(i in 1:k) {
    aux1[i] = logkpost_re(X=X, mu=r_q[i,1], sigma2=exp(r_q[i,2]),
      nu=1/(1+exp(-r_q[i,3])), m=m, V=V, a=a,
      d=d)
  }
  aux2 = aux1/d_q
  w = aux2/sum(aux2)
  return(w)
}

w1 = weiSIR(sam, r1_q, m, V, a, d, d1_q); sum(w1)
w2 = weiSIR(sam, r2_q, m, V, a, d, d2_q); sum(w2)
w3 = weiSIR(sam, r3_q, m, V, a, d, d3_q); sum(w3)

# Geração das amostras da densidade aproximada pelo método SIR com os pesos
# calculados acima:

set.seed(122019)
am1_mu_p = sample(x=r1_q[,1], size=am1, replace=T, prob=w1)
set.seed(122019)
am1_s2_p = sample(x=exp(r1_q[,2]), size=am1, replace=T, prob=w1)
set.seed(122019)
am1_nu_p = sample(x=1/(1+exp(-r1_q[,3])), size=am1, replace=T, prob=w1)

set.seed(122019)
am2_mu_p = sample(x=r2_q[,1], size=am2, replace=T, prob=w2)
set.seed(122019)
am2_s2_p = sample(x=exp(r2_q[,2]), size=am2, replace=T, prob=w2)
set.seed(122019)
am2_nu_p = sample(x=1/(1+exp(-r2_q[,3])), size=am2, replace=T, prob=w2)

set.seed(122019)
am3_mu_p = sample(x=r3_q[,1], size=am3, replace=T, prob=w3)
set.seed(122019)
am3_s2_p = sample(x=exp(r3_q[,2]), size=am3, replace=T, prob=w3)
set.seed(122019)
am3_nu_p = sample(x=1/(1+exp(-r3_q[,3])), size=am3, replace=T, prob=w3)

```

```
# Média; variância; assimetria; curtose e histogramas das amostras geradas:
```

```
library(moments)
```

```
st_sam_post = function(Xpar) {  
  medi = mean(Xpar)  
  vari = var(Xpar)  
  assi = skewness(Xpar)  
  curt = kurtosis(Xpar)  
  return(list(media=medi, variancia=vari, assimetria=assi, curtose=curt))  
}
```

```
st_sam_post(am1_mu_p)  
hist(am1_mu_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(mu)), ylab="")  
st_sam_post(am2_mu_p)  
hist(am2_mu_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(mu)), ylab="")  
st_sam_post(am3_mu_p)  
hist(am3_mu_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(mu)), ylab="")
```

```
st_sam_post(am1_s2_p)  
hist(am1_s2_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(sigma^2)), ylab="")  
st_sam_post(am2_s2_p)  
hist(am2_s2_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(sigma^2)), ylab="")  
st_sam_post(am3_s2_p)  
hist(am3_s2_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(sigma^2)), ylab="")
```

```
st_sam_post(am1_nu_p)  
hist(am1_nu_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(nu)), ylab="")  
st_sam_post(am2_nu_p)  
hist(am2_nu_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(nu)), ylab="")  
st_sam_post(am3_nu_p)  
hist(am3_nu_p, breaks=50, prob=T, main="",  
     xlab=expression(paste(nu)), ylab="")
```

O Método de Monte Carlo via Cadeias de Markov (MCMC)

Cenários de tamanhos para amostragem da "posteriori" via MCMC com passos MH:

```
am1 = 500; am2 = 5000; am3 = 50000
```

Geração de 3 amostras de uma uniforme padrão para o critério de aceitação no
algoritmo de Metropolis-Hastings:

```
set.seed(122019); up1 = runif(am1)
set.seed(122019); up2 = runif(am2)
set.seed(122019); up3 = runif(am3)
```

Ponto inicial da cadeia:

```
x = c(10.86, log(0.50), log(0.14/0.86))
```

Vetores para receber amostras "a posteriori" da distribuição desejada:

```
am1_mu_p = am1_s2_p = am1_nu_p = numeric(am1)
am2_mu_p = am2_s2_p = am2_nu_p = numeric(am2)
am3_mu_p = am3_s2_p = am3_nu_p = numeric(am3)
```

Geração das amostras "a posteriori" via com passos MH:

```
XparMH = function(X, m, V, a, d, x, var_d, up, seed) {
  k = length(up)
  am_mu_p = am_s2_p = am_nu_p = numeric(k)
  cont=0
  set.seed(122019)
  for(i in 1:k) {
    y = rmvnorm(1,mean=c(x[1],x[2],x[3]),sigma=diag(var_d))
    auy = exp(logkpost_re(sam,y[1],exp(y[2]),1/(1+exp(-y[3])),m,V,a,d))
    aux = exp(logkpost_re(sam,x[1],exp(x[2]),1/(1+exp(-x[3])),m,V,a,d))
    acep = min(c(1, auy/aux))
    if(up[i] <= acep) {
      am_mu_p[i] = y[1]
      am_s2_p[i] = exp(y[2])
      am_nu_p[i] = 1/(1+exp(-y[3]))
      x = y; cont = cont + 1
    }
    else {
      am_mu_p[i] = x[1]
      am_s2_p[i] = exp(x[2])
      am_nu_p[i] = 1/(1+exp(-x[3]))
    }
  }
  return(list(XparMH_mu=am_mu_p[(0.2*k + 1):k],
    XparMH_s2=am_s2_p[(0.2*k + 1):k],
    XparMH_nu=am_nu_p[(0.2*k + 1):k], taxa_acep=cont/k))
}
```



```

am1_p = XparMH(X=sam, m=m, V=V, a=a, d=d,
  x=x, var_d=c(0.0022, 0.0065, 0.0203), up=up1, seed=122019)
am2_p = XparMH(X=sam, m=m, V=V, a=a, d=d,
  x=x, var_d=c(0.0022, 0.0065, 0.0203), up=up2, seed=122019)
am3_p = XparMH(X=sam, m=m, V=V, a=a, d=d,
  x=x, var_d=c(0.0022, 0.0065, 0.0203), up=up3, seed=122019)

# Taxa de aceitação do algoritmo em cada cenário de tamanho amostral:

am1_p$taxa_acep; am2_p$taxa_acep; am3_p$taxa_acep

# Média; variância; assimetria; curtose e histogramas das amostras geradas:

st_sam_post(am1_p$XparMH_mu)
hist(am1_p$XparMH_mu, breaks=50, prob=T, main="",
  xlab=expression(paste(mu)), ylab="")
st_sam_post(am2_p$XparMH_mu)
hist(am2_p$XparMH_mu, breaks=50, prob=T, main="",
  xlab=expression(paste(mu)), ylab="")
st_sam_post(am3_p$XparMH_mu)
hist(am3_p$XparMH_mu, breaks=50, prob=T, main="",
  xlab=expression(paste(mu)), ylab="")

st_sam_post(am1_p$XparMH_s2)
hist(am1_p$XparMH_s2, breaks=50, prob=T, main="",
  xlab=expression(paste(sigma^2)), ylab="")
st_sam_post(am2_p$XparMH_s2)
hist(am2_p$XparMH_s2, breaks=50, prob=T, main="",
  xlab=expression(paste(sigma^2)), ylab="")
st_sam_post(am3_p$XparMH_s2)
hist(am3_p$XparMH_s2, breaks=50, prob=T, main="",
  xlab=expression(paste(sigma^2)), ylab="")

st_sam_post(am1_p$XparMH_nu)
hist(am1_p$XparMH_nu, breaks=50, prob=T, main="",
  xlab=expression(paste(nu)), ylab="")
st_sam_post(am2_p$XparMH_nu)
hist(am2_p$XparMH_nu, breaks=50, prob=T, main="",
  xlab=expression(paste(nu)), ylab="")
st_sam_post(am3_p$XparMH_nu)
hist(am3_p$XparMH_nu, breaks=50, prob=T, main="",
  xlab=expression(paste(nu)), ylab="")

```

```
# Gráficos de convergência e de autocorrelação das cadeias geradas:
```

```
library(coda)
```

```
par(mar = c(5,5,3,2))
```

```
traceplot(mcmc(am1_p$XparMH_mu), ylab=expression(paste(mu)))
```

```
plot(acf(am1_p$XparMH_mu), main="")
```

```
traceplot(mcmc(am2_p$XparMH_mu), ylab=expression(paste(mu)))
```

```
plot(acf(am2_p$XparMH_mu), main="")
```

```
traceplot(mcmc(am3_p$XparMH_mu), ylab=expression(paste(mu)))
```

```
plot(acf(am3_p$XparMH_mu), main="")
```

```
traceplot(mcmc(am1_p$XparMH_s2), ylab=expression(paste(sigma^2)))
```

```
plot(acf(am1_p$XparMH_s2), main="")
```

```
traceplot(mcmc(am2_p$XparMH_s2), ylab=expression(paste(sigma^2)))
```

```
plot(acf(am2_p$XparMH_s2), main="")
```

```
traceplot(mcmc(am3_p$XparMH_s2), ylab=expression(paste(sigma^2)))
```

```
plot(acf(am3_p$XparMH_s2), main="")
```

```
traceplot(mcmc(am1_p$XparMH_nu), ylab=expression(paste(nu)))
```

```
plot(acf(am1_p$XparMH_nu), main="")
```

```
traceplot(mcmc(am2_p$XparMH_nu), ylab=expression(paste(nu)))
```

```
plot(acf(am2_p$XparMH_nu), main="")
```

```
traceplot(mcmc(am3_p$XparMH_nu), ylab=expression(paste(nu)))
```

```
plot(acf(am3_p$XparMH_nu), main="")
```