UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

DEPARTAMENTO DE ESTATÍSTICA

Introdução à modelagem geoestatística de dados de contagem: Estimação dos parâmetros em diferentes algoritmos de MCMC

Autor: Lucas da Cunha Godoy

Orientador: Professor Dr. Fernando Hepp Pulgati

Porto Alegre, 24 de Junho de 2016.

Universidade Federal do Rio Grande do Sul

Instituto de Matemática e Estatística
Departamento de Estatística

Introdução à modelagem geoestatística de dados de contagem: Estimação dos parâmetros em diferentes algoritmos de MCMC

Autor: Lucas da Cunha Godoy

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

Banca Examinadora:
Professor Dr. Fernando Hepp Pulgati
Professor Dr. Cleber Bisognin

Porto Alegre, 24 de Junho de 2016.

Agradecimentos

Agradeço à Professora Jandyra Fachel, pelos conselhos e paciência; ao Professor Sergio Bassanesi, por me incentivar a prosseguir nesta caminhada; ao Dr. Eric Brown, por me ajudar com os códigos em Stan; ao Professor Fernando Pulgati, pela orientação e, por fim, à minha família por todo carinho e suporte dedicados à mim.

Sumário

1. Introdução

Os métodos estatísticos de modelagem eram inicialmente baseados nas suposições de independência entre as observações e de normalidade das variáveis resposta. Na falta desta segunda suposição, tinham e ainda têm o suporte do Teorema Central do Limite. Entretanto nem sempre é possível se aproveitar de suposições restritivas como estas. A partir desta problemática e do desenvolvimento dos recursos computacionais, começaram a surgir métodos capazes de lidar com dados de outras distribuições, como os modelos lineares generalizados propostos por *Nelder e Wedderburn*[21]. Este modelo revolucionário pode ser visto como uma extensão menos restritiva do modelo linear clássico, pois a única restrição em relação à distribuição de probabilidade da variável resposta é que ela pertença a família exponencial.

Apesar dos modelos lineares generalizados serem modelos que resolvem um grande número de problemas de modelagem eles não são capazes de lidar com estruturas de dependência entre as observações, mas uma extensão desta classe de modelos chamada Modelos Lineares Mistos Generalizados, proposta por *Breslow e Clayton*[13], é capaz de lidar com observações que apresentam alguma dependência, através da inserção de um efeito aleatório ao modelo.

Os Modelos Lineares Generalizados Geoestatísticos propostos por *Diggle, Tawn e Moye*[10] *apud Diggle e Ribeiro*[9] podem ser vistos como uma extensão dos modelos mistos de *Breslow e Clayton*[13], onde as medidas da variável resposta estão relacionadas espacialmente. Esta relação é representada pela inserção de um Processo Estocástico Subjacente no modelo, que pode ser visto com um efeito aleatório.

Estes modelos podem ser aplicados nas mais diversas áreas, tais como ecologia, epidemiologia, geologia e demografia. Alguns exemplos de aplicações podem ser facilmente encontrados na literatura, tais como: *Highfield, Ward e Laffan* [1], estes autores usaram geoestatística para estimar a distribuição espacial de veados com a doença *foot-and-mouth* e também fazer predições; o exemplo da concentração de radionuclídeo na ilha de *Rongelap*, primeiramente analisado por *Diggle* [2] e, por fim, o exemplo de contagems de erva daninha no sudoeste da Suécia presente em *Guillot, Lorén e Rudemo*[3] e analisado no trabalho de *Jing*[4].

O objetivo principal deste estudo é oferecer aos pesquisadores não somente uma introdução ao Modelo Poisson Log-linear para dados espacializados, mas também uma comparação, no que diz respeito à estimação de parâmetros, dos pacotes computacionais mais usados na literatura para modelagem geoestatística de dados de contagem. Nesta comparação serão

considerados os seguintes aspectos: esforço computacional, dificuldades na modelagem e eficiência na estimação. Para isso será usado um banco de dados simulado, no qual os verdadeiros parâmetros do modelo são conhecidos.

Este trabalho está estruturado como segue: No Capítulo 2 são apresentadas as definições básicas para a compreensão dos modelos geoestatísticos, além dos próprios modelos, o método e o cenário de simulação. No Capítulo 3 estarão as análises descritivas dos dados simulados e os resultados da estimação dos parâmetros de interesse nos diferentes pacotes computacionais. E, por fim, no Capítulo 4, os resultados são comparados e discutidos.

2. Metodologia

2.1. Dados Geoestatísticos

Os dados geoestatísticos, segundo a descrição de *Cressie*[8], têm a seguinte configuração:

$$(Y_i, \boldsymbol{x_i}): i = 1, \dots, n.$$

Onde, Y_i é uma variável aleatória que representa uma característica de interesse que está associada à localização $\boldsymbol{x_i}$, que está contida em uma região espacial $\boldsymbol{D} \in \mathbb{R}^{\mathbf{2}}$; e um fenômeno espacial continuamente distribuído representado pelo processo estocástico subjacente $\{S(\boldsymbol{x}): \boldsymbol{x} \in \boldsymbol{D} \in \mathbb{R}^{\mathbf{2}}\}$. Definidas estas duas principais características, é assumido que Y_i pode ser uma medida direta ou uma medida indireta estatisticamente correlacionada com $S(\boldsymbol{x_i})$.

2.2. O Processo Subjacente $S(x)$

É usual, em técnicas geoestatísticas, independentemente do modelo que esteja sendo considerado, assumir que o processo $S(\boldsymbol{x})$ seja um Processo Gaussiano com média μ, variância $\sigma^2 = Var\{S(\boldsymbol{x})\}$ e Corr$\{S(\boldsymbol{x}), S(\boldsymbol{x'})\} = \rho(d)$, onde $d = \| \boldsymbol{x} - \boldsymbol{x'} \|$ e $\|.\|$ representa a norma euclidiana. Nestes casos, os processos gaussianos garantem uma série de propriedades desejáveis neste tipo de análise, ver *Cressie*[8].

Outra característica importante que o Processo subjacente deve ter é a estacionariedade, que assim como a estrutura espacial, será definida nas próximas seções.

2.3. Estacionariedade

Estacionariedade de Segunda ordem

Um processo é considerado estacionário de segunda ordem se sua média é constante para todos os locais x da região de estudo D e sua variância é finita, ou seja,

$$E\{S(\boldsymbol{x})\} = \mu, \forall\, \boldsymbol{x} \in D,$$

$$Var\{S(\boldsymbol{x})\} < \infty,$$

além disso,

$$Cov\{S(\boldsymbol{x}), S(\boldsymbol{x'})\} = C(\boldsymbol{x} - \boldsymbol{x'}), \forall\, \boldsymbol{x}, \boldsymbol{x'} \in D,$$

onde a função $C(.)$ é conhecida como covariograma ou função de covariância espacial. Ou seja, esta segunda suposição implica que a covariância do processo de interesse entre dois locais depende apenas da distância existente entre eles.

Estacionariedade Estrita

A estacionariedade estrita é definida pela seguinte relação:

$$F_{\boldsymbol{x_1},\ldots,\boldsymbol{x_n}}(s_1,\ldots,s_n) = P\{S(\boldsymbol{x_1}) \leq s_1,\ldots,S(\boldsymbol{x_n}) \leq s_n\}$$
$$= P\{S(\boldsymbol{x_1}+\boldsymbol{h}) \leq s_1,\ldots,S(\boldsymbol{x_n}+\boldsymbol{h}) \leq s_n\},$$

onde, $n > 1$; $\boldsymbol{x} = (\boldsymbol{x_1},\ldots,\boldsymbol{x_n})$ representam locais, ou seja, coordenadas dentro da região de estudo; $F_{x_1,\ldots,x_n}$representa a função de distribuição acumulada global, isto é, de toda a região de estudo e, por fim, $\boldsymbol{h}$ é uma quantidade fixa que representa uma translação na região de estudo. A afirmação feita sugere que a lei de distribuição de probabilidade de um processo não se altere quando este sofrer translações. Neste caso o processo é dito ser fortemente estacionário.

Estacionariedade Intrínseca

A estacionariedade intrínseca é caracterizada pelas seguintes suposições:

$$E\{S(\boldsymbol{x}) - S(\boldsymbol{x}')\} = 0,$$
$$Var\{S(\boldsymbol{x}) - S(\boldsymbol{x}')\} = 2\gamma(\mathrm{d}),$$

onde d representa a distância entre os locais $\boldsymbol{x}$ e $\boldsymbol{x}'$. A quantidade $\gamma(\mathrm{d})$ é conhecida como semivariograma.

2.4. Isotropia

Quando um processo estocástico como o definido em *2.3*. é estacionário de segunda ordem e sua função de covariância espacial $C(\boldsymbol{x}-\boldsymbol{x}')$depende apenas de $\| \boldsymbol{x}-\boldsymbol{x}' \|$, ele é dito isotrópico. Caso a função de covariância espacial de uma mesma distância variar para diferentes ângulos o processo é classificado como anisotrópico.

2.5. Estrutura de Dependência Espacial

O comportamento padrão de estrutura espacial considera que a correlação espacial $\rho(d)$ entre $S(\boldsymbol{x})$ e $S(\boldsymbol{x}')$ diminui quando a distância $d = \| \boldsymbol{x}-\boldsymbol{x}' \|$ aumenta, ou seja,

$$\lim_{d \to \infty} \rho(d) \to 0.$$

Neste contexto é importante ressaltar que existe mais de uma opção para quantificar a estrutura de covariância espacial. As opções mais utilizadas na literatura são a função de correlação espacial, a função de covariância espacial e o semivariograma, os quais serão definidos a seguir. Estas três diferentes funções que têm um objetivo em comum possuem relações explícitas. É usual definir um modelo de correlação espacial, de acordo com famílias

de funções propostas pela literatura, que garanta que a matriz de covariâncias seja positiva definida, que permite garantir variâncias não-negativas. E, a partir desta escolha de família de correlação espacial, deve-se usar ou estimadores da covariância espacial ou estimadores do semivariograma para estimar os parâmetros da estrutura espacial.

Correlação espacial

Comumente representada pela letra grega ρ, a função de correlação espacial é definida a partir de famílias de funções propostas pela literatura que dependem apenas das distâncias d. As mais conhecidas e usadas são:

Família Matérn

A Família Matérn(*Figura 1*) é representada por,

$$\rho(d) = \{2^{k-1}\Gamma(k)\}^{-1}(d/\phi)^k K_k(d/\phi),$$

onde,

$K_k(.)$ é uma função modificada de Bessel de ordem k;

$\phi > 0$, é o parâmetro de escala com a dimensão da distância que controla o alcance da dependência espacial;

$k > 0$, chamado de ordem do processo, é um parâmetro que, além de ter influência no alcance da dependência espacial, também é responsável pela diferenciabilidade do processo de interesse $S(x)$. Especificamente $S(x)$ é $k - 1$ vezes diferenciável em média quadrática.

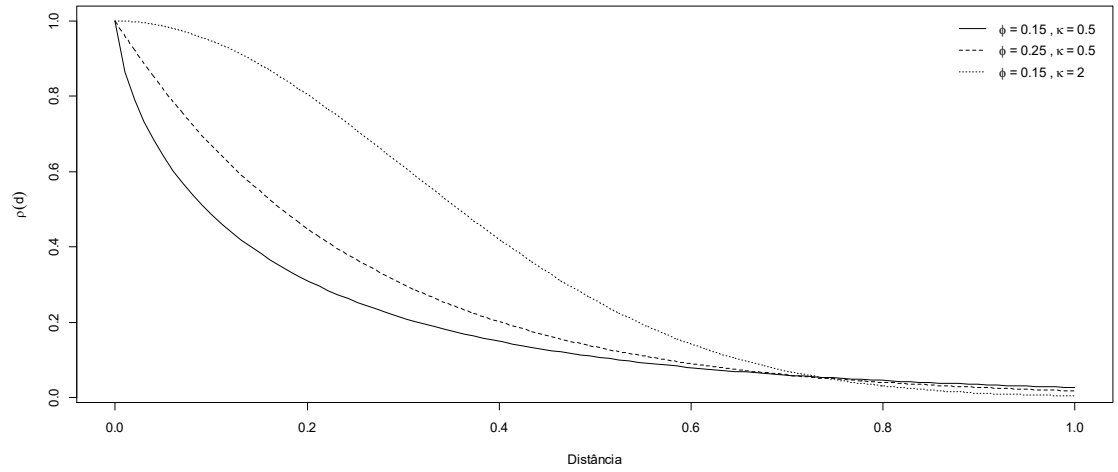

Figura 1- Correlação espacial: Família Matérn, com diferentes parâmetros.

Família Exponencial Potência

Esta família é definida pela seguinte função de correlação

$$\rho(d) = \exp\{-(d/\phi)^k\}.$$

Os parâmetros desta família são análogos aos parâmetros da família Matérn, entretanto os valores que k pode assumir estão no intervalo $(0,2]$, ver *Figura 2*.

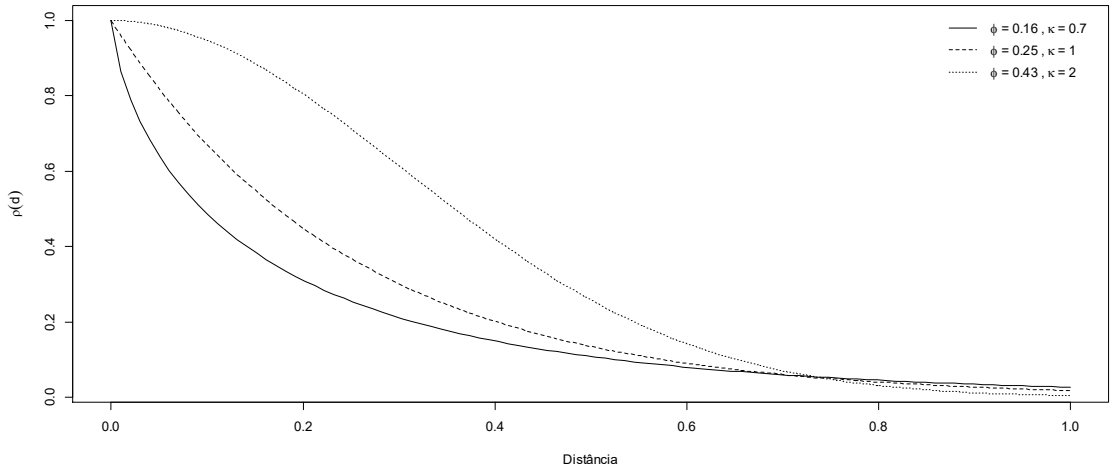

Figura 2 - Correlação espacial: Família Exponencial Potência, com diferentes parâmetros.

Família Esférica

Uma outra família amplamente usada na literatura de geoestatística clássica é a família esférica(*Figura 3*), definida por

$$\rho(d) = \begin{cases} 1 - \frac{3}{2}(d/\phi) + \frac{1}{2}(d/\phi)^3, se\ 0 \leq d \leq \phi \\ 0, caso\ contrário, \end{cases}$$

onde $\phi > 0$, é o parâmetro de alcance da dependência espacial. Está função só é diferenciável quando $d = \phi$, isto causa dificuldades no uso de técnicas que usam estimação de máxima verossimilhança *(Warnes e Ripley*[22]*; Mardia and Watkins*[23] *apud Diggle e Ribeiro*[9]).

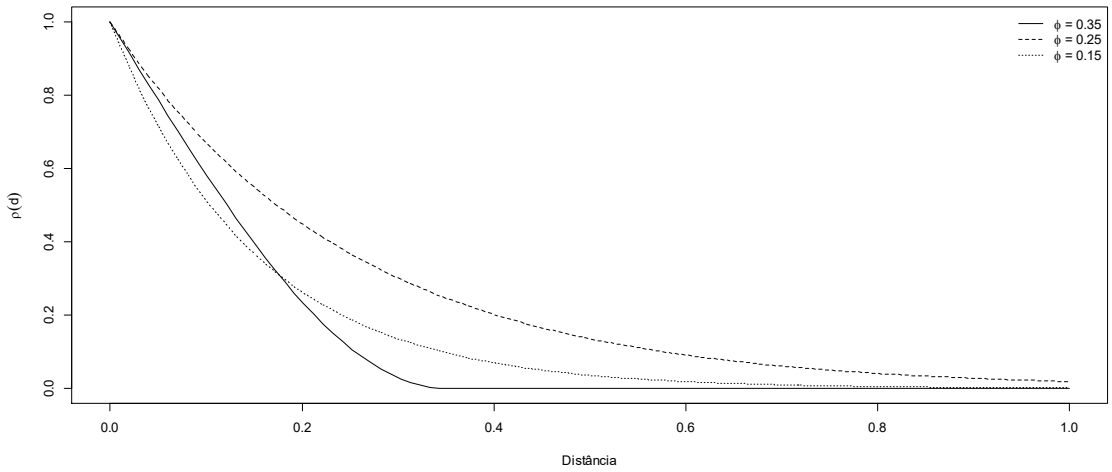

Figura 3 - Correlação espacial: Família Esférica, com diferentes parâmetros.

Covariância Espacial

A função de covariância espacial(*Figura 4*) é proporcional à função de correlação espacial, portanto sua forma dependerá diretamente do modelo de correlação espacial adotado e é definida por

$$C(d) = \sigma^2 \rho(d).$$

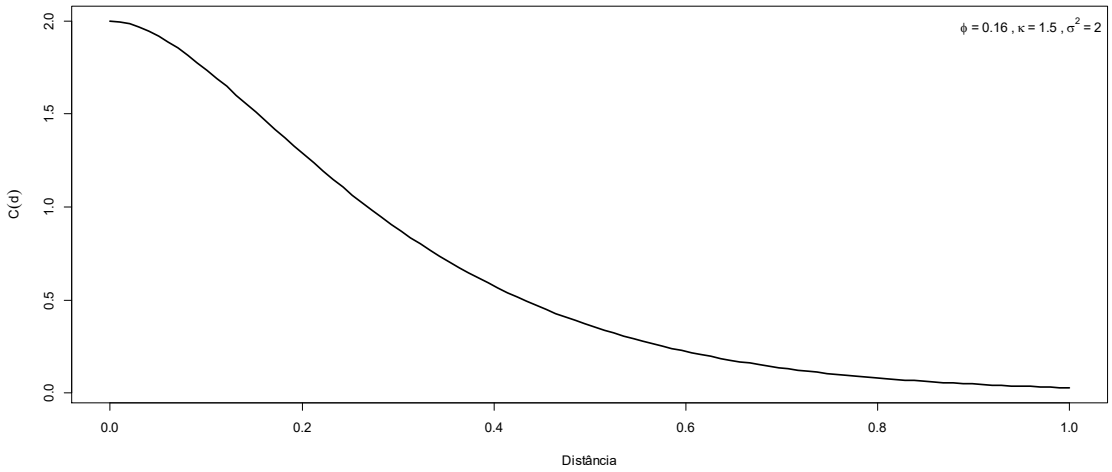

Figura 4 - Exemplo de Covariância Espacial da Família Matérn.

Semivariograma

O semivariograma (*Figura 5*) é considerado a principal ferramenta da geoestatística clássica. Na geoestatística clássica, são estimados os parâmetros da estrutura espacial do processo através de modelos feitos a partir do semivariograma amostral. O semivariograma é definido por

$$\gamma(d) = \frac{1}{2} Var\{S(\boldsymbol{x}) - S(\boldsymbol{x}')\}.$$

Caso o processo seja estacionário e isotrópico o semivariograma pode ser definido como

$$\begin{aligned}\gamma(d) &= \frac{1}{2}(Var\{S(\boldsymbol{x})\} + Var\{S(\boldsymbol{x}')\} - 2C(d)) \\ &= \frac{1}{2}(\sigma^2 + \sigma^2 - 2C(d)) \\ &= \frac{1}{2}(2\sigma^2 - 2\sigma^2\rho(d)) \\ &= \sigma^2\big(1 - \rho(d)\big).\end{aligned}$$

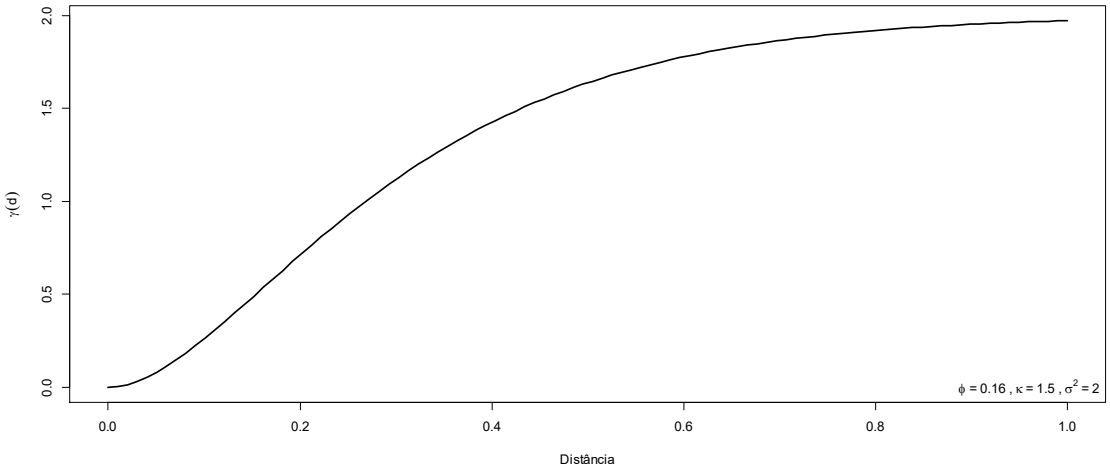

Figura 5 - Exemplo de um Semivariograma da Família Matérn.

Note que o semivariograma pode ser visto como uma espécie de inversão da função de covariância espacial (*Figura 6*).

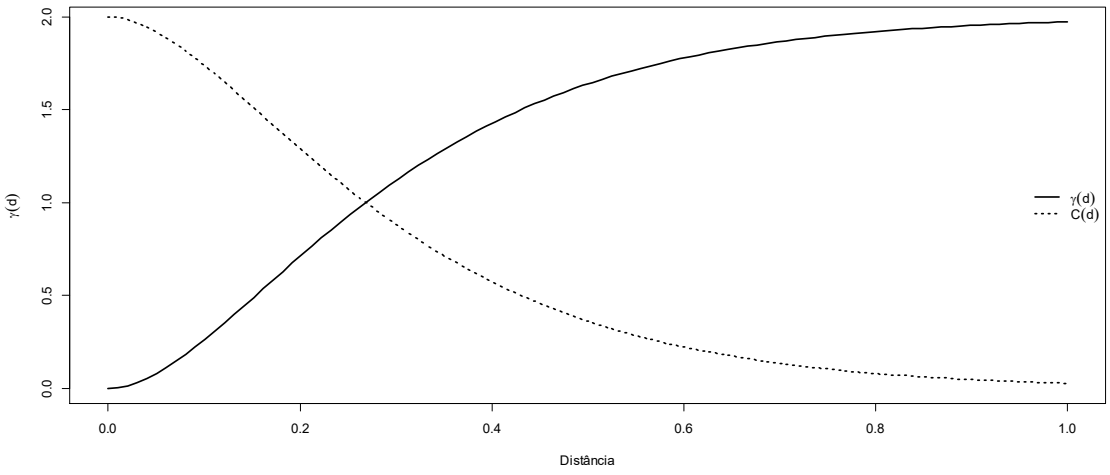

Figura 6 - Ilustração da relação existente entre a Covariância Espacial e o Semivariograma.

Efeito Pepita

Este efeito, que pode ser identificado através de uma descontinuidade na origem do semivariograma estimado (*Figura 7*), é tido na literatura como uma medida responsável pela variabilidade em pequena escala. Ou seja, ao calcularmos o semivariograma assumimos que

$$\lim_{d \to 0} \gamma(d) \to 0,$$

isto é o mesmo que afirmar que caso exista mais de uma medida Y em um mesmo local $\boldsymbol{x}$, esta medida não terá variabilidade. Entretanto tal suposição não é razoável, pois mesmo que não haja variabilidade da medida em um mesmo local, podem haver problemas de medição ou outras causas de variação em pequena escala. Logo o efeito pepita, representado por $\tau^2 \geq 0$, pode ser visto como uma medida de erro.

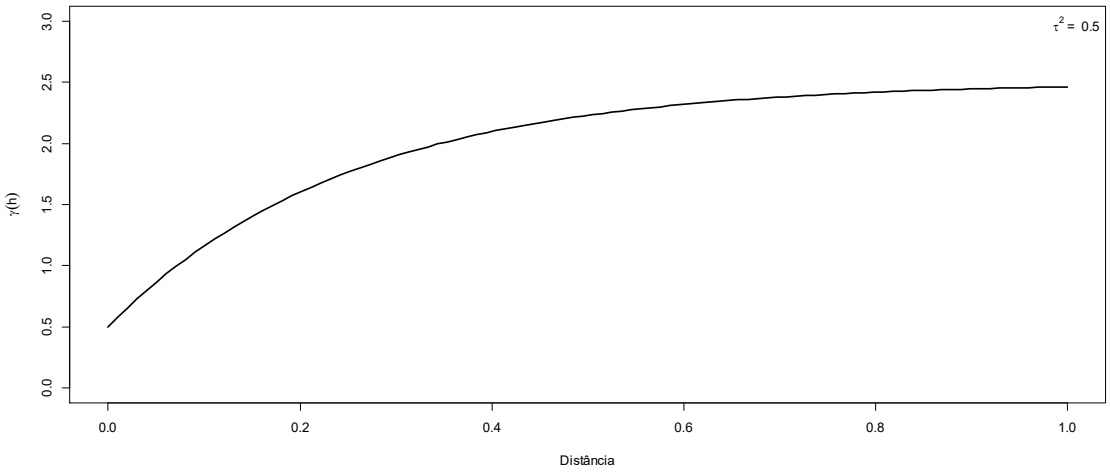

Figura 7 - Exemplo de Semivariograma com efeito pepita.

2.6. Tendência

Conforme vimos nas seções anteriores, o processo latente $S(x)$ e Y são aproximadamente iguais ou têm alguma relação estatística. Também foi visto que uma das condições de estacionariedade do processo latente $S(x)$ é que sua média seja constante para todo $\boldsymbol{x}$, ou seja, $E\{S(\boldsymbol{x})\} = \mu$. Para que esta suposição seja verdadeira é necessário que $E\{Y(\boldsymbol{x})\} = \mu$ independentemente do local $\boldsymbol{x}$. Entretanto, frequentemente Y não tem média constante para toda região de estudo. Quando isto ocorre necessita-se que a tendência de Y seja removida. Os métodos tradicionais de remoção de tendência são os modelos lineares. Feito isso, o processo latente $S(\boldsymbol{x})$ é analisado através dos resíduos deste modelo, pois estes sim tem média constante em toda região de estudo.

2.7. Modelos Lineares Geoestatísticos Generalizados

Esta classe de modelos surgiu da necessidade de modelar dados georeferenciados que não fossem oriundos de uma distribuição Normal. A técnica foi proposta por *Diggle, Tawn e Moye* [6] e se trata de uma tentativa de adaptar o Modelo Gaussiano (ver *Cressie*[8]) através da combinação das técnicas de Modelos Lineares Generalizados, Modelos Mistos Generalizados e o próprio Modelo Gaussiano.

Para esta modelagem é utilizada uma função de ligação (*link*) (para mais detalhes ver *Nelder e Wedderburn*[21]) e um efeito aleatório. No caso em que Y segue uma distribuição Poisson pode ser que tenhamos que inserir um efeito aleatório adicional que seria responsável por modelar a chamada *overdispersion*. Estes modelos podem ser vistos como um caso especial dos modelos mistos e são descritos da seguinte maneira

$$Y_i | \, S(\boldsymbol{x_i}) \sim \mathcal{P}(\mu_i), i = 1, \dots n.$$
$$\mu_i = \, g(\boldsymbol{X}'\beta + S(\boldsymbol{x_i})).$$

Onde,

$S(x_i)$, é um processo gaussiano seguindo as definições da Seção (2.3);

$Y_i | \, S(\boldsymbol{x_i})$ é uma variável aleatória independente e identicamente distribuída com média populacional μ_i. Note que, dependendo da distribuição de $Y_i | \, S(x_i)$, podem haver parâmetros adicionais;

$g(.)$, é a inversa da função link;

$\boldsymbol{X}$, é uma matriz com uma coluna de constante em 1 e as outras colunas correspondentes às covariáveis.

Os modelos lineares geoestatísticos generalizados (MLGG) mais usados são o modelo Binomial Logístico, para variáveis binárias, e o modelo Poisson Log-linear, para dados de contagem. Apenas o segundo será considerado neste trabalho.

2.7.1. O Semivariograma nos Modelos Lineares Generalizados Geoestatísticos

Considerado a ferramenta mais importante na geoestatística quando se está lidando com dados gaussianos o semivariograma não tem a mesma importância no contexto de Modelos Lineares Generalizados Geoestatísticos. Isso se deve principalmente a dois fatos: primeiro, o semivariograma é uma ferramenta baseada em momentos de segunda ordem; segundo, o semivariograma da variável resposta não tem relação explícita com o semivariograma do processo subjacente.

Supondo que $S(\boldsymbol{x})$ é um processo gaussiano estacionário e isotrópico com média 0 e variância σ^2, e que as observações Y_i condicionais à $S(\boldsymbol{x_i})$ são independentes e identicamente distribuídas com esperanças condicionais $\mu_i = g(\boldsymbol{X'}\beta + S(\boldsymbol{x_i}))$ e variâncias condicionais $v_i = v(\mu_i)$. A partir destas suposições o semivariograma de Y pode ser definido através de esperança condicional como

$$\gamma_Y(d) = E\left[\tfrac{1}{2}\left(Y_i - Y_j\right)^2\right]$$

$$= \frac{1}{2}E_S\left[E_y\left[\left(Y_i - Y_j\right)^2 \middle| S(.)\right]\right]$$

$$= \frac{1}{2}E_S\left[E_y\left[{Y_i}^2 - 2Y_iY_j + {Y_j}^2 \middle| S(.)\right]\right]$$

$$= \tfrac{1}{2}E_S\left[\{g\left(\boldsymbol{X'}\beta + S(\boldsymbol{x_i})\right) - g\left(\boldsymbol{X'}\beta + S(\boldsymbol{x_j})\right)\}^2 + v\left(g\left(\boldsymbol{X'}\beta + S(\boldsymbol{x_i})\right)\right) + v\left(g\left(X'\beta + S(\boldsymbol{x_j})\right)\right)\right]$$

$$\boldsymbol{\gamma_Y(d) = \frac{1}{2}(E_S\left[\{g\left(X'\beta + S(x_i)\right) - g\left(X'\beta + S(x_j)\right)\}^2\right] + 2E_S[v(g(X'\beta + S(x_i)]),} \quad (1)$$

onde a última igualdade está amparada pela suposição de que a distribuição marginal de $S(\boldsymbol{x_i})$ é a mesma para todos os locais $\boldsymbol{x_i}$ (ver *Diggle e Ribeiro*[9]). O último termo da igualdade(*Equação 1*) é uma constante, e será representado por $2\bar{\tau}^2$. *Diggle e Ribeiro*[9] usaram séries de Taylor de primeira ordem para aproximar o primeiro termo do lado direito da igualdade e obtiveram os seguintes resultados

$$g\left(\boldsymbol{X'}\beta + S(\boldsymbol{x_i})\right) \approx g(\boldsymbol{X'}\beta) + S(\boldsymbol{x_i})g'(\boldsymbol{X'}\beta),$$

de onde segue que

$$\boldsymbol{\gamma_Y(d) \approx g'(X'\beta)^2\gamma_S(d) + 2\bar{\tau}^2.} \quad (2)$$

A partir da *Equação 2* conclui-se que, o semivariograma na escala de Y é aproximadamente proporcional ao semivariograma do processo latente gaussiano, com a adição de um termo que é análogo ao efeito pepita.

Outra conclusão importante destes cálculos é que o cálculo do semivariograma de Y é uma ferramenta útil para verificar a existência tanto da estrutura espacial quanto do efeito pepita, mas não é aconselhável usá-lo para a estimação dos parâmetros referentes à estrutura espacial do processo $S(x)$.

2.8. Estimação

Como consequência da suposição feita nos modelos mistos generalizados de que $Y = (Y_1, \dots, Y_n)$ condicionado em um efeito aleatório $S = (S_1, \dots, S_n)$[1] é uma variável aleatória independente e igualmente distribuída pode-se escrever a função de verossimilhança desta variável de forma explícita.

Seja θ, o conjunto de parâmetros que determinam a distribuição condicional de Y dado S e θ, então

$$L(\theta|S) = \prod_{i=1}^{n} f_i(y_i|S,\theta).$$

Agora, considerando $g(S;\phi)$ como a distribuição conjunta de S com parâmetro ϕ. Então, com um enfoque clássico, a verossimilhança das variáveis observadas Y é obtida através das marginais do efeito não observável S, conduzindo à verossimilhança do modelo misto,

$$L(\theta,\phi) = \int_S \prod_{i=1}^{n} f_i(y_i|S,\theta)g(s,\phi)ds. \quad (3)$$

No caso em que os efeitos aleatórios são mutuamente independentes o problema é resolvido através da substituição da integral múltipla por um produto de integrais unidimensionais. Entretanto, no caso dos modelos geoestatísticos, onde o efeito aleatório tem a mesma dimensão da variável resposta, os métodos numéricos para a resolução da integral na *Equação 3* falham. A partir desta problemática os métodos mais usados na literatura para estimação dos parâmetros dos modelos geoestatísticos generalizados são métodos bayesianos baseados em simulações.

2.9. Estimação Bayesiana

No caso em que funções de verossimilhança não têm uma resolução analítica fechada a inferência bayesiana se mostra uma ferramenta importante e poderosa. Através de métodos de simulação ela é capaz de obter as distribuições preditivas e distribuições a posteriori, pelas quais é possível fazer a inferência sobre os parâmetros desejados.

Nos modelos lineares geoestatísticos generalizados alguns métodos tradicionais de MCMC, como Gibb's Sampling e Metropolis-Hasting, costumam apresentar problemas de convergência e autocorrelação. Os pacotes estatísticos que aqui serão utilizados utilizam diferentes abordagens do algoritmo Metropolis-Hasting, tais como Langevin-Hasting, ver *Diggle e Ribeiro*[9] *apud Jing*[4], e No-U-turn (NUTS), ver *Hoffman e Gelman*[20].

---

[1] Por simplicidade de notação, em alguns casos, adotaremos: $S_i = S(x_i)$.

2.10. O Modelo Poisson log-linear

O modelo de interesse deste estudo é o Poisson log-linear, e se trata de um caso especial dos modelos lineares geoestatísticos generalizados que é direcionado especificamente para dados de contagem. Considerando a configuração dos modelos lineares geoestatísticos generalizados feitos na seção anterior, podemos escrever o modelo Poisson log-linear como

$$Y_i|S(x) \sim Poisson(\mu_i)$$
$$ln(\mu_i) = X'\beta + S(x_i),$$

onde X', β e $S(x)$ são definidos conforme as suposições das seções anteriores e $\ln(.)$ é a função link deste modelo.

O semivariograma aproximado deste modelo pode ser escrito como

$$\gamma_Y \approx \exp\{X'\beta\}^2 \gamma_S + 2\bar{\tau}^2.$$

É de extrema importância notar que o semivariograma da variável Y deve ser usado apenas com a finalidade de verificar a existência da dependência espacial e do efeito pepita no processo subjacente, pois γ_Y é uma aproximação e, apesar de ser proporcional à γ_S, a relação entre os dois semivariogramas não é linear.

2.11. Pacotes Computacionais

O pacote *geoRglm*

O pacote *geoRglm*, desenvolvido por *Christensen e Ribeiro*[7], é uma extensão do pacote *geoR* que foi desenvolvido por *Diggle e Ribeiro*[12] e sua proposta é realizar as análises geoestatísticas de dados não-gaussianos, especificamente para dados binários e de contagem. O algoritmo usado por este método é o algoritmo Langevin-Hasting, que é uma extensão do Metropolis-Hasting, que usa informações dos gradientes das distribuições a posteriori.

Este pacote, em conjunto com o *geoR,* além de realizar modelagem geoestatísticas, tem uma série de funções para visualização, análise descritiva e simulação de dados. Na parte da modelagem ele é capaz tanto de estimar os parâmetros espaciais quanto realizar predições, as distribuições a priori para os parâmetros espaciais presentes neste pacote são dadas pela seguinte tabela.

Tabela 1: Prioris disponíveis no pacote geoRglm.

Parâmetro	Prioris
β	*Normal*
σ^2	*Uniforme,* χ^2 *inversa com parâmetro de escala, Recíproca*
ϕ	*Uniforme, Exponencial, Recíproca e* $Recíproca^2$

Já a lista de famílias de função de correlação espacial presentes no pacote é muito extensa e pode ser consultada no *cran-R*[11].

O ajuste de modelos no *geoRglm*, e também no *geoCount*, têm um grande complicador, as taxas de aceitação de S e de ϕ que, segundo *Christensen e Ribeiro*[17], devem ser ajustadas em aproximadamente 0.6 e 0.25, respectivamente, e dependem de parâmetros que devem ser informados a priori. A relação entre estas taxas e parâmetros é desconhecida, muita instável e carece de informações tanto na literatura quanto na página de ajuda dos pacotes computacionais. Um aspecto interessante da modelagem no *geoRglm* é que, através da discretização da distribuição a priori do parâmetro ϕ, ele acaba sendo muito mais rápido que os outros dois pacotes abordados neste trabalho.

O pacote *geoCount*

O *geoCount***,** proposto por *Jing*[17], tem como finalidade a análise de modelos geoestatísticos generalizados, para isso o autor desenvolveu uma variação do algoritmo proposto por *Christensen e Ribeiro*[7] que diminui a autocorrelação dos parâmetros. Ao contrário do *geoRglm*, este pacote possibilita que sejam geradas cadeias paralelas no algoritmo de MCMC.

O *geoCount* possui funções para modelagem geoestatística, visualização e simulação de dados georreferenciados. Em relação ao *geoRglm* ele possui uma quantidade menor de famílias de correlação espacial disponíveis, pois neste pacote só estão presentes as funções da família Matérn, Exponencial e Esférica. As distribuições a priori disponíveis estão na Tabela 2.

Tabela 2: Prioris disponíveis no pacote geoCount.

Parâmetros	Prioris
β	*Normal não informativa*
σ	*Halft, Gama Inversa e Recíproca*
ϕ	*Uniforme*

Note que, este pacote estima σ e não σ^2.

Em relação às taxas de aceitação citadas na subseção do pacote *geoRglm*, este pacote tem uma modelagem ainda mais complexa. Pois é necessário informar cinco parâmetros para o ajuste das taxas de aceitação para que quatro taxas sejam ajustadas em valores próximos do ideal. Segundo *Jing*[4], os valores ideais para as taxas são 0.575 para as duas primeiras, correspondentes às taxas de aceitação para S e β, e 0.25 para as duas últimas, que são as taxas de σ e ϕ.

O pacote *rstan*

O pacote *rstan* é na verdade uma interface em *R* do software *Stan*. Ao contrário dos pacotes citados anteriormente, o *rstan* é uma ferramenta que não foi desenvolvida especialmente para a modelagem geoestatística e sim para inferência bayesiana em geral, os algoritmos de MCMC disponíveis neste software são o *Hamiltonean Monte Carlo* e o *No-U-Turn*. Ainda pouco explorado na área de interesse deste estudo o *rstan* surge como uma ferramenta com potencial para a modelagem geoestatística, pois ele é muito mais flexível no que se diz respeito tanto às distribuições a priori quanto à estrutura do modelo.

2.12. Simulações

Para avaliar a capacidade de estimação de cada um dos algoritmos de MCMC aqui mencionados, será feito um experimento com base em simulação. Neste experimento será simulada uma amostra no seguinte cenário: $n = 100$, $\beta = \exp\{1.61\}$, $\sigma^2 = 0.7$, $\phi = 0.3$ e considerando que a função de correlação do processo subjacente é uma função de correlação

exponencial. O objetivo é verificar qual pacote computacional tem um melhor desempenho na estimação dos parâmetros do modelo Poisson log-linear.

A malha amostral destas simulações considera que tanto a coordenada X quanto a coordenada Y estão no intervalo de 0 a 1 e igualmente espaçadas, *Figura 8*.

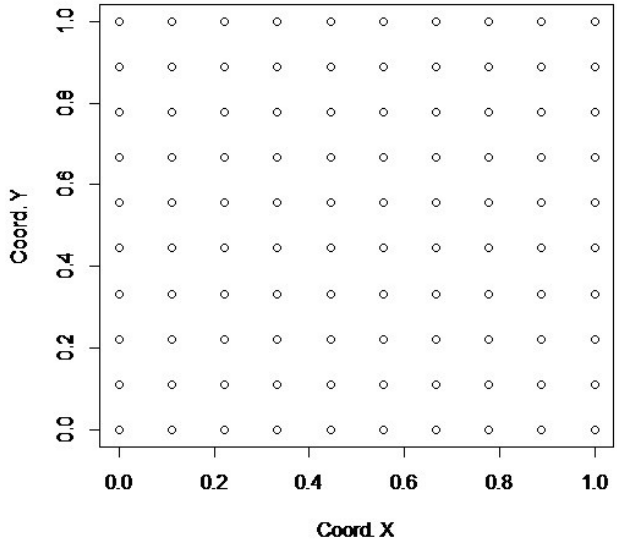

Figura 8 - Malha amostral simulada.

Para simular uma variável que siga uma distribuição Poisson e tenha um processo estocástico subjacente responsável pela estrutura espacial, necessitamos simular primeiro o processo subjacente e, posteriormente, através das realizações do processo simuladas, a variável de contagem.

2.12.1. Simulação do Processo Subjacente

Existem diversas metodologias para a simulação de um processo gaussiano, as mais populares são técnicas baseadas em manipular a função de autocovariância do processo, tais como decomposição espectral e decomposição de cholesky(*ver Diggle e Ribeiro*[9]). Portanto, considerando que o processo é espacialmente distribuído, estacionário e isotrópico, os passos para simulação são:

1) Simular uma malha amostral, isto é, definir os pares de coordenadas (x, y);
2) Definir a função de covariância espacial e seus parâmetros;
3) Definida a função de covariância espacial, devemos calcular a matriz de covariâncias populacional deste processo;
4) Definir a média μ da variável de interesse, note que aqui consideraremos que esta variável tem média constante, mas a metodologia no caso em que existe tendência é semelhante;

5) Simular n realizações independentes e com distribuição Normal com média 0 e variância 1 que chamaremos de Z;

Neste passo temos duas opções, que são ilustradas a seguir.

Método da Decomposição Espectral

Neste método primeiro devemos resolver a o sistema de equações

$$\mathbf{det}(\boldsymbol{\Sigma} - \boldsymbol{I\lambda}) = 0, \tag{4}$$

onde,

$\boldsymbol{\Sigma} = \sigma^2\rho$ é a matriz de covariâncias $n \times n$;

$\boldsymbol{I}$ é a matriz identidade $n \times n$;

$\boldsymbol{\lambda}$ é um vetor com n elementes, conhecidos como autovalores.

Após revolver o sistema de equações da *Equação 4* e obter os autovalores deve se encontrar os autovetores tais que

$$\boldsymbol{\Sigma}(1 - \boldsymbol{I\lambda})\boldsymbol{e} = 0.$$

Depois de obtidos os autovalores λ_i e autovetores e_i da matriz de covariâncias do processo, é necessário calcular

$$\boldsymbol{A} = \boldsymbol{U\Lambda^{1/2}}, \tag{5}$$

onde, $\boldsymbol{U}$ é uma matriz $n \times n$ que contém os autovetores e $\boldsymbol{\Lambda^{1/2}}$ é uma matriz diagonal $n \times n$ em que os elementos da diagonal principal são $\sqrt{\lambda_i}$.

Finalmente, para obter a simulação para S tem-se que

$$\boldsymbol{S} = \boldsymbol{AZ},$$

onde, $\boldsymbol{A}$ está definida na *Equação 5* e Z possui distribuição $N(0, 1)$.

Método da Decomposição de Cholesky

Neste método a única diferença em relação ao anterior é o modo de obtenção da matriz $\boldsymbol{A}$. Aqui, para obtê-la, considera-se que

$$\boldsymbol{\Sigma} = \mathbf{LL}',$$

onde, $\boldsymbol{L}$ é uma matriz triangular inferior. Então consideramos que $\boldsymbol{A} = \boldsymbol{L}$, e repetimos o último passo do método da Decomposição Espectral.

2.12.2. Simulação da Variável Resposta

Depois de feita a simulação das realizações do Processo Gaussiano Subjacente, a simulação da variável resposta se torna algo muito simples. Para isto basta simular, em cada local $x_i, i = 1, \dots, n$, a realização de uma variável aleatória Y_i com a seguinte distribuição:

$$Y_i \sim Poisson\left(e^{\beta + S_i}\right),$$

Onde e é a inversa da função link e $S_i = S(x_i)$ é a simulação da realização do processo S na região $i \in \{1, \dots, n\}$.

2.13. Ajuste dos Modelos

Com a finalidade de explorar os pontos fortes de cada pacote computacional, o número de *iterações*, o *burn-in* e o *lag* utilizados em cada um deles foi diferente. Pois o objetivo principal aqui é ajustar um modelo parcimonioso para cada pacote e expor as dificuldades e vantagens de cada um

Nesta etapa os maiores desafios são: encontrar um *burn-in* e um *lag* que levem as cadeias à convergência com a menor autocorrelação dos parâmetros possível e, para o *geoRglm* e *geoCount*, ajustar as taxas de aceitação.

Para que haja um critério objetivo para convergência das cadeias foi usado o Diagnóstico de Convergência de *Heidelberger e Welch*[6]. Este diagnóstico de convergência é composto por dois testes: O primeiro é baseado na estatística Cramer-von-Misses para testar a hipótese nula de que os valores amostrados vêm de uma distribuição estacionária; caso a hipótese nula seja rejeitada deve-se obter mais simulações. No caso em que a hipótese nula é aceita, o teste informa, se necessário, a partir de qual iteração os dados devem ser considerados; Já o segundo teste chamado de *Half-width test* calcula um intervalo de confiança(95%) para a média, usando apenas a porção das iterações que passaram no teste de estacionariedade, e então metade do comprimento deste intervalo é comparado com a estimativa da média. Se a razão da "meia-medida" com a média for maior que um erro especificado, então o teste conclui que a amostra não tem um tamanho suficiente para estimar a média com uma boa acurácia.

Será usada a moda para a estimação pontual dos parâmetros e, se a função de distribuição a posteriori for simétrica, serão usados os percentis *2.5%* e *97.5%* para estimar o intervalo de credibilidade. Caso contrário, se a distribuição a posteriori for assimétrica, será usado o *Highest Posterior Density (HPD)*.

3. Resultados

3.1. Análise descritiva dos dados simulados

Os dados foram simulados conforme a metodologia descrita nas seções anteriores. Para a simulação dos dados foi usada a função *simData* do pacote *geoCount*, que usa a Decomposição de Cholesky. A seguir serão apresentadas medidas de resumo e gráficos que representem os dados simulados para que se possa fazer uma análise descritiva.

Na Tabela 3 já temos informações interessantes para a etapa de modelagem, tais como um intervalo para o parâmetro ϕ, que controla a dependência espacial e conceitualmente não pode ser maior que a distância máxima observada na amostra, e uma estimativa inicial para o parâmetro β, que pode ser obtido através do logaritmo neperiano da média da variável resposta, também é importante notar que temos um problema de *overdispersion*, pois a variância é maior que a média.

Tabela 3: Resumo da amostra simulada

Resumo das Distâncias		**Resumo da Variável Resposta**	
Mínima	**Máxima**	**Média**	**Variância**
0.1111	1.4142	7.82	38.15

Outra etapa importante é a verificação de *outliers*, que não deve ser feita apenas com uma análise dos gráficos tipo boxplot, pois quando se trata de modelagem de dados espacializados os *outliers* são tratados de uma maneira diferente. Ou seja, caso haja um valor extremo na nossa amostra, mas com valores relativamente altos na sua vizinhança este valor não é considerado um *outlier*.

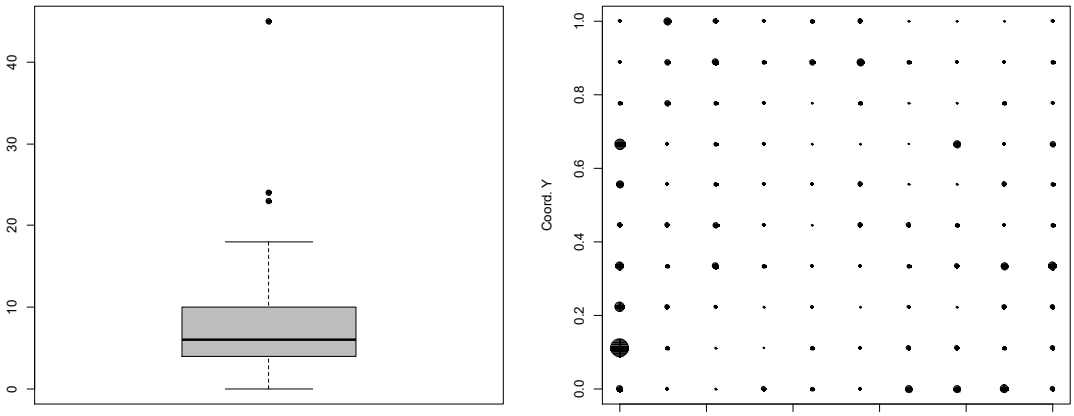

Figura 9 - Boxplot e dstribuição espacial da variável resposta.

Na *Figura 9*, através do boxplot identificaríamos alguns *outliers* na nossa amostra, no entanto com o gráfico de dispersão à direita, onde os tamanhos dos círculos correspondem à grandeza das medidas na variável resposta, podemos ver que, como os valores extremos estão próximos de outros valores altos, estes não devem ser considerados *outliers*. Mais ainda, este comportamento indica a existência de uma dependência espacial no processo subjacente.

Usando o semivariograma da variável resposta(*Figura 10*) reforça-se a hipótese da existência de uma dependência espacial, também pode-se identificar a possível existência de um efeito pepita, o qual pode ter sido causado pelo problema de *overdispersion*.

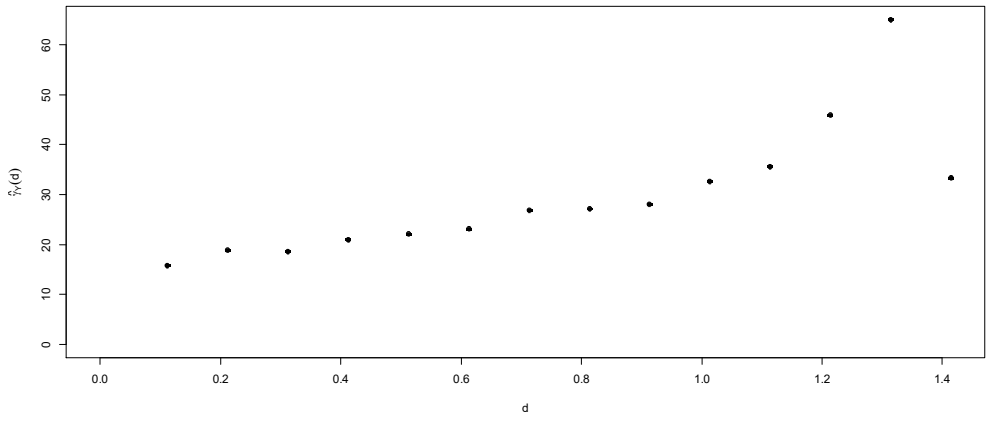

Figura 10: Semivariograma da variável resposta.

3.2. Resultados do pacote geoRglm

Neste pacote foi utilizado um *burn-in = 200000* e um *lag = 800* para gerar amostras de tamanho *1500*. Mesmo que seja necessário a simulação de um número muito grande de amostras o algoritmo utilizado se mostrou muito rápido, no entanto, como podemos ver na *Figura 11*, nem o *lag = 800* foi capaz de reduzir a autocorrelação dos parâmetros.

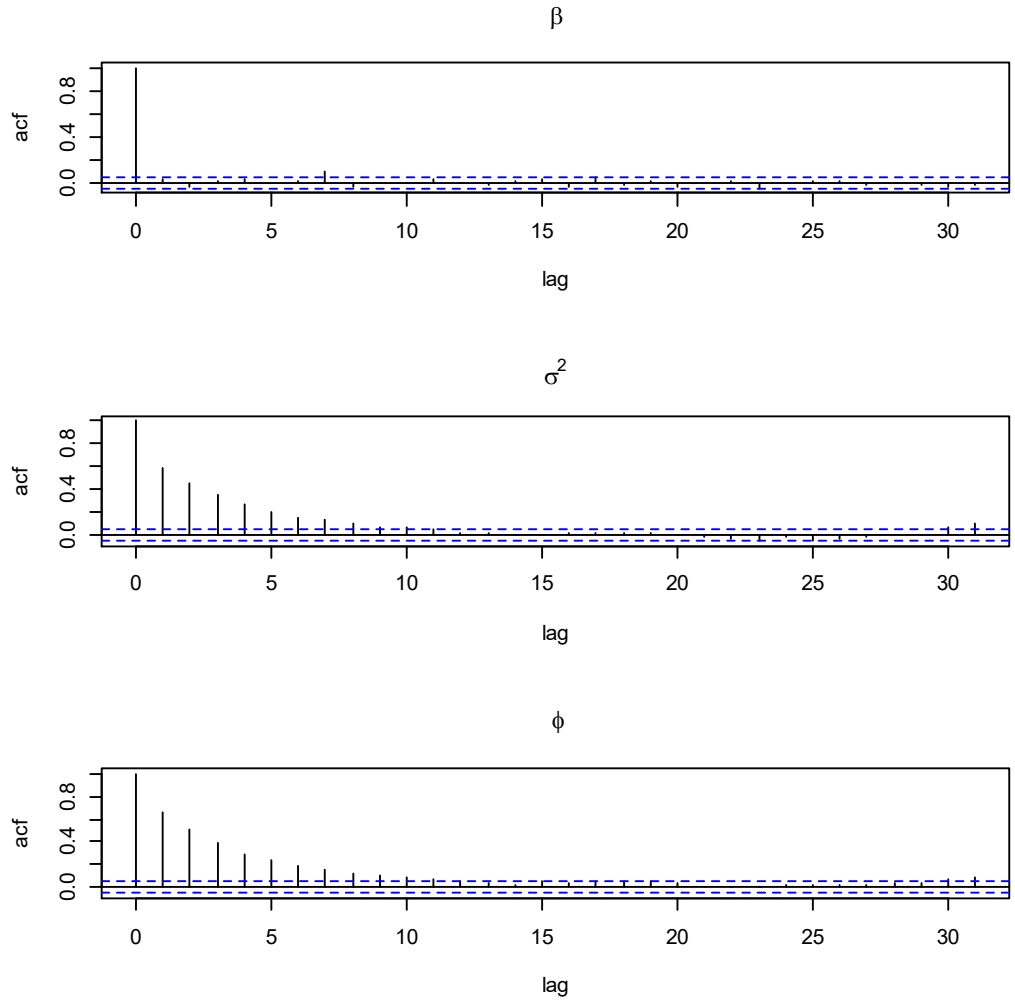

Figura 11 - Autocorrelação dos parâmetros.

Contudo, apenas a autocorrelação dos parâmetros foi insatisfatória neste pacote, pois, além de rápido, ele passou nos diagnósticos de convergência (ver Tabela 4 e *Figura 12*) e obteve boas estimativas para os verdadeiros parâmetros dos dados com uma baixa variabilidade, ver Tabela 5, onde estão contidas as medidas de resumo das distribuições a posteriori dos parâmetros de interesse. Na simulação dos processos subjacentes o pacote se mostrou eficiente tanto na questão da convergência, quanto na questão da autocorrelaçao (*Figura 13*).

Tabela 4: Diagnóstico de convergência de Heidelberger e Welch.

Diagnósticos de Heidelberger e Welch		
Estacionariedade		**Acurácia da Média**
Parâmetro	**p-valor**	**Half-width(erro = 0,1)**
β	0.727	0.0268
σ^2	0.812	0.0461
ϕ	0.448	0.0259

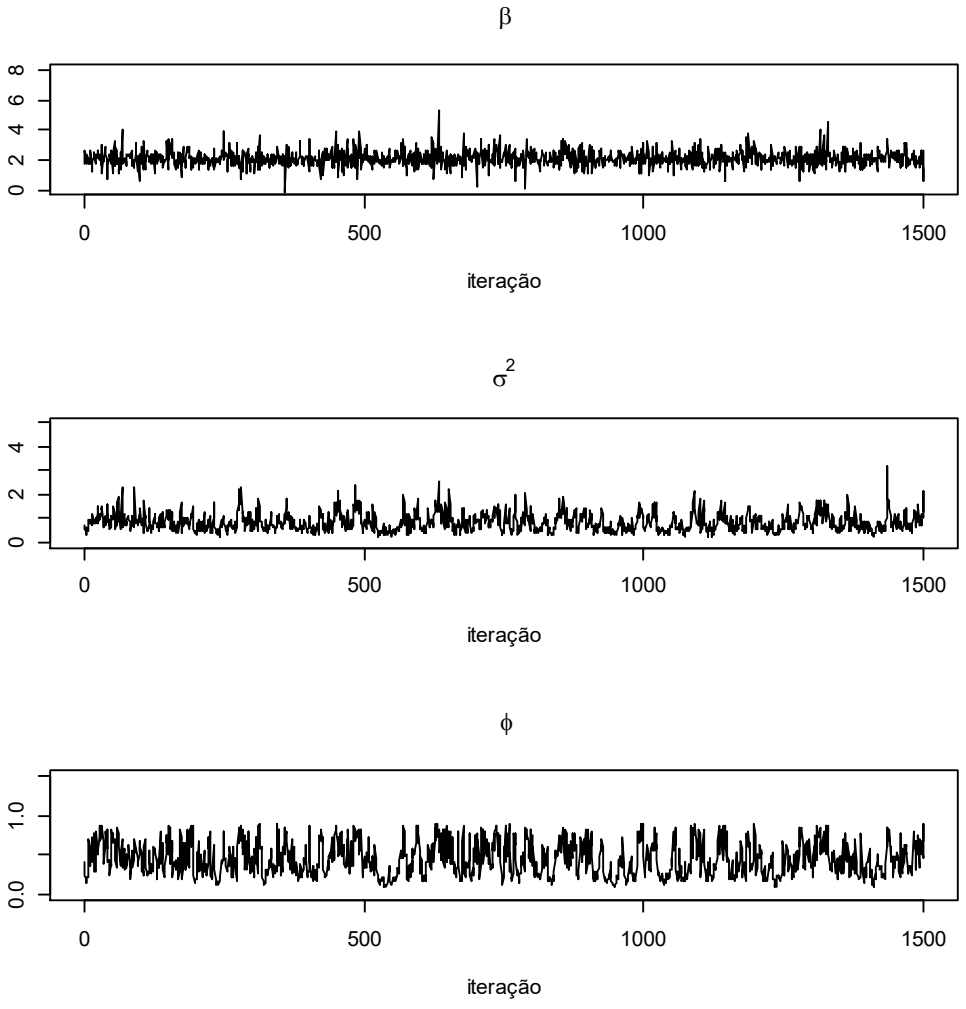

Figura 12 - Cadeias dos parâmetros.

Tabela 5: Medidas de resumo das distribuições a posteriori de cada parâmetro.

	$\boldsymbol{\beta}$	$\boldsymbol{\sigma^2}$	$\boldsymbol{\phi}$
Média	2.1149	0.8473	0.4501
Desvio-padrão	0.5034	0.3767	0.2027
Erro-padrão	0.0130	0.0097	0.0052
Moda	1.9837	0.6337	0.2482
2.5%	1.1709	0.3538	0.1425
Mediana	2.0818	0.7689	0.4175
97.5%	3.2245	1.7676	0.8526

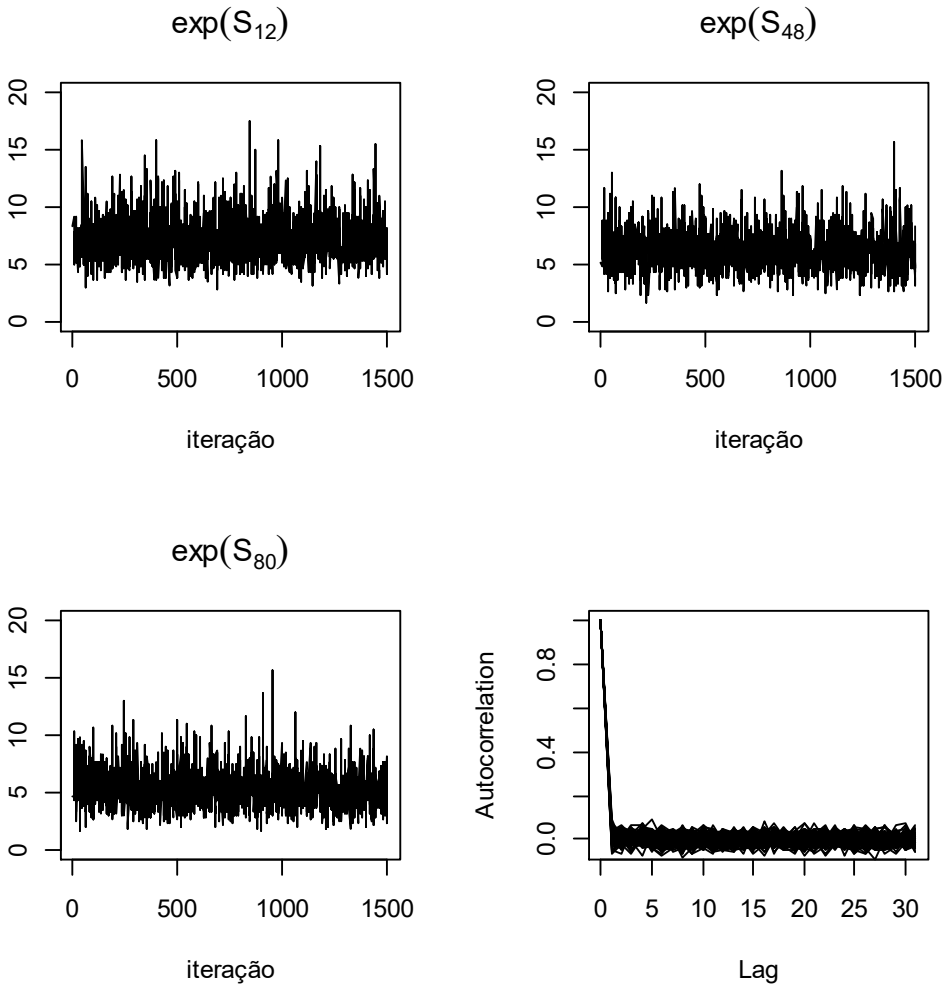

Figura 13 - Cadeias das realizações 12,48 e 80 do processo subjacente; Autocorrelação das realizações do processo subjacente.

3.3. Resultados do pacote geoCount

O *geoCount* foi o pacote mais difícil para se fazer a modelagem, pois a etapa de ajuste das taxas de aceitação é realmente muito instável. Estas taxas podem inclusive variar até com o número de simulações. Outro problema para as simulações de MCMC deste método é que ele é muito lento. Apesar das dificuldades na modelagem o pacote mostrou-se eficiente na estimação dos parâmetros, ver Tabela 6, onde estão contidas as informações das distribuições a posteriori dos parâmetros de interesse. Com um *burn-in* = 5000 e um *lag* = 10, o algoritmo conseguiu baixar a autocorrelação dos parâmetros (*Figura 14)* e atingir a convergência (*Figura 15)*, passando inclusive nos testes de diagnóstico, onde não foram rejeitadas as hipóteses de que as cadeias são estacionárias *(*Tabela 7).

Tabela 6: Resumo das distribuições a posteriori.

	$\boldsymbol{\beta}$	$\boldsymbol{\sigma^2}$	$\boldsymbol{\phi}$
Média	1.9543	1.5223	0.3124
Desvio-padrão	0.6412	1.0100	0.2274
Erro-padrão	0.0166	0.0261	0.0058
Moda	1.8907	0.8504	0.1386
2.5%	0.7537	0.5147	0.0747
Mediana	1.9042	1.1244	0.2221
97.5%	3.4342	4.2400	0.8450

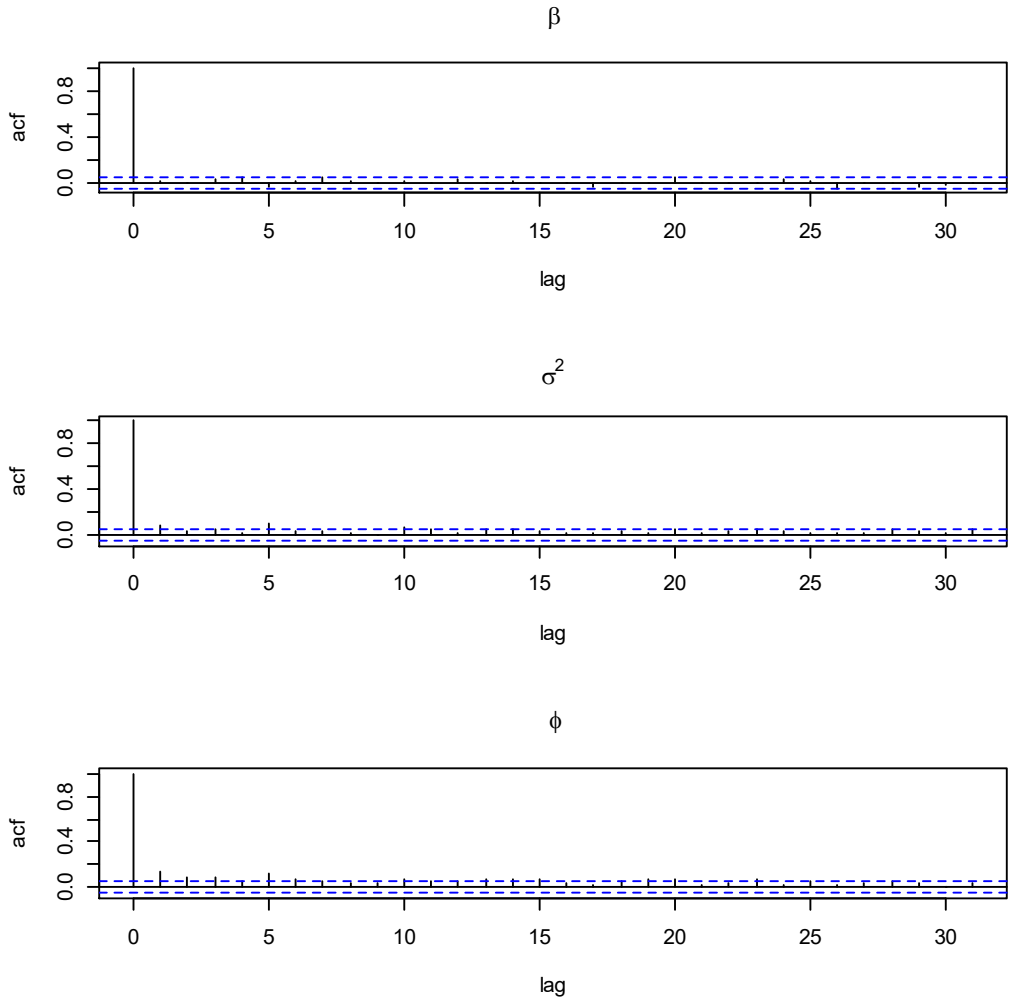

Figura 14 - Autocorrelação dos parâmetros.

Tabela 7: Diagnóstico de Heidelberger e Welch.

Diagnósticos de Heidelberger e Welch		
Estacionariedade		**Acurácia da Média**
Parâmetro	**p-valor**	**Half-width(erro = 0,1)**
β	0.4401	0.0661
σ^2	0.2458	0.0174
ϕ	0.1847	0.0259

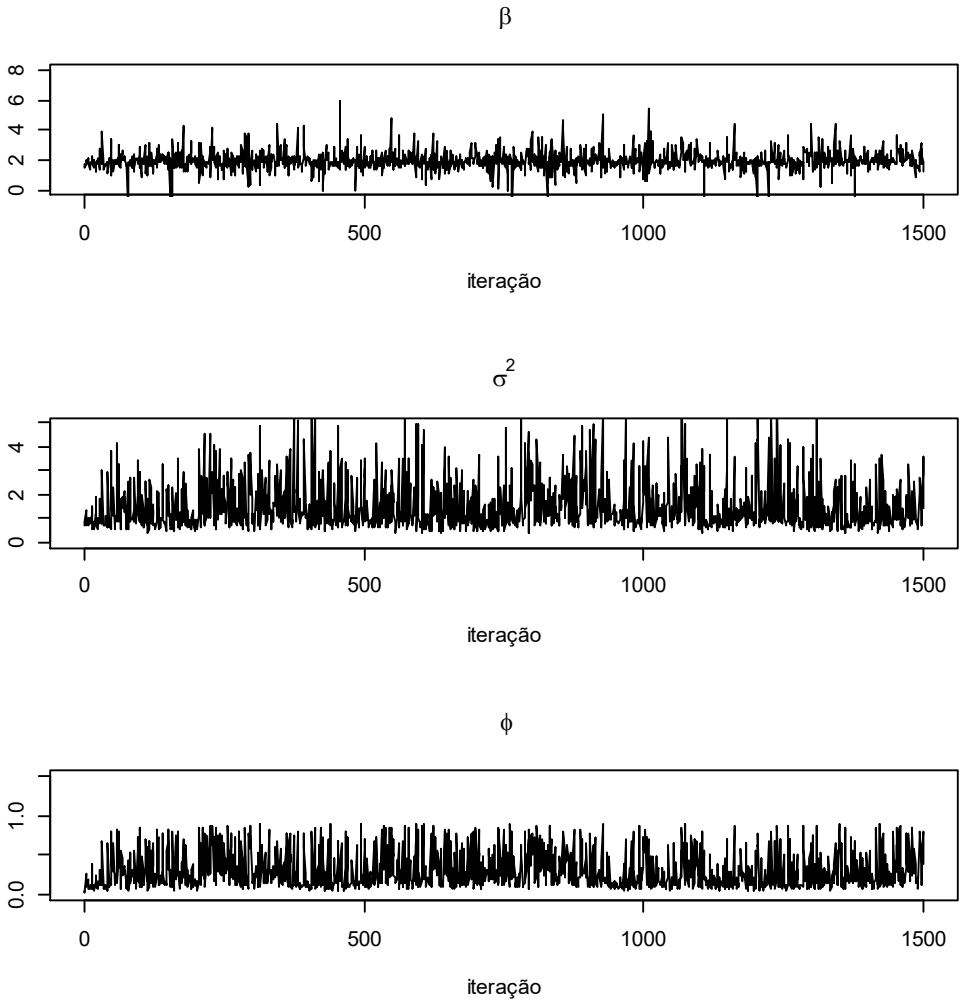

Figura 15 - Cadeias dos parâmetros.

Já na estimação das realizações do processo subjacente, o *geoCount* não se mostrou eficiente (ver *Figura 16*), pois além das cadeias não estarem convergindo a autocorrelação das realizações do processo é muito alta.

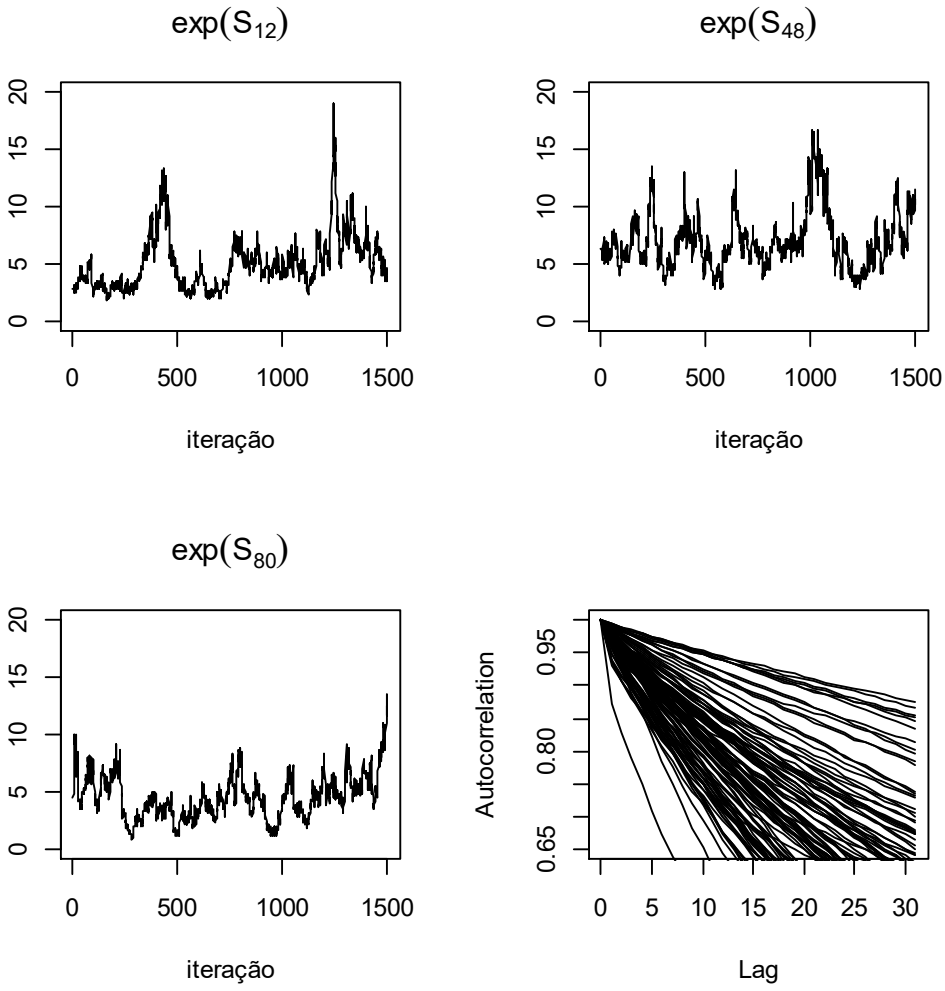

Figura 16 - Cadeias das realizações 12,48 e 80 do processo subjacente; Autocorrelação das realizações do processo subjacente.

3.4. Resultados do pacote rstan

Este pacote computacional teve resultados interessantes, pois ele foi capaz de reduzir a autocorrelação dos parâmetros (ver *Figura 17*) e do processo (ver *Figura 19*), bem como atingir a convergência (ver *Figura 18* e Tabela 8). Esta convergência, no caso das realizações do processo subjacente é discutível por existir uma alta variabilidade, com um *burn-in = 15000* e um *lag = 10*. Também se mostrou muito eficiente na estimação de parâmetros (ver Tabela 9), entretanto, por existir uma carência muito grande de trabalhos nesta área, este método precisa ser melhor investigado para que se tenha certeza da sua eficiência. Outro ponto é que pode ser difícil implementar as opções de predição neste pacote.

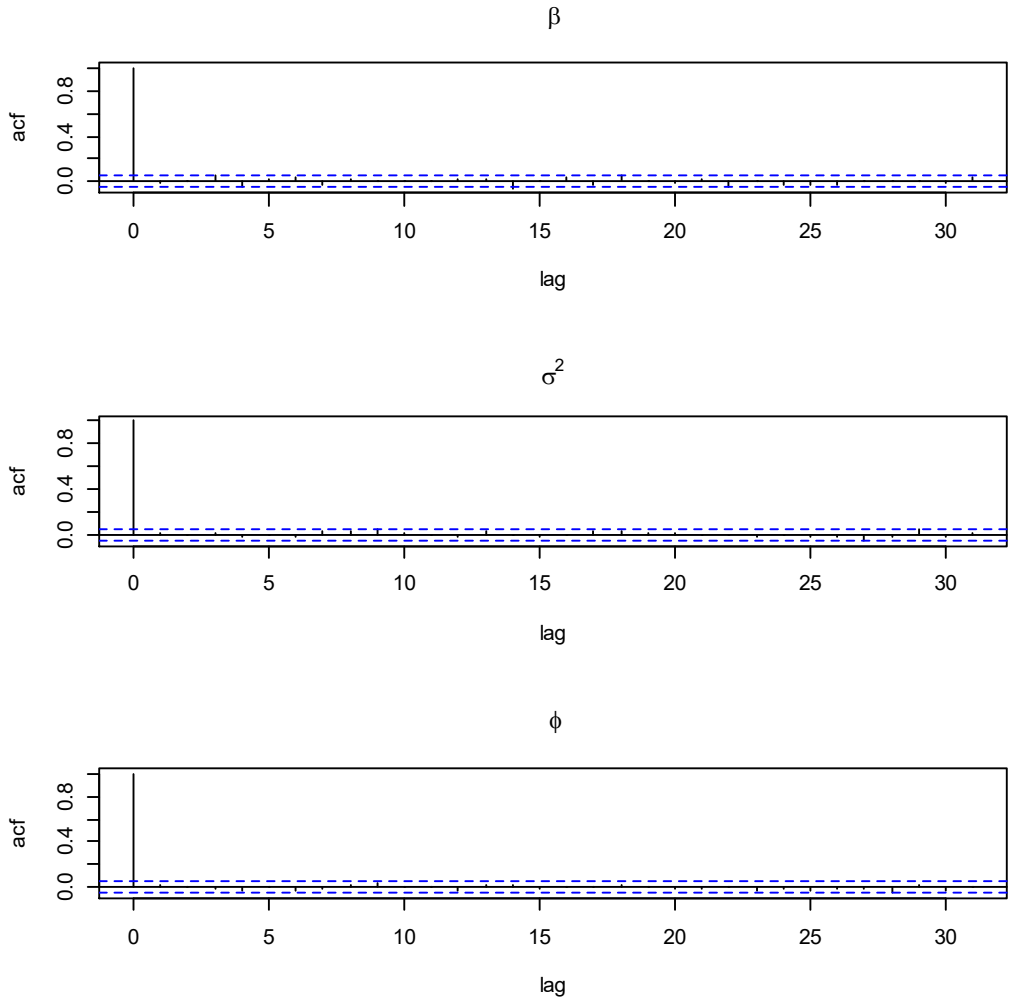

Figura 17 - Autocorrelação dos parâmetros.

Tabela 8: Diagnóstico de convergência de Heidelberger e Welch,

Diagnósticos de Heidelberger e Welch		
Estacionariedade		**Acurácia da Média**
Parâmetro	**p-valor**	**Half-width(erro = 0,1)**
β	0.3687	0.0225
σ^2	0.0637	0.0191
ϕ	0.6442	0.0107

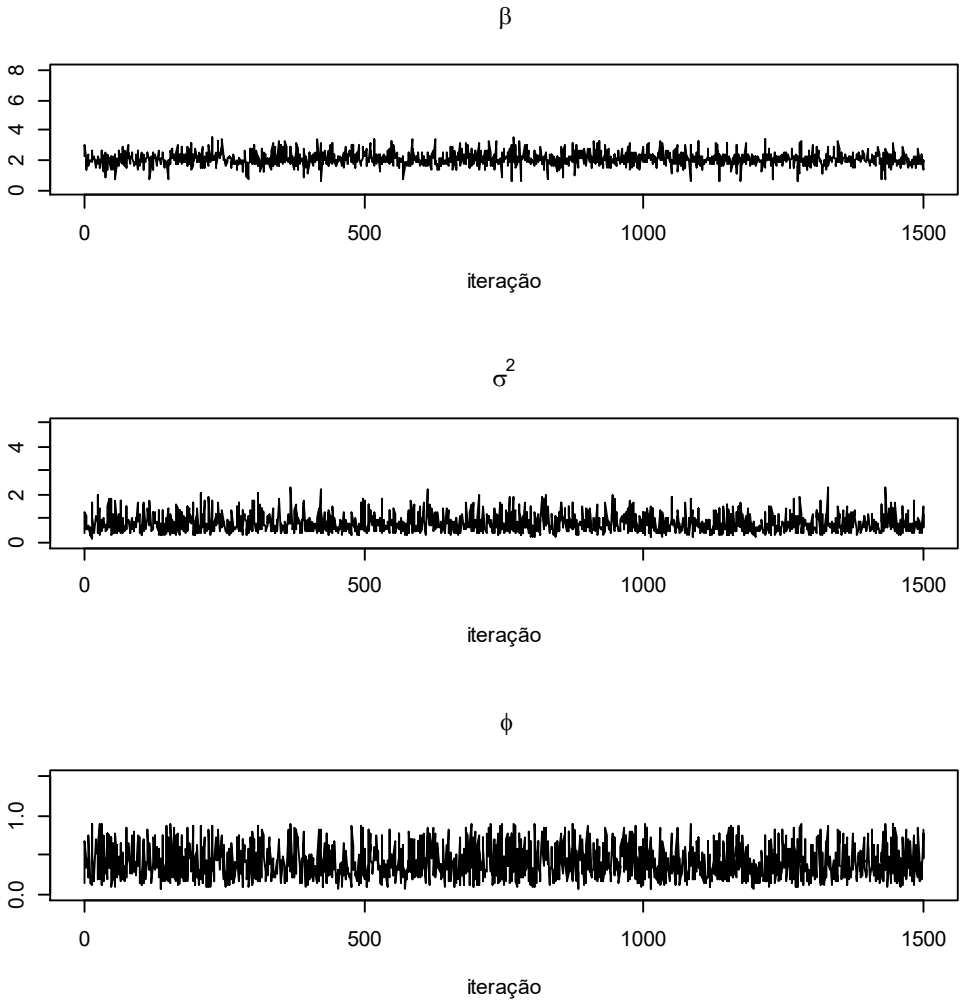

Figura 18 - Cadeias dos parâmetros.

Tabela 9: Resumo das distribuições a posteriori dos parâmetros.

	$\boldsymbol{\beta}$	$\boldsymbol{\sigma^2}$	$\boldsymbol{\phi}$
Média	2.0740	0.8002	0.4092
Desvio-padrão	0.4383	0.3776	0.2106
Erro-padrão	0.0113	0.0097	0.0054
Moda	1.9556	0.5058	0.2278
2.5%	1.3223	0.3350	0.1171
Mediana	2.0336	0.7071	0.3639
97.5%	3.0983	1.7324	0.8511

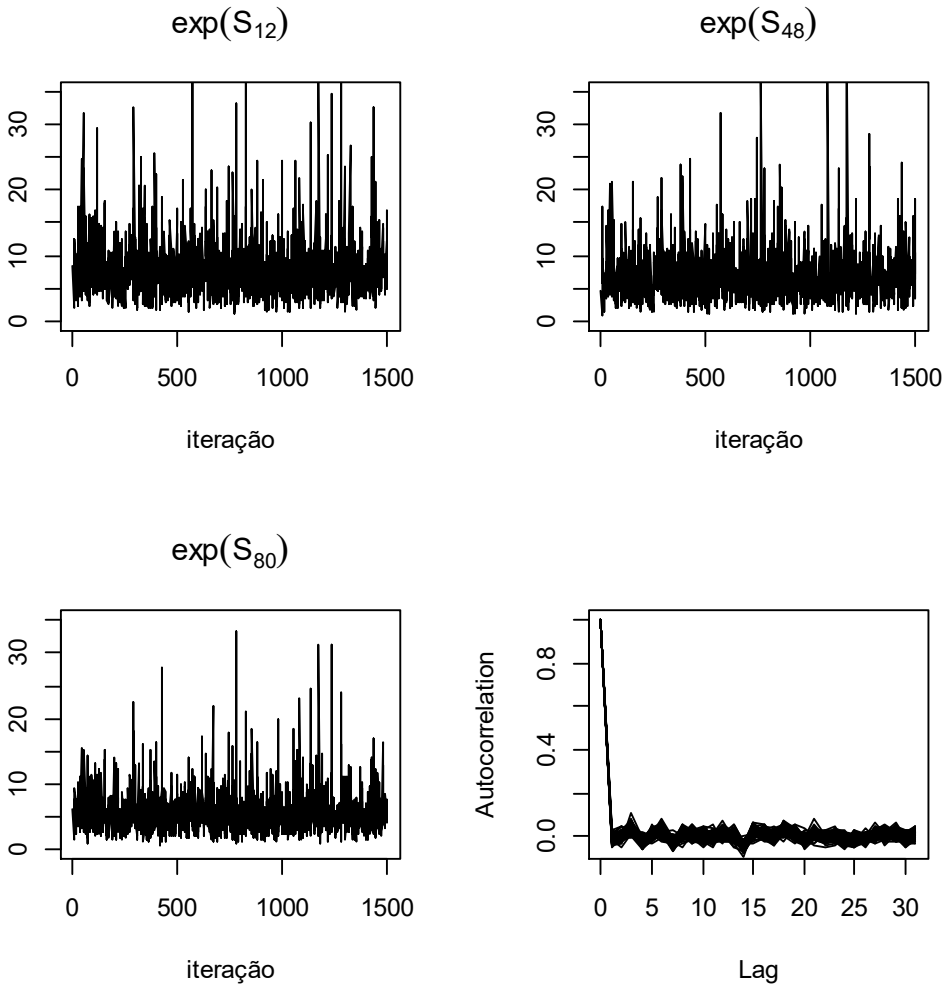

Figura 19 - Cadeias das realizações 12,48 e 80 do processo subjacente; Autocorrelação das realizações do processo subjacente.

3.5. Comparação dos resultados dos pacotes computacionais

Como todos os pacotes computacionais passaram no diagnóstico de convergência, pode-se supor que, neste quesito não existe a superioridade de algum dos três algoritmos em relação aos outros.

O algoritmo usado no pacote *geoRglm* se mostrou ineficaz na redução da autocorrelação dos parâmetros, enquanto o *NUTS*, algoritmo usado pelo *rstan*, apresentou resultados muito bons neste quesito. Na *Figura 20* pode ser vista a diferença da autocorrelação dos parâmetros de cada algoritmo.

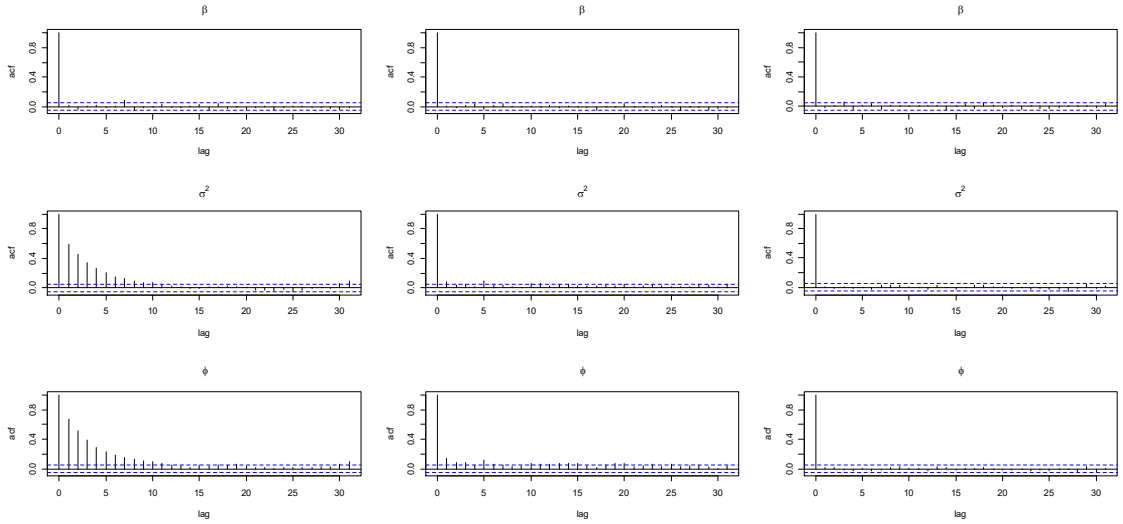

Figura 20 - À esquerda a autocorrelação dos parâmetros do algoritmo feito pelo pacote *geoRglm*, no centro do *geoCount* e, à direita, do *rstan*.

É desejável que o comportamento das cadeias e da autocorrelação das simulações do processo subjacente seja tão bom quanto o dos parâmetros. O algoritmo que se mostrou mais eficiente nesta questão foi o usado pelo *geoRglm*, enquanto os dois outros algoritmos se mostraram ineficientes para fazer com que as realizações do processo chegassem à convergência (*Figura 21)*. No caso do *geoCount*, também observou-se que o algoritmo não é capaz de reduzir a autocorrelação das realizações dos processos subjacentes, ver *Figura 22*.

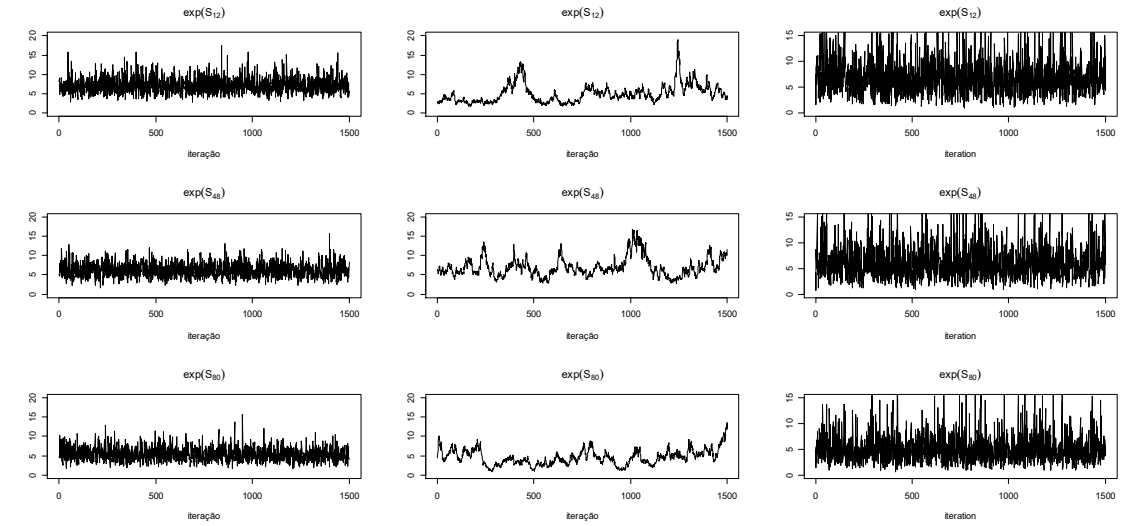

Figura 21 - Cadeias da exponencial das realizações 12,48 e 80 do processo subjacente. À esquerda o *geoRglm*, no centro o *geoCount* e à direita o *rstan*.

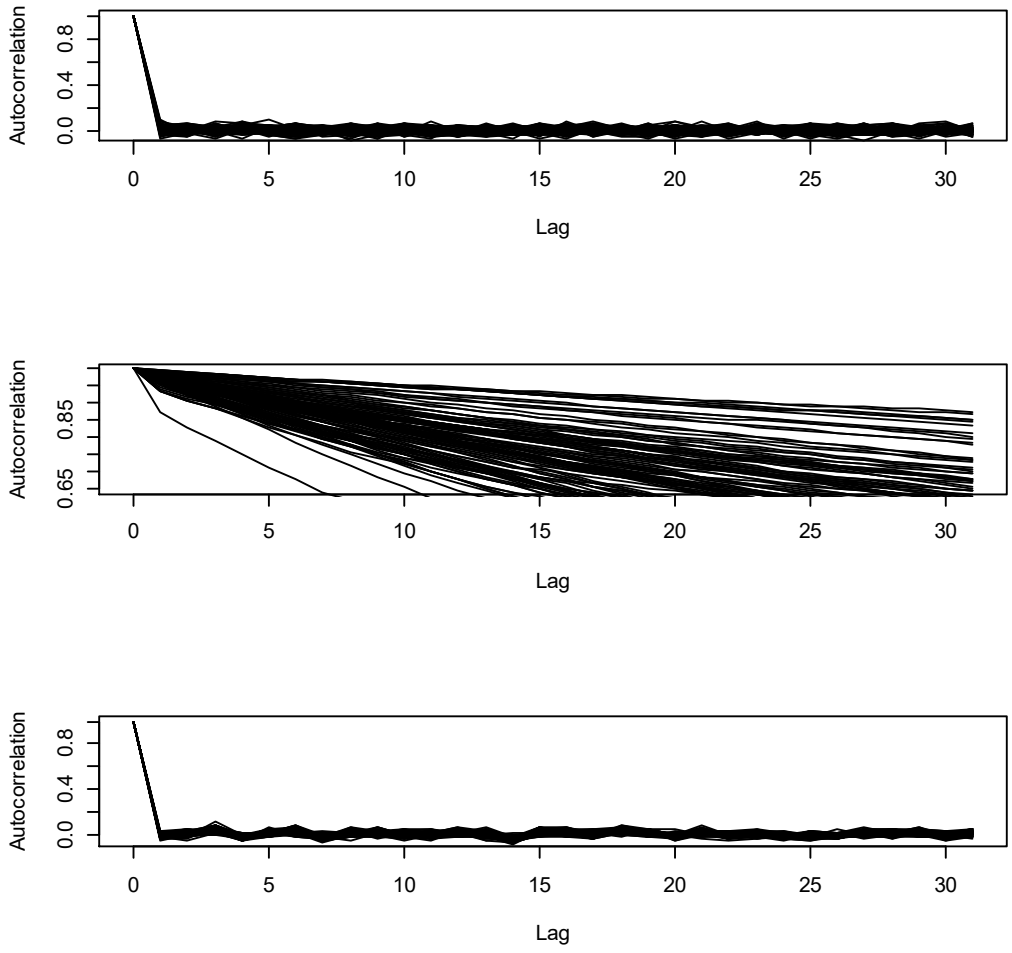

Figura 22 - Gráfico que resume a autocorrelação de todas as realizações dos processos subjacentes. Ordem (de cima para baixo): *geoRglm*, *geoCount* e *rstan*.

Outros dois aspectos que devem ser levados em conta são o tempo computacional de cada algoritmo e, obviamente, o quão bem este estimou os parâmetros para os quais desejávamos obter estimativas. Na *Tabela 10*, pode ser visto que o algoritmo usado *geoRglm* é muito mais rápido que os outros dois.

Tabela 10: Número de iterações e tempo decorrido de cada algoritmo.

Pacote	iterações	Tempo(em minutos)
geoRglm	1400000	2.6748
geoCount	25000	4.6103
Rstan	30000	21.0465

Na Tabela 11 temos a moda da distribuição a posteriori de cada parâmetro em cada pacote computacional, bem como os intervalos de credibilidade, usando percentis para o parâmetro β, e o *Highest Posterior Density(HPD)* para os parâmetros σ^2 e ϕ, tendo em vista que ambos possuem distribuição a posteriori assimétrica.

Tabela 11: Estimativa dos parâmetros pela moda e por intervalo de credibilidade em cada pacote computacional.

		$\boldsymbol{\beta}$	
Pacote	**Moda**	**IC (95%)**	
geoRglm	1.9837	1.1709	3.2245
geoCount	1.8907	0.7537	3.4342
Rstan	1.9556	1.3223	3.0983
		$\boldsymbol{\sigma^2}$	
Pacote	**Moda**	**IC (95%)**	
geoRglm	0.6337	0.3097	1.6187
geoCount	0.8504	0.3691	3.5587
Rstan	0.5058	0.2633	1.5909
		$\boldsymbol{\phi}$	
Pacote	**Moda**	**IC (95%)**	
geoRglm	0.2482	0.1560	0.8590
geoCount	0.1386	0.0306	0.8051
Rstan	0.5058	0.0924	0.8169

Na Tabela 12 são computados o Vício (verdadeiro valor do parâmetro subtraído de sua estimativa) e a amplitude dos intervalos de credibilidade. Entende-se que o algoritmo que obtiver menor vício, em módulo, e também a menor amplitude do intervalo de credibilidade é mais eficiente.

Tabela 12: Vício dos Estimadores pontuais e amplitudes dos intervalos de confiança.

$\boldsymbol{\beta}$		
Pacote	**Vício**	**Amplitude – IC**
geoRglm	0.3737	2.0536
geoCont	0.2807	2.6805
rstan	0.3456	1.7760
$\boldsymbol{\sigma^2}$		
Pacote	**Vício**	**Amplitude – IC**
geoRglm	0.1163	1.3100
geoCont	-0.1004	3.1896
rstan	0.2442	1.3277
$\boldsymbol{\phi}$		
Pacote	**Vício**	**Amplitude – IC**
geoRglm	-0.0518	0.7030
geoCont	-0.1614	0.7745
rstan	0.2058	0.7245

4. Discussão

Ao final deste estudo parece difícil apontar um dos pacotes computacionais utilizados como o melhor, entretanto ficou evidente que eles são bem diferentes.

O algoritmo utilizado pelo *geoRglm* teve como trunfo a velocidade, a baixa amplitude dos intervalos de credibilidade, o vício pequeno para o parâmetro responsável pela dependência espacial e também a convergência e baixa autocorrelação das simulações das realizações do processo subjacente. Seu ponto fraco foi a dificuldade em conseguir diminuir a autocorrelação dos parâmetros.

Já o algoritmo utilizado pelo *geoCount*, que trata de uma reparametrização do algoritmo do *geoRglm*, não tem nenhum ganho expressivo em relação aos outros algoritmos. Além de ser muito difícil ajustar suas taxas de aceitação, o algoritmo se mostrou menos eficiente que os outros na estimação dos parâmetros, por conta da alta amplitude do seus intervalos de credibilidade. Apesar das cadeias dos parâmetros terem atingido a convergência e terem baixa autocorrelação o método utilizado por este algoritmo não mostrou essa eficiência nas simulações das realizações do processo subjacente, pois, após uma análise gráfica, estas não parecem convergir e tem uma autocorrelação muito alta se comparada a dos outros pacotes.

O método utilizado pelo *rstan* inicialmente apresentou resultados animadores, pois além de ser um método que não apresenta o mesmo problema com as taxas de aceitação, se mostrou eficiente na questão da convergência das cadeias e na ausência de autocorrelação dos parâmetros, assim como na ausência de autocorrelação nas simulações das realizações do processo subjacente. No entanto, as cadeias destas simulações aparentam ter alta variabilidade e ser muito instáveis. Outro ponto fraco deste algoritmo foi que ele é o mais demorado, levando aproximadamente 21 minutos para obter 30000 iterações. A estimação dos parâmetros também foi outro aspecto em que este algoritmo deixou a desejar (ver Tabela 12), pois, com exceção da estimativa para β, foi o algoritmo que apresento os maiores valores para os vícios dos estimadores. Vale ressaltar que este algoritmo é muito novo, tem muito potencial e é um bom objeto de estudo para trabalhos futuros.

Referências Bibliográficas

[1] Highfield, L.;Ward, M e Laffan, S. (2008). Representation of animal distributions in space: how geostatistical estimates impact simulation modeling of foot-and-mouth disease spread. Veterinary Research, BioMed Central, 39 (2), pp.1-14.

[2]Diggle P.J., Harper L. e Simon S. L. (1997). Geostatistical analysis of residual contamination from nuclear testing. In: Statistics for the environment 3: pollution assesment and control (eds. V. Barnet and K. F. Turkmann), Wiley, Chichester, 89-107.

[3] Guillot G., Lor´en N. e Rudemo M. (2009). "Spatial Prediction of Weed Intensities from Exact Count Data and Image-Based Estimates." Journal of the Royal Statistical Society C, 58, 525–542.

[4] Jing L. (2011). "Bayesian Model Checking for Generalized Linear Spatial Models for Count Data." Unpublished Ph.D. Dissertation, The University of Texas at San Antonio.

[5] Hoffman M.D. e Gelman A. (2011). "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". arXiv 1111:4246.

[6] Heidelberger P. e Welch PD (1981)." A spectral method for confidence interval generation and run length control in simulations". Comm. ACM. 24, 233-245.

[7] Christensen O.F. e Ribeiro P.J. (2002). "geoRglm: A Package for Generalized Linear Spatial Models." R News, 2(2), 26–28.

[8] Cressie N. (1993). Statistics for Spatial Data. Revised edition. John Wiley & Sons, New York.

[9] Diggle P.J. e Ribeiro P.J. (2007). Model-Based Geostatistics. Springer-Verlag, New York.

[10] Diggle P.J., Tawn J.A. e Moyeed R.A. (1998). "Model-Based Geostatistics." Journal of the Royal Statistical Society C, 47, 299–326.

[11] R Core Team (2016). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

[12] Ribeiro P.J. e Diggle P.J. (2001). “geoR: A Package For Geostatistical Analysis.” R News, 1(2), 15–18.

[13] Breslow N.E. e Clayton D.G. (1993). Approximate inference in generalized linear mixed models, Journal of the American Statistical Association 88: 9–25.

[14] Stan Development Team (2016). Stan Modeling Language Users Guide and Reference Manual, Version 2.9.0.

[15] Stan Development Team (2016). RStan: the R interface to Stan, Version 2.9.0.

[16] Plummer M., Best N., Cowles K. e Vines k. (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC, R News, vol 6, 7-11.

[17] Jing L. e De Oliveira V. (2015). geoCount: An R Package for the Analysis of Geostatistical Count Data. Journal of Statistical Software, 63(11), 1-33.

[18] Gamerman D. e Lopes H. (2006). MCMC – Stochastic Simulation for Bayesian Inference. Chapman & Hall/CRC.

[19] Gelman A., Carlin J. B., Stern H. S. e Rubin D.B. (2003). Bayesian Data Analysis, second edition, Chapman and Hall, London.

[20] Hoffman M. D. e Gelman A. (2011). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. arXiv, 1111.4246.

[21] Nelder J. e Wedderburn R.(1972). "Generalized Linear Models". Journal of the Royal Statistical Society. Series A (General) (Blackwell Publishing) 135 (3): 370–384.

[22] Warnes J.J. e Ripley B.D. (1987). Problems with likelihood estimation of covariance functions of spatial Gaussian processes, Biometrika 74: 640–642.

[23] Mardia K.V. e Watkins A.J. (1989). On multimodality of the likelihood in the spatial linear model, Biometrika 76: 289–296.

ANEXO

Programa em R

```
############################################################
##          Gráficos das Estruturas de dependência espacial          ##
############################################################

library(geoR)

h = seq(0,1,l = 100)

#################################
##        Correlação Espacial          ##
#################################

#####################################
##        Familia Matern - Figura 1          ##
#####################################

rho = cov.spatial(h, cov.model= "matern",
        cov.pars=c(2,0.25),kappa = 0.5)/2

rho2 = cov.spatial(h, cov.model= "matern",
        cov.pars=c(2,0.16),kappa = 1.5)/2

rho3 = cov.spatial(h, cov.model= "matern",
        cov.pars=c(2,0.13),kappa = 2.5)/2


plot(h, rho, type = "l", ylab = expression(rho(d)),
xlab = "Distância", xlim = c(0,1))

lines(h, rho2, type = "l", lty = 2)

lines(h, rho3, type = "l", lty = 3)

legend("topright",c(expression(paste(phi," = 0.15"," , ", kappa," = 0.5")),
expression(paste(phi," = 0.25"," , ", kappa," = 0.5")),
expression(paste(phi," = 0.15"," , ", kappa," = 2"))) ,lwd = rep(1,4), lty = c(1,2,3), bty = "n")
```

```
################################################
##       Família Exponencial Potencia - Figura 2          ##
################################################

rho = cov.spatial(h, cov.model= "powered.exponential",
          cov.pars=c(2,0.16),kappa = 0.7)/2

rho2 = cov.spatial(h, cov.model= "powered.exponential",
          cov.pars=c(2,0.25),kappa = 1)/2

rho3 = cov.spatial(h, cov.model= "powered.exponential",
          cov.pars=c(2,0.43),kappa = 2)/2


plot(h, rho, type = "l", ylab = expression(rho(d)),
xlab = "Distância", xlim = c(0,1))

lines(h, rho2, type = "l", lty = 2)

lines(h, rho3, type = "l", lty = 3)

legend("topright",c(expression(paste(phi," = 0.16"," , ", kappa," = 0.7")),
expression(paste(phi," = 0.25"," , ", kappa," = 1")),
expression(paste(phi," = 0.43"," , ", kappa," = 2"))) ,lwd = rep(1,4), lty = c(1,2,3), bty = "n")

########################################
##        Família Esférica - Figura 3          ##
########################################

rho = cov.spatial(h, cov.model= "spherical",
          cov.pars=c(2,0.35))/2

rho2 = cov.spatial(h, cov.model= "matern",
          cov.pars=c(2,0.25))/2

rho3 = cov.spatial(h, cov.model= "matern",
          cov.pars=c(2,0.15))/2
```

```
plot(h, rho, type = "l", ylab = expression(rho(d)),
xlab = "Distância", xlim = c(0,1))

lines(h, rho2, type = "l", lty = 2)

lines(h, rho3, type = "l", lty = 3)

legend("topright",c(expression(paste(phi," = 0.35")),expression(paste(phi," = 0.25")),
expression(paste(phi," = 0.15"))),lwd = 1, lty = c(1,2,3), bty = "n")

############################################################
##          Função de Covariância Espacial - Matern - Figura 4          ##
############################################################

sigma2 = 2

C = sigma2*rho2

plot(h, C, type = "l", lwd = 2, ylab = expression(C(d)),
xlab = "Distância", xlim = c(0,1))

legend("topright",expression(paste(phi," = 0.16"," , ", kappa," = 1.5", " , ", sigma^2, " = 2")), bty = "n")

##############################################
##          Semivariograma - Matern - Figura 5          ##
##############################################

svar <- sigma2*(1-rho2)

plot(h, svar, type = "l", lwd = 2, ylab = expression(gamma(d)),
xlab = "Distância", xlim = c(0,1))

legend("bottomright",expression(paste(phi," = 0.16"," , ", kappa," = 1.5", " , ", sigma^2, " = 2")), bty = "n")


#####################################################
##          Semivariograma e covariancia - Figura 6          ##
#####################################################

svar <- sigma2*(1-rho2)
```

```
plot(h, svar, type = "l", lwd = 2, ylab = expression(gamma(d)),
xlab = "Distância", xlim = c(0,1))

lines(h, C, lwd = 2, lty = 3)

legend("right",c(expression(paste(gamma(d))),expression(paste(C(d)))),lwd = c(2,2),lty = c(1,3), bty = "n")

###################################
##        Efeito Pepita - Figura 7          ##
###################################

pepita = 0.5

variog = pepita + (sigma2)*(1-rho)

plot(h, variog, type = "l", lwd = 2, ylab = expression(gamma(h)),
xlab = "Distância", xlim = c(0,1),ylim = c(0,3))

lines(h, C, lty = 3)

legend("topright",expression(paste(tau^2,' =  0.5')) , bty = "n")

#########################################################
##        Fim da Seção "Estruturas de Dependência Espacial       ##
#########################################################

########################################
##        Carregando pacotes exigidos     ##
########################################

library(geoCount)
library(geoR)
library(geoRglm)
library(scatterplot3d)
library(coda)

#######################
##        Simulacao          ##
#######################
```

```
set.seed(1) ## Semente usada nas simulacoes

mi1 <- exp(1.61) ## beta = 1.61
sigmaqs1 <- 0.75
phi1 <- .3

## Funcao do pacote geoCount que gera coordenadas em um grid
## x = comprimento de x, nx = número de pontos em x
## y e ny análogos

sample_coords <- locGrid(x = 1,y = 1,nx = 10,ny = 10)

################################################
##            Malha Amostral Simulada - Figura 8         ##
################################################

plot(sample_coords, xlab = "coord x", ylab = "coord y")


## Simulando os dados a partir da funcao simData do pacote geoCount

y_c1 <- simData(sample_coords, L = 0, X = NULL, beta = 1.61,
cov.par = c(sqrt(sigmaqs1), phi1,1), rho.family = "rhoPowerExp",
Y.family = "Poisson")

## Esta funcao usa como default o metodo da decomposicao de choelsky.
## Na sua saida temos os dados de contagem
## e as realizacoes simuladas do processo subjacente

## Armazenando as saidas da funcao simData e as coordenadas em um unico
## dataframe.

dat_c1 <- data.frame(X = sample_coords[,1],Y = sample_coords[,2],
y = y_c1$data, S = y_c1$latent)

######################
##   Fim da simulacao. ##
######################
```

```
############################
##        Analise Descritiva  ##
############################


dat_geo <- as.geodata(dat_c1, coords.col = 1:2, data.col = 3) ## Formato geodata

summary(dat_geo) ## Aqui podemos obter o resumo das distancias

(c(media = mean(dat_c1$y),var = var(dat_c1$y))) ## Obtendo media e variancia das contagens

############################
##        Gerando Figura 9   ##
############################

par(mfrow = c(1,2))
boxplot(dat_c1$y, col = "gray", pch = 19)
plotData(dat_c1$y, dat_c1[,1:2], pch = 19, xlab = "Coord. X",
ylab = "Coord. Y")

############################
##        Gerando Figura 10    ##
############################

c <-  seq(d_min, d_max, (d_max - d_min)/13)

vy <- variog(dat_geo,estimator.type = "modulus", uvec = c)

plot(vy, xlab = "d", ylab = expression(hat(gamma)[Y](d)))

###################################
##   Fim da Analise Descritiva   ##
###################################


#################################
##        Resultados: geoRglm      ##
#################################
```

```
## Modelagem

## Informa a tendencia e a familia de correlacao espacial utilizada

m_c1 <- model.glm.control(trend.d = "cte", cov.model = "exponential")

## Parametros do MCMC: os parametros S.scale e phi.scale sao obtidos atraves
## de tentativa e erro.

mc_c1 <- mcmc.control(S.scale = 0.0138, phi.scale = .0096752, phi.start = .0055,
                burn.in = 200000, thin = 800, n.iter = 800*1500)

## Informa as prioris a serem consideradas para cada parametro e seus
## respectivos hiperparametros
## é necessario informar uma discretizacao da priori de phi.

pri_gR_c1 <- prior.glm.control(beta.prior = "flat", phi.prior = "uniform",
                    phi.discrete = seq(0.1, 0.9, by = 0.001),
                    sigmasq.prior = "sc.inv.chisq",
                    sigmasq = .7, df.sigmasq = 2)

## Objeto que recebera as saidas do modelo

pkb_c1 <- pois.krige.bayes(coords = dat_c1[,1:2], data = dat_c1$y,
                  units.m = rep(1, 100), model = m_c1,
                                      prior = pri_gR_c1,mcmc.input = mc_c1)

## Percentis 2.5%; 50% e 97.5% da posteriori de cada parametro
quantile(pkb_c1$posterior$beta$sample, probs = c(0.025, 0.5, 0.975))
quantile(pkb_c1$posterior$sigmasq$sample, probs = c(0.025, 0.5, 0.975))
quantile(pkb_c1$posterior$phi$sample, probs = c(0.025, 0.5, 0.975))

## Moda de cada paramtro
findMode(pkb_c1$posterior$beta$sample)
findMode(pkb_c1$posterior$sigmasq$sample)
findMode(pkb_c1$posterior$phi$sample)

## Transforma a saida em um objeto mcmc
## para usar a funcao de diagnostico
```

```
pbkmcmc <- mcmc(cbind(beta = pkb_c1$posterior$beta$sample, s2 = pkb_c1$posterior$sigmasq$sample,
phi = pkb_c1$posterior$phi$sample))

## Diagnostico de Heidelberger e Welch
heidel.diag(pbkmcmc, eps=0.1, pvalue=0.05)

## Computando medidas de resumo das posterioris
heidel.diag(pbkmcmc, eps=0.1, pvalue=0.05)

## Obtendo HPD
HPDinterval(pbkmcmc, prob = 0.95)

##################################
##        Gerando a Figura 11       ##
##################################

par(mfrow = c(3, 1))
acf(pkb_c1$posterior$beta$sample, xlab = "lag", ylab = "acf" , main = expression(beta))
acf(pkb_c1$posterior$sigmasq$sample, xlab = "lag", ylab = "acf" , main = expression(sigma^2))
acf(pkb_c1$posterior$phi$sample, xlab = "lag", ylab = "acf" , main = expression(phi))

##################################
##        Gerando a Figura 12       ##
##################################

par(mfrow = c(3, 1))
plot(pkb_c1$posterior$beta$sample, type = "l", main = expression(beta),
    xlab = "iteração", ylim = c(0, 8), ylab = " ")
plot(pkb_c1$posterior$sigmasq$sample, type = "l", main = expression(sigma^2),
    xlab = "iteração", ylim = c(0, 5), ylab = " ")
plot(pkb_c1$posterior$phi$sample, type = "l", main = expression(phi),
    xlab = "iteração", ylim = c(0, 1.5), ylab = " ")

##################################
##        Gerando a Figura 13       ##
##################################

par(mfrow = c(2,2))
plot(pkb_c1$posterior$simulations[12, ], type = "l", ylab = " ", ylim = c(0, 20),
    xlab = "iteração", main = expression(exp(S[12])))
```

```
plot(pkb_c1$posterior$simulations[48, ], type = "l", ylab = " ", ylim = c(0, 20),
    xlab = "iteração", main = expression(exp(S[48])))
plot(pkb_c1$posterior$simulations[80, ], type = "l", ylab = " ", ylim = c(0, 20),
    xlab = "iteração", main = expression(exp(S[80])))
plotACF(pkb_c1$posterior$simulations)

##################################
##         Resultados: geoCount     ##
##################################

## Modelagem

## Informa a distribuicao de Y, a familia da funcao de correlacao espacial,
## o numero de iteracoes, as prioris, hiperparametros e os parametros de escala
## Parametros de escala obtidos atraves da tentativa e erro
input_c1 <- MCMCinput(Y.family = "Poisson", rho.family = "rhoPowerExp",
                      run = 10000, run.S = 1, phi.bound = c(0, 1.4),
                      priorSigma = "InvGamma",parSigma = c(1,1),
                      ifkappa = 0, initials = list(c(1.61), sqrt(.7), .29,1),
                      scales = c(0.003005752, 1.65^2 + 0.4, 0.9, 0.8, 0.15))

## Objeto que armazena as saidas da modelagem
post_c1 <- runMCMC(loc = sample_coords, Y = y_c1$data, L = 0, X = NULL,
                   MCMCinput = input_c1)

## Comando para realizar o burn-in e aplicar o lag desejado.
post_c1.cut <- cutChain(post_c1, burnin=5000, thinning = 10)

## Percentis 2.5%; 50% e 97.5% da posteriori de cada parametro
quantile(post_c1.cut$m.posterior, probs = c(0.025, 0.5, 0.975))
quantile(post_c1.cut$s.posterior^2, probs = c(0.025, 0.5, 0.975))
quantile(post_c1.cut$s.posterior, probs = c(0.025, 0.5, 0.975))

## Moda de cada paramtro
findMode(post_c1.cut$m.posterior)
findMode(post_c1.cut$s.posterior^2)
findMode(post_c1.cut$a.posterior)



## Transforma a saida em um objeto mcmc
```

```
## para usar a funcao de diagnostico
post_c1.mcmc <- mcmc(cbind(sigma2=post_c1.cut$s.posterior^2, phi=post_c1.cut$a.posterior,
beta=post_c1.cut$m.posterior))

## Diagnostico de Heidelberger e Welch
heidel.diag(post_c1.mcmc, eps=0.1, pvalue=0.05)

## Obtendo resumos das posterioris
summary(post_c1.mcmc)

## Obtendo HPD
HPDinterval(post_c1.mcmc, prob = 0.95)

##################################
##        Gerando a Figura 14       ##
##################################

par(mfrow = c(3, 1))
acf(post_c1.cut$m.posterior, xlab = "lag", ylab = "acf" , main = expression(beta))
acf(post_c1.cut$s.posterior^2, xlab = "lag", ylab = "acf" , main = expression(sigma^2))
acf(post_c1.cut$a.posterior, xlab = "lag", ylab = "acf" , main = expression(phi))

##################################
##        Gerando a Figura 15       ##
##################################

par(mfrow = c(3, 1))
plot(post_c1.cut$m.posterior, type = "l", main = expression(beta),
    xlab = "iteração", ylim = c(0, 8), ylab = " ")
plot(post_c1.cut$s.posterior^2, type = "l", main = expression(sigma^2),
    xlab = "iteração", ylim = c(0, 5), ylab = " ")
plot(post_c1.cut$a.posterior, type = "l", main = expression(phi),
    xlab = "iteração", ylim = c(0, 1.5), ylab = " ")

##################################
##        Gerando a Figura 16       ##
##################################

par(mfrow = c(2,2))
plot(exp(post_c1.cut$S.posterior[12, ]), type = "l", main = expression(exp(S[12])),
```

```
        xlab = "iteração", ylab = " ", ylim = c(0, 20))
plot(exp(post_c1.cut$S.posterior[48, ]), type = "l", main = expression(exp(S[48])),
        xlab = "iteração", ylab = " ", ylim = c(0, 20))
plot(exp(post_c1.cut$S.posterior[80, ]), type = "l", main = expression(exp(S[80])),
        xlab = "iteração", ylab = " ", ylim = c(0, 20))
plotACF(post_c1.cut$S.posterior)

###################################
##          Resultados: rstan      ##
###################################

## Carregando pacote

library(rstan)

## Modelagem

## Especificando os dados, o modelo, as prioris e os hiperparametros
model_code <- '
data {
  int<lower=1> N;
  vector[N] X;
  vector[N] Y;
  int<lower=0> y[N];
}
transformed data {
  matrix[N,N] D;
  vector[N] zeros;
  for(i in 1:N) {
    for(j in 1:N) {
      if( i == j ) {
        D[i,j] <- 0.0;
      } else {
        D[i,j] <- sqrt( pow(X[i]-X[j], 2.0) + pow(Y[i]-Y[j], 2.0) );
      }
    }
  }
  for(i in 1:N) {
    zeros[i] <- 0.0;
  }
```

```
}
parameters {
  real beta;
  real<lower=0> sigmasq;
  real<lower=0> phi;
  vector[N] zeta;
}
transformed parameters {
}
model {
  matrix[N,N] Sigma;
  beta    ~ normal(0,100);
  sigmasq ~ inv_gamma(1,1);
  phi     ~ uniform(0,0.9);
  Sigma  <- sigmasq * exp(-D/phi);
  zeta    ~ multi_normal(zeros, Sigma);
  y       ~ poisson(exp(beta+zeta));
}
generated quantities{
}
'
data <- list(
  N=nrow(dat_c1),
  X=dat_c1[,1],
  Y=dat_c1[,2],
  y=dat_c1[,3]
  )

## objeto que armazenara as saidas do modelo.
## Aqui devem ser informados o numero de iteracoes, o lag, o numero de cadeias
## e os chutes iniciais de cada parametro. Por default o burnin e metade do
## numero de iteracoes
fit <- stan(model_code=model_code, data=data, iter=30000, thin=10, chains=1, seed=1,
init=function(x){list(beta=1.6,sigmasq=.7,phi=.25,zeta=rep(0,nrow(dat_c1)))},
nondiag_mass=TRUE)

## Verificando diagnosticos especificos do algoritmo
summary(do.call(rbind, args = get_sampler_params(fit, inc_warmup = FALSE)),
digits = 2)
```

```
## Extraindo as posterioris que deseja-se analisar
coda <- extract(fit, inc_warmup = FALSE)

## Percentis 2.5%; 50% e 97.5% da posteriori de cada parametro
quantile(coda$beta, probs = c(0.025, 0.5, 0.975))
quantile(coda$sigmasq, probs = c(0.025, 0.5, 0.975))
quantile(coda$phi, probs = c(0.025, 0.5, 0.975))

## Moda de cada paramtro
findMode(coda$beta)
findMode(coda$sigmasq)
findMode(coda$phi)

## Transforma a saida em um objeto mcmc
## para usar a funcao de diagnostico
coda.mcmc <- mcmc(cbind(beta = coda$beta, sigmasq = coda$sigmasq, phi = coda$phi))

## Diagnostico de Heidelberger e Welch
heidel.diag(coda.mcmc, eps=0.1, pvalue=0.05)

## Obtendo medidas de resumo das posterioris
summary(coda.mcmc)

## Obtendo HPD
HPDinterval(coda.mcmc, prob = 0.95)

##################################
##        Gerando a Figura 17       ##
##################################

par(mfrow = c(3, 1))
acf(coda$beta, xlab = "lag", ylab = "acf" , main = expression(beta))
acf(coda$sigmasq, xlab = "lag", ylab = "acf" , main = expression(sigma^2))
acf(coda$phi, xlab = "lag", ylab = "acf" , main = expression(phi))

##################################
##        Gerando a Figura 18       ##
##################################

par(mfrow = c(3, 1))
```

```
plot(coda$beta, type = "l", main = expression(beta),
    xlab = "iteração", ylim = c(0, 8), ylab = " ")
plot(coda$sigmasq, type = "l", main = expression(sigma^2),
    xlab = "iteração", ylim = c(0, 5), ylab = " ")
plot(coda$phi, type = "l", main = expression(phi),
    xlab = "iteração", ylim = c(0, 1.5), ylab = " ")

#################################
##        Gerando a Figura 19          ##
#################################

par(mfrow = c(2,2))
plot(exp(coda$zeta[,12]+mean(coda$beta)), type = "l", ylab = " ", ylim = c(0, 35),
    xlab = "iteration", main = expression(exp(S[12])))
plot(exp(coda$zeta[,48]+mean(coda$beta)), type = "l", ylab = " ", ylim = c(0, 35),
    xlab = "iteration", main = expression(exp(S[48])))
plot(exp(coda$zeta[,80]+mean(coda$beta)), type = "l", ylab = " ", ylim = c(0, 35),
    xlab = "iteration", main = expression(exp(S[80])))
plotACF(t(exp(coda$zeta)))

############################
##  Comparacao dos pacotes    ##
############################

#################################
##        Gerando a Figura 20       ##
#################################

par(mfrow = c(3,3))
acf(pkb_c1$posterior$beta$sample, xlab = "lag", ylab = "acf" , main = expression(beta))
acf(post_c1.cut$m.posterior, xlab = "lag", ylab = "acf" , main = expression(beta))
acf(coda$beta, xlab = "lag", ylab = "acf" , main = expression(beta))

acf(pkb_c1$posterior$sigmasq$sample, xlab = "lag", ylab = "acf" , main = expression(sigma^2))
acf(post_c1.cut$s.posterior^2, xlab = "lag", ylab = "acf" , main = expression(sigma^2))
acf(coda$sigmasq, xlab = "lag", ylab = "acf" , main = expression(sigma^2))

acf(pkb_c1$posterior$phi$sample, xlab = "lag", ylab = "acf" , main = expression(phi))
acf(post_c1.cut$a.posterior, xlab = "lag", ylab = "acf" , main = expression(phi))
acf(coda$phi, xlab = "lag", ylab = "acf" , main = expression(phi))
```

```
#################################
##        Gerando a Figura 21        ##
#################################

par(mfrow = c(3,3))
plot(pkb_c1$posterior$beta$sample, type = "l", main = expression(beta),
    xlab = "iteração", ylim = c(0, 8), ylab = " ")
plot(post_c1.cut$m.posterior, type = "l", main = expression(beta),
    xlab = "iteração", ylim = c(0, 8), ylab = " ")
plot(coda$beta, type = "l", main = expression(beta),
    xlab = "iteração", ylim = c(0, 8), ylab = " ")

plot(pkb_c1$posterior$sigmasq$sample, type = "l", main = expression(sigma^2),
    xlab = "iteração", ylim = c(0, 5), ylab = " ")
plot(post_c1.cut$s.posterior^2, type = "l", main = expression(sigma^2),
    xlab = "iteração", ylim = c(0, 5), ylab = " ")
plot(coda$sigmasq, type = "l", main = expression(sigma^2),
    xlab = "iteração", ylim = c(0, 5), ylab = " ")

plot(pkb_c1$posterior$phi$sample, type = "l", main = expression(phi),
    xlab = "iteração", ylim = c(0, 1.5), ylab = " ")
plot(post_c1.cut$a.posterior, type = "l", main = expression(phi),
    xlab = "iteração", ylim = c(0, 1.5), ylab = " ")
plot(coda$phi, type = "l", main = expression(phi),
    xlab = "iteração", ylim = c(0, 1.5), ylab = " ")

#################################
##        Gerando a Figura 22        ##
#################################

par(mfrow = c(3,1))
plotACF(pkb_c1$posterior$simulations)
plotACF(post_c1.cut$S.posterior)
plotACF(t(exp(coda$zeta)))

#### Fim
```