

UNIVERSIDADE FEDERAL DO PARANÁ
DEPARTAMENTO DE ESTATÍSTICA
CURSO DE BACHARELADO EM ESTATÍSTICA

Ademir Muller

SIMULAÇÃO ESTOCÁSTICA: O MÉTODO DE MONTE CARLO

ORIENTADOR:

Prof. Benito Orlando Olivares Aguilera

CURITIBA – PR

2008

Ademir Muller

SIMULAÇÃO ESTOCÁSTICA: O MÉTODO DE MONTE CARLO

Trabalho de Conclusão de
Curso apresentado como
requisito parcial para
aprovação na disciplina
Laboratório de Estatística II.

CURITIBA – PR

2008

SIMULAÇÃO ESTOCÁSTICA: O MÉTODO DE MONTE CARLO**ADEMIR MULLER**

Aprovado em ____/____/____.

BANCA EXAMINADORA

Prof. Benito Olivares Aguilera (orientador)
Depto. Estatística/ UFPR

Prof. Joel Corrêa da Rosa
Depto. Estatística/ UFPR

CONCEITO FINAL: _____

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1. Simulação Estocástica	1
1.2. Método de Monte Carlo.....	2
1.2.1 Breve resenha	2
1.2.2 A metodologia	6
1.2.3 Outros métodos que utilizam Monte Carlo.....	8
CAPÍTULO 2 - CÁLCULO DE INTEGRAIS DEFINIDAS.....	10
2.1. Descrição do Método.....	10
2.2. A escolha da distribuição na redução da variância.....	11
2.3. Simulação de variáveis aleatórias contínuas	12
2.4. Um exemplo numérico	14
2.5. Acerca da estimativa do erro absoluto.....	17
CAPÍTULO 3 - MONTE CARLO COM CADEIA DE MARKOV	19
3.1. Introdução.....	19
3.2. Método.....	19
3.3. Algoritmo de Metropolis-Hastings	22
3.4. Amostrador de Gibbs.....	28
3.5. Avaliação da Convergência.....	33
CAPÍTULO 4 – APLICAÇÃO: ANÁLISE CUSTO-EFICIÊNCIA DA PRÓTESE DE REPOSIÇÃO DA BACIA.....	36
4.1. Justificativa.....	36
4.2. Modelo Estatístico	37
4.3. Parâmetros do Modelo.....	39
4.4. Custo - Benefício	40
4.5. Análise Probabilística de Sensibilidade.....	42
4.5.1. Sensibilidade para heterogeneidade de pacientes: caso parâmetro θ fixo	42
4.5.2. Sensibilidade para parâmetros incertos de subgrupos de pacientes fixando s	43
4.5.3. Sensibilidade associada para parâmetros incertos e heterogeneidade	45
4.5.4. Exemplo: Sensibilidade para heterogeneidade e parâmetro incerto	46
CONCLUSÃO.....	49
REFERÊNCIAS BIBLIOGRAFICAS	50
Comandos do R	52
Comandos do WinBUGS	53

RESUMO

Este trabalho apresenta um método bastante popular de simulação estocástica: o Método de Monte Carlo. Os computadores de alta velocidade e as linguagens especiais de programação tornaram a simulação uma poderosa técnica para obter soluções aproximadas de muitos problemas complexos, ou analiticamente insolúveis. O Método de Monte Carlo é um mecanismo que gera dados a partir de um gerador de números aleatórios e das distribuições de frequências de interesse, as quais caracterizam os processos estocásticos considerados pelo modelo de simulação utilizado. O Método de Monte Carlo via Cadeia de Markov (MCMC) é utilizado para simular diversos modelos probabilísticos. Esta simulação é um processo que projeta o comportamento de um sistema real e conduz experimentos de amostragem, com o propósito de entender o comportamento do sistema ou avaliar várias estratégias para sua operação. Envolve a geração e observação da “história artificial” do sistema e permite traçar inferências a respeito das características operacionais do sistema real. Uma aplicação do método é feita para simular custos e benefícios esperados para um grupo de pacientes que utilizaram prótese na reposição da bacia.

Palavras-chave: Simulação estocástica, Método de Monte Carlo, Cadeias de Markov

CAPÍTULO 1 - INTRODUÇÃO

1.1. Simulação Estocástica

O termo simulação refere-se a tratamento de problemas reais através de reproduções em ambientes controlados pelo pesquisador. Esses ambientes muitas vezes são equipamentos computacionais, embora possam ser também reproduções do sistema em estudo em uma escala menor. Um exemplo é o estudo do movimento de marés em reservatórios de laboratório. O movimento da água é causado por um dispositivo controlado por um motor elétrico. Nesse caso, está se estudando um sistema onde todos os seus componentes são conhecidos e seu comportamento tem um caráter determinístico.

Em outros sistemas temos alguns ou todos os componentes sujeitos a efeitos aleatórios, ou seja, não podem ser descritos de forma exata por uma regra matemática, mas apenas através do uso de estruturas probabilísticas. Esses sistemas aparecem freqüentemente no mundo real e o seu processo de simulação é estocástico, ou seja, é baseado em distribuições de probabilidade.

Do ponto de vista formal, esses sistemas podem ser vistos como coleções de variáveis aleatórias e, embora a estrutura dessas variáveis possa ser bastante complexa dependendo do problema, o objetivo se resume a reproduzir essas variáveis aleatórias dentro de um ambiente sob nosso controle. Para isso, é utilizado tipicamente um computador dotado da capacidade de reprodução dessas variáveis.

O ponto de partida para simulações estocásticas é a construção de um gerador de números aleatórios. Usualmente, esse mecanismo gera números uniformemente em um

intervalo $[0,M]$, para um dado valor M . Podemos também limitar essa geração a um número no intervalo $[0,1]$ pela divisão por M .

É importante frisar, entretanto, que todos os métodos atualmente disponíveis são baseados em mecanismos determinísticos e iterativos, portanto a nomenclatura mais adequada é “pseudo-aleatório”, para deixar claro que esses mecanismos geram números com comportamento similar aos efetivamente aleatórios. O algoritmo em si é completamente determinístico e seu conhecimento permite prever com certeza o próximo valor gerado, mas se não se assume conhecimento do mecanismo e por causa da enorme amplitude do gerador, pode-se assumir que os números gerados possuem as propriedades básicas necessárias a um número realmente aleatório: uniformidade no intervalo unitário e independência entre seqüências de valores. Uma discussão mais detalhada sobre o assunto pode ser encontrada em Bustos e Frery (1992), Ripley (1987) e Devroye (1986).

1.2. Método de Monte Carlo

1.2.1 Breve resenha

O Método de Monte Carlo fornece soluções aproximadas para uma série de problemas matemáticos executando experimentos de amostragem estatística em um computador. Notavelmente, o método aplica-se tanto a problemas com conteúdo não probabilístico quanto àqueles com estrutura probabilística. Apenas isto não faz do Método de Monte Carlo vantajoso sobre outros métodos de aproximação, porém, entre todos os métodos numéricos disponíveis para a avaliação de n -pontos no espaço m -dimensional que produzem uma solução aproximada, o Método de Monte Carlo teve erro absoluto de estimação da ordem de $n^{-1/2}$, enquanto que todos os outros têm erros que decrescem, na melhor hipótese à taxa de $n^{-1/m}$. Esta propriedade dá ao Método de Monte Carlo uma considerável eficiência

computacional. Ajustes combinatórios ilustram esta propriedade muito bem. Visto que a solução exata para o problema combinatório com m elementos frequentemente tem custo computacional que aumentam exponencialmente ou superexponencialmente com m , o Método de Monte Carlo frequentemente fornece uma solução estimada com erro tolerável e um custo que aumenta não mais rápido que um polinômio de grau m .

Bastante simples no conceito, o Método de Monte Carlo incorpora três desenvolvimentos históricos da ciência matemática distintos, mas relacionados. Primeiro, os jogos de chances motivaram os matemáticos do século dezessete e dezoito a pensar resultados de ensaios sucessivos como uma seqüência de eventos aleatórios. Observando que a média de uma função de variáveis aleatórias contínuas toma a forma de uma integral, estatísticos do século dezenove e vinte reconhecem que, a princípio, poder-se-ia extrair números aleatórios, transformá-los de acordo com regras prescritas e derivar uma solução aproximada para uma integral, num problema que intrinsecamente não tem conteúdo probabilístico. No século dezenove uma segunda linha de pesquisa desenvolvida por Lord Rayleigh (1899) mostrou que um passeio aleatório unidimensional sem barreiras absorventes poderia fornecer uma solução aproximada para uma equação diferencial parabólica. Para demonstrar que em condições apropriadas uma particular equação de diferenças finitas poderia produzir uma solução aproximada para o problema de Dirichlet, Courant mostrou que a forma recursiva da solução para o passeio aleatório bidimensional, em uma área quadriculada dentro de uma região fechada cujos pontos fronteira fossem estados absorventes, produz exatamente a mesma equação em diferenças. Logo depois, Kolmogorov (1931) mostrou a relação entre processos estocásticos Markovianos e certas equações integro-diferenciais. Petrowski (1933) generalizou o resultado de Courant mostrando a conexão assintótica entre o passeio aleatório cuja seqüência de posições forma uma cadeia de Markov e a solução para uma equação

diferencial elíptica parcial. Ele chamou esta formulação de “problema de Dirichlet generalizado”.

A essa altura, ficou evidente a significância de observar que soluções para problemas encontrados em processos estocásticos, freqüentemente correspondem a soluções que aparecem no estudo de equações diferenciais parciais, e que então, poderia se empregar o método das equações em diferença para fornecer soluções aproximadas ao problema estocástico original.

Durante o desenvolvimento da energia atômica na era pós Segunda Guerra Mundial, a terceira linha de pesquisa evoluiu. Cientistas precisavam resolver problemas de difusão de nêutrons ou transporte através de um meio isotrópico. Esses problemas multidimensionais provaram-se extremamente complexos para o método tradicional de equações em diferença. Dado que já havia sido estabelecida a analogia entre equações diferenciais parciais e integrais com processos estocásticos, John Von Neumann e Stanislaw Ulam sugeriram que experimentos amostrais usando modelos de passeios aleatórios e executando-os nos recentemente desenvolvidos computadores digitais poderiam fornecer aproximações utilizáveis para as soluções desejadas. Esta proposta inverte o sentido do raciocínio. Em vez de usar a incômoda equação em diferenças para fornecer soluções a problemas probabilísticos, usar experimentos amostrais para fornecer soluções para as equações integro-diferenciais, em que não necessariamente temos uma base probabilística. Além disso, os computadores fornecem o meio para colocar esta sugestão em prática. Mais tarde, o conceito de utilização de experimentos amostrais no computador veio para prevalecer em muitas disciplinas científicas, mais notavelmente em química, engenharia, pesquisa operacional, física e estatística. Em pesquisa operacional, a maioria dos modelos de filas simples usa gerações aleatórias em computadores para fornecer soluções aproximadas. Em física, o

Método de Monte Carlo veio ser reconhecido como a uma única aproximação eficiente para fornecer soluções úteis aos problemas da estrutura e do transporte de nêutrons.

Na década de 70, o advento da Teoria da Complexidade Computacional proporcionou argumentos mais precisos e persuasivos para utilização do Método de Monte Carlo. A teoria identificou uma classe de problemas para a qual o tempo de cálculo da solução exata para o problema nesta classe cresce, ao menos, exponencialmente com m . Muitos problemas comumente encontrados pertencem a esta classe e, se não existir uma estrutura especial que possa conduzir a um algoritmo para cálculo exato, aparece um alto custo de avaliação. A existência desta classe substancial de problemas deu um novo curso para a idéia de que o Método de Monte Carlo pode ofertar uma solução competitiva. A questão para resolver era se o Método de Monte Carlo poderia estimar a solução para problemas nesta classe, dentro de uma exatidão estatística especificada, em um tempo limitado superiormente por um polinômio de grau m . Numerosos exemplos agora sustentam esta questão. Karp(1985) mostra esta propriedade para estimar a confiabilidade em uma rede multi-terminal com falhas aleatórias. Dyer(1989) estabelece isto para estimar o volume de um corpo convexo num espaço euclidiano m -dimensional. Broder(1986) e Jerrum e Sinclair(1988) estabelecem a propriedade para estimar o número perfeito de combinações em um gráfico bipartido.

A simplicidade do conceito disfarça a sofisticação matemática que o Método de Monte Carlo pode apresentar. Em particular, Von Neumann, Ulam, e outros reconheceram que é possível modificar o Método de Monte Carlo padrão de modo a produzir uma solução para o problema original com erro específico consideravelmente menor, em termos de tempo computacional, do que gerando diretamente o passeio aleatório correspondente ao problema original. Algumas dessas técnicas de redução de variância já eram conhecidas dos estatísticos que atuavam na área de amostragem, mas outras técnicas devem sua origem ao Método de

Monte Carlo. Coletivamente esses procedimentos vêm representar o foco central do Método de Monte Carlo.

No início da era dos computadores, com a relativa lentidão da primeira geração deles, a aplicação de técnicas de redução de variância foi um ingrediente essencial para qualquer aplicação bem sucedida do Método de Monte Carlo. De qualquer forma, tanto antes quanto hoje, uma análise cuidadosa deve ser realizada para determinar a técnica de redução de variância para cada problema. Independente de tudo existe problemas de tamanho substancial e, se o uso de um microcomputador, estação de trabalho ou computador de grande porte irá possibilitar a obtenção de soluções estimadas com alta exatidão estatística depende criticamente de como será aproveitada a informação disponível.

1.2.2 A metodologia

Para executar o Método de Monte Carlo, há três passos básicos que devem ser seguidos. O primeiro passo envolve o estabelecimento das populações de interesse. Estas populações devem ter certos parâmetros (por exemplo, média, desvio padrão, moda, etc.) e podem apresentar vários comportamentos distribucionais (por exemplo, Normal, Exponencial, Uniforme, etc.).

O segundo passo para executar o Método de Monte Carlo é obter amostras aleatórias dessas populações e calcular as estatísticas de interesse. Para obter as amostras é preciso gerar computacionalmente uma seqüência de números aleatórios utilizando algum gerador disponível. Geradores de números aleatórios produzem comumente uma série de valores segundo uma distribuição Uniforme no intervalo (0,1), porém, podem-se obter números aleatórios em qualquer intervalo (0,K) fazendo uma transformação da seguinte forma:

$$L = \text{Inteiro}[(R * K) + 1],$$

onde L é um número inteiro variando de 1 até K e R é um número gerado no intervalo $(0,1)$.

Como apontado na Seção 1.1, os números gerados por esta técnica são pseudo-aleatórios, porém, eles geralmente são suficientes para obter aproximações razoáveis de números aleatórios verdadeiros, e podem ser usados para obter amostras aleatórias de alguma população de interesse.

Só a modo de exemplo, o Software R dispõe de vários geradores de números aleatórios, tais como o gerador de Wichmann-Hill, cujo período (extensão sem repetição) é da ordem de 6.9536×10^{12} números, o Super-Duper cujo período é da ordem de 4.6×10^{18} e o Marsaglia-Multicarry, com um amplo período de mais de 2^{60} e que, aparentemente, tem passado em todos os testes de aleatoriedade realizados. Existe atualmente geradores de período muito longo, como o Mersenne-Twister, cujo período é da ordem de $2^{19937-1}$.

O terceiro passo para executar o Método de Monte Carlo é criar a distribuição de frequência da estatística de interesse. Esta distribuição é chamada de distribuição aleatória empírica, e pode ser comparada com a distribuição aleatória teórica da estatística.

A distribuição empírica, às vezes, pode ser comparada com outra distribuição empírica, ou seja, o pesquisador pode executar dois Métodos de Monte Carlo, por exemplo, um no qual nenhuma das suposições foi violada, e outro no qual alguma suposição foi violada. Então, podem-se fazer comparações entre as duas distribuições aleatórias empíricas.

O foco deste trabalho é apresentar os conceitos iniciais do método, com ênfase na aplicação em Cadeias de Markov. O Capítulo 2 é dedicado a ilustrar o método no cálculo de integrais definidas, seguindo de perto o enfoque de Sobol. O Capítulo 3 mostra o Método de Monte Carlo via Cadeias de Markov e suas duas principais ferramentas: o algoritmo de Metropolis-Hastings e o Amostrador de Gibbs. Já o Capítulo 4 apresenta uma aplicação prática do método na área médica.

1.2.3 Outros métodos que utilizam Monte Carlo

Desde sua formulação por Metropolis e Ulam (1949) diversas metodologias foram criadas utilizando a idéia original do Método de Monte Carlo. Na medida em que recursos computacionais cada vez mais poderosos foram viáveis, diversas modificações ou sofisticações do método apareceram.

Algumas delas são:

Simulação direta de Monte Carlo: modela os fluxos de fluidos usando simulação que representam um grande número de moléculas em uma simulação probabilística para resolver a equação de Boltzmann. Atualmente o método tem sido aplicado para a solução dos fluxos que vão desde a estimativa do lançamento espacial, reentrada aerodinâmica e modelagem de micro-sistemas eletromecânicos.

Monte Carlo Cinético: Este método calcula simulações no intuito de simular o tempo de evolução de alguns processos que ocorrem na natureza. Normalmente são processos que ocorrem com uma determinada taxa conhecida. É importante compreender que estas taxas são entradas para o algoritmo de Monte Carlo Cinético.

Monte Carlo Quântico: é uma grande classe de algoritmos que simulam sistemas quânticos com a idéia de resolver o problema de “muitos corpos”. Usa-se o método de Monte Carlo para controlar muitas integrais dimensionais. Monte Carlo Quântico permite representar diretamente o efeito de “muitos corpos” em funções de onda, com um custo estatístico que pode ser reduzido com um tempo maior de simulação.

Quase Monte Carlo: Também chamado de seqüências de baixa discrepância são usadas como substituto para distribuição uniforme de números aleatórios. O “quase” modifica o uso para denotar mais claramente que os valores para uma seqüência de baixa discrepância não são aleatórios nem pseudoaleatórios, mas seqüências com alguma propriedade de variáveis aleatórias e em certas aplicações o Quase Monte Carlo com baixa discrepância é uma importante vantagem.

Monte Carlo com Cadeias de Markov: é uma classe de algoritmos que recolhe amostras de distribuições de probabilidade baseadas em uma cadeia de Markov que possui uma distribuição de equilíbrio. O estado da cadeia depois de um grande número de iterações é então usado como uma amostra da distribuição desejada.

Localização por Monte Carlo: Usado em robótica e sensores, é um método para determinar a posição de um robô dado um mapa do seu ambiente.

Otimização estocástica: são métodos de otimização que incorporam elementos aleatórios, quer em dados do problema (a função objeto), no próprio algoritmo ou em ambos. O conceito contrasta com os métodos de otimização determinística, onde os valores da função objeto são assumidos para ser exato, bem como o cálculo é totalmente determinado pelos valores amostrados.

CAPÍTULO 2 - CÁLCULO DE INTEGRAIS DEFINIDAS

2.1. Descrição do Método

Considere a integral definida

$$I = \int_a^b g(x)dx ,$$

onde $g(x)$ é uma função definida sobre o intervalo $a < x < b$.

O objetivo é calcular de forma aproximada a integral acima.

Seja $p_X(x)$ uma densidade de probabilidade (por enquanto arbitrária) definida em (a,b) , isto é, uma função satisfazendo as relações:

a) a densidade $p(x)$ é positiva em (a,b) , isto é,

$$p(x) > 0, \quad x \in (a, b) ;$$

b) a integral de $p(x)$ tomada sobre o intervalo (a, b) é igual a um, quer dizer,

$$\int_a^b p(x)dx = 1 .$$

Se, tendo denotado por X a variável aleatória dada pela densidade $p_X(x)$, tomarmos

$$Y = \frac{g(x)}{p_X(x)} ,$$

então, temos:

$$E(Y) = \int_a^b \frac{g(x)}{p_X(x)} p_X(x) dx = \int_a^b g(x) dx = I .$$

Consideremos N variáveis aleatórias independentes e identicamente distribuídas

Y_1, Y_2, \dots, Y_N e apliquemos o Teorema central do Limite a sua soma. Nesse caso

$$P\left\{\left|\frac{1}{N}\sum_{j=1}^N Y_j - I\right| < 3\sqrt{\frac{V(Y)}{N}}\right\} \approx 0,997.$$

Logo, se X_1, X_2, \dots, X_N forem N valores independentes de X , para N suficientemente grande:

$$\frac{1}{N}\sum_{j=1}^N \frac{g(x_j)}{p_X(x_j)} \approx I,$$

sendo que o erro da estimativa quase certamente não excederá $3\sqrt{\frac{V(Y)}{N}}$.

2.2. A escolha da distribuição na redução da variância

Qualquer que seja a variável X dada em (a, b) se terá

$$E(Y) = E\left(\frac{g(x)}{p_X(x)}\right) = I$$

Todavia, como resulta da relação

$$V(Y) = E(Y^2) - [E(Y)]^2 = E(Y^2) - I^2 = \int_a^b \frac{g^2(x)}{p_X(x)} dx - I^2,$$

a escolha particular de X pode servir a minimizar a variância $V(Y)$ e simultaneamente, o erro da estimativa.

A variância em apreço assume seu mínimo quando $p_X(x)$ é proporcional a $|g(x)|$.

Logo, é razoável escolher $p_X(x)$ de modo que o quociente desta função por $|g(x)|$ não se desvie muito de uma constante e que, ao mesmo tempo, a simulação de X não resulte demasiado laboriosa.

Cabe salientar que, na prática, o método de Monte Carlo não teria nenhuma vantagem sobre os tradicionais métodos de quadratura, que fornecem grande exatidão para calcular integrais unidimensionais, mas no caso de integrais múltiplas, o método de Monte Carlo torna-se indiscutivelmente superior, pois as fórmulas de integração ficam extremamente complexas ou ineficientes, enquanto que o método de Monte Carlo permanece o mesmo.

2.3. Simulação de variáveis aleatórias contínuas

Para aplicação do Método de Monte Carlo no cálculo de integrais definidas é necessário ainda compreender como deve ser realizada a simulação de variáveis aleatórias contínuas.

Suponhamos que se trate da simulação de uma variável aleatória X , distribuída num intervalo (a, b) com a densidade $p(x)$.

Mostremos que se pode simular X mediante a solução da equação

$$\int_a^x p(x)dx = g$$

encontrando a partir de cada valor de g , o respectivo valor de X .

Neste intuito, examinaremos a função

$$y = \int_a^x p(x)dx$$

que, em virtude das propriedades de uma densidade de probabilidade,

$$p(x) > 0;$$

$$\int_a^b p(x)dx = 1.$$

satisfará as condições

$$y(a)=0, \quad y(b)=1$$

e

$$y'(x)=p(x)>0,$$

significando isto que $y(x)$ é monótona crescente entre 0 e 1. Logo, qualquer reta da forma $y=\alpha$, com $0 < \alpha < 1$, terá um único ponto de interseção com o gráfico da função $y(x)$. A abscissa deste ponto, que será a única solução da equação

$$\int_a^x p(x)dx = \alpha,$$

toma-se como sendo o valor de X .

Mostremos agora que estes valores simulam de fato a variável de densidade $p(x)$. Aos pontos x ,

$$a' < x < b',$$

de qualquer intervalo (a', b') compreendido em (a, b) correspondem os pontos $y=y(x)$ satisfazendo as desigualdades

$$y(a') < y < y(b').$$

Assim, X assumirá um valor no intervalo $a' < x < b'$ se, e somente se, α tiver assumido um valor em $y(a') < \alpha < y(b')$. Logo,

$$P\{a' < X < b'\} = P\{y(a') < \alpha < y(b')\}.$$

A variável α estando uniformemente distribuída em $(0,1)$, vale

$$P\{y(a') < \alpha < y(b')\} = y(b') - y(a') = \int_{a'}^{b'} p(x)dx,$$

assim

$$P\{a' < X < b'\} = \int_{a'}^{b'} p(x) dx.$$

A última relação significa que a densidade da variável aleatória X , simulada por meio da

equação $\int_a^x p(x) dx = g$, tem densidade de probabilidade $p(x)$.

Exemplo: Sabe-se que variável aleatória Y se distribui uniformemente num intervalo (a,b) , se a sua densidade de probabilidade permanecer constante neste intervalo. Neste caso,

$$p(x) = \frac{1}{b-a} \quad \text{para } a < x < b.$$

A simulação de Y se efetua, de acordo com $\int_a^x p(x) dx = g$, a partir de

$$\int_a^y \frac{dx}{b-a} = g.$$

acha-se a expressão explícita

$$Y = a + g(b-a),$$

que é uma transformação linear de ?.

2.4. Um exemplo numérico

Trata-se do cálculo aproximado da integral

$$I = \int_0^{p/2} \text{sen}(x) dx$$

cujo valor exato é

$$\int_0^{p/2} \text{sen}(x) dx = [-\cos(x)]_0^{p/2} = 1$$

Empregaremos sucessivamente, a título de ilustração, duas variáveis aleatórias X distintas, a saber, a variável uniformemente distribuída em $(0, p/2)$, isto é, dada pela densidade

$$p_X(x) = \frac{2}{p}, \text{ e a variável } X \text{ dada pela densidade } p_X(x) = \frac{8x}{p^2}.$$

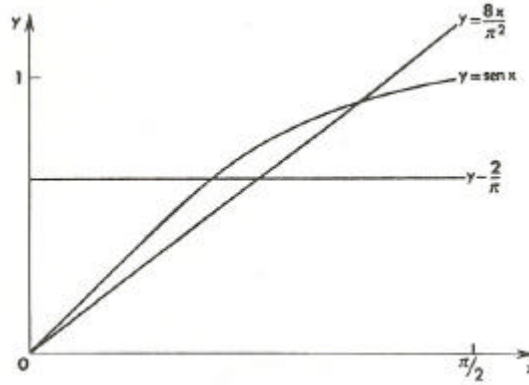


Figura 1: A função $y = \text{sen } x$ e duas densidades.

Da figura 1, segue que a segunda densidade (igual a $8x/p^2$) é quem melhor obedece ao critério dado na Seção 2.2 acerca da proporcionalidade entre $p_X(x)$ e $\text{sen } x$. Isto permite conjecturar que é mais conveniente basear-se na segunda densidade.

(a) O caso da variável X dada por $p_X(x) = \frac{2}{p}$ em $(0, p/2)$. Esta variável se simulará

pela fórmula $Y = a + g(b - a)$, com $a=0$ e $b=p/2$, isto é, se terá

$$X = \frac{pg}{2}.$$

A relação $\frac{1}{N} \sum_{j=1}^N \frac{g(x_j)}{p_X(x_j)} \approx I$ será dada por

$$I \approx \frac{p}{2N} \sum_{j=1}^N \text{sen}(x_j).$$

Se, com $n=10$, a título de valores de I tomarmos números com três casas decimais obtidos agrupando três a três os algarismos gerados aleatoriamente, chegaremos, após duas etapas de cálculo à Tabela 1.

Tabela 1: Valores obtidos para a densidade $U(0, p/2)$.

J	1	2	3	4	5	6	7	8	9	10
g_j	0,865	0,159	0,566	0,079	0,155	0,664	0,345	0,655	0,812	0,332
x_j	1,359	0,250	0,124	0,889	0,243	1,043	0,542	1,029	1,275	0,521
$sen(x_j)$	0,978	0,247	0,124	0,776	0,241	0,864	0,516	0,857	0,957	0,498

o resultado final será

$$I \approx \frac{p}{2N} \sum_{j=1}^N sen(x_j) = 0,952.$$

(b) Se X se der pela densidade $p_X(x) = \frac{8x}{p}$, a simulação procederá pela fórmula

$$\int_a^x p(x) dx = g,$$

$$\int_0^x \frac{8x}{p} dx = g,$$

donde

$$X = \frac{p}{2} \sqrt{g}.$$

A relação $\frac{1}{N} \sum_{j=1}^N \frac{g(x_j)}{p_X(x_j)} \approx I$ chegará a ser

$$I \approx \frac{\mathbf{p}^2}{8N} \sum_{j=1}^N \frac{\text{sen}(x_j)}{x_j}.$$

A partir dos resultados intermediários obtidos pelos valores de θ empregados no caso (a) e mostrados na Tabela 2,

Tabela 2: Valores obtidos para a densidade $8x/\mathbf{p}^2$.

J	1	2	3	4	5	6	7	8	9	10
g_j	0,865	0,159	0,566	0,079	0,155	0,664	0,345	0,655	0,812	0,332
x_j	1,461	0,626	0,442	1,182	0,618	1,280	0,923	1,271	1,415	0,905
$\frac{\text{sen}(x_j)}{x_j}$	0,680	0,936	0,968	0,783	0,937	0,748	0,863	0,751	0,698	0,868

encontra-se que

$$I \approx \frac{\mathbf{p}^2}{8N} \sum_{j=1}^N \frac{\text{sen}(x_j)}{x_j} = 1,016.$$

Como se tinha previsto, a aplicação da segunda densidade conduz a um resultado mais preciso.

2.5. Acerca da estimativa do erro absoluto

De acordo com que se observou na Seção 2.1, o erro absoluto da estimativa praticamente não pode exceder $3\sqrt{\frac{V(Y)}{N}}$. Porém, o erro será quase sempre sensivelmente inferior a este número, sendo razoável o emprego, a título de sua estimativa, do chamado erro provável \mathbf{d}_{prv} ,

$$d_{prv} = 0,675 \sqrt{\frac{V(Y)}{N}}.$$

Com relação ao conceito de erro provável pode-se mostrar que, para uma variável aleatória normal ? individualizada pelos parâmetros a e s , vale $\int_{a-r}^{a+r} p_z(x) dx = 0,5$, onde $r = 0,675s$.

O erro real dependerá da amostra de números aleatórios empregada e poderá ser várias vezes superior ou inferior ao erro provável. Assim, este último fornece apenas a ordem do erro possível.

A Tabela 3 ilustra o resultado do cálculo da variância $V(Y)$ corresponde a cada um dos procedimentos de solução do exemplo numérico. Como era de se esperar, a ordem do erro observado e a do erro provável é a mesma.

Tabela 3: Cálculo da variância para ambas as densidades do exemplo.

Procedimento	$V(Y)$	d_{prv}	Erro observado
(a)	0,256	0,103	0,048
(b)	0,016	0,027	0,016

$$(a) V(Y) \approx \frac{\mathbf{p}^2}{9.4} \left[\sum_{j=1}^{10} (\text{sen}(x_j))^2 - \frac{1}{10} \left(\sum_{j=1}^{10} \text{sen}(x_j) \right)^2 \right] = \frac{\mathbf{p}^2}{36} (4,604 - 3,670) = 0,256$$

$$(b) V(Y) \approx \frac{\mathbf{p}^4}{9.64} \left[\sum_{j=1}^{10} \left(\frac{\text{sen}(x_j)}{x_j} \right)^2 - \frac{1}{10} \left(\sum_{j=1}^{10} \frac{\text{sen}(x_j)}{x_j} \right)^2 \right] = \frac{\mathbf{p}^4}{576} (6,875 - 6,777) = 0,016$$

CAPÍTULO 3 - MONTE CARLO COM CADEIA DE MARKOV

3.1. Introdução

No método simples de Monte Carlo se obtém uma amostra da distribuição de importância em um único passo. Os valores são gerados de forma independente e não há preocupação com a convergência do algoritmo, bastando que o tamanho da amostra seja suficientemente grande. Por isso estes métodos são chamados não iterativos. No entanto, em muitos problemas pode ser bastante difícil, ou mesmo impossível, encontrar uma densidade de importância que seja simultaneamente uma boa aproximação da densidade verdadeira e que seja fácil de ser amostrada. Os métodos de Monte Carlo com cadeias de Markov (MCMC) são uma alternativa aos métodos não iterativos em problemas complexos. A ideia ainda é obter uma amostra da distribuição a posteriori e calcular estimativas amostrais de características desta distribuição. A diferença é que aqui usaremos técnicas de simulação iterativa, baseadas em cadeias de Markov, e assim os valores gerados não serão mais independentes. Nesta seção serão apresentados os métodos MCMC mais utilizados, o amostrador de Gibbs e o algoritmo de Metropolis-Hastings. A ideia básica é simular um passeio aleatório no espaço de θ que converge para uma distribuição estacionária, que é a distribuição de interesse no problema. Uma discussão mais geral sobre o tema pode ser encontrada, por exemplo, em Gamerman (1997) e Gamerman e Lopes (2006).

3.2. Método

Em geral, amostrar X_i independentemente da distribuição $p(\cdot)$ não é factível, visto que $p(\cdot)$ pode ser muito diferente das distribuições padrões, tais como: Normal, Beta, Gama, etc.

Entretanto $\{X_t\}$ não precisa ser necessariamente independente. As $\{X_t\}$ podem ser geradas por qualquer processo que gera amostras através do suporte de $\mathbf{p}(\cdot)$ numa proporção correta. Uma forma para construir essa amostra é através de uma cadeia de Markov que possua $\mathbf{p}(\cdot)$ como distribuição estacionária (invariante), ou seja, que possua $\mathbf{p}(\cdot)$ como distribuição de equilíbrio. Nem sempre tal distribuição existe.

Dependência de Markov é um conceito atribuído ao matemático russo Andrei Andreivich Markov, quem no início do século 20 pesquisou a alternância de vogais e consoantes no poema “Onegim” de Poeshkin. Ele desenvolveu um modelo probabilístico no qual resultados sucessivos dependiam dos resultados de seus predecessores somente através do predecessor imediato.

$$X^{(0)} X^{(1)} X^{(2)} \dots X^{(n-2)} X^{(n-1)} X^{(n)}$$

Esse modelo permitiu, a ele, obter boas estimativas da frequência de vogais no poema. Quase simultaneamente, o matemático francês Henri Poincaré estudou seqüências de variáveis aleatórias que eram de fato cadeias de Markov.

Uma cadeia de Markov é um tipo especial de processo estocástico que cuida da caracterização de seqüência de variáveis aleatórias que apresentam a dependência de Markov. De especial interesse é a dinâmica e o comportamento limite da seqüência.

Um processo estocástico pode ser definido como uma coleção de variáveis aleatórias $\{X^{(t)} \in S \mid t \in T\}$ para algum T. O conjunto $\{X^{(t)} \in S \mid t \in T\}$ é definido como um processo estocástico com espaço de estados S e conjunto de índices T. Simplificando, uma cadeia de Markov é um processo estocástico, em que, dado o estado presente, passado e futuro são independentes. Essa propriedade pode ser formalizada através de:

$$\Pr[X^{(n+1)} \in A \mid X^{(n)} = x, X^{(n-1)} \in A_{n-1}, \dots, X^{(0)} \in A_0] = \Pr[X^{(n+1)} \in A \mid X^{(n)} = x],$$

para quaisquer conjuntos $A_0, \dots, A_{n-1}, A \subset S$ e $x \in S$ (Gamerman, 1997).

Suponha a seqüência de variáveis aleatórias $\{X_0, X_1, X_2, \dots\}$ tal que em cada tempo $t \geq 0$ o próximo estado X_{t+1} é amostrado da distribuição $P(X_{t+1} | X_t)$, a qual depende somente do estado atual da cadeia X_t , isto é, dado X_t , o próximo estado X_{t+1} não depende mais da história da cadeia $\{X_0, X_1, X_2, \dots, X_{t-1}\}$. Esta cadeia é chamada de *cadeia de Markov* e $P(\cdot | \cdot)$, é chamado de “kernel” (ou núcleo) de transição da cadeia e não depende de t se a cadeia for homogênea no tempo.

Os métodos Monte Carlo com Cadeias de Markov requerem ainda que a cadeia seja:

- Homogênea, i.e. as probabilidades de transição de um estado para outro não dependem do tempo (invariantes);
- Irredutível, i.e. cada estado pode ser atingido a partir de qualquer outro estado em um número finito de iterações;
- Aperiódica, i.e. não haja estados absorventes.

Sujeito a condição de regularidade a cadeia irá gradualmente esquecendo seu estado inicial $P^{(t)}(\cdot | X_0)$, sendo que $P^{(t)}(X_t | X_0)$ é a distribuição de X_t dado X_0 , e convergirá para uma única distribuição estacionária ou invariante, que não depende de t ou X_0 . Esta distribuição estacionária será denotada por $f(\cdot)$. Com o crescimento de t os pontos amostrados $\{X_t\}$ se parecerão mais e mais com amostras dependentes de $f(\cdot)$. Assim, depois de se desprezarem m iterações (“*burn-in*”), os pontos $\{X_t, t = m + 1, \dots, n\}$ serão aproximadamente amostras dependentes de $f(\cdot)$. Pode-se agora utilizar esta amostra para estimar a esperança $E[g(X)]$ em que X tem distribuição $f(\cdot)$. Descartando o “*burn-in*” para os cálculos de estimador tem-se:

$$\bar{g} = \frac{1}{n - m} \sum_{t=m+1}^n g(X_t), \quad (3.1)$$

que é a chamada de média ergódica.

A equação (3.1) mostra como uma cadeia de Markov pode ser utilizada para estimar $E[g(X)]$, em que a esperança é tomada sobre a distribuição estacionária $f(\cdot)$. Esta parece fornecer uma solução para o problema, mas primeiro é necessário descobrir como construir cadeias de Markov tal que a sua distribuição estacionária $f(\cdot)$ seja precisamente a conjunta a *posteriori* $p(\mathbf{q} | y)$, isto é, $f(x) = p(\mathbf{q} | y)$.

Apresentam-se a seguir dois algoritmos: Metropolis-Hastings e Amostrador de Gibbs, que permitem gerar cadeias de Markov cuja distribuição de equilíbrio seja precisamente $p(\mathbf{q} | y)$.

3.3. Algoritmo de Metropolis-Hastings

O algoritmo de Metropolis-Hastings permite gerar uma amostra da distribuição conjunta a *posteriori* $p(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k | y)$, a partir das distribuições condicionais completas, que podem possuir forma fechada ou não (Metropolis et al., 1953; Hastings, 1970; Chib & Greenberg, 1995).

Os fatos principais desse método para gerar amostras de uma distribuição com densidade $p(\mathbf{q} | y) = p(\mathbf{q})$ são:

- i. os cálculos dependem de $p(\mathbf{q})$ somente através da razões da forma $\frac{p(\mathbf{q}')}{p(\mathbf{q})}$ em que \mathbf{q} e \mathbf{q}' são pontos amostrais. Assim, a constante normalizadora não precisa ser conhecida, pois será cancelada no quociente;
- ii. nenhuma fatoração de $p(\mathbf{q})$ será necessária;
- iii. os métodos são facilmente implementados, e

iv. a seqüência de amostras é obtida através de uma cadeia de Markov.

Seja a cadeia de Markov $\{\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_z, \dots\}$. Para o algoritmo de Metropolis-Hastings, suponha que atualmente a cadeia está em estado $\mathbf{q}_t = \mathbf{q}$. Então, para cada tempo $t \geq 0$, o próximo estado \mathbf{q}_{t+1} é escolhido, primeiramente, amostrando um ponto candidato \mathbf{q}' da distribuição proposta $q(\cdot | \mathbf{q}_t)$ (a distribuição proposta pode depender do ponto atual, \mathbf{q}_t).

O ponto candidato \mathbf{q}' é então aceito com probabilidade $\mathbf{a}(\mathbf{q}, \mathbf{q}')$ sendo que

$$\mathbf{a}(\mathbf{q}, \mathbf{q}') = \min \left\{ 1, \frac{p(\mathbf{q}')q(\mathbf{q} | \mathbf{q}')}{p(\mathbf{q})q(\mathbf{q}' | \mathbf{q})} \right\}.$$

Se o ponto candidato \mathbf{q}' for aceito, o próximo estado será $\mathbf{q}_{t+1} = \mathbf{q}'$. Se o candidato for rejeitado, a cadeia não se moverá, isto é, $\mathbf{q}_{t+1} = \mathbf{q}$.

Todo algoritmo em que é gerada uma cadeia reversível através de uma região de aceitação da forma acima, será chamado do tipo Metropolis-Hastings. A otimalidade do método pode ser analisada em termos da minimização da variância assintótica dos estimadores de momento.

O algoritmo Metropolis-Hastings (Gilks et al, 1997) pode ser resumido como:

Inicie \mathbf{q}_0 ;

Faça $t=0$

Repita {

Amostre um ponto \mathbf{q}' de $q(\cdot | \mathbf{q})$

Amostre um valor U de uma uniforme (0,1)

Se $U \leq \mathbf{a}(\mathbf{q}, \mathbf{q}')$ faça $\mathbf{q}_{t+1} = \mathbf{q}'$

caso contrário faça $\mathbf{q}_{t+1} = \mathbf{q}$

incremente t

}

$$\text{em que } \mathbf{a}(\mathbf{q}, \mathbf{q}') = \min \left\{ 1, \frac{p(\mathbf{q}')q(\mathbf{q} | \mathbf{q}')}{p(\mathbf{q})q(\mathbf{q}' | \mathbf{q})} \right\}.$$

Em outras palavras, em cada estágio da cadeia, um novo candidato é proposto de acordo com a função geradora. Se este candidato é aceito, entende-se que a cadeia anda (ocorre transição), senão, a cadeia fica parada e o candidato é descartado.

O algoritmo Metropolis-Hastings é bastante geral, e pode, pelo menos a princípio, ser implementado com qualquer distribuição $p(\mathbf{q})$ e para qualquer proposta. Entretanto, sob o ponto de vista prático a escolha da proposta é crucial para efeitos de convergência para a distribuição posteriori.

Casos Especiais

Um caso particular é quando a distribuição proposta não depende do estado atual da cadeia, i.e. $p(\mathbf{q}' | \mathbf{q}) = q(\mathbf{q}')$. Em geral, $q(\cdot)$ deve ter uma aproximação de $p(\cdot)$, mas é mais seguro se $q(\cdot)$ tiver caudas mais pesadas do que $p(\cdot)$. A probabilidade de aceitação agora passa a ser

$$\mathbf{a}(\mathbf{q}, \mathbf{q}') = \min \left\{ 1, \frac{p(\mathbf{q}')q(\mathbf{q})}{p(\mathbf{q})q(\mathbf{q}')} \right\}.$$

Note que embora os valores de \mathbf{q}' sejam gerados de forma independente a cadeia resultante não será independente e identicamente distribuída já que a probabilidade de aceitação ainda depende de \mathbf{q} .

Outro caso particular é chamado algoritmo Metropolis e considera apenas propostas simétricas, i.e., $q(\mathbf{q}'|\mathbf{q}) = q(\mathbf{q}|\mathbf{q}')$ para todos os valores de \mathbf{q} e \mathbf{q}' . Neste caso a probabilidade de aceitação se reduz para

$$\mathbf{a}(\mathbf{q}, \mathbf{q}') = \min\left\{1, \frac{p(\mathbf{q}')}{p(\mathbf{q})}\right\}.$$

Um algoritmo Metropolis muito utilizado é baseado em um passeio aleatório de modo que a probabilidade da cadeia mover-se de \mathbf{q} para \mathbf{q}' depende apenas da distância entre eles, i.e. $q(\mathbf{q}'|\mathbf{q}) = q(\|\mathbf{q} - \mathbf{q}'\|)$. Neste caso, se usarmos uma distribuição proposta com variância \mathbf{s}^2 duas situações extremas podem ocorrer,

1. se \mathbf{s}^2 for muito pequena os valores gerados estarão próximos do valor atual e quase sempre serão aceitos. Mas levará muitas iterações até o algoritmo cobrir todo o espaço do parâmetro;
2. valores grandes de \mathbf{s}^2 levam a uma taxa de rejeição excessivamente alta e a cadeia se movimenta muito pouco.

Nas duas situações o algoritmo torna-se ineficiente e na prática temos que tentar vários valores de \mathbf{s}^2 .

De um modo geral $\mathbf{q} = (q_1 \dots q_d)'$ será um vetor de parâmetros de dimensão d . Neste caso, pode ser computacionalmente mais eficiente dividir \mathbf{q} em k blocos $\{q_1 \dots q_k\}$ e dentro de cada iteração teremos o algoritmo aplicado k vezes. Definindo o vetor $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_k)$ que contém todos os elementos de x exceto x_i suponha que a iteração $t+1$ os blocos $1, 2, \dots, i-1$ já foram atualizados i.e.

$$x_{-i} = (x_1^{(t+1)}, \dots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \dots, x_k^{(t)}).$$

Para atualizar a i -ésima componente, um valor de x_i é gerado da distribuição proposta $q(\cdot | x_i, x_{-i})$ e este valor candidato é aceito com probabilidade

$$\mathbf{a}(x_i, x_i') = \min \left\{ 1, \frac{p(x_i' | x_{-i})q(x_i | x_i', x_{-i})}{p(x_i | x_{-i})q(x_i' | x_i, x_{-i})} \right\}.$$

Exemplo: Numa certa população de animais sabe-se que cada animal pode pertencer a uma dentre quatro linhagens genéticas com probabilidades

$$p_1 = \frac{1}{2} + \frac{\mathbf{q}}{2}, p_2 = \frac{1-\mathbf{q}}{4}, p_3 = \frac{1-\mathbf{q}}{4}, p_4 = \frac{\mathbf{q}}{4}.$$

sendo $0 < \mathbf{q} < 1$ um parâmetro desconhecido. Para qualquer $\mathbf{q} \in (0,1)$ é fácil verificar que $p_i > 0, i=1,2,3,4$ e $p_1 + p_2 + p_3 + p_4 = 1$. Observando-se n animais dentre os quais y_i pertencem à linhagem i estão o vetor aleatório $Y = (y_1, y_2, y_3, y_4)$ tem distribuição multinomial com parâmetros n, p_1, p_2, p_3, p_4 e portanto

$$p(y | \mathbf{q}) = \frac{n!}{y_1! y_2! y_3! y_4!} p_1^{y_1} p_2^{y_2} p_3^{y_3} p_4^{y_4}$$

$$\mathbf{a} (2 + \mathbf{q})^{y_1} (1 - \mathbf{q})^{y_2 + y_3} \mathbf{q}^{y_4}$$

Atribuindo uma priori $\mathbf{q} \sim U(0,1)$ segue que a posteriori é proporcional à expressão acima.

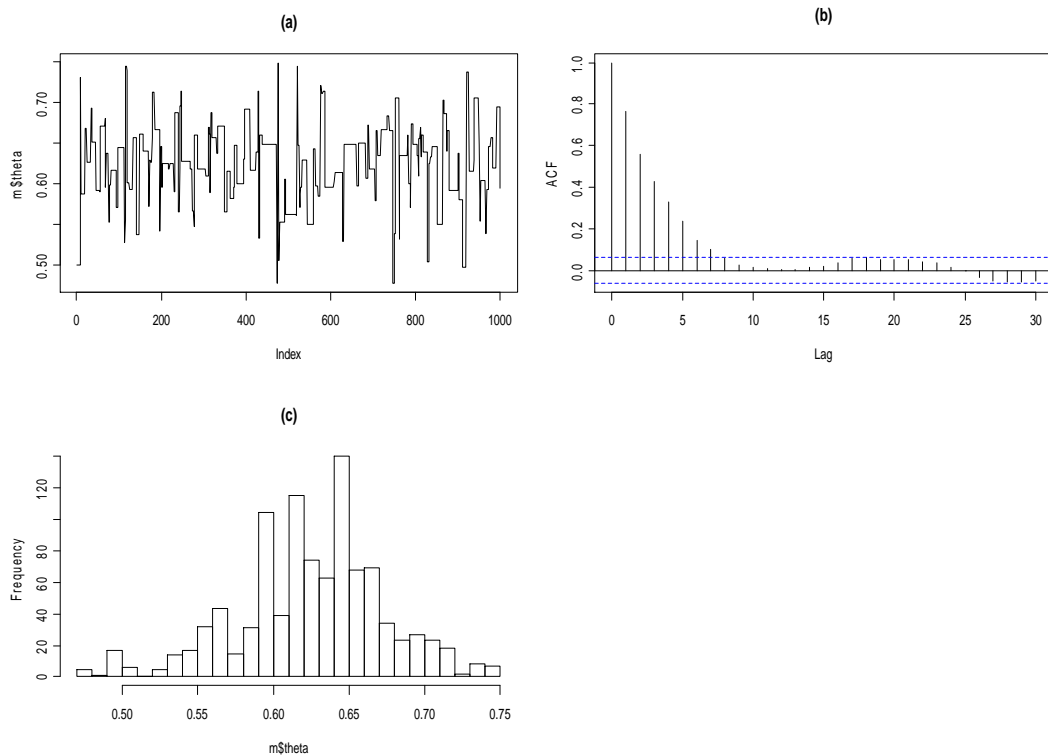
Tomando a distribuição $U(0,1)$ como proposta então $q(\mathbf{q}) = 1, \forall \mathbf{q}$ e a probabilidade se simplifica para

$$\mathbf{a}(\mathbf{q}, \mathbf{q}') = \min \left\{ 1, \frac{p(x | \mathbf{q}')}{p(x | \mathbf{q})} \right\} = \min \left\{ 1, \left(\frac{2 + \mathbf{q}'}{2 + \mathbf{q}} \right)^{y_1} \left(\frac{1 - \mathbf{q}'}{1 - \mathbf{q}} \right)^{y_2 + y_3} \left(\frac{\mathbf{q}'}{\mathbf{q}} \right)^{y_4} \right\}.$$

Foram observados 197 animais com os números de animais nas categorias dados por $y=(125,18,20,34)$ e foi gerada uma cadeia de Markov com 1000 valores de ?. Os valores simulados e as primeiras 30 autocorrelações amostrais de ? estão na Figura 2. A cadeia parece ter convergido após algumas iterações e podemos descartar os 100 primeiros valores (esta foi a nossa amostra de aquecimento-“burn-in”). Note também que a cadeia é altamente

correlacionada ao longo das iterações e isto é devido à alta taxa de rejeição por causa da escolha de q . Os resultados foram obtidos através do programa R.

FIGURA 2: (a) 1000 valores simulados de θ . (b) 30 primeiras autocorrelações amostrais após aquecimento. (c) histograma dos valores simulados após aquecimento.



Exemplo 2: Suponha que queremos simular valores $X \sim N(0,1)$ propondo valores $Y \sim N(x, s^2)$. Neste caso as densidades propostas no numerador e denominador de

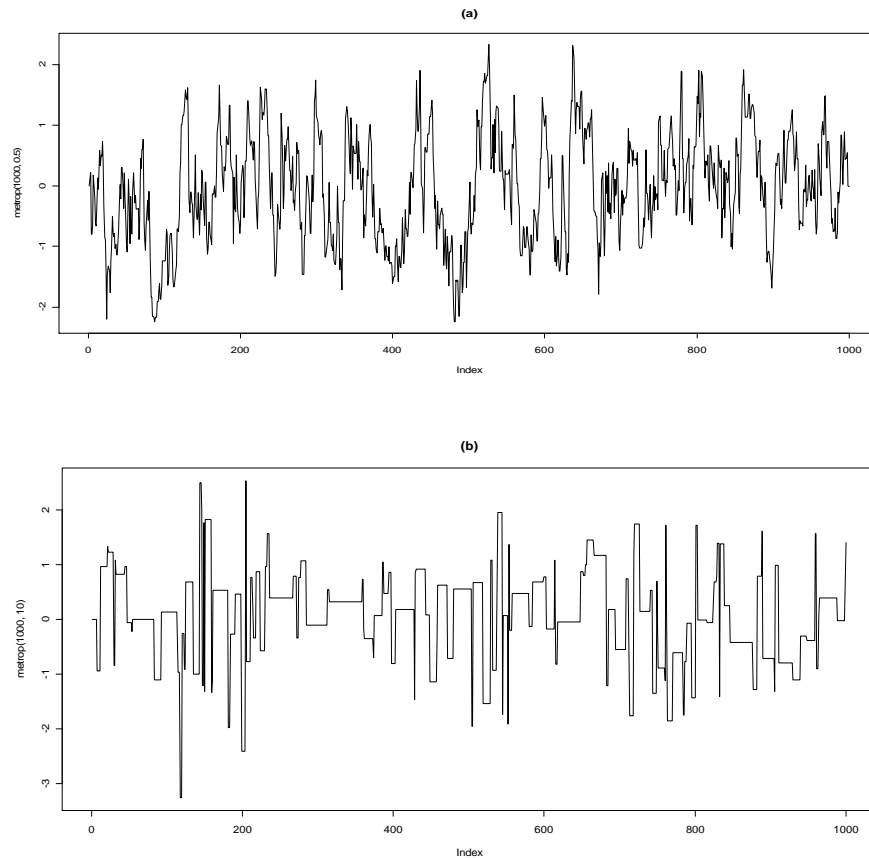
$\mathbf{a}(\mathbf{q}, \mathbf{q}') = \min \left\{ 1, \frac{p(\mathbf{q}')q(\mathbf{q} | \mathbf{q}')}{p(\mathbf{q})q(\mathbf{q}' | \mathbf{q})} \right\}$ se simplificam e a probabilidade de aceitação fica

$$\mathbf{a}(x, x') = \min \left\{ 1, \exp \left(-\frac{1}{2} (y^2 - x^2) \right) \right\}.$$

Fixando os valores $s=0,5$ e $s=10$ foram simuladas as cadeias que aparecem na Figura 3. Note que o valor de s teve um grande impacto na taxa de aceitação do algoritmo. Isto ocorre

porque com $s=0,5$ a distribuição proposta está muito mais próxima da distribuição de interesse do que com $s=10$.

Figura 3: 1000 valores simulados para o Exemplo 2 usando o algoritmo de Metropolis-Hastings com (a) $s=0,5$ e (b) $s=10$.



3.4. Amostrador de Gibbs

O amostrador de Gibbs é um caso especial do Metropolis-Hastings em que o valor aleatório é sempre aceito. Resta a tarefa de especificar como construir uma cadeia de Markov em que valores convergem para a distribuição de interesse. A chave para o amostrador de Gibbs é se considerar a distribuição condicional univariada, quando todas as variáveis aleatórias são mais fáceis de simular do que distribuições conjuntas complexas. Também nesse caso, tem formas simples (normais, qui-quadradas inversas ou outras distribuições a

priori comuns). Assim, simular n variáveis aleatórias seqüencialmente para as n condicionais univariadas que geram um vetor simples n -dimensional em uma passagem simples usando a distribuição conjunta completa.

Para introdução do amostrador de Gibbs, considere uma variável aleatória bivariada (x, y) , e suponha que desejamos calcular uma ou ambas as marginais, $p(x)$ e $p(y)$. A idéia do amostrador é simples, considerar uma seqüência de distribuições condicionais, $p(x|y)$ e $p(y|x)$, que é obtida da marginal por integração da densidade conjunta $p(x, y)$, e $p(x) = \int p(x, y)dy$. A amostragem inicia com algum valor y_0 para y e obtém-se x_0 como uma observação da variável aleatória da distribuição condicional $p(x|y = y_0)$. A amostragem então usa x_0 para gerar um novo valor para y_1 , esboçando a distribuição condicional baseada no valor x_0 , $p(y|x = x_0)$. A amostragem continua conforme

$$x_i \sim p(x|y = y_{i-1})$$

$$y_i \sim p(y|x = x_i)$$

Repetindo este processo k vezes, gerando uma seqüência de Gibbs para um k grande, aonde um subconjunto de pontos (x_j, y_j) para $1 \leq j \leq m < k$ são tirados da nossa simulação esboçada da distribuição conjunta completa. Para obter o total de m amostras de interesse (onde cada ponto na amostra é um vetor de dois parâmetros), uma amostra da cadeia (i) depois de um aquecimento suficiente para remover os efeitos do valor inicial e (ii) no conjunto de interesse (sempre n amostras) seguintes ao aquecimento. A seqüência de Gibbs converge para uma distribuição estacionária (equilíbrio) que é independente dos valores iniciais e a construção desta distribuição estacionária gera a distribuição de interesse que estamos tentando simular.

Exemplo: A distribuição a seguir foi obtida em Casella e George (1992). Suponha que a distribuição conjunta para $x = 0, 1, \dots, n$ e $0 \leq y \leq 1$ é dada por

$$p(x, y) = \frac{n!}{(n-x)!x!} y^{x+a-1} (1-y)^{n-x+b-1}$$

Note que x é discreto enquanto y é contínuo. Enquanto a densidade conjunta é complexa, as densidades condicionais são distribuições simples. Para verificar isto, primeiro vamos lembrar que a variável aleatória binomial z tem uma densidade proporcional

$$p(z | q, n) \propto \frac{q^z (1-q)^{n-z}}{z!(n-z)!} \text{ para } 0 \leq z \leq n$$

onde $0 < q < 1$ é o parâmetro de sucesso e n o número de ações, e denotamos $z \sim B(n, p)$. Da mesma forma lembramos que a densidade para $z \sim \text{Beta}(a, b)$, uma distribuição beta com modelo de parâmetro a e b dado por

$$p(z | a, b) \propto z^{a-1} (1-z)^{b-1} \text{ para } 0 \leq z \leq 1$$

Com estas distribuições de probabilidade em mãos, note que a distribuição condicional para x (tratando y como uma constante fixa) é $x | y \sim B(n, y)$, enquanto $y | x \sim \text{Beta}(x + \mathbf{a}, n - x + \mathbf{b})$.

O poder do Amostrador de Gibbs é que estimando uma seqüência dessas variáveis aleatórias condicionais univariadas (uma binomial e outra beta) pode-se estimar algum aspecto de outra distribuição marginal. Supondo $n=10$ e $a=1, \beta=2$. Começando a amostragem com $y_0 = 1/2$, e tomando a amostra através de três iterações completas.

- (i) x_0 é obtido gerando variável aleatória $B(n, y_0) = B(10, 1/2)$, fornecendo $x_0 = 5$ em nossa simulação.
- (ii) y_1 é obtido de uma variável aleatória $\text{Beta}(x_0 + \mathbf{a}, n - x_0 + \mathbf{b}) = \text{Beta}(5 + 1, 10 - 5 + 2)$, fornecendo $y_1 = 0,33$.

(iii) x_1 é a realização de uma variável aleatória $B(n, y_1) = B(10, 0.33)$, fornecendo $x_1 = 3$.

(iv) y_2 é obtido de uma variável aleatória $Beta(x_1 + \mathbf{a}, n - x_1 + \mathbf{b}) = Beta(3 + 1, 10 - 3 + 2)$, fornecendo $y_2 = 0,56$.

(v) x_2 é a realização de uma variável aleatória $B(n, y_2) = B(10, 0.56)$, fornecendo $x_2 = 7$.

Nossa particular realização da seqüência de Gibbs depois de três iterações é da forma (5 , 0.5), (3 , 0.33), (7 , 0.56). Podemos continuar este processo para gerar uma cadeia longa. Obviamente, os valores iniciais da cadeia são altamente dependentes do valor y_0 escolhido para iniciar a cadeia. Esta dependência diminui conforme a seqüência cresce e começamos a registrar a seqüência depois um número suficiente de iterações de aquecimento ter ocorrido para remover algum efeito das condições iniciais.

Quando duas ou mais variáveis são usadas, a amostragem é estendida de maneira óbvia. O valor da k -ésima variável é descrita como a distribuição $p(\mathbf{q}^{(k)} | \Theta^{(-k)})$ onde $\Theta^{(-k)}$ denota um vetor contendo todas as variáveis com exceção da variável k . Entretanto, durante a i -ésima iteração da amostra, para obter o valor de $p(\mathbf{q}^{(k)} | \Theta^{(-k)})$ descreve-se a distribuição da forma

$$\mathbf{q}_i^{(k)} \sim p(\mathbf{q}^{(k)} | \mathbf{q}^{(1)} = \mathbf{q}_i^{(1)}, \dots, \mathbf{q}^{(k-1)} = \mathbf{q}_i^{(k-1)}, \dots, \mathbf{q}^{(k+1)} = \mathbf{q}_{i-1}^{(k+1)}, \dots, \mathbf{q}^{(n)} = \mathbf{q}_{i-1}^{(n)})$$

Por exemplo, se temos quatro variáveis, (w, x, y, z) , o amostrador será

$$w_i \sim p(w | x = x_{i-1}, y = y_{i-1}, z = z_{i-1})$$

$$x_i \sim p(x | w = w_i, y = y_{i-1}, z = z_{i-1})$$

$$y_i \sim p(y | w = w_i, x = x_i, z = z_{i-1})$$

$$z_i \sim p(z | w = w_i, x = x_i, y = y_i)$$

Gelfand e Smith (1990) ilustram o amostrador de Gibbs para tratar uma ampla variedade de estatísticas, enquanto Smith e Roberts (1993) mostram o casamento do Amostrador de Gibbs com a estatística Bayesiana (para obter distribuições a posteriori). Uma boa introdução é dada por Casella e George (1992), enquanto mais detalhes pode ser obtidos em Tanner (1996), Besag et al. (1995), e Lee (1997).

Usando o Amostrador de Gibbs para aproximação de Distribuições Marginais:

Algum aspecto de interesse para as marginais pode ser calculado através das m realizações da seqüência de Gibbs. Por exemplo, a esperança da função f da variável aleatória x é aproximada por

$$E[f(x)]_m = \frac{1}{m} \sum_{i=1}^m f(x_i)$$

Esta é a aproximação de Monte Carlo para $f(x)$, pois $E[f(x)]_m \rightarrow E[f(x)]$ quando $m \rightarrow \infty$. Da mesma forma, a estimação de Monte Carlo para outras funções com n variáveis $(\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(n)})$, é dada por

$$E[f(\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(n)})]_m = \frac{1}{m} \sum_{i=1}^m f(\mathbf{q}_i^{(1)}, \dots, \mathbf{q}_i^{(n)})$$

Continuação do exemplo:

Embora a seqüência de tamanho 3 calculada anteriormente seja muito pequena para ser propriamente uma seqüência de Gibbs, como ilustração podemos utilizá-la para o cálculo de estimativas de Monte Carlo. As estimativas das médias para x e y são

$$\bar{x}_3 = \frac{5+3+7}{3} = 5, \quad \bar{y}_3 = \frac{0,5+0,33+0,56}{3} = 0,46$$

Similarmente $(\bar{x}^2)_3 = 27,67$ e $(\bar{y}^2)_3 = 0,22$, fornecendo as estimativas para a variância de x e y

$$Var(x)_3 = (\bar{x}^2)_3 - (\bar{x}_3)^2 = 2,67$$

$$Var(y)_3 = (\bar{y}^2)_3 - (\bar{y}_3)^2 = 0,25$$

3.5. Avaliação da Convergência

Os métodos MCMC são uma ótima ferramenta para resolução de muitos problemas práticos para análise Bayesiana. Porém, algumas questões relacionadas à convergência nestes métodos ainda merecem bastante pesquisa.

Entretanto, uma questão que pode surgir é “quantas iterações deve ter o processo de simulação para garantir que a cadeia convergiu para o estado equilíbrio?”. Como a cadeia não é inicializada na distribuição estacionária, uma prática comum é usar um período de aquecimento (“burn-in”) (Gilks et al., 1996). A cadeia é rodada por $L + M$ iterações, sendo as primeiras L iterações iniciais descartadas. Espera-se que depois deste período de aquecimento a cadeia tenha esquecido os valores iniciais e convergido para a distribuição de equilíbrio; a amostra resultante de tamanho M será uma amostra da distribuição de equilíbrio. Para eliminar uma possível auto-correlação das cadeias selecionadas a partir do aquecimento a cada k iterações, o tamanho de k será chamada de *lag*.

O grau de correlação da amostra final afetará a acurácia do estimador de Monte Carlo baseado na amostra. O conceito de tempo de autocorrelação (Madras e Sokal, 1998) é usado para quantificar este efeito. Assumindo que a cadeia tenha alcançado o equilíbrio, seja \mathbf{q}^t o valor da cadeia no tempo t . A autocorrelação $\mathbf{r}_g(k)$ no *lag* k para alguma função $g(\mathbf{q})$ é definida por

$$\mathbf{r}_g(k) = \frac{E[g(\mathbf{q}^t)g(\mathbf{q}^{t+k})] - E[g(\mathbf{q}^t)]^2}{V[g(\mathbf{q}^t)]}$$

A esperança é com relação à densidade $\mathbf{p}(\mathbf{q}^t)$. O tempo de autocorrelação, \mathbf{t}_g para a função g é definida como

$$\mathbf{t}_g = \sum_{-\infty}^{\infty} \mathbf{r}_g(k)$$

Se $M \gg \mathbf{t}_g$, então a aproximação para a variância de

$$\frac{1}{M} \sum_{t=1}^M g(\mathbf{q}_t),$$

o estimador da esperança de $g(\mathbf{q})$, é $V[g(\mathbf{q})]t_g / M$ o qual é maior que o estimador baseado em uma amostra tamanho M de observações independentes e identicamente distribuídas. Em outras palavras, o número efetivo de amostras independentes em uma cadeia de tamanho M é aproximadamente M/t_g . Métodos para estimar t_g da saída do MCMC podem ser encontrados em Geyer (1992).

Observe que o tempo de autocorrelação é uma estimativa da eficiência da cadeia de Markov uma vez que alcança o equilíbrio, e portanto, a estacionariedade e não uma estimativa de quantas iterações são necessárias para a cadeia alcançar a distribuição estacionária.

Para avaliar a convergência dos métodos MCMC faz-se uso de alguns critérios que existem na literatura. As técnicas mais populares são as de Geweke (1992) e Heidelberger e Welch (1983) que usam resultados baseados em análise espectral, Raftery e Lewis (1992) que permite calcular quantas iterações são necessárias para uma cadeia atingir a distribuição estacionária através da estimação de quantis posteriores com uma precisão previamente fixada e Gelman e Rubin (1992) que usa resultados baseados na análise de variância clássica para duas ou mais cadeias simuladas com valores iniciais diferentes. Estes métodos e outros foram comparados no trabalho de Cowles e Carlin (1996), aonde se chegou a conclusões de que não se pode afirmar qual deles é o mais eficiente. As técnicas de Geweke, Heidelberger-Welch, Raftery-Lewis, Gelman-Rubin e outras estão implementadas no pacote CODA (Cowles e Carlin, 1996) executável no R.

Também podemos fazer uma avaliação informal de convergência (Gelfand e Smith, 1990). Eles sugeriram técnicas gráficas para verificação da convergência. Após um número suficientemente grande N de iterações em m cadeias paralelas, forma-se uma amostra de n e pode-se construir um histograma de qualquer uma de suas componentes (ou suas funções).

Esse histograma pode ser suavizado através de alguma técnica de alisamento ou não. O mesmo procedimento pode ser repetido após $N+k$ iterações. Se não houver diferença perceptível a olho nu entre os gráficos obtidos após N e $N+k$ iterações, então se conclui pela convergência das cadeias. O valor de k não pode ser muito pequeno, pois a correlação inerente à cadeia de Markov estará exercendo sua influência e não se poderá dizer que a similaridade é devida a convergência ou à correlação do processo. Valores de k muito grande são desnecessários, pois se há suspeita de convergência após N iterações não há necessidade de ir muito além na cadeia apenas para uma verificação. Tipicamente, valores entre 10 e 50 iterações são apropriadas.

Ainda na linha de verificação gráfica é possível observar a trajetória de uma única cadeia ao longo das iterações. Se o gráfico após um período inicial apresenta repetidamente o mesmo comportamento qualitativo e quantitativo então pode se concluir pela convergência da cadeia. Similarmente, pode-se usar uma versão gráfica de teorema ergódico e plotar as médias ergódicas ao invés dos valores gerados. Convergência é verificada se o gráfico demonstra um comportamento assintótico.

Essas técnicas de monitoração devem ser usadas com muita cautela e sempre acompanhadas de alguma fundamentação teórica. Técnicas gráficas podem ser ilusórias indicando uma constância que pode não ser tão evidente sob outra escala. Além disso, muitas cadeias podem exigir um comportamento similar ao de convergência sem que a convergência tenha sido atingida.

CAPÍTULO 4 – APLICAÇÃO: ANÁLISE CUSTO-EFICIÊNCIA DA PRÓTESE DE REPOSIÇÃO DA BACIA

4.1. Justificativa

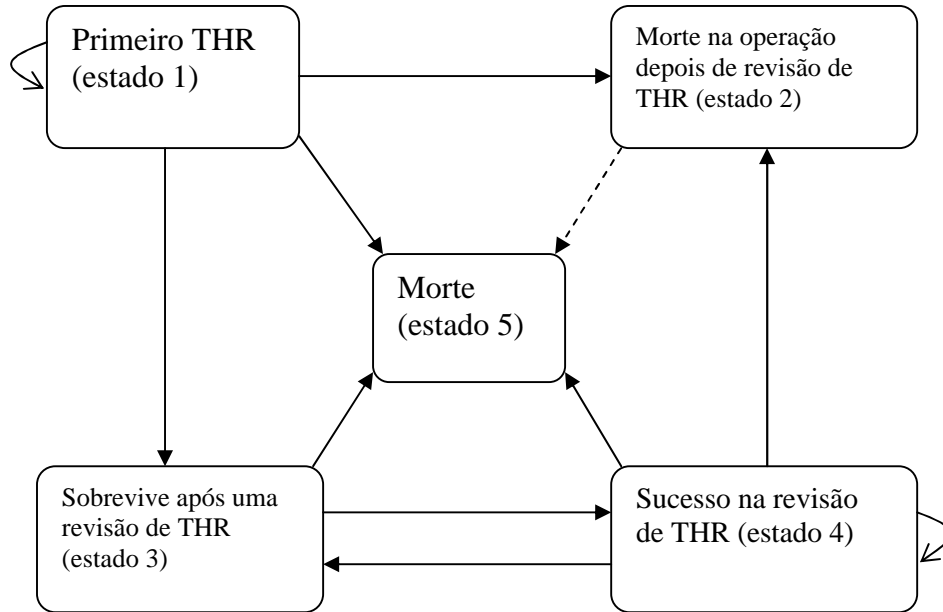
Nosso exemplo em curso diz respeito à seleção de próteses na reposição total da bacia: este é um procedimento ortopédico muito comum, com um potencial benéfico em termos de alívio de dor e melhora das funções físicas. A bacia é a articulação em que o osso da coxa (fêmur) se une ao osso do quadril (pélvis). A substituição total da bacia é uma cirurgia que repõe as partes desgastadas ou danificadas da articulação da bacia. As superfícies danificadas da articulação são removidas e substituídas por uma articulação artificial. Entretanto, há vários produtos avaliados e sendo usados, com limitada evidencia de sua eficiência relativa, particularmente em termos de suas taxas de revisão para diferentes subpopulações. Visto que próteses variam consideravelmente no custo, o National Institute of Clinical Evidence for England and Wales (NICE) do Reino Unido, tem-se direcionado para o custo-eficiência de diferentes tipos de próteses, fazendo uso substancial de análises prévias apresentada por Fitzpatrick (1991) e outros.

Aqui usamos o modelo para consequência seguinte a reposição da bacia que é baseado em Fitzpatrick e ilustra a estrutura aproximada para as várias origens de incerteza. Ele usa evidências nas taxas de revisão relativas para diferentes próteses citadas na avaliação de NICE. Entretanto, nosso propósito é no desenvolvimento da metodologia estatística, e assim nossos resultados não devem ser usados como contribuição de qualquer maneira para escolha da prótese apropriada.

4.2. Modelo Estatístico

Nosso modelo para predição do custo - benefício seguindo a reposição da bacia é um modelo de Markov discreto a tempo discreto. O primeiro ciclo ($t=1$) é assumido para iniciar imediatamente depois da primeira operação de reposição total da bacia (THR); pacientes podem também morrer na operação ou pós-operação; em cada caso eles entram no estado 5 (morte), de outro modo podem ficar no estado 1. Em ciclos subsequentes, pacientes sobreviventes ficam no estado 1 até morrerem por outras causas (passando ao estado 5), ou sua reposição de bacia falha e exigem uma operação THR de revisão. Pacientes sofrendo uma operação de revisão entram em um ou dois estados, dependendo de sua morte pós-operatória (estado 2) ou sobrevivida (estado 3). Pacientes sobreviventes passam para o estado 4 (sucesso na revisão THR) no ciclo seguinte, a menos que morram de outras causas (passam para o estado 5). Pacientes no estado 4 permanecem ali até também morrer por outras causas (estado 5) ou necessitar de outra operação de revisão THR. Em cada caso este progresso volta ao estado 2 ou 3 mais adiante. Também assumimos a transição do estado 2 para o estado 5 no ciclo seguinte à morte operatória depois da revisão THR. Isto pode parecer artificial, mas é necessário para evitar múltiplas contagens para revisão de custos no momento que pacientes ficam no estado 2. A Figura 4 ilustra os vários estados e as possíveis transições entre estados.

Figura 4: Modelo Markoviano para resultados após reposição total da bacia



Transições entre estados são definidas em estrutura de tempo (ciclo longo) de um ano. O vetor de estados de probabilidade no ciclo $t=1$ é $\mathbf{p}_1 = (1 - I_{op}, 0, 0, 0, I_{op})$. Então consideramos um avanço de 59 ciclos para o modelo, escolhendo garantir que pacientes no grupo idade jovem com $t=1$ deveriam morrer no fim de 60 ciclos (anos). A matriz Λ de transição de probabilidades para $t=2, \dots, 60$ é mostrada abaixo. Aqui Λ_{ijk} é a probabilidade de estar no estado j no ano $t-1$ e se movendo para o estado k no início do ano t ; I_{op} é a taxa de mortalidade na operação; \mathbf{g}_t é o risco para revisão no ano t ; I_t é a taxa de mortalidade em t anos depois da primeira operação e \mathbf{r} é a taxa de revisão, que é assumida como constante.

$$\Lambda = \begin{bmatrix} 1 - \mathbf{g}_t - I_t & I_{op} \mathbf{g}_t & (1 - I_{op}) \mathbf{g}_t & 0 & I_t \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 - I_t & I_t \\ 0 & \mathbf{r} I_{op} & \mathbf{r} (1 - I_{op}) & 1 - \mathbf{r} - I_t & I_t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4.3. Parâmetros do Modelo

Seguiremos Fitzpatrick adotando o uso da prótese de Charnley como análise base, assumindo como uma constante na taxa de mortalidade pós-operatória, $I_{op}=0,01$, e a constante taxa de revisão, $r=0,04$. Além disso, assumiremos o aumento linear no risco de revisão $g_t = h(t-1)$ (i.e. sem reposição no primeiro ano), mas diferente de Fitzpatrick, deixaremos o aumento anual h para depender da idade e sexo do paciente. Como base da taxa de revisão para prótese de Charnley nos registros de reposição da bacia suíço, e assumindo maior taxa de revisão para homens e pessoas jovens, teremos $h=0,0022$ para homens < 65 anos, $h=0,0017$ para mulheres < 65 anos, $h=0,0016$ para homens = 65 anos e $h=0,0012$ para mulheres = 65 anos. Finalmente assumiremos que pacientes sobreviventes a operação THR terão a mesma taxa de mortalidade da população em geral e usaremos as taxas nacionais de sexo-específico por idade da UK Office for National Statistics, reproduzidos na Tabela 4. Também é mostrada na Tabela 4 a distribuição idade-sexo para pacientes que receberam o primeiro THR usando a prótese Charnley no Reino Unido. Estes dados serão usados na análise permitindo heterogeneidade para pacientes.

Tabela 4: Taxa de mortalidade de sexo-específico por idade, e distribuição por sexo e idade de pacientes que receberam o primeiro THR no Reino Unido

	Taxa de Mortalidade		% receptores de THR	
	Homens	Mulheres	Homens	Mulheres
<45	0,0017	0,0011	2%	2%
45-54	0,0044	0,0028	3%	4%
55-64	0,0138	0,0081	7%	10%
65-74	0,0379	0,0220	13%	22%
75-84	0,0912	0,0578	10%	26%
>84	0,01958	0,1503	0%	1%

4.4. Custo - Benefício

Estimativas dos custos de primeira operação e da operação de revisão de THR usando a prótese de Charnley foram obtidas de Fitzpatrick. Para um paciente típico, o custo inicial do THR é $C_p = \text{£}4052$, e o custo de revisão do THR é $C_p = \text{£}5290$, e usamos uma taxa de desconto para custos ocorridos nos anos futuros de $d_c = 6\%$ por ano.

A qualidade de vida relativa (HRQL) é medida pela qualidade ajustada de vida por anos (QALYs) se baseia no grau de severidade da dor que os pacientes sentem em diferentes estados do modelo. Baseado nos resultados de estudos Canadenses, Fitzpatrick, assume valores $u_1 = 1$, $u_2 = 0,69$, $u_3 = 0,38$, $u_4 = 0,19$ para os HRQL dos pacientes nos estados sem dor, dor suave, moderada ou severa, respectivamente. Assume-se que depois de uma operação THR bem sucedida, 80% dos pacientes não sentiram dor e 20% sentiram dor moderada. Para pacientes que pertencem a reposição de bacia com falha, assume-se que 15% sentem dor

severa e 85% sentem dor moderada no ano que precede o ano da operação de revisão, e entre aqueles que sentiram dor moderada e dor severa no ano da operação tivemos 50% para ambos os grupos. Calculamos então QALYs para cada estado em nosso modelo de Markov como segue:

$$QALY_1 = 0,8u_1 + 0,2u_2 = 0,938$$

$$QALY_2 = 0 + 1,06(0,85u_3 + 0,14u_4 - 0,8u_1 - 0,2u_2) = -0,622$$

$$QALY_3 = (u_3 + u_4) / 2 + 1,06(0,85u_3 + 0,15u_4 - 0,8u_1 - 0,2u_2) = -0,377$$

$$QALY_4 = 0,8u_1 + 0,2u_2 = 0,938$$

$$QALY_5 = 0$$

Notamos de forma anômala o aparecimento de valores negativos para os estados 2 e 3, os quais representam uma subtração da qualidade para o ano que precede os pacientes que precisaram de uma operação de revisão. Para custos, descontamos QALYs (e também, expectativa de vida) nos anos futuros a taxa de $d_b = 6\%$ por ano: entretanto, supor uma taxa de desconto diferente para benefícios e custos pode ser mais razoável.

Na Tabela 5 temos os custos esperados e benefícios para cada subgrupo, calculados ambos de forma fechada e via simulação de Monte Carlo (calculada através do software WinBUGS). Estimativas de Monte Carlo da variabilidade são expressas pelos desvios padrões amostrais para estes custos e benefícios. As estimativas simuladas baseadas nas expectativas concordam bem com o resultado exato para cada subgrupo, e o desvio padrão mostra variabilidade substancial entre efeitos atingido por pacientes individualmente. Enquanto os custos esperados são razoavelmente constantes nos subgrupos, há clara heterogeneidade nos benefícios esperados.

Agora poderemos calcular todas as esperanças $m_{s,q}$ na forma fechada, ignorando a variabilidade entre indivíduos homogêneos.

4.5. Análise Probabilística de Sensibilidade

4.5.1. Sensibilidade para heterogeneidade de pacientes: caso parâmetro θ fixo

Supondo que desejamos uma medida global do custo-eficiência através da população completa, mas com um resumo da variabilidade devido à heterogeneidade dos pacientes. Estamos inclinados nesta etapa a considerar o parâmetro θ do modelo ser conhecido. Permitindo s ter uma distribuição $p(s | \mathbf{q})$ com média e variância:

$$E_{s|\mathbf{q}} [m_{s\mathbf{q}}] = m_{\mathbf{q}}$$

$$Var_{s|\mathbf{q}} [m_{s\mathbf{q}}] = v_{\mathbf{q}}$$

onde $E_{s|\mathbf{q}}$ representa uma esperança com respeito a distribuição de s para θ fixo. Dessa forma $m_{\mathbf{q}}$ e $v_{\mathbf{q}}$ são a média e a variância do resultado esperado para sub-populações, para θ fixo. Se $m_{s\mathbf{q}}$ é viável na forma fechada para cada um dos conjuntos finitos dos subgrupos, então $m_{\mathbf{q}}$ e $v_{\mathbf{q}}$ podem ser obtidos pelo cálculo direto.

Para o exemplo de prótese de bacia, as sub-populações S compreendem os grupos por idade e sexo mostrados na Tabela 4. Cálculos para $m_{\mathbf{q}}$ e $v_{\mathbf{q}}$ podem ser mostrados computando os pesos médios e variâncias de ambos na forma fechada ou esperanças por Monte Carlo para cada subgrupo, onde os pesos são dados por porcentagens do subgrupo na Tabela 5. Ao final das três linhas da Tabela 5 mostra-se, respectivamente $m_{\mathbf{q}}$, $\sqrt{v_{\mathbf{q}}}$ e o coeficiente de variação $\sqrt{v_{\mathbf{q}}} / m_{\mathbf{q}}$ para cada um dos três resultados de interesse; estes resumos de custos e benefícios esperados de THR e suas variabilidades dos subgrupos de pacientes. Estas medidas enfatizam a consistência razoável dos custos, mas variabilidade substancial dos benefícios dos subgrupos.

4.5.2. Sensibilidade para parâmetros incertos de subgrupos de pacientes fixando s

Agora vamos considerar uma situação na qual estamos interessados nos subgrupos individuais, mas queremos resumir as conseqüências para parâmetros incertos. Tendo a distribuição $p(\mathbf{q} | s)$ para cada subgrupo s fornecendo

$$E_{\mathbf{q}|s} [m_{sq}] = m_s$$

$$Var_{\mathbf{q}|s} [m_{sq}] = v_s$$

onde $E_{\mathbf{q}|s}$ representa uma esperança com respeito a distribuição de \mathbf{q} para um dado subgrupo s . Desse modo m_s e v_s são a média e a variância dos resultados esperados para subpopulações específicas assumindo para incerteza \mathbf{q} .

Assumindo que m_{sq} é viável na forma fechada, m_s e v_s podem ser estimados simulando valores para \mathbf{q} de $p(\mathbf{q} | s)$, avaliando m_{sq} e tomando a média amostral e variância v_s . Esta é a aplicação natural do método de Monte Carlo para uma população homogênea, a qual se tornou um instrumento padrão na análise de risco. Isto está implementado como macros para Excel, também em softwares comerciais como RISK e Crystal Ball. Também pode ser usado gratuitamente o software WINBUGS. Os resultados aparecem a seguir. A terceira coluna para cada resultado fornece o desvio padrão amostral, estimado através de integração de Monte Carlo.

Tabela 5: Custos e benefícios esperados de THR para subgrupos de pacientes com parâmetros fixos, calculados exatamente e usando simulação de Monte Carlo, e global, permitindo subgrupos heterogêneos com parâmetros fixos..

Subgrupos	Custos (£)			Expectativa de Vida (anos)			QALYS		
	Exato	Monte Carlo		Exato	Monte Carlo		Exato	Monte Carlo	
	$m_{sq}^{[C]}$	$m_{sq}^{[C]}$	$\sqrt{v_{sq}^{[C]}}$	$m_{sq}^{[L]}$	$m_{sq}^{[L]}$	$\sqrt{v_{sq}^{[L]}}$	$m_{sq}^{[Q]}$	$m_{sq}^{[Q]}$	$\sqrt{v_{sq}^{[Q]}}$
Homens									
35-44	5.781	5.793	1.892	14,5	14,5	2,9	13,2	13,2	2,6
45-54	5.417	5.435	1.889	12,7	12,7	3,4	11,6	11,6	3,0
55-64	4.989	4.974	1.659	10,3	10,3	3,7	9,5	9,5	3,3
65-74	4.466	4.454	1.211	7,7	7,7	3,5	7,2	7,2	3,2
75-84	4.263	4.250	905	5,4	5,4	3,0	5,0	5,1	2,8
>84	4.193	4.203	806	4,1	4,1	3,0	3,8	3,9	2,7
Mulheres									
35-44	5.626	5.641	1.835	15,1	15,2	2,6	13,8	13,8	2,4
45-54	5.350	5.346	1.765	13,7	13,7	3,0	12,5	12,6	2,8
55-64	5.002	5.020	1.666	11,6	11,6	3,6	10,7	10,7	3,2
65-74	4.487	4.484	1.242	9,1	9,0	3,7	8,4	8,4	3,4
75-84	4.282	4.277	955	6,5	6,4	3,3	6,0	6,0	3,1
>84	4.212	4.209	812	5,0	5,0	3,4	4,6	4,6	3,2
Global	Exato	Monte Carlo		Exato	Monte Carlo		Exato	Monte Carlo	
Média, m_q	4.603	4.600		8,7	8,7		8,0	8,0	
SD, $\sqrt{v_q}$	403	409		2,6	2,6		2,3	2,3	
CV= $\frac{\sqrt{v_q}}{m_q}$	0,09	0,09		0,30	0,30		0,29	0,29	

4.5.3. Sensibilidade associada para parâmetros incertos e heterogeneidade

Quando desejamos simultaneamente investigar sensibilidade para heterogeneidade e parâmetros incertos, precisamos então considerar a distribuição associada $p(s, \mathbf{q})$ que fornece o resumo estatístico

$$E_{sq} [m_{sq}] = m_s$$

$$Var_{sq} [m_{sq}] = v_s$$

que quantificam a esperança e variância dos resultados dos subgrupos de pacientes e possíveis parâmetros.

Os resumos globais m e v podem ser obtidos de duas maneiras, correspondente expressando $p(s, \mathbf{q})$ como $p(s | \mathbf{q})p(\mathbf{q})$ ou como $p(\mathbf{q} | s)p(s)$.

1. Podemos condicionar os parâmetros e determinar sobre os subgrupos com respeito a $p(s | \mathbf{q})$ seguido por simulação de incerteza $p(\mathbf{q})$. Isto é talvez a ordem natural quando o comportamento de subgrupos individuais é considerado sem importância. Então obtemos padrões idênticos.

$$m = E_{\mathbf{q}} [m_{sq} (m_{sq})] = E_{\mathbf{q}} [m_{\mathbf{q}}]$$

$$v = E_{\mathbf{q}} [Var_{sq} (m_{sq})] + Var_{\mathbf{q}} [E_{sq} (m_{sq})] = E_{\mathbf{q}} [v_{\mathbf{q}}] + Var_{\mathbf{q}} [m_{\mathbf{q}}] = v_{H1} + v_{P1}. \quad (4.1)$$

Mais tarde podemos considerar como uma decomposição da variância total v no resultado em duas componentes correspondendo para heterogeneidade de pacientes (v_{H1}) e parâmetro de incerteza (v_{P1}) respectivamente. Assumindo que $m_{\mathbf{q}}$ e $v_{\mathbf{q}}$ podem ser obtidos de forma fechada, a decomposição pode ser obtida usando estimação de Monte Carlo para quantidades desejadas.

2. Quando subgrupos individuais são mais importantes, é natural para a primeira condição nos subgrupos e parâmetros simulados para $p(\mathbf{q} | s)$ seguindo para determinação com respeito a $p(s)$. Obtemos

$$m = E_s [m_{q|s} (m_{q|s})] = E_s [m_s]$$

$$v = E_s [Var_{q|s} (m_{q|s})] + Var_s [E_{q|s} (m_{q|s})] = E_s [v_s] + Var_s [m_s] = v_{P2} + v_{H2}. (4.2)$$

Esta decomposição final da variância total v no resultado esperado em componentes correspondentes a parâmetros incertos (v_p) e heterogeneidade (v_H).

Usando nossa suposição padrão, m_s e v_s podem ser obtidos pelas estimativas de Monte Carlo, enquanto v_p e v_H são calculados diretamente de m_s e v_s usando a priori discreta $p(s)$. Assim a porcentagem de variabilidade para as duas fontes podem ser calculadas.

A segunda aproximação será mostrada mais apropriadamente e ilustrada com aproximações no exemplo a seguir.

4.5.4. Exemplo: Sensibilidade para heterogeneidade e parâmetro incerto

Um parâmetro relevante no nosso modelo para prognóstico seguindo THR é o risco de revisão h . Pode ser razoável assumir uma incerteza de $\pm 50\%$ sobre nosso risco de revisão assumido (que denotaremos agora por h_0) para cada grupo de idade e sexo. Temos aproximadamente um intervalo de 95% para $(h_0 t / 1,5, h_0 t \cdot 1,5)$ o risco de revisão, que representamos como uma distribuição normal para o log do risco

$$\log h \sim N(\log h_0, 0,2^2)$$

A Tabela 6 fornece a esperança m_s e desvio padrão $\sqrt{v_s}$ para custos e benefícios de cada subgrupo, permitindo incerteza no risco de revisão. A parte inferior desta tabela fornece a esperança global e variância para cada resultado dos subgrupos e a distribuição de risco, avaliada usando, respectivamente, Eq. (4.1) e Eq. (4.2), e tomando juntos $p(s)$ e $p(s|\mathbf{q})$ iguais para a distribuição idade e sexo fornecida na Tabela 6.

É claro que uma considerável incerteza sobre a taxa de risco de revisão tenha pequena influencia na expectativa de vida ou QALYS, mas leva para uma substancial sensibilidade nos custos. Quando combinado com influencia da heterogeneidade, parâmetro de incerteza é responsável apenas por 6,5% ou menos da variância total dos custos e menos de 0,01% para a variância total dos benefícios.

Tabela 6: Esperança e desvio padrão para custos e benefícios de THR para subgrupos de pacientes com parâmetro incerto, e global para parâmetro incerto e heterogeneidade de pacientes.

Subgrupos	Custos (£)		Expectativa Vida (anos)		QALYS	
	$m_s^{[C]}$	$\sqrt{v_s^{[C]}}$	$m_s^{[L]}$	$\sqrt{v_s^{[L]}}$	$m_s^{[Q]}$	$\sqrt{v_s^{[Q]}}$
Homens						
35-44	5.787	231	14,5	0,0052	13,2	0,060
45-54	5.425	202	12,7	0,0037	11,6	0,052
55-64	4.997	152	10,3	0,0021	9,5	0,038
65-74	4.472	75	7,7	0,0008	7,2	0,019
75-84	4.266	40	5,4	0,0003	5,1	0,010
>84	4.196	27	4,1	0,0002	3,9	0,007
Mulheres						
35-44	5.636	218	15,1	0,0052	13,8	0,057
45-54	5.359	194	13,7	0,0038	12,6	0,050
55-64	5.010	154	11,7	0,0024	10,7	0,039
65-74	4.493	79	9,1	0,0010	8,4	0,020
75-84	4.285	44	6,5	0,0004	6,0	0,011
>84	4.215	31	5,0	0,0003	4,6	0,008
Geral(c/ $p(s q)p(q)$)						
Esperança geral m		4.609		8,7		8,0
Var. para incerteza		1,013		0,0000002		0,00006
Var p/ heterogeneidade		174,400		6,7		5,5
Var. Total		175,413		6,7000002		5,50006
% var p/ heterog.		99,4%		99,9%		99,9%
Geral(c/ $p(q s)p(s)$)						
Esperança geral m		4,609		8,7		8,0
Var. para incerteza		11,473		0,0000002		0,00006
Var p/ heterogeneidade		163,953		6,7		5,5
Var. Total		175,426		6,7000002		5,50006
% var p/ heterog.		93,5%		99,9%		99,9%

CONCLUSÃO

Em muitas situações reais, a complexidade ou dimensionalidade do problema exige a utilização de ferramentas de simulação. É o caso, por exemplo, de serem exigidos cálculos de integrais multidimensionais sobre domínios difíceis de serem especificados.

O Método de Monte Carlo apresenta-se como uma técnica de simulação conceitualmente simples, mas de grande poder e aplicabilidade a diversas áreas do conhecimento.

A disponibilidade de computadores com capacidade de processamento cada vez maior permite utilizar alternativas mais sofisticadas e eficientes do Método de Monte Carlo, tal como os chamados métodos Quase-Monte Carlo.

O Método de Monte Carlo para Cadeias de Markov (MCMC) tem encontrado uma larga escala de aplicações, principalmente em Estatística Bayesiana, Física Computacional e Genética.

Na aplicação prática feita neste trabalho pode-se constatar a eficiência do método de Monte Carlo. As estimativas simuladas para custos, expectativa de vida e qualidade ajustada de vida para pacientes que realizaram reposição da prótese da bacia foram muito próximas dos valores calculados de forma exata.

Citando Ehlers (2007): “Apesar da sua grande utilidade, os métodos de simulação devem ser aplicados com cautela. Devido à facilidade com que os recursos computacionais podem ser utilizados hoje em dia, corremos o risco de apresentar uma solução para o problema errado (erro tipo 3-ver Mosteller) ou uma solução ruim para o problema certo. Assim, os métodos computacionalmente intensivos não devem ser vistos como substitutos do pensamento crítico sobre o problema por parte do pesquisador”.

REFERÊNCIAS BIBLIOGRÁFICAS

- BESAG, J. E. (1993). Bayesian inference for agricultural. *Bulletin of the International Statistical Institute* , 55, 121:136.
- BUSTOS, O., & FRERY, A. (1992). *Simulação estocástica: teoria e algoritmos*. Rio de Janeiro: IMPA.
- CASELLA, G. a. (1992). Explaining the Gibbs Sampler. *The American Statistician* , 46, 167-174.
- COWLES, M. K. (1996). Markov Chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association* , 91, 883–904.
- DEVROYE, L. (1986). *Non-Uniform Random Variate Generation*. New York: Springer-Verlag.
- EHLERS, R. S. (2007). *Introdução à Inferência Bayesiana*. Curitiba: Universidade Federal do Paraná, Departamento de Estatística.
- FITZPATRICK, R. (1991). Surveys of patient satisfaction. I. Important general considerations. *BMJ* , 302:887-9.
- GAMERMAN, D. a. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference*. London: Chapman & Hall/CRC.
- GAMERMAN, D. (1997). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. London: Chapman and Hall.
- GELFAND, A. a. (1990). Sampling-Based approaches to calculating marginal densities. *Journal of the American Statistical Association* , 85, 398–409.
- GELMAN, A. a. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science* , 7, 457–511.
- GEWEKE, J. (1994). *Bayesian comparison of econometric models*. Working Paper, Federal Reserve Bank of Minneapolis Research Department.
- GILKS, W. R., RICHARDSON, S., & SPIEGELHALTER, D. J. (1997). *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.
- GLASSERMAN, P. (2004). *Monte Carlo methods in financial engineering*. New York: Springer-Verlag.
- HEIDELBELGER, P. a. (1983). Simulation run length control in the presence of a initial transient. *Operations Research* , 31, 1109–1144.

- JERRUM, M. (1988). Conductance and the rapid mixing property for Markov chains: the approximation of the permanent resolved. *Proc. 20th ACM Symp. on Theory of Computing* , 235:243.
- KARP, R. a. (1985). Monte carlo algorithms for the planar multiterminal network reliability problem. *Journal of Complexity* , 1:45-64.
- KOLMOGOROV, A. N. (1931). Eine Verallgemeinerung des Laplace-Liapunovschen Stazes. *Izv. Akad. Nauk SSSR Ser. Fiz-Mat* , 959-962.
- LEE, E. K. (1997). *Computational experience of a general purpose mixed 0/1 integer programming solver (mipsol). Technical report, Industrial and Systems Engineering.* Atlanta, GA: Georgia Institute of Technology.
- METROPOLIS, N., & ULAM, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association* , 335-341.
- MOSTELLER, F. (1948). A k-Sample Slippage Test for an Extreme Population. *The Annals of Mathematical Statistics*, Vol.19, No.1, pp.58-65.
- PETROVSKI, I. (1933). Sur le topologie des courbes reelles et algebriques. *C. R. Acad. Sci. Paris* , 1270:1272.
- RAFTERY, A. E. (1992). How many iterations in the Gibbs sampler? In J. M. Bernardo, J. O. Berger, A. P. David e A. F. M. Smith (Eds.), *Bayesian Statistics. Vol. 4.* (pp. 763–774). , 763–774.
- RAYLEIGH, L. (1899). ‘On the transmission of light through an atmosphere containing small particles in suspension, and on the origin of the blue sky. *The Philosophical Magazine* , 47, 375-383.
- RIPLEY, B. D. (1987). *Stochastic Simulation.* New York: Miley.
- SMITH, A. a. (1993). Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *J. Roy. Statist. Soc. Ser. , B* 55, 3-24.
- SOBOL, I. M. (1994). *A Primer for the Monte Carlo Method.* Russia: CRC Press.
- SPIEGELHALTER, D. J., & BEST, N. G. (1 de November de 2002). Bayesian approaches to multiple sources of evidence and uncertainty in complex cost-effectiveness modeling.
- TANNER, M. (1996). *Tools for statistical inference: methods for the exploration of posterior distributions and likelihood functions.* New York: Springer, 3rd Edition. Chapter 4, pages 64-89.
- WALSH, B. (26 de April de 2004). Markov Chain Monte Carlo and Gibbs Sampling.

ANEXOS

Comandos do R

```
# Exemplo 1 – Metropolis Hastings
p = function(x,y) (2+x)^y[1] * (1-x)^(y[2]+y[3]) * x^y[4]
metr = function(n,y,p,start){
  theta = matrix(NA, nrow=n)
  theta[1] = start
  for (i in 2:n) {
    x = runif(1)
    A = p(x,y)/p(theta[i-1],y)
    prob = min(1,A)
    u = runif(1)
    taxa = 0
    if (u < prob) {
      theta[i] = x
      taxa = taxa + 1
    }
    else theta[i] = theta[i-1]
  }
  taxa = taxa/n
  return(list(theta=theta,taxa=round(taxa,2)))
}
```

```
# Exemplo 2 – Metropolis Hastings
metrop = function(n,sigma){
  x = matrix(NA,nrow=n)
  x[1] = 0
  for (i in 2:n){
    y = rnorm(1,x[i-1],sigma)
    prob = min(1,exp(-0.5*(y^2-x[i-1]^2)))
    u = runif(1)
    if (u < prob) x[i] = y else x[i] = x[i-1]
  }
  return(x)
}
```

Comandos do WinBUGS

Modelo 1: EStimação na forma fechada para cada estrato - fornece os resultados para a coluna "exato" na Tabela 2.

Não são necessárias iterações para este modelo: apenas carregar os dados e compilar, então obtém-se valores para C, mean.C, sd.C, BL, mean.BL, sd.BL, BQ, mean.BQ and sd.BQ usando a ferramenta "node" no menu "Info".

Modelo

```

model {

for(k in 1:K) { # loop para os estratos

# Equações de custos e benefícios na forma fechada:
#####

# Custos
for(t in 1:N) {
  ct[k,t] <- inprod(pi[k,t,], c[])/pow((1+delta.c), (t-1))
}
C[k] <- C0 + sum(ct[k,])

# Beneficios - Expectativa de vida
for(t in 1:N) {
  blt[k,t] <- inprod(pi[k,t,], bl[])/pow((1+delta.b), (t-1))
}
BL[k] <- sum(blt[k,])

# Beneficios - QALYs
for(t in 1:N) {
  bqt[k,t] <- inprod(pi[k,t,], bq[])/pow((1+delta.b), (t-1))
}
BQ[k] <- sum(bqt[k,])

# Probabilidades no modelo de Markov:
#####

# Matriz de Transição
for(t in 2:N) {
  Lambda[k,t,1,1] <- 1 - gamma[k,t] - lambda[k,t]
  Lambda[k,t,1,2] <- gamma[k,t] * lambda.op
  Lambda[k,t,1,3] <- gamma[k,t] *(1-lambda.op)
  Lambda[k,t,1,4] <- 0
  Lambda[k,t,1,5] <- lambda[k,t]

  Lambda[k,t,2,1] <- 0

```

```

Lambda[k,t,2,2] <- 0
Lambda[k,t,2,3] <- 0
Lambda[k,t,2,4] <- 0
Lambda[k,t,2,5] <- 1

Lambda[k,t,3,1] <- 0
Lambda[k,t,3,2] <- 0
Lambda[k,t,3,3] <- 0
Lambda[k,t,3,4] <- 1 - lambda[k,t]
Lambda[k,t,3,5] <- lambda[k,t]

Lambda[k,t,4,1] <- 0
Lambda[k,t,4,2] <- rho * lambda.op
Lambda[k,t,4,3] <- rho * (1 - lambda.op)
Lambda[k,t,4,4] <- 1 - rho - lambda[k,t]
Lambda[k,t,4,5] <- lambda[k,t]

Lambda[k,t,5,1] <- 0
Lambda[k,t,5,2] <- 0
Lambda[k,t,5,3] <- 0
Lambda[k,t,5,4] <- 0
Lambda[k,t,5,5] <- 1

gamma[k,t] <- h[k] * (t-1)
}

# Probabilidade marginal da existência em cada estado no tempo 1
pi[k,1,1] <- 1-lambda.op ; pi[k,1,2]<-0 ; pi[k,1,3] <- 0 ; pi[k,1,4]<-0; pi[k,1,5] <-
lambda.op

# Probabilidade marginal da existência em cada estado no tempo t>1
for(t in 2:N) {
  for(s in 1:S) {
    pi[k,t,s] <- inprod(pi[k,(t-1),], Lambda[k,t,s])
  }
}

}

# Média e desvio padrão dos custos e benefícios sobre estratos
#####

mean.C <- inprod(p.strata[], C[])
for(k in 1:12) { dev.C[k] <- pow(C[k] - mean.C, 2) }
var.C <- inprod(p.strata[], dev.C[])
sd.C <- sqrt(var.C)

mean.BL <- inprod(p.strata[], BL[])
for(k in 1:12) { dev.BL[k] <- pow(BL[k] - mean.BL, 2) }
var.BL <- inprod(p.strata[], dev.BL[])

```



```

BQ[9]    10.69584501325796
BQ[10]   8.431190404035165
BQ[11]   6.004466120882278
BQ[12]   4.640733878882862
mean.BQ  8.016253305629553
sd.BQ    2.338376751447872

```

Modelo 2: Estimativas MC para cada estrato - fornece resultados para colunas "Monte Carlo" na Tabela 2

n = 10000 iterações (1 por paciente simulado) são necessárias para este modelo; fornece C, mean.C, BL, mean.BL, BQ, mean.BQ.

Modelo

```

model <- function (){

for(k in 1:K) { # loop para os estratos

# Equações de custo e Benefício
#####

# Custos
for(t in 1:N) {
  ct[k,t] <- inprod(y[k,t,], c[])/pow((1+delta.c), (t-1))
}
C[k] <- C0 + sum(ct[k,])

# Benefícios - Expectativa de vida
for(t in 1:N) {
  blt[k,t] <- inprod(y[k,t,], bl[])/pow((1+delta.b), (t-1))
}
BL[k] <- sum(blt[k,])

# Benefícios - QALYs
for(t in 1:N) {
  bqt[k,t] <- inprod(y[k,t,], bq[])/pow((1+delta.b), (t-1))
}
BQ[k] <- sum(bqt[k,])

# Probabilidades no modelo de Markov:
#####

# Matriz de transição
for(t in 1:N) {
  Lambda[k,t,1,1] <- 1 - gamma[k,t] - lambda[k,t]

```

```

Lambda[k,t,1,2] <- gamma[k,t] * lambda.op
Lambda[k,t,1,3] <- gamma[k,t] *(1-lambda.op)
Lambda[k,t,1,4] <- 0
Lambda[k,t,1,5] <- lambda[k,t]

Lambda[k,t,2,1] <- 0
Lambda[k,t,2,2] <- 0
Lambda[k,t,2,3] <- 0
Lambda[k,t,2,4] <- 0
Lambda[k,t,2,5] <- 1

Lambda[k,t,3,1] <- 0
Lambda[k,t,3,2] <- 0
Lambda[k,t,3,3] <- 0
Lambda[k,t,3,4] <- 1 - lambda[k,t]
Lambda[k,t,3,5] <- lambda[k,t]

Lambda[k,t,4,1] <- 0
Lambda[k,t,4,2] <- rho * lambda.op
Lambda[k,t,4,3] <- rho * (1 - lambda.op)
Lambda[k,t,4,4] <- 1 - rho - lambda[k,t]
Lambda[k,t,4,5] <- lambda[k,t]

Lambda[k,t,5,1] <- 0
Lambda[k,t,5,2] <- 0
Lambda[k,t,5,3] <- 0
Lambda[k,t,5,4] <- 0
Lambda[k,t,5,5] <- 1

gamma[k,t] <- h[k] * (t-1)
}

# Probabilidade marginal da existência em cada estado no tempo 1
pi[k,1,1] <- 1-lambda.op ; pi[k,1,2]<-0 ; pi[k,1,3] <- 0 ; pi[k,1,4]<-0; pi[k,1,5] <-
lambda.op

# Estado para cada indivíduo no estrato k para tempo t=1
y[k,1,1:S] ~ dmulti(pi[k,1,], 1)

# Estado para cada indivíduo no estrato k para tempo t>1
for(t in 2:N) {
  for(s in 1:S) {
    pi[k,t,s] <- inprod(y[k,(t-1),], Lambda[k,t,s]) # probabilidades amostrais
  }
  y[k,t,1:S] ~ dmulti(pi[k,t,], 1)
}
}

# Média para custos e benefícios sobre estratos

```



```
#####
```

```
mean.C <- inprod(p.strata[], C[])
mean.BL <- inprod(p.strata[], BL[])
mean.BQ <- inprod(p.strata[], BQ[])
```

```
}
```

Data

```
list(N = 60,           # Numero de ciclos
     K = 12,          # Numero de estratos para idade-sexo
     S = 5,           # Numero de estados no modelo de Markov
     rho = 0.04,      # taxa de revisão
     lambda.op = 0.01, # taxa de mortalidade pós operatória
     # risco de revisão específica para idade-sexo:
     h = c(.0022, 0.0022, 0.0022, 0.0016, 0.0016, 0.0016, 0.0017, 0.0017, 0.0017, 0.0012,
            0.0012, 0.0012),
     C0 = 4052,       # Custos iniciais para operação primária
     c = c(0, 5290, 5290, 0, 0), # Custos adicionais associados a cada estado (zero, exceto para
     # revisão nos estados 2 e 3)
     bl = c(1,0,1,1,0), # Expectativa de vida associados a cada estado (um, exceto para
     # morte nos estados 2 e 5)
     bq = c(0.938, -0.622, -0.3387, 0.938, 0), # QALYs associadas a cada estado
     delta.c = 0.06,  # Abatimento nos custos
     delta.b = 0.06,  # Abatimento na saúde
     # Probabilidade de reposição da baquia por idade e sexo
     p.strata = c(0.02, 0.03, 0.07, 0.13, 0.10, 0.00, 0.02, 0.04, 0.10, 0.22, 0.26, 0.01),
     lambda = structure(.Data = c(0.0017, 0.0017, 0.0017, 0.0017, 0.0017, 0.0044, 0.0044,
     0.0044, 0.0044, 0.0044, 0.0044, 0.0044, 0.0044, 0.0044, 0.0044, 0.0138,
     0.0138, 0.0138, 0.0138, 0.0138, 0.0138, 0.0138, 0.0138, 0.0138, 0.0138,
     0.0379, 0.0379, 0.0379, 0.0379, 0.0379, 0.0379, 0.0379, 0.0379, 0.0379,
     0.0379, 0.0912, 0.0912, 0.0912, 0.0912, 0.0912, 0.0912, 0.0912, 0.0912,
     0.0912, 0.0912, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958,
     0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.0044,
     0.0044, 0.0044, 0.0044, 0.0138, 0.0138, 0.0138, 0.0138, 0.0138,
     0.0138, 0.0138, 0.0138, 0.0138, 0.0138, 0.0379, 0.0379, 0.0379, 0.0379,
     0.0379, 0.0379, 0.0379, 0.0379, 0.0379, 0.0379, 0.0912, 0.0912, 0.0912,
     0.0912, 0.0912, 0.0912, 0.0912, 0.0912, 0.1958, 0.1958,
     0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958,
     0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958,
     0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958,
     0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958,
     0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958,
     0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958,
     0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958, 0.1958,
     0.1958, 0.1958, 0.0379, 0.0379, 0.0379, 0.0379, 0.0379, 0.0379, 0.0912, 0.0912,
```



```

0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503, 0.1503,
), .Dim=c(12,60))
)

```

Resultados

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
BL[1]	14.47	2.891	0.02988	5.212	15.37	16.91	1	10000
BL[2]	12.74	3.358	0.03319	3.673	13.78	16.38	1	10000
BL[3]	10.31	3.68	0.03863	1.943	11.11	15.37	1	10000
BL[4]	7.737	3.53	0.02904	1.0	7.802	13.78	1	10000
BL[5]	5.436	2.991	0.02868	1.0	5.212	12.16	1	10000
BL[6]	4.145	2.953	0.03025	1.0	3.673	11.48	1	10000
BL[7]	15.15	2.621	0.02599	7.21	15.95	17.13	1	10000
BL[8]	13.74	3.045	0.02986	4.465	14.76	16.76	1	10000
BL[9]	11.63	3.551	0.03378	1.943	12.76	16.05	1	10000
BL[10]	9.025	3.685	0.03704	1.0	9.853	15.08	1	10000
BL[11]	6.439	3.325	0.03087	1.0	5.917	13.78	1	10000
BL[12]	4.984	3.446	0.03392	1.0	4.465	13.04	1	10000
BQ[1]	13.16	2.63	0.02753	4.889	13.97	15.54	1	10000
BQ[2]	11.62	3.04	0.03077	3.445	12.57	14.96	1	10000
BQ[3]	9.45	3.335	0.03488	1.823	10.05	14.0	1	10000
BQ[4]	7.16	3.241	0.02697	0.938	7.318	12.71	1	10000
BQ[5]	5.051	2.76	0.02639	0.938	4.889	11.09	1	10000
BQ[6]	3.852	2.718	0.02787	0.938	3.445	10.42	1	10000
BQ[7]	13.83	2.404	0.02385	6.395	14.54	15.82	1	10000
BQ[8]	12.58	2.774	0.02715	4.188	13.37	15.43	1	10000
BQ[9]	10.68	3.237	0.0304	1.823	11.57	14.76	1	10000
BQ[10]	8.361	3.388	0.03342	0.938	9.242	13.85	1	10000
BQ[11]	5.986	3.069	0.02838	0.938	5.55	12.5	1	10000
BQ[12]	4.637	3.184	0.03122	0.938	4.188	11.97	1	10000
C[1]	5793.0	1892.0	17.52	4052.0	5359.0	10650.0	1	10000
C[2]	5435.0	1889.0	17.68	4052.0	4052.0	10440.0	1	10000
C[3]	4974.0	1659.0	18.42	4052.0	4052.0	9663.0	1	10000
C[4]	4454.0	1211.0	11.66	4052.0	4052.0	8242.0	1	10000
C[5]	4250.0	905.4	7.962	4052.0	4052.0	7781.0	1	10000
C[6]	4203.0	806.4	7.597	4052.0	4052.0	7183.0	1	10000
C[7]	5641.0	1835.0	15.51	4052.0	5087.0	10500.0	1	10000
C[8]	5346.0	1765.0	16.74	4052.0	4701.0	10200.0	1	10000
C[9]	5020.0	1666.0	18.57	4052.0	4052.0	9746.0	1	10000
C[10]	4484.0	1242.0	12.48	4052.0	4052.0	8385.0	1	10000

C[11]	4277.0	955.0	9.092	4052.0	4052.0	7781.0	1	10000	
C[12]	4209.0	811.9	8.303	4052.0	4052.0	7183.0	1	10000	
mean.BL	8.668	1.392		0.01182	5.957	8.68	11.38	1	10000
mean.BQ	7.999	1.282		0.01092	5.495	8.005	10.5	1	10000
mean.C	4600.0	469.0		4.031	4088.0	4465.0	5825.0	1	10000

Desvio padrão global para estimativas de Monte Carlo da Tabela 2 são apenas para ponderar o desvio padrão das medias específicas por estrato.

Model 3: Estimativas MC para cada estrato, permitindo para parâmetro de incerteza no risco de revisão, fornecendo resultados para Tabela 3

n = 10000 iterações (1 por grupo de simulação) são necessários para este modelo; fornece C, BL, BQ para media e desvio padrão em cada subgrupo de resultados na Tabela 3.

Para resultados na parte inferior da Tabela 3:

Aproximação 1: $p(s, \theta) = p(s | \theta) p(\theta)$

m_theta e v_theta são a média e variância através dos subgrupos para fornecer valores de theta

=> no código BUGS, calcula a media e variância para C[k], BL[k], BQ[k] através dos subgrupos em cada iteração, então obtendo a expectativa de Monte Carlo no fim do processo
=> fornece mean.C, mean.BL, mean.BQ, var.C, var.BL, var.BQ

Aproximação 2: $p(s, \theta) = p(\theta | s) p(s)$

Média global, m = média ponderada para médias posteriores de C[k], BL[k], BQ[k] => calculado após executar o código BUGS

var exata para incerteza, vP2 = média ponderada para variâncias posteriores de C[k], BL[k], BQ[k] => calculado após executar o código BUGS

var exata para heterogeneidade = vH2 = variância ponderada para médias posteriores de C[k], BL[k], BQ[k] => calculado após executar o código BUGS

Modelo

model {

for(k in 1:K) { # loop para os estratos

Equações de custos e benefícios

#####

Custos

for(t in 1:N) {

ct[k,t] <- inprod(pi[k,t], c[])/pow((1+delta.c), (t-1))

}

C[k] <- C0 + sum(ct[k,])

```

# Beneficios - Expectativa de vida
for(t in 1:N) {
  blt[k,t] <- inprod(pi[k,t,], bl[])/pow((1+delta.b), (t-1))
}
BL[k] <- sum(blt[k,])

# Beneficios - QALYs
for(t in 1:N) {
  bqt[k,t] <- inprod(pi[k,t,], bq[])/pow((1+delta.b), (t-1))
}
BQ[k] <- sum(bqt[k,])

# Probabilidades no modelo de Markov:
#####

# Matriz de transição
for(t in 2:N) {
  Lambda[k,t,1,1] <- 1 - gamma[k,t] - lambda[k,t]
  Lambda[k,t,1,2] <- gamma[k,t] * lambda.op
  Lambda[k,t,1,3] <- gamma[k,t] *(1-lambda.op)
  Lambda[k,t,1,4] <- 0
  Lambda[k,t,1,5] <- lambda[k,t]

  Lambda[k,t,2,1] <- 0
  Lambda[k,t,2,2] <- 0
  Lambda[k,t,2,3] <- 0
  Lambda[k,t,2,4] <- 0
  Lambda[k,t,2,5] <- 1

  Lambda[k,t,3,1] <- 0
  Lambda[k,t,3,2] <- 0
  Lambda[k,t,3,3] <- 0
  Lambda[k,t,3,4] <- 1 - lambda[k,t]
  Lambda[k,t,3,5] <- lambda[k,t]

  Lambda[k,t,4,1] <- 0
  Lambda[k,t,4,2] <- rho * lambda.op
  Lambda[k,t,4,3] <- rho * (1- lambda.op)
  Lambda[k,t,4,4] <- 1 - rho - lambda[k,t]
  Lambda[k,t,4,5] <- lambda[k,t]

  Lambda[k,t,5,1] <- 0
  Lambda[k,t,5,2] <- 0
  Lambda[k,t,5,3] <- 0
  Lambda[k,t,5,4] <- 0
  Lambda[k,t,5,5] <- 1

  gamma[k,t] <- h[k] * (t-1)
}

```

```
# Probabilidade marginal da existência em cada estado no tempo 1
pi[k,1,1] <- 1-lambda.op ; pi[k,1,2]<-0 ; pi[k,1,3] <- 0 ; pi[k,1,4]<-0; pi[k,1,5] <-
lambda.op
```

```
# Probabilidade marginal da existência em cada estado no tempo t>1
for(t in 2:N) {
  for(s in 1:S) {
    pi[k,t,s] <- inprod(pi[k,(t-1),], Lambda[k,t,,s])
  }
}
}
```

```
# risco de revisão específica por idade-sexo
for(k in 1:K) {
  logh[k] ~ dnorm(logh0[k], tau)
  h[k] <- exp(logh[k])
}
}
```

```
# Calculo da media e variancia através dos estratos em cada iteração
# (Fornece média global e variancia usando a aproximação 1)
```

```
mean.C <- inprod(p.strata[], C[])
mean.BL <- inprod(p.strata[], BL[])
mean.BQ <- inprod(p.strata[], BQ[])

for(k in 1:12) {
  C.dev[k] <- pow(C[k]-mean.C , 2)
  BL.dev[k] <- pow(BL[k]-mean.BL , 2)
  BQ.dev[k] <- pow(BQ[k]-mean.BQ , 2)
}
var.C <- inprod(p.strata[], C.dev[])
var.BL <- inprod(p.strata[], BL.dev[])
var.BQ <- inprod(p.strata[], BQ.dev[])

}
```

Data

```
list(N = 60,          # Numero de ciclos
     K = 12,         # Numero de estratos por idade-sexo
     S = 5,          # Numero de eastados no modelo de Markov
     rho = 0.04,     # taxa de revisão
     lambda.op = 0.01, # taxa de mortalidade pós operatória
     # risco de revisão específica para idade-sexo:
     logh0 = c(-6.119, -6.119, -6.119, -6.438, -6.438, -6.438, -6.377, -6.377, -6.377, -6.725, -
6.725, -6.725),
     tau = 25,       # variância inversa refletindo a incerteza aproximadamente o log do
risco de revisão (= 1 / 0.2^2)
```


BL[4]	7.737	7.832E-4	7.842E-6	7.735	7.737	7.738	1	10000
BL[5]	5.405	3.237E-4	2.943E-6	5.405	5.405	5.406	1	10000
BL[6]	4.101	2.136E-4	2.169E-6	4.101	4.101	4.102	1	10000
BL[7]	15.13	0.005161	5.362E-5	15.12	15.14	15.14	1	10000
BL[8]	13.69	0.003836	3.409E-5	13.69	13.7	13.7	1	10000
BL[9]	11.65	0.002427	2.862E-5	11.64	11.65	11.65	1	10000
BL[10]	9.1	9.526E-4	9.262E-6	9.098	9.1	9.101	1	10000
BL[11]	6.46	4.226E-4	4.001E-6	6.459	6.46	6.461	1	10000
BL[12]	4.988	2.918E-4	2.873E-6	4.988	4.988	4.989	1	10000
BQ[1]	13.17	0.06027	5.015E-4	13.05	13.17	13.28	1	10000
BQ[2]	11.59	0.05198	5.214E-4	11.48	11.59	11.68	1	10000
BQ[3]	9.469	0.03841	3.949E-4	9.386	9.472	9.537	1	10000
BQ[4]	7.157	0.01873	1.875E-4	7.116	7.158	7.189	1	10000
BQ[5]	5.019	0.009921	9.019E-5	4.997	5.02	5.036	1	10000
BQ[6]	3.812	0.006748	6.85E-5	3.797	3.813	3.824	1	10000
BQ[7]	13.82	0.05718	5.931E-4	13.7	13.82	13.92	1	10000
BQ[8]	12.53	0.05006	4.447E-4	12.43	12.53	12.62	1	10000
BQ[9]	10.69	0.03914	4.619E-4	10.61	10.7	10.76	1	10000
BQ[10]	8.43	0.01989	1.934E-4	8.386	8.431	8.464	1	10000
BQ[11]	6.004	0.01083	1.026E-4	5.98	6.004	6.022	1	10000
BQ[12]	4.64	0.007627	7.509E-5	4.623	4.641	4.653	1	10000
C[1]	5787.0	230.8	1.921	5357.0	5781.0	6259.0	1	10000
C[2]	5425.0	202.1	2.027	5058.0	5414.0	5850.0	1	10000
C[3]	4997.0	151.6	1.559	4730.0	4984.0	5324.0	1	10000
C[4]	4472.0	74.94	0.7504	4342.0	4466.0	4634.0	1	10000
C[5]	4266.0	40.06	0.3641	4198.0	4263.0	4355.0	1	10000
C[6]	4196.0	27.27	0.2768	4149.0	4194.0	4257.0	1	10000
C[7]	5636.0	218.0	2.261	5230.0	5631.0	6079.0	1	10000
C[8]	5359.0	193.5	1.719	5010.0	5348.0	5768.0	1	10000
C[9]	5010.0	153.5	1.812	4734.0	5002.0	5332.0	1	10000
C[10]	4493.0	79.14	0.7691	4356.0	4487.0	4666.0	1	10000
C[11]	4285.0	43.46	0.4115	4210.0	4282.0	4380.0	1	10000
C[12]	4215.0	30.63	0.3015	4162.0	4212.0	4283.0	1	10000
mean.BL	8.687	4.508E-4	4.579E-6	8.686	8.687	8.688	1	10000
mean.BQ	8.015	0.008052	8.236E-5	7.998	8.015	8.03	1	10000
mean.C	4609.0	31.82	0.3256	4550.0	4608.0	4674.0	1	10000
var.BL	6.714	0.002979	2.954E-5	6.708	6.714	6.72	1	10000
var.BQ	5.466	0.03819	3.709E-4	5.389	5.467	5.539	1	10000
var.C	174400.0	28440.0	285.7	124400.0	172500.0	235400.0	1	10000

Nota: resultados para o painel inferior da Tabela 3 (aproximação 1) para custos são fornecidos por

m = média posterior para mean.C = 4609

vP1 = variância posterior para mean.C = 31.82 * 31.82

vH1 = média posterior para var.C = 174400

'Modelo' para calcular a média global (m), var exata para incerteza (vP2) e var exata para heterogeneidade (vH2) usando aproximação 2

Iteração não são necessárias - apenas compilar o modelo, ler dados, e iniciar, então usar a ferramenta "node" no menu "info" menu para obter valores para mC, mBL, mBQ, vP2.C, vP2.BL, vP2.BQ, vH2.C, vH2.BL, vH2.BQ, TC, TBL, TBQ, pcC, pcBL, pcBQ.

```

model {

  # resultado média geral (m)
  mC <- inprod(p.strata[], C[])
  mBL <- inprod(p.strata[], BL[])
  mBQ <- inprod(p.strata[], BQ[])

  # variancia exata para incerteza, vP
  for(k in 1:12) {
    VC[k] <- sdC[k]*sdC[k]
    VBL[k] <- sdBL[k]*sdBL[k]
    VBQ[k] <- sdBQ[k]*sdBQ[k]
  }
  vP2.C <- inprod(p.strata[], VC[])
  vP2.BL <- inprod(p.strata[], VBL[])
  vP2.BQ <- inprod(p.strata[], VBQ[])

  # variância exata para heterogeneidade, vH
  for(k in 1:12) { devC[k] <- pow(C[k] - mC, 2) }
  vH2.C <- inprod(p.strata[], devC[])
  for(k in 1:12) { devBL[k] <- pow(BL[k] - mBL, 2) }
  vH2.BL <- inprod(p.strata[], devBL[])
  for(k in 1:12) { devBQ[k] <- pow(BQ[k] - mBQ, 2) }
  vH2.BQ <- inprod(p.strata[], devBQ[])

  # Porcentagem para variância total exata para heterogeneidade
  TC <- vP2.C + vH2.C
  pcC <- vH2.C/TC
  TBL <- vP2.BL + vH2.BL
  pcBL <- vH2.BL/TBL
  TBQ <- vP2.BQ + vH2.BQ
  pcBQ <- vH2.BQ/TBQ

}

```

Data (médias posteriores desvio padrão posterior para C, BL and BQ executando o modelo 3)

```

list(
# médias posteriores
BL=c(14.48, 12.7, 10.34, 7.737, 5.405, 4.101, 15.13, 13.69, 11.65, 9.1, 6.46, 4.988),
BQ = c(13.17, 11.59, 9.469, 7.157, 5.019, 3.812, 13.82, 12.53, 10.69, 8.43, 6.004, 4.64),
C = c(5787.0, 5425.0, 4997.0, 4472.0, 4266.0, 4196.0, 5636.0, 5359.0, 5010.0, 4493.0,
4285.0, 4215.0),
# desvios padrões posteriores

```

```

sdBL =
c(0.005166,0.003677,0.002138,0.0007832,0.0003237,0.0002136,0.005161,0.003836,0.00242
7,0.0009526,0.0004226,0.0002918),
sdBQ =
c(0.06027,0.05198,0.03841,0.01873,0.009921,0.006748,0.05718,0.05006,0.03914,0.01989,0.
01083,0.007627),
sdC = c(230.8, 202.1, 151.6, 74.94, 40.06, 27.27, 218.0, 193.5, 153.5, 79.14, 43.46, 30.63),
p.strata = c(0.02, 0.03, 0.07, 0.13, 0.10, 0.00, 0.02, 0.04, 0.10, 0.22, 0.26, 0.01)
# estratos ponderados (prob de reposição de bacia por idade e sexo)
p.strata = c(0.02, 0.03, 0.07, 0.13, 0.10, 0.00, 0.02, 0.04, 0.10, 0.22, 0.26, 0.01)
)

```

Resultados

```

mC      4609.03
mBL     8.687390000000001
mBQ     8.01488
vP2.C   11472.793425
vP2.BL  3.3068250474E-6
vP2.BQ  7.493649773900001E-4
vH2.C   163953.4291
vH2.BL  6.713258897899999
vH2.BQ  5.464389485600001
TC      175426.222525
TBL     6.713262204725046
TBQ     5.465138850577391
pcC     0.9346004647431485
pcBL    0.999999507419054
pcBQ    0.9998628827193821

```