



Stéfane Lele

## **BioBayes: Pacote de análises Bayesianas para ambiente R**

Maringá - Paraná  
Fevereiro, 2019

Stéfane Lele

## **BioBayes: Pacote de análises Bayesianas para ambiente R**

Dissertação apresentada ao Programa de Pós-Graduação em Bioestatística do Centro de Ciências Exatas da Universidade Estadual de Maringá como requisito para a obtenção do título de Mestre em Bioestatística.

Universidade Estadual de Maringá – UEM

Departamento de Estatística

Programa de Pós-Graduação em Bioestatística

Orientador: Robson Marcelo Rossi

Maringá - Paraná

Fevereiro, 2019

---

Stéfane Lele

BioBayes: Pacote de análises Bayesianas para ambiente R/ Stéfane Lele. – Maringá - Paraná, Fevereiro, 2019-

Orientador: Robson Marcelo Rossi

Dissertação (Mestrado) – Universidade Estadual de Maringá – UEM

Departamento de Estatística

Programa de Pós-Graduação em Bioestatística, Fevereiro, 2019.

---

Stéfane Lele

## **BioBayes: Pacote de análises Bayesianas para ambiente R**

Dissertação apresentada ao Programa de Pós-Graduação em Bioestatística do Centro de Ciências Exatas da Universidade Estadual de Maringá como requisito para a obtenção do título de Mestre em Bioestatística.

Trabalho aprovado.

Maringá - Paraná, 28 de fevereiro de 2019.

---

**Robson Marcelo Rossi**  
Orientador (PBE - UEM)

---

**Elias Nunes Martins**  
(DZO - UEM)

---

**Carlos Aparecido dos Santos**  
(PBE - UEM)

Maringá - Paraná  
Fevereiro, 2019

---

# Agradecimentos

---

Os agradecimentos pelo desenvolvimento deste trabalho envolvem pessoas que participaram do meu crescimento profissional, pessoal e/ou ambos.

Agradeço plenamente ao meu orientador Robson M. Rossi, que além de compartilhar seu conhecimento, sempre me desafia com novos objetivos, acreditando em minha capacidade e empenho.

Assim também agradeço à minha família, pois como meu pai e minha mãe, desempenharam indiscutivelmente seus papéis em ouvir, amparar e caminhar ao meu lado, independente da circunstância.

Ao meu filho Gabriel, que fez todos os esforços que podia, dentro de sua capacidade de criança, para compreender minha necessidade de incontáveis horas de estudo, ausência e cansaço. Para ele deixo minha dedicação em um futuro melhor.

Ao amor da minha vida trazido pelo meu caminho na Estatística, Diogo Rossoni, que me mostra diariamente o amor em sua plenitude.

Ao Oilson Gonzatto, que há muitos anos compartilha comigo seu conhecimento, nunca negando esforços para contribuir com meu crescimento.

À Cooper Card (empresa e colaboradores), que tornaram possível a conclusão deste trabalho, me confiando seu banco de dados e auxiliando no aprimoramento desta dissertação.

Agradeço também aos meus amigos, em especial Luis Angeli e Mônica Marin, que se fizeram presentes em cada detalhe do percurso, trazendo o bem que a amizade proporciona.

Por fim, agradeço aos órgãos de fomento Capes e Fundação Araucária pelo apoio financeiro no decorrer dos meses de estudo, me isentando de demais preocupações financeiras em grande parte desta etapa.

*"Science as a candle in the Dark".  
(Carl Sagan)*

# Resumo

Este trabalho mostra detalhes a respeito da idealização, desenvolvimento e funcionamento de um pacote construído em ambiente R para análises de dados com enfoque bayesiano, denominado BioBayes. O objetivo principal é facilitar as análises bayesianas junto ao programa OpenBUGS (programa independente para análises bayesianas), permitindo que o usuário tenha maior flexibilidade em aplicações práticas como em relação ao tamanho amostral, se desprendendo da suposição assintótica, além da possibilidade de inserção de conhecimento prévio, entre outros. Por meio do programa R, foram implementadas em sua linguagem própria, funções para análises estatísticas que utilizam tanto as funcionalidades do R (como pacotes anteriormente desenvolvidos), quanto o programa OpenBUGS. A coleção de funções de distribuições disponíveis incluem alguns modelos de distribuições de probabilidade básicas, além de modelos para comparações de duas populações, análises de variância, modelos de regressão linear, log-linear e não-lineares, incluindo alguns modelos de dose-resposta, de curvas de crescimento e de lactação. Após as análises, um relatório técnico no formato HTML é gerado automaticamente, contendo gráficos a respeito das cadeias, estimativas, densidades *a posteriori* para os parâmetros, análises de convergência, DIC (*Deviance Information Criterion*), e o modelo utilizado internamente em linguagem bug. Ao final deste trabalho, observou-se que o pacote desenvolvido trabalha de forma prática e eficiente em relação à obtenção das distribuições dos parâmetros do modelo estabelecido e funções dos mesmos, às estimativas e às análises gráficas *a posteriori*.

**Palavras-chaves:** modelagem. OpenBUGS. *posteriori*.

# Abstract

This work shows details about idealization, development and operation of the package being build in the software R for bayesian-focused data analysis, called BioBayes. The main goal is to facilitate the bayesian analysis jointly to OpenBUGS software (independent software for bayesian analysis), allowing the user to have more flexibility in practical applications like unconcern about the sample size, prior knowledge insertion, and others. Through the program R, were implemented in your own language, models for statistical analysis using both the R features (like packages previously developed) and the OpenBUGS program. The collection of available distribution functions include since basic models of probability distribution, in addition to means comparisons models, analysis of variance, linear, non-linear and log-linear regression models (including for dose-response), growth and lactation curves. After the analysis, a technical report in the format HTML is automatically generated, containing graphics about the chains, estimates, posteriori densities for the parameters, convergence analysis, DIC (Deviance Information Criterion), and the internally used model in the language bug. At the end, it was observed that the developed package works in practical and efficient way in relation to obtaining the distributions and functions of parameters, estimates, and posteriori graphical analysis.

**Key-words:** modelagem. OpenBUGS. posteriori.



---

## Lista de ilustrações

---

Figura 1 – Incidência da palavra “ <i>bayesian</i> ” em artigos da base PubMed. . . . .	16
Figura 2 – Interferência na <i>posteriori</i> causada por <i>prioris</i> Beta não-informativas e informativas com verossimilhança binomial. . . . .	20
Figura 3 – Diagrama de funcionamento do pacote BioBayes. . . . .	27
Figura 4 – Diagrama de dispersão entre as variáveis X (independente) e Y (dependente) para os dados da Tabela 3. . . . .	34
Figura 5 – Gráfico dos traços das cadeias de Markov geradas a partir da simulação e da autocorrelação para cada parâmetro, respectivamente. . . . .	34
Figura 6 – Gráfico das densidades <i>a posteriori</i> geradas a partir da simulação e dos testes de convergência segundo os critérios de Heidelberg e Welch para a convergência, respectivamente. . . . .	35
Figura 7 – Estimativas <i>a posteriori</i> geradas pelo modelo de regressão quadrática. .	35
Figura 8 – Modelo quadrático em formato *.bug já implementando no pacote BioBayes. . . . .	36
Figura 9 – Gráfico dos ajustes do modelo quadrático pelo método bayesiano, frequentista, e a comparação simultânea entre ambos. . . . .	37
Figura 10 – <i>Boxplot</i> dos dados de três populações normais com variâncias heterogêneas. . . . .	39
Figura 11 – Gráfico das cadeias de Markov geradas a partir da simulação e da autocorrelação para cada parâmetro, respectivamente. . . . .	39
Figura 12 – Gráfico das densidades <i>a posteriori</i> geradas a partir da simulação e dos testes de convergência segundo os critérios de Heidelberg e Welch, respectivamente. . . . .	40
Figura 13 – Estimativas <i>a posteriori</i> geradas do modelo para populações normais com variâncias heterogêneas. . . . .	40
Figura 14 – Modelo ANOVA heterocedástica em formato *.bug já implementando no pacote BioBayes. . . . .	41

---

## Lista de tabelas

---

Tabela 1 – Famílias conjugadas para algumas distribuições mais usadas* . . . . .	22
Tabela 2 – Funções disponíveis no pacote BioBayes com suas respectivas descrições. . . . .	30
Tabela 3 – Dados hipotéticos para análise a partir de um modelo de regressão quadrática. . . . .	32
Tabela 4 – DIC para o modelo quadrático. . . . .	35
Tabela 5 – Dados hipotéticos para comparação de médias <i>a posteriori</i> de três populações normais. . . . .	38
Tabela 6 – DIC do modelo para populações normais com variâncias heterogêneas. . . . .	40

---

# Sumário

---

Introdução . . . . .	11
Objetivos . . . . .	14
<b>1 Capítulo 1 . . . . .</b>	<b>15</b>
Pacote BioBayes . . . . .	15
1.1 Referencial teórico . . . . .	15
1.1.1 A inferência bayesiana . . . . .	15
1.1.2 O Teorema de Bayes . . . . .	17
1.1.3 Distribuições <i>a priori</i> não-informativas . . . . .	18
1.1.4 Distribuições <i>a priori</i> conjugadas . . . . .	18
1.1.5 Obtenção da distribuição <i>a posteriori</i> via métodos computacionais . . . . .	23
1.1.5.1 O método Monte Carlo via Cadeias de Markov (MCMC) . . . . .	23
1.1.6 Estimação bayesiana . . . . .	24
1.1.7 Comparação de modelos . . . . .	25
1.2 Materiais e Métodos . . . . .	26
1.3 Resultados . . . . .	29
1.3.1 Funcionalidades do pacote BioBayes . . . . .	29
1.3.1.1 Exemplo 1: O modelo de regressão quadrática . . . . .	32
1.3.1.2 Exemplo 2: O modelo para testar médias de populações normais com variâncias heterogêneas . . . . .	37
1.4 Considerações finais . . . . .	42
<b>2 Capítulo 2 . . . . .</b>	<b>43</b>
BioBayes Package for Bayesian Analysis . . . . .	43
<b>Referências . . . . .</b>	<b>61</b>
<b>Anexos . . . . .</b>	<b>65</b>
<b>ANEXO A Manual do pacote BioBayes . . . . .</b>	<b>66</b>

## Introdução

O primeiro registro do pensamento bayesiano, desenvolvido pelo reverendo Thomas Bayes, ocorreu por volta de 1763, onde a ideia de agregar informações já conhecidas *a priori* foi desencadeada no teorema atualmente conhecido como Teorema de Bayes, utilizado para o cálculo da probabilidade de um evento condicionado à ocorrência de um outro.

Sua obra foi significativamente corrigida e editada por Richard Price, tal qual enviou o manuscrito de Bayes denominado “*An essay towards solving a problem in the doctrine of chances*” (BAYES et al., 1763) para ser lido postumamente pela Royal Society<sup>1</sup> e posteriormente reproduzido pela revista *Biometrika*.

Apesar de o Teorema de Bayes ser flexível e de fácil aplicação, permitindo a inclusão de um conhecimento *a priori*, e que esta pode ser modificada a cada nova evidência, o estudo aprofundado nesta área como, por exemplo, a inferência bayesiana, só mostrou seus primeiros registros com os trabalhos de Harold Jeffreys (1967), onde há informações sobre a primeira edição de seu livro, que ocorreu no ano de 1939. Salvo o trabalho de Jeffreys a respeito da inferência bayesiana, a necessidade de simulações com inúmeras iterações para a estimação dos parâmetros, eram na época inviáveis sem recursos computacionais, fazendo com que o pensamento inferencial bayesiano ficasse em suspenso.

Apenas a partir de 1980, devido ao desenvolvimento tecnológico e computacional advindos da terceira revolução industrial (COUTINHO, 1992) é que as técnicas bayesianas começaram a se mostrar uma boa alternativa em relação à metodologia frequentista (clássica), considerada até então como o método padrão nas análises de dados. Dessa forma, vários trabalhos começaram a surgir na época, tanto teóricos como estudos sobre expansões assintóticas como, por exemplo, para a média *a posteriori* (LINDLEY, 1980), testes de hipóteses e relação entre hipóteses clássicas e bayesianas a partir de distribuições *a priori* não-informativas (BERNARDO, 1980), *bootstrap* bayesiano (simulando a distribuição *a posteriori* do parâmetro ao invés de simular a distribuição amostral de uma estatística, estimando um parâmetro) (RUBIN, 1981), estudos de modelos *outliers* e distribuições *a priori* (WEST, 1984), ou mesmo em muitos trabalhos do ponto de vista aplicado.

Após o crescimento de trabalhos na área bayesiana, foi-se percebendo as particularidades e vantagens de sua utilização como, por exemplo, a inexistência do conceito de repetição nesta metodologia (BLASCO, 2001; FARIA et al., 2007). Já que o parâmetro é considerado aleatório no conceito bayesiano (diferentemente do frequentista, onde o estimador é quem tem distribuição), sua informação é dada pela maior probabilidade deste parâmetro assumir determinado valor. A utilização desta probabilidade na inferência diminui a dependência das estimativas em relação ao número de observações, requerendo uma quantidade menor destas (SILVA, 2006) além de o “viés” (que produz estimativas viadas) também não existir, já que não há repetição do experimento (BLASCO, 2001).

<sup>1</sup> Real Sociedade de Londres para o Melhoramento do Conhecimento Natural, fundada no ano de 1660 com intuito de promover o conhecimento científico <<https://royalsociety.org/about-us/history/>>.

Outra vantagem é que mesmo trabalhando com um número de observações pequenas, não são gerados intervalos de confiança infinitos, como ocorre na inferência clássica, conseguindo obtê-los por meio de métodos iterativos (NOGUEIRA et al., 2003). O mesmo pode ser feito com as estimativas para os parâmetros quando a obtenção das distribuições marginais forem demasiado complexas ou não-factíveis analiticamente, podendo-se fazer uso de métodos computacionais intensivos de simulação como o *Metropolis-Hastings* (METROPOLIS et al., 1953; HASTINGS, 1970), amostrador de Gibbs em cadeias de Markov (GEMAN; GEMAN, 1984), Monte Carlo Híbrido ou Hamiltoniano (NEAL et al., 2011), pacotes do programa R como MCMCglimm<sup>2</sup>, BRugs<sup>3</sup>, INLA<sup>4</sup> (integrada de Laplace por modelos aninhados) (RUE et al., 2009) e *procedures* do programa SAS como, por exemplo, a proc MCMC<sup>5</sup> e a proc BCHOICE<sup>6</sup>.

Há diversas outras vantagens na utilização da metodologia bayesiana, como aceitar a realidade dos dados, sem supor que estes provenham de uma distribuição normal (ROSSI, 2011); incorporar informações anteriormente adquiridas (*a priori*) a respeito do parâmetro, enriquecendo o processo de inferência (NOGUEIRA et al., 2003), entre inúmeras outras que podem ser encontradas na literatura.

Após a descoberta de tais vantagens pela comunidade estatística, segundo os dados do *site* PubMed<sup>7</sup>, o emprego de técnicas bayesianas cresceu de forma exponencial (Figura 1, p.16), mas isto só foi possível devido a criação de programas computacionais específicos.

Juntamente com o advento de novas tecnologias e o aumento da procura pelo método bayesiano, surgiu uma ferramenta para as análises, um programa chamado *BUGS* (*Bayesian inference Using Gibbs Sampling*) (LUNN et al., 2009), que em 1997 teve sua versão disponibilizada para *Windows*, o WinBUGS, em que Lykou e Ntzoufras (2011) apresentam de forma muito bem específica em seu trabalho “*WinBUGS: a tutorial*”, as funcionalidades do programa. Esta ferramenta até hoje é utilizada na área, como pode ser verificado, por exemplo, no trabalho de Yang et al. (2018), “*Chinese herbal injections for heart failure*” que compara a partir do método bayesiano, o efeito de diversos tipos de injeções de ervas chinesas em pacientes com insuficiência cardíaca.

Em 2005, foi criado o OpenBUGS<sup>8</sup>, uma variante de código aberto e gratuita do WinBUGS.

“Na verdade, este desenvolvimento destaca uma das diferenças fundamentais entre o WinBUGS e o OpenBUGS. Ambas as versões classificam a distribuição

<sup>2</sup> <<https://cran.r-project.org/web/packages/MCMCglimm/MCMCglimm.pdf>>

<sup>3</sup> <<https://cran.r-project.org/web/packages/BRugs/BRugs.pdf>>

<sup>4</sup> <[www.r-inla.org](http://www.r-inla.org)>

<sup>5</sup> <<https://support.sas.com/rnd/app/stat/procedures/mcmc.html>>

<sup>6</sup> <<https://support.sas.com/rnd/app/stat/procedures/bchoice.html>>

<sup>7</sup> <<https://www.ncbi.nlm.nih.gov/pubmed/>>

<sup>8</sup> <[www.openbugs.net](http://www.openbugs.net)>

condicional completa de cada parâmetro a ser atualizado, mas enquanto o WinBUGS usa um mapeamento de classificação de um-para-um para atualizar o método, o OpenBUGS trabalha através de uma lista de métodos de atualização disponíveis, geralmente em ordem crescente de generalidade e aloca métodos para todos os parâmetros para os quais o método é apropriado” (LUNN et al., 2009).

A partir da linguagem R criada por Ross Ihaka e Robert Gentleman em 1993<sup>9</sup> e após, com o aprimoramento de várias pessoas do mundo que contribuíram com a submissão de pacotes para sub-rotinas ou áreas de estudo específicas, o R se tornou uma das maiores linguagens utilizadas pela comunidade estatística, e que se mantém em crescimento.

Ainda em 2005, foi desenvolvido um pacote para o programa R, denominado “BRugs” com interface totalmente interativa com o OpenBUGS, utilizando a amostragem MCMC para análises bayesianas<sup>10</sup>. Segundo Lunn et al. (2009), um dos objetivos da criação do OpenBUGS era

“[...]desacoplar a funcionalidade principal e a interface do usuário, para facilitar o uso do programa de outros ambientes. A saída mais significativa desse braço de pesquisa foi o BRugs, uma interface que permite que o BUGS seja usado interativamente de dentro do R. [...] a beleza do BRugs é que ele é totalmente interativo e a saída do BUGS pode ser acessada no meio da análise para explorar todo o potencial do R na geração de gráficos de diagnóstico, por exemplo.

Algumas desvantagens observadas ao utilizar o OpenBUGS, é que este necessita do modelo estatístico escrito em um arquivo separado, no formato \*.bug, tendo que ser modificado o modelo original ou criado uma cópia para cada alteração que se deseje fazer nos valores dos parâmetros ou nas distribuições *a priori*. Além disso, há grande dificuldade em relação à leitura de grandes arquivos de dados. Outra desvantagem revelada pelo criador do “BRugs” foi de que o projeto opera de forma estável em R de 32-bits para sistema operacional *Windows*, e que para os casos de 64-bits no *Windows* ou execução em Linux, o desempenho é menos eficiente. Além disso, o passo-a-passo do programa é realizado em “*click point*”, tornando-o cansativo ao usuário, principalmente pelo fato de que a qualquer erro, o processo deve ser reiniciado.

Dadas as informações supracitadas, deseja-se com este trabalho, apresentar em linguagem R, um pacote similar ao BRugs, denominado BioBayes, que sirva como mais uma ferramenta alternativa para análises bayesianas, porém de fácil utilização, contendo

---

<sup>9</sup> <<https://www.r-project.org/>>

<sup>10</sup> <<https://cran.r-project.org/web/packages/BRugs/index.html>>.

uma coleção de modelos estatísticos básicos disponíveis ao usuário, gerando inferências a respeito dos parâmetros e relatório dos mesmos.

Além disso, aplicá-lo à uma análise de dados reais, utilizando o modelo de regressão logística para análise de risco na concessão ao crédito, identificando as variáveis capazes de corroborar com o modelo que mensura o risco de um cliente se tornar inadimplente.

## Objetivos

Os objetivos deste trabalho envolvem facilidades aos pesquisadores que desejam fazer uso de métodos bayesianos, tais como:

- Implementar em linguagem R, funções para análises estatísticas básicas com enfoque bayesiano, utilizando conjuntamente o programa OpenBUGS;
- Disponibilizar um conjunto de funções referentes às distribuições de probabilidade básicas, testes de médias de duas populações, análises de variância e modelos de regressão, em um pacote denominado BioBayes, propiciando ferramentas que facilitem o ingresso ao estudo bayesiano da comunidade acadêmica e demais interessados, difundindo o método de forma simples e prática.
- Empregar a ferramenta desenvolvida à uma análise de um problema real do cotidiano, verificando sua efetiva execução.

---

## Seção 1

---

# Capítulo 1

---

## Pacote BioBayes

### 1.1 Referencial teórico

#### 1.1.1 A inferência bayesiana

A inferência bayesiana começou a se fortalecer na década de 80, a partir do avanço tecnológico (COUTINHO, 1992), pois é a partir desta data que começaram a aparecer trabalhos na área, tanto teóricos como do ponto de vista prático.

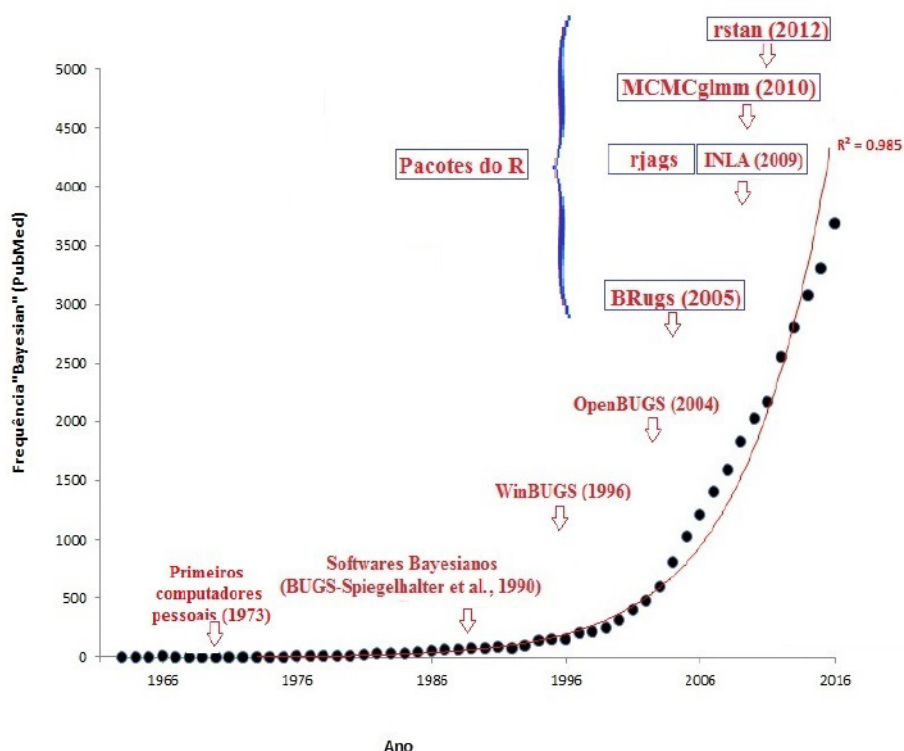
Pela Figura 1 é possível visualizar o crescimento exponencial do uso do método por meio da incidência da palavra “*bayesian*” em artigos científicos da base PubMed, além da evolução dos pacotes e programas de análises bayesianas.

Segundo Ntzoufras (2011), até o final da década de 80, a estatística bayesiana era considerada apenas uma alternativa interessante ao método clássico (frequentista), com a principal diferença de que a bayesiana considera os parâmetros como variáveis aleatórias, escritos em uma distribuição prévia denominada *priori*, que ao ser combinada com a verossimilhança tradicional por um fator multiplicativo, gera a distribuição *a posteriori* do vetor de parâmetros de interesse.

O método frequentista, até então, supria as necessidades do pesquisador até certo ponto, pois possuía algumas desvantagens em casos específicos como a suposição de normalidade para qualquer tipo de dados, mesmo que estes não fossem, fugindo da realidade que os mesmos apresentavam, ou comumente em estudos envolvendo componentes genéticos de melhoramento animal e vegetal em que as estimativas de verossimilhança são complexas, os recursos computacionais do método frequentista para obtê-las são limitados, ou até mesmo em casos de tamanho amostral insuficiente, as estimativas intervalares geram intervalos de confiança infinitos.



Figura 1 – Incidência da palavra “bayesian” em artigos da base PubMed.



Fonte: Slides de aula da disciplina Estatística Bayesiana Aplicada, ministrada pelo professor Robson Marcelo Rossi ao Programa de Mestrado em Bioestatística - UEM.

É sabido que no método clássico, é possível obter informações a respeito de um parâmetro de interesse a partir da amostra, por meio da função de verossimilhança, permitindo ao pesquisador fazer estimações a respeito de um parâmetro  $\theta$  apenas utilizando informação amostral. Porém, segundo Paulino et al. (2003), a informação já conhecida a respeito do parâmetro de interesse  $\theta$  é de grande importância, apesar de o verdadeiro  $\theta$  ser ainda desconhecido, fazendo menção à importância da distribuição *a priori*, descrita por  $f(\theta)$ . Os autores ainda afirmam que se  $\theta$  é desconhecido, este é incerto, e toda a incerteza deve ser quantificada em termos de probabilidade.

A inferência bayesiana, ao contrário da frequentista, trata o vetor de parâmetros  $\theta$  como quantidades aleatórias, e permite a inclusão de conhecimentos prévios do pesquisador (obtidos por meio de análises anteriores, experiência no ramo, revisões de literatura ou metanálise), incorporadas na forma de modelos probabilísticos para  $\theta$ , antes mesmo dos dados terem sido coletados (MACEDO, 2015).

Outra vantagem do método é que pelo fato do parâmetro ser considerado aleatório e ser atribuída a ele uma probabilidade, esta faz com que se diminua a dependência das estimativas em relação ao número de observações, requerendo um número menor destas.

Mesmo sob esta circunstância, é possível obter estimações intervalares plausíveis, devido aos métodos iterativos, tais quais também permitem calcular as estimativas para os parâmetros quando as distribuições marginais forem demasiado complicadas (NOGUEIRA et al., 2003). Outra vantagem a ser considerada é que pela ausência de repetição do experimento, o “viés” que produz estimativas viciadas, deixa de existir (BLASCO, 2001).

Considere que  $Y$  seja a variável resposta e  $f(y | \theta)$  seja a função densidade de probabilidade, onde  $\theta$  é um vetor de parâmetros. Dada uma amostra independente e identicamente distribuída (i.i.d.), contendo  $n$  observações, descrita por  $y = [y_1, y_2, \dots, y_n]^T$ , a função de verossimilhança (FV) do modelo que contém a informação proveniente da amostra observada, tem a seguinte forma:

$$f(y | \theta) = \prod_{i=1}^n f(y_i | \theta) \quad (1.1)$$

(NTZOUFRAS, 2011).

### 1.1.2 O Teorema de Bayes

O Teorema de Bayes, desenvolvido por Bayes et al. (1763) mostra o princípio de como a distribuição *a posteriori* é o resultado da operação bayesiana, de forma multiplicativa, entre a distribuição *a priori* e a função de verossimilhança.

**O Teorema de Bayes:** Considerando dois eventos distintos  $A$  e  $B$ , em que  $A = A_1 \cup A_2 \cup \dots \cup A_n$  onde  $A_i \cap A_j = \emptyset$  para cada  $i \neq j$ , então o Teorema de Bayes fornece uma expressão para a probabilidade condicional de  $A_i$ , dado que  $B$  tenha ocorrido, descrita como

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{P(B)} = \frac{P(B | A_i)P(A_i)}{\sum_{i=1}^n P(B | A_i)P(A_i)} \quad (1.2)$$

Segundo Hoffmann-Jorgensen (1994), a Expressão 1.2 pode ser reescrita de uma forma geral, utilizando a regra de Bayes descoberta por Pierre-Simon de Laplace, como

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \propto P(B | A)P(A) \quad (1.3)$$

(NTZOUFRAS, 2011).

De forma análoga, de acordo com o Teorema de Bayes, a distribuição *a posteriori* pode ser escrita como

$$f(\theta | y) = \frac{f(y | \theta)f(\theta)}{f(y)} \propto f(y | \theta)f(\theta) \quad (1.4)$$

em que  $f(\theta)$  representa a distribuição *a priori* e  $f(y | \theta)$  é a função de verossimilhança, mostrada na Equação 1.1.

### 1.1.3 Distribuições *a priori* não-informativas

Usualmente, distribuições *a priori* não-informativas são utilizadas, entre algumas razões, quando o pesquisador não tem informações prévias a respeito dos dados e também para permitir a comparação com as estimativas frequentistas. Neste caso, é necessário determinar uma distribuição (juntamente com seus hiperparâmetros<sup>1</sup>), que não influencie na função de verossimilhança, ou seja,  $f(\theta) = 1$ , para que apenas  $f(y | \theta)$  seja responsável pelas estimativas, assim como ocorre na inferência frequentista.

As distribuições *a priori*, não precisam necessariamente serem uma função densidade de probabilidade (f.d.p.) ou mesmo integráveis. Anteriormente ao avanço computacional, foram desenvolvidas teorias algébricas que geravam *prioris* que, em geral, conjugavam ou que, pelo menos, tornavam a *posteriori* identificável. Os métodos mais utilizados na literatura para obtê-las eram os de Jeffreys, de Bayes-Laplace e de Box-Tiao (PAULINO et al., 2003). Em geral, as distribuições *a priori* geradas pelas regras de Bayes-Laplace e Jeffreys são do tipo impróprias, ou seja, estas não são integráveis, acomodando probabilidades infinitas, como por exemplo, a distribuição  $Beta(0, 0)$ , considerada uma *priori* não-informativa de Jeffreys do tipo imprópria. No entanto, a distribuição  $Beta(1/2, 1/2)$  também é uma *priori* não-informativa de Jeffreys, todavia do tipo própria.

Atualmente, não há mais a necessidade desse esforço algébrico devido aos recursos computacionais desenvolvidos. Ao se trabalhar com *prioris* no contexto atual, deve-se apenas respeitar o espaço paramétrico das distribuições, já que a capacidade tecnológica está apta a solucionar algumas operações.

### 1.1.4 Distribuições *a priori* conjugadas

Para se determinar uma distribuição *a priori*, é necessário verificar que tanto a *priori* como a *posteriori* devem pertencer à mesma classe de distribuições. Dessa forma, para se atualizar a informação prévia a respeito dos dados, é necessário apenas uma mudança nos hiperparâmetros (EHLERS, 2005). Quando isto ocorre, dá-se o nome de *prioris* conjugadas.

**Definição 1:** Se  $F = \{p(x | \theta), \theta \in \Theta\}$  é uma classe de distribuições amostrais, então uma classe de distribuições  $P$  é conjugada a  $F$  se

$$\forall p(x | \theta) \in F \quad e \quad p(\theta) \in P \rightarrow p(\theta | x) \in P \quad (1.5)$$

Um exemplo simples para melhor entendimento é o modelo discreto binomial. Segundo Rossi (2011), se o parâmetro de interesse for a proporção  $\theta$ , desconhecido, ou seja,

$$0 < \theta < 1,$$

<sup>1</sup> Os parâmetros indexadores da família de distribuições *a priori* são chamados de hiperparâmetros, para que não sejam confundidos com os parâmetros de interesse  $\theta$ . <<http://www.leg.ufpr.br/~paulojus/CE227/ce227/node3.html>>

de indivíduos com certa característica, tem-se que para o  $i$ -ésimo indivíduo:

$$Y_i \sim \text{Bernoulli}(\theta)$$

com  $\theta$  representando a probabilidade de sucesso, ou seja, do indivíduo vir a ter a característica. Para uma amostra de tamanho  $n$  com reposição desta população, supõe-se que se obteve  $t$  = número de indivíduos observados com a característica de interesse, logo,  $t \sim \text{Bin}(n, \theta)$ , ou seja,

$$f(y \mid \theta) = P(T = t \mid \theta) = \binom{n}{t} \theta^t (1 - \theta)^{n-t}, \quad t = 0, 1, 2, \dots, n. \quad (1.6)$$

Neste caso, supõe-se que a distribuição de probabilidade *a priori*, seja uma Beta (já que nela  $0 < \theta < 1$ ), com  $\alpha$  e  $\beta$  conhecidos ( $\alpha > 0, \beta > 0$ ), isto é,

$$f(\theta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1} \quad (1.7)$$

onde  $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}$ , tal que  $\Gamma(n) = (n - 1)!$ .

A multiplicação entre  $f(y \mid \theta)$  e  $f(\theta)$  resulta em uma *posteriori*  $f(\theta \mid y)$  conjugada com função densidade de probabilidade  $\text{Beta}(\alpha + t, \beta + n - t)$ , em que  $t = \sum_{i=1}^n y_i$  é o total de sucessos observados na amostra. A obtenção esmiuçada da *posteriori* pode ser vista a seguir nas Equações 1.8 a 1.10.

$$\begin{aligned} f(\theta \mid y) &= f(y \mid \theta) f(\theta) \\ &= \binom{n}{t} \theta^t (1 - \theta)^{n-t} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1} \end{aligned} \quad (1.8)$$

É possível utilizar o *kernel* (núcleo) das funções, ou seja, somente os termos que contemplam o parâmetro, pois

$$\begin{aligned} \binom{n}{t} \theta^t (1 - \theta)^{n-t} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1 - \theta)^{\beta-1} &\propto \theta^t (1 - \theta)^{n-t} \theta^{\alpha-1} (1 - \theta)^{\beta-1} \\ &\propto \theta^{t+\alpha-1} (1 - \theta)^{n-t+\beta-1} \end{aligned} \quad (1.9)$$

Sabendo que uma distribuição  $\text{Beta}(\alpha, \beta)$  tem a forma  $\theta^{\alpha-1} (1 - \theta)^{\beta-1}$ , podemos olhar para a Expressão 1.9 de modo que

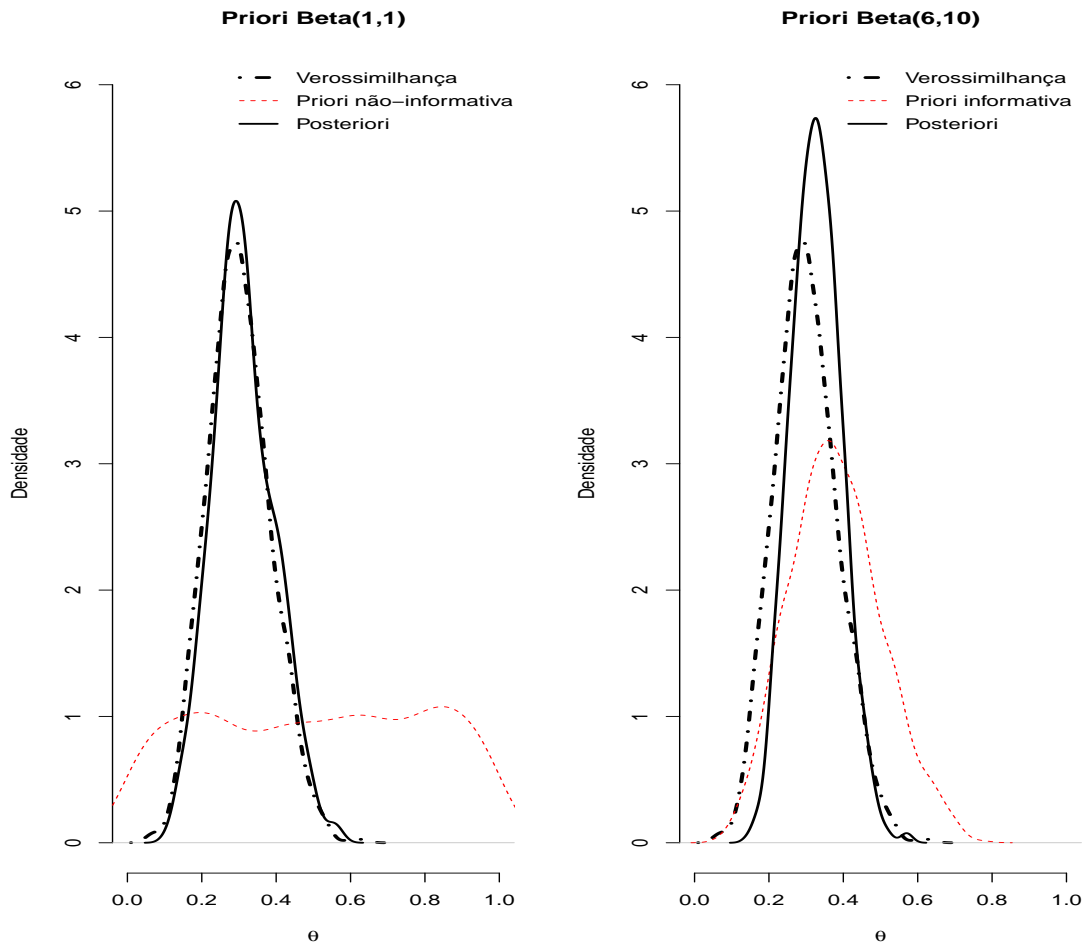
$$f(\theta \mid y) \propto \theta^{(\alpha+t)-1} (1 - \theta)^{(\beta+n-t)-1} \sim \text{Beta}(\alpha + t, \beta + n - t) \quad (1.10)$$

Como a média  $E(\theta)$  na distribuição Beta é dada por  $\frac{\alpha}{\alpha + \beta}$ , então a média *a posteriori* neste caso é dada por

$$E[\theta | y] = \frac{\alpha + t}{\alpha + t + (\beta + n - t)} = \frac{\alpha + t}{\alpha + \beta + n} \quad (1.11)$$

Pode-se verificar o papel de interferência causada na *posteriori* de uma distribuição conjugada com verossimilhança binomial e *priori* Beta (não-informativa e informativa, respectivamente) (Figura 2).

Figura 2 – Interferência na *posteriori* causada por *prioris* Beta não-informativas e informativas com verossimilhança binomial.



Existem conjugações para outras distribuições *a priori*, já sugeridas na literatura, e algumas delas são apresentadas na Tabela 1.

A partir dos recursos computacionais disponíveis atualmente, testar conjugações, modificar parâmetros das distribuições *a priori* e obter a distribuição *a posteriori* é consideravelmente simples, pois a partir da criação do programa WinBUGS/OpenBUGS já não se

faz necessário que o pesquisador conjugue as distribuições para gerar as cadeias, recurso do qual estes são programados a fazer.

Tabela 1 – Famílias conjugadas para algumas distribuições mais usadas\*.

Variável/Modelo	$\propto F.V$	Distribuição a priori	Distribuição a posteriori
$Y_i \sim \text{Bernoulli}(\theta)$	$\theta^t (1 - \theta)^{n-t},$ $t = \sum_{i=1}^n y_i$	$\theta \sim \text{Beta}(\alpha, \beta)$	$\theta \mid t \sim \text{Beta}(\alpha + t, \beta + n - t)$
$Y_i \sim \text{Poisson}(\theta)$	$\theta^t e^{-n\theta}$	$\theta \sim \text{Gama}(\alpha, \beta)$	$\theta \mid t \sim \text{Gama}(\alpha + t, \beta + n)$
$Y_i \sim \text{Normal}(\mu, \tau)$	$\tau^{n/2} \exp \left\{ -\frac{\tau}{2} \sum_{i=1}^n (y_i - \mu)^2 \right\}$  $\tau = \frac{1}{\sigma^2}$	$\mu \sim \text{Normal}(0, s^2)$	$\mu \mid y, \tau \sim \text{Normal} \left( \frac{1}{n\tau + \frac{1}{s^2}} \bar{y} n\tau, \frac{1}{n\tau + \frac{1}{s^2}} \right)$
		$\tau \sim \text{Gama}(\alpha, \beta)$	$\tau \mid y, \mu \sim \text{Gama} \left( \alpha + \frac{n}{2}, \beta + \frac{1}{2} \sum_{i=1}^n (y_i - \mu)^2 \right)$
$Y_i \sim \text{Exponencial}(\theta)$	$\theta^n e^{-\theta t}$	$\theta \sim \text{Gama}(\alpha, \beta)$	$\theta \mid t \sim \text{Gama}(\alpha + n, \beta + t)$

\*Conjugações para outras distribuições podem ser encontradas em Nizoufras (2011).

### 1.1.5 Obtenção da distribuição *a posteriori* via métodos computacionais

A sigla BUGS (**B**ayesian inference **U**sing **G**ibbs **S**ampling) deu o nome aos programas da coleção BUGS, como por exemplo, o WinBUGS (SPIEGELHALTER et al., 1996), programa estatístico para inferência bayesiana, que utiliza amostragem de Gibbs por meio de cadeias de Markov (MCMC - *Markov Chain Monte Carlo*). Após sua versão aprimorada que surgiu oito anos mais tarde, o OpenBUGS, juntamente com o pacote BRugs do programa R, apesar das novidades advindas, conservam o mesmo método de amostragem.

Os métodos computacionais para obtenção da *posteriori* incluem o Método de Monte Carlo Simples que simplesmente calcula uma integral no intervalo  $(a, b)$  e obtém um valor esperado; Métodos de Reamostragem que podem ser utilizados como ferramenta computacional alternativa para quando uma distribuição é demasiadamente difícil ou impossível de se simular valores. Entre os métodos de reamostragem, estão o Método de Rejeição e a Reamostragem Ponderada. Estes métodos geram valores em duas etapas, primeiramente a partir de uma distribuição auxiliar conhecida (Smith e Gelfand (1992) sugerem que esta seja a distribuição *a priori*), e na segunda utilizando um mecanismo de correção para que os valores gerados sejam representativos da *posteriori*. Tais métodos apresentados são considerados não-iterativos.

Desde que o tamanho da amostra seja suficientemente grande, não existe preocupação em relação à convergência do algoritmo, já que os valores são gerados independentemente uns dos outros.

O método MCMC é uma ferramenta alternativa para quando se quer encontrar uma densidade com boa aproximação da *posteriori* simultaneamente à fácil amostragem, sem que o processo seja demasiado complicado (EHLERS, 2005).

#### 1.1.5.1 O método Monte Carlo via Cadeias de Markov (MCMC)

No método MCMC, mantém-se o desejo de obtenção de uma amostra da distribuição *a posteriori* para que se faça inferências a respeito do parâmetro de interesse  $\theta$ . Porém neste caso é utilizada simulação iterativa, baseada em cadeias de Markov, onde os dados gerados não são mais independentes uns dos outros.

O algoritmo *Metropolis-Hastings* é um processo iterativo, que gera um valor de uma distribuição auxiliar e é verificada a probabilidade de aceitação/rejeição do mesmo. Esta correção garante a convergência da cadeia para a distribuição desejada (*posteriori*). De forma mais aprofundada, ao se construir a cadeia de Markov segundo Hastings (1970) juntamente com a generalização proposta por Metropolis et al. (1953), o algoritmo, em cada tempo  $t$ , escolherá o próximo passo  $X_{t+1}$  pela primeira amostragem de um ponto candidato  $Y$  a partir de uma distribuição proposta. Se o ponto é aceito no critério de aceitação/rejeição, o próximo passo será  $X_{t+1} = Y$ . Se não, a cadeia não se move, permanecendo com  $X_{t+1} = X_t$ . Este processo é repetido até que se obtenha uma distribuição



estacionária  $\phi(\cdot)$  que será considerada a distribuição de interesse *a posteriori*. O algoritmo detalhado com demais informações sobre as probabilidades de aceitação/rejeição pode ser visto em Spiegelhalter et al. (1996).

O método de *Gibbs Sampling*, introduzido por Geman e Geman (1984), utilizado no OpenBUGS, é apenas um caso específico do algoritmo *Metropolis-Hastings*, onde são geradas cadeias de Markov para amostras, em que estas são correlacionadas com amostras próximas, não havendo critério de aceitação-rejeição.

Ao simular um passeio aleatório no espaço amostral de  $\theta$ , o início da cadeia deve ser descartado por não representar com acurácia a distribuição desejada. Porém ao se utilizar uma longa cadeia, há uma estabilização substancial nas oscilações, fazendo com que  $\theta$  convirja para uma distribuição estacionária, obtendo-se assim, a distribuição *a posteriori*. Mais detalhes sobre os algoritmos MCMC podem ser encontrados em Gamerman (1997).

Para que se tenha informações a respeito da convergência da cadeia, foram criados critérios para tal, como o critério de Geweke (1991), Raftery e Lewis (1991) e Heidelberger e Welch (1983), que verificam o equilíbrio da cadeia gerada, indicando se esta é estacionária. Dessa forma, o pesquisador pode aumentar o número de iterações caso estas não tenham sido suficientes para a estabilização, assim como alterar os valores de *burn* (valores que serão descartados no início da cadeia, comumente com grandes oscilações) e do intervalo de reamostragem (intervalo no qual valores são retirados da cadeia para eliminar a autocorrelação e utilizados para a estabilização), até que o método indique convergência. O pacote “coda” do R, por exemplo, disponibiliza tais critérios.

### 1.1.6 Estimação bayesiana

A partir da obtenção da distribuição *a posteriori*, é possível retirar toda a informação a respeito do parâmetro  $\theta$ , de forma gráfica através de densidades, histogramas, *boxplots*, entre outros, de forma extendida analisando numericamente todos os valores gerados pela simulação ou de forma resumida, por meio de estimações pontuais como média, mediana, desvios-padrão, além de estimações intervalares como intervalos de credibilidade e HPD (*Highest Posterior Density*), por exemplo, definidos a seguir:

*C é um intervalo de credibilidade de  $100(1 - \alpha)\%$ , ou nível de credibilidade  $(1 - \alpha)$ , para  $\theta$ , se  $P(\theta \in C) \geq 1 - \alpha$ .*

Por exemplo, quando  $\alpha = 5\%$ ,  $ICr(95\%) = [P_{2,5\%}; P_{97,5\%}]$ , em que  $P$  indica os percentis. Porém, quando se está interessado em intervalos de menor comprimento, é possível obtê-los tomando valores de  $\theta$  com maior densidade *a posteriori*. Intervalos obtidos a partir deste critério, são chamados de intervalos HPD (*Highest Posterior Density*), que podem ser definidos como:

*Um intervalo de credibilidade C de  $100(1 - \alpha)\%$  para  $\theta$  é de máxima densidade a posteriori (HPD), se  $C = \{\theta \in \Theta : p(\theta | x) \geq k(a)\}$  onde  $k(a)$  é a maior constante tal que  $P(\theta \in C) \geq 1 - \alpha$ .*

Para maiores esclarecimentos sobre estimações intervalares, consultar Paulino et al. (2003).

Segundo Pagano e Gauvreau (2004), um intervalo de confiança diz mais a respeito de um parâmetro do que apenas sua abrangência. É possível aferir se uma diferença de médias (ou variâncias) é significativa ou não, ou se um parâmetro específico é relevante em determinado modelo, por exemplo. Este cenário é muito discutido em modelos de regressão, pois conforme o número de parâmetros aumenta, outros fatores atrelados a ele também aumentam, como por exemplo a soma de quadrados total do modelo, causando menor acurácia na análise. Assim, as estimações intervalares são de extrema importância para se tomar decisões.

Considera-se uma diferença não-significativa, ou uma influência não-significativa, quando o intervalo de credibilidade ou HPD contém o valor zero. Isto indica que o parâmetro pode assumir o valor zero ou estar tão próximo deste que não corrobora significativamente com o modelo ou mesmo que não é possível discriminar uma diferença entre amostras.

Tais inferências são necessárias para que o pesquisador seja capaz de tomar suas decisões de forma a reduzir a incerteza e aumentar sua probabilidade de acerto, fatores importantes em problemas reais, onde tempo, mão de obra e investimentos devem ser controlados.

Devido à grande quantidade de tipos de dados, são necessários diversos modelos estatísticos para que se analisem tais problemas, pois cada qual tem suas especificações e exigem modelos que se adaptem à elas.

### 1.1.7 Comparação de modelos

Após as estimações dos parâmetros, suponha que o pesquisador tenha um grupo de modelos candidatos que parecem plausíveis em relação ao ajuste dos dados. Quando se faz necessário optar por um ou outro modelo, usualmente no caso bayesiano se utiliza o DIC (*Deviance Information Criterion*) (SPIEGELHALTER et al., 2002), considerada uma generalização de modelagem hierárquica do AIC (*Akaike Information Criterion*) e do BIC (*Bayesian Information Criterion*).

Assim como o AIC e o BIC, o DIC é uma aproximação assintótica conforme o tamanho da amostra se torna suficientemente grande.

Tendo a *deviance* definida por

$$D(\theta) = -2\log(p(y | \theta)) + C \quad (1.12)$$

onde  $y$  são os dados,  $\theta$  é o vetor de parâmetros desconhecidos do modelo,  $p(y | \theta)$  é a função de verossimilhança e  $C$  é uma constante que é cancelada ao se comparar modelos diferentes, não sendo necessário ser conhecida.

Considerando  $\bar{D}$  como a média *a posteriori* da *deviance* e  $\hat{D}$  como um ponto estimado da *deviance* ao se substituir  $\theta$  por  $\bar{\theta}$ , ou seja

$$\hat{D} = -2\log(p(y | \bar{\theta})), \quad (1.13)$$

o número de parâmetros do modelo segundo Spiegelhalter et al. (2002), é obtido pela expressão

$$pD = \bar{D} - \hat{D}. \quad (1.14)$$

Dessa forma, o *Deviance Information Criterion* (DIC) é calculado por

$$DIC = pD + \bar{D} = \hat{D} + 2pD. \quad (1.15)$$

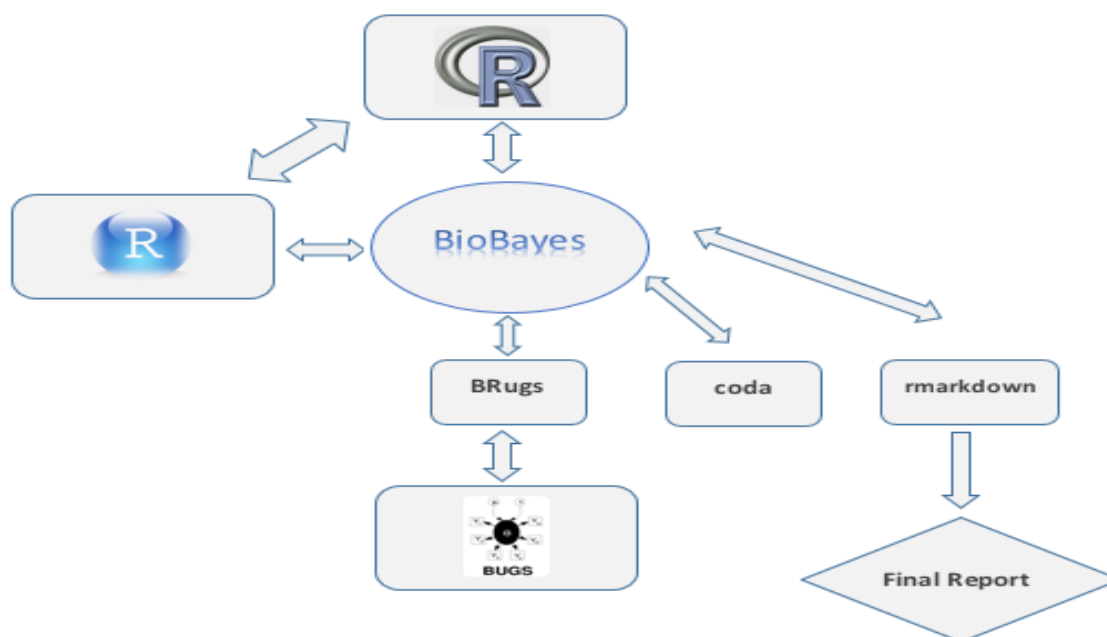
A partir da obtenção do valor de DIC para os modelos candidatos, este é interpretado da mesma forma que o AIC e BIC, considerando que o modelo com menor DIC é mais parcimonioso. Este fato ocorre pois os modelos são influenciados tanto pelo  $\bar{D}$  que favorece o bom ajuste, quanto pelo número efetivo de parâmetros  $pD$ . Dessa forma,  $\bar{D}$  diminui à medida que o número de parâmetros de um modelo aumenta, então o termo  $pD$  compensa esta penalização favorecendo modelos com menor número de parâmetros.

A vantagem de se utilizar o DIC em relação a outros critérios como AIC e BIC é de que este pode ser facilmente calculado a partir das amostras geradas pela simulação MCMC. Já o AIC e BIC requerem o cálculo da máxima verossimilhança, a qual não está facilmente disponível nas simulações.

## 1.2 Materiais e Métodos

A partir do vínculo criado entre o programa OpenBUGS e o R, por meio do pacote “BRugs” (Figura 3), foi possível desenvolver um algoritmo para análises bayesianas denominado BioBayes, que além de gerar as cadeias via MCMC *a posteriori* para os parâmetros de interesse, também gera, em formato HTML, um relatório completo de informações obtidas a partir da análise de determinado modelo. Este recurso é possível devido à funcionalidade de um pacote existente no R, chamado “rmarkdown”, ideal para construção de relatórios em formato *Word*, PDF, ou HTML.

Figura 3 – Diagrama de funcionamento do pacote BioBayes.



Fonte: Autoria própria.

A utilização do RStudio<sup>2</sup>, assim como um grupo de pacotes já existentes nele, são necessários para que o BioBayes funcione, entre eles:

- **BRugs** para obtenção de cadeias geradas via MCMC e cálculos de estimativas bayesianas;
- **fitdistplus** para a estimação dos valores iniciais utilizados na geração das cadeias via MCMC;
- **data.table** para a organização das cadeias geradas;
- **coda** para análise de convergência das cadeias MCMC;
- **rmarkdown** para geração do relatório em formato HTML com as principais informações das análises;
- **ggplot2** para a geração de gráficos no relatório;
- **RColorBrewer** para determinação das cores utilizadas no relatório.

Uma vez criada a função principal, denominada `BioBayes.fit`, esta será chamada em todos os tipos de modelos estatísticos criados posteriormente para as análises de dados. A `BioBayes.fit` carrega os argumentos necessários para que o usuário determine suas preferências a respeito da simulação, como:

<sup>2</sup> <<https://www.rstudio.com/>>

- `bugmodel`: nome do modelo desejado para a análise;
- `database`: dados que estão entrando no modelo, em formato de lista;
- `priors`: hiperparâmetros para as *priors*, em formato de lista;
- `parameters`: vetor com os nomes de saída dos valores após a análise bayesiana (definidos no modelo `*.bug`);
- `initials`: valores iniciais para os parâmetros a serem estimados, em formato de lista;
- `iter`: número de iterações para gerar as cadeias MCMC;
- `burn`: número de observações a serem descartadas no início (período de adaptação) do processo MCMC;
- `jump`: salto para eliminar possível autocorrelação na cadeia simulada;
- `Round`: número de casas após a vírgula a serem mostradas;
- `seed`: semente de geração fixada;
- `title`: título dos resultados, exemplo “Resultados Inferenciais Bayesianos”;
- `save_html`: Para não salvar, utiliza-se o comando `NULL`. Qualquer outra forma de preencher este campo fará com que o relatório seja salvo;
- `render`: Para gerar o HTML, utiliza-se `TRUE`.

Para mais detalhes a respeito dos argumentos da função principal, consultar o manual do pacote BioBayes (ANEXO A).

Por trás das escolhas do usuário, a `BioBayes.fit` ainda contém a implementação de algoritmos para gerar as tabelas com as estimativas dos parâmetros no relatório final, com seus respectivos nomes em cada coluna (tais quais variam, baseado na escolha do modelo), informações sobre os critérios de convergência entre outros detalhes necessários.

Do mesmo modo, há a implementação de todos os recursos gráficos necessários para o relatório final, com os comandos para gerar os gráficos das cadeias de Markov, de autocorrelação, densidades das distribuições *a posteriori*, entre outros, obtidos pelo pacote `ggplot2`.

Após a criação do algoritmo principal necessário para que o pacote funcione, foram implementados os modelos estatísticos a serem utilizados nas análises de dados, cada um com suas especificações, como por exemplo, a distribuição *a priori*, já intrínseca e imutável de cada um. Caso deseje, o usuário poderá determinar apenas os valores de seus

parâmetros, caso a queira informativa ou não, mas não será possível modificar o tipo de distribuição.

A escolha das distribuições *a priori* para a construção do pacote se baseia, em geral, nas famílias conjugadas, apresentadas na Tabela 1, portanto, têm largo uso na literatura. Entretanto, algumas são definidas de acordo com a restrição paramétrica, como por exemplo, dos modelos de regressão não-linear.

Para cada modelo estatístico implementado, existe uma descrição teórica, que encontra-se disponível no *help* do próprio pacote, capaz de ajudar o usuário a compreender a forma e a parametrização utilizada em cada caso. Há também, para todos os modelos inseridos, pelo menos um exemplo com a forma de entrada dos dados e demais especificações necessárias, também disponível juntamente com a descrição teórica.

Para cada diferente modelo de análise, há um algoritmo em linguagem *bug* (do mesmo modo em que eram utilizados no programa OpenBUGS), porém este não fica mais a cargo do usuário, pois é implementado dentro de cada função existente no pacote BioBayes. O usuário só terá acesso ao modelo *bug* utilizado ao final da análise, onde este é apresentado juntamente com as demais informações no relatório final.

## 1.3 Resultados

Após a inserção dos códigos necessários para implementação do pacote, o BioBayes resultou em um projeto com funções básicas para análise, que podem ser divididas em quatro módulos:

1. Funções de distribuições de probabilidade básicas;
2. Modelos para testar médias de duas populações;
3. Modelos para testar  $k > 2$  médias de populações normais;
4. Modelos de regressão.

O pacote já encontra-se disponível na plataforma GitHub pelo *link*:

[<https://github.com/rmrossidesuem/BioBayes>](https://github.com/rmrossidesuem/BioBayes).

### 1.3.1 Funcionalidades do pacote BioBayes

Nesta primeira versão do BioBayes, são, no total, 38 funções para análises de diversos tipos de dados, com suas respectivas descrições especificadas individualmente na Tabela 2.

Tabela 2 – Funções disponíveis no pacote BioBayes com suas respectivas descrições.

Módulo	Nome da função	Descrição
Distribuições de probabilidade	binomial_model	Modelo para a distribuição binomial.
	poisson_model	Modelo para a distribuição de Poisson.
	beta_model	Modelo para a distribuição beta.
	exponential_model	Modelo para a distribuição exponencial.
	gamma_model	Modelo para a distribuição gama.
Modelos para testar duas médias	normal_model	Modelo para a distribuição normal.
	binomial_diff_model	Modelo para diferença de proporções entre duas populações binomiais independentes.
	poisson_diff_model	Modelo para diferença de médias entre duas populações independentes com distribuição Poisson.
	beta_diff_model	Modelo para diferença de médias entre duas populações beta independentes.
	gamma_diff_model	Modelo para diferença de médias entre duas populações gama independentes.
	exponential_diff_model	Modelo para diferença de médias entre duas populações exponenciais independentes.
	normal_homo_diff_model	Modelo para diferença de médias entre duas populações normais independentes com variâncias iguais.
	normal_hetero_diff_model	Modelo para diferença de médias entre duas populações normais independentes com variâncias diferentes.
	normal_paired_model	Modelo para diferença de médias entre duas populações normais dependentes (pareadas).
Modelos para testar $k > 2$ médias de populações normais	anova_homo_model	Modelo para uma análise de variância para populações normais com variâncias homogêneas.
	anova_hetero_model	Modelo para uma análise de variância para populações normais com variâncias heterogêneas.

Módulo	Nome da função	Descrição
Modelos de regressão	linear_reg_model	Modelo de regressão linear simples.
	linear_multi_reg_model	Modelo de regressão linear múltipla.
	quadratic_reg_model	Modelo de regressão quadrática.
	quadratic_control_reg_model	Modelo de regressão quadrática versus grupo controle.
	bin_reg_model_logit	Modelo de regressão binomial com função de ligação logit.
	bin_multi_reg_model_logit	Modelo de regressão binomial múltipla com função de ligação logit.
	bin_multi_reg_model_probit	Modelo de regressão binomial múltipla com função de ligação probit.
	bin_multi_reg_model_loglog	Modelo de regressão binomial múltipla com função de ligação loglog.
	bin_multi_reg_model_cloglog	Modelo de regressão binomial múltipla com função de ligação cloglog.
	bin_multi_reg_model_cauchy	Modelo de regressão binomial múltipla com função de ligação Cauchy.
	bin_reg_model_logit_DR	Modelo de regressão binomial simples com função de ligação logit para dose-resposta.
	bin_reg_model_probit_DR	Modelo de regressão binomial simples com função de ligação probit para dose-resposta.
	bin_reg_model_cloglog_DR	Modelo de regressão binomial simples com função de ligação cloglog para dose-resposta.
	bin_reg_model_loglog_DR	Modelo de regressão binomial simples com função de ligação loglog para dose-resposta.
	bin_reg_model_cauchy_DR	Modelo de regressão binomial simples com função de ligação Cauchy para dose-resposta.
	linear_reg_model_poisson	Modelo de regressão linear simples com erros Poisson.
	linear_multi_reg_model_poisson	Modelo de regressão linear múltipla com erros Poisson.
	logistic_normal_model	Modelo de crescimento não-linear Logístico.
	gompertz_normal_model	Modelo de crescimento não-linear de Gompertz.
	brody_normal_model	Modelo de crescimento não-linear de Brody.
	vonbertalanffy_normal_model	Modelo de crescimento não-linear de Von Bertalanffy.
	wood_normal_model	Modelo para curva de lactação de Wood.



Para exemplificar o funcionamento e a saída HTML gerada pelo pacote, dois exemplos serão considerados: (1) o modelo de regressão quadrática e (2) o modelo para testar três médias de populações normais com variâncias heterogêneas.

#### 1.3.1.1 Exemplo 1: O modelo de regressão quadrática

A partir dos dados apresentados na Tabela 3 e após a entrada dos mesmos no programa RStudio, há a verificação do usuário de que estes podem ser ajustados por uma regressão quadrática (Figura 4) com os seguintes pressupostos:

$$Y \sim Normal(\mu, \tau) \text{ em que } \mu \text{ é o preditor linear e } \tau > 0 \text{ a precisão } (\sigma^2 = \frac{1}{\tau}, \text{ segundo a parametrização})$$

$$\mu = \beta_0 + \beta_1 x + \beta_2 x^2$$

e distribuições *a priori*

$$\beta_k \sim Normal(\mu\beta_k, \tau\beta_k) \text{ com } \mu\beta_k \text{ real, } \tau\beta_k > 0 \text{ e } k = 0, 1, 2;$$

$$\tau \sim Gama(a, b) \text{ com } a > 0, b > 0,$$

além de coordenadas do ponto de mínimo ou máximo da parábola estimada para os pontos críticos de  $x$  e  $y$  dadas pelas funções dos parâmetros:

$$PC_x = -\frac{\beta_1}{2\beta_2}$$

$$PC_y = -\frac{\beta_1^2 - 4\beta_0\beta_2}{4\beta_2}.$$

Tabela 3 – Dados hipotéticos para análise a partir de um modelo de regressão quadrática.

id	x	y	id	x	y	id	x	y	id	x	y	id	x	y
1	1	1,52	11	2	2,71	21	3	1,07	31	4	0,51	41	5	-2,15
2	1	1,68	12	2	2,28	22	3	1,49	32	4	0,15	42	5	-1,90
3	1	2,58	13	2	2,30	23	3	1,09	33	4	0,75	43	5	-2,43
4	1	1,84	14	2	2,16	24	3	1,24	34	4	0,74	44	5	-0,72
5	1	1,86	15	2	1,82	25	3	1,29	35	4	0,71	45	5	-1,20
6	1	2,66	16	2	2,99	26	3	0,76	36	4	0,64	46	5	-2,36
7	1	2,03	17	2	2,35	27	3	2,02	37	4	0,58	47	5	-2,00
8	1	1,17	18	2	1,12	28	3	1,68	38	4	0,27	48	5	-2,03
9	1	1,46	19	2	2,45	29	3	1,03	39	4	0,15	49	5	-1,41
10	1	1,58	20	2	1,86	30	3	2,23	40	4	0,11	50	5	-1,84

Logo, determina-se como argumento na função principal o `modelo_reg_quadratica`, além da entrada de dados em formato lista, assim como especificações sobre os parâmetros das distribuições *a priori* (se esta for informativa ou não), o nome dos parâmetros

utilizados pelo modelo (ver ANEXO A, p. 51), os valores iniciais, o número de iterações desejadas, tal qual o número de *burn* e salto que o usuário achar pertinente. Para este exemplo foram utilizadas 100.000 iterações, 10.000 de *burn*, 100 de salto, e distribuições *a priori* não-informativas, como:

$$\begin{aligned} &Normal(0, 10^{-6}) && \text{para os coeficientes } \beta_k, k = 0, 1, 2 \text{ e} \\ &Gama(10^{-3}, 10^{-3}) && \text{para } \tau. \end{aligned}$$

Os valores iniciais foram dados pelo parâmetros estimados na regressão quadrática frequentista, isto é,  $\beta_0 = 0,8378$ ,  $\beta_1 = 1,4009$  e  $\beta_2 = -0,3839$ , por meio da função `lm` do R:

```
fit_freq <- lm(y~x+I(x^2))
```

Call:

```
lm(formula = y ~ x + I(x^2))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	0.83776	0.32002	2.618	0.0119	*
x	1.40085	0.24388	5.744	6.55e-07	***
I(x^2)	-0.38392	0.03988	-9.627	1.08e-12	***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4719 on 47 degrees of freedom

Multiple R-squared: 0.9071, Adjusted R-squared: 0.9031

F-statistic: 229.3 on 2 and 47 DF, p-value: < 2.2e-16

Variance\_freq =  $0.4719^2 = 0.2226896$

Após as simulações, obtém-se um relatório HTML com informações sobre a análise na forma mostrada pelas Figuras 5 a 8 e pela Tabela 4.

As estimativas pontuais e intervalares geradas pelo método bayesiano são mostradas na Figura 7, contendo média (*Mean*), desvio-padrão da média (*Std. Errors*), mediana (*Median*), intervalos de confiança ( $ICr(95\%)$ ) e intervalos de alta densidade ( $HPD(95\%)$ ) para os parâmetros  $\beta_k$ ,  $\tau$ ,  $\sigma^2$ , além de um valor predito (*y.pred*) para um  $x_0$  (à escolha do usuário, neste caso  $x_0 = 2.5$ ), e também para os pontos críticos (*PC.x* e *PC.y*), coordenadas do ponto de máximo/mínimo da equação quadrática estimada.

Figura 4 – Diagrama de dispersão entre as variáveis X (independente) e Y (dependente) para os dados da Tabela 3.

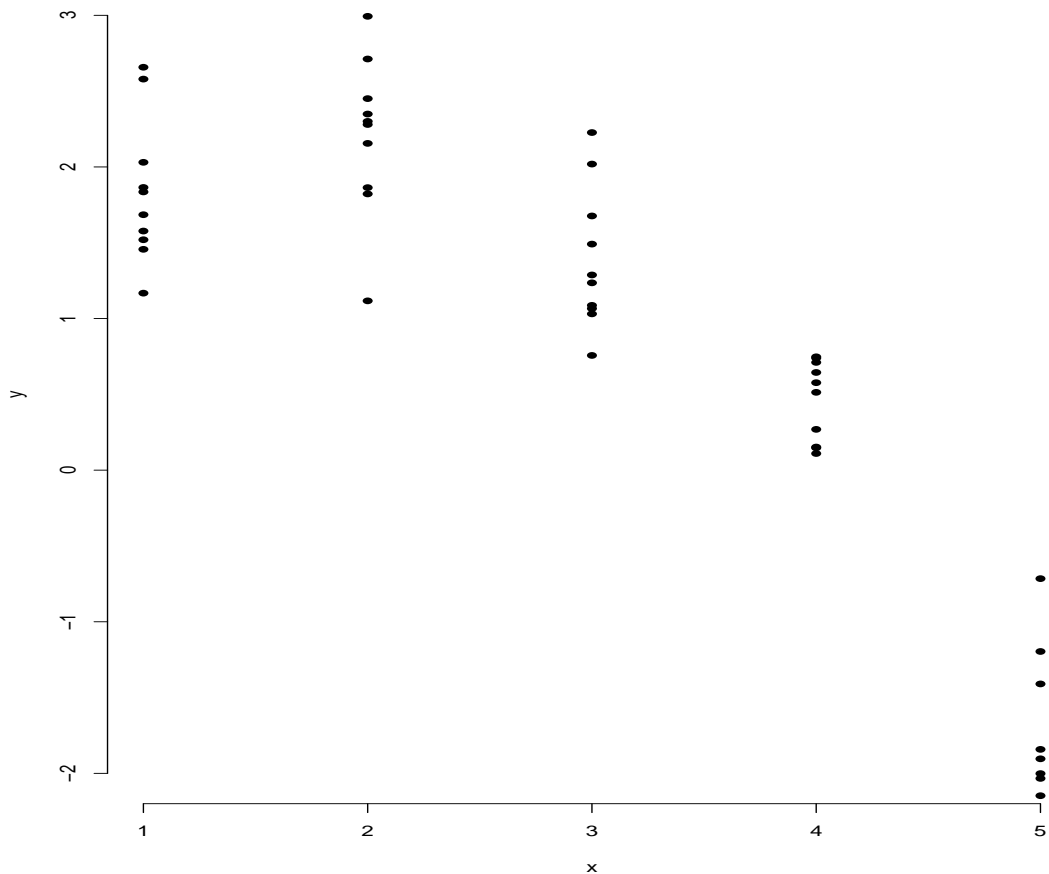
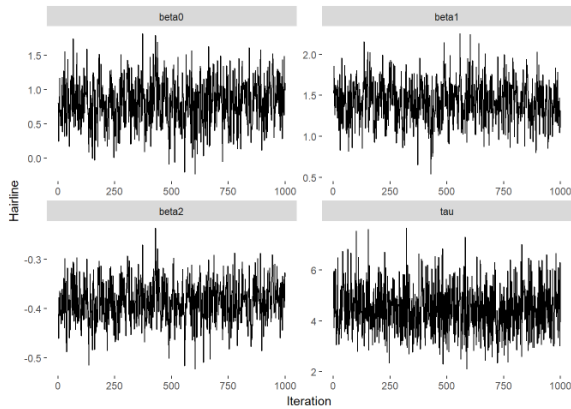


Figura 5 – Gráfico dos traços das cadeias de Markov geradas a partir da simulação e da autocorrelação para cada parâmetro, respectivamente.

Simulation Results

Bayesian Inferential Results

Chain graphics



Autocorrelation

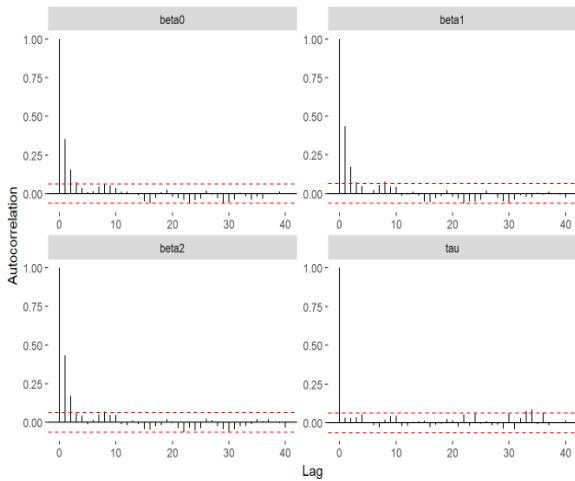


Figura 6 – Gráfico das densidades *a posteriori* geradas a partir da simulação e dos testes de convergência segundo os critérios de Heidelberg e Welch para a convergência, respectivamente.

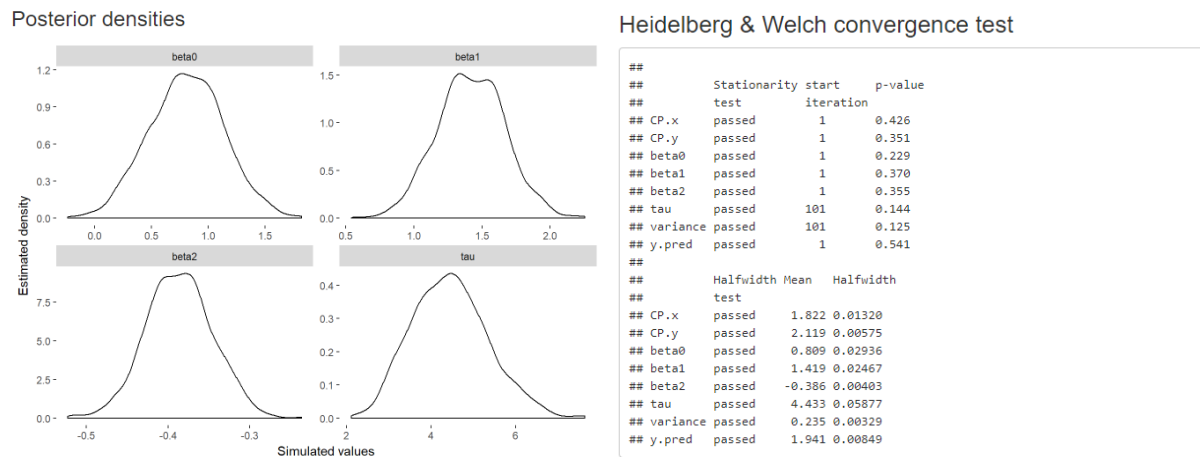


Figura 7 – Estimativas *a posteriori* geradas pelo modelo de regressão quadrática.

### Summary of posterior distributions

	Mean	Std. Error	Median	ICr 95%	HPD 95%
beta0	0.809	0.328	0.811	(0.174; 1.46)	(0.167; 1.438)
beta1	1.419	0.249	1.419	(0.93; 1.915)	(0.965; 1.936)
beta2	-0.386	0.041	-0.387	(-0.465; -0.308)	(-0.465; -0.308)
tau	4.451	0.898	4.418	(2.903; 6.376)	(2.8; 6.202)
variance	0.234	0.05	0.226	(0.157; 0.345)	(0.154; 0.333)
y.pred	1.941	0.105	1.944	(1.732; 2.136)	(1.741; 2.141)
CP.x	1.822	0.143	1.836	(1.526; 2.064)	(1.562; 2.1)
CP.y	2.119	0.096	2.118	(1.925; 2.309)	(1.945; 2.319)

Tabela 4 – DIC para o modelo quadrático.

DIC	Dbar	Dhat	DIC	pD
y	67,89	63,79	71,99	4,097
total	67,89	63,79	71,99	4,097

O modelo \*.bug utilizado internamente pelo pacote é mostrado ao final, juntamente com as demais informações no relatório HTML. A forma com que este aparece pode ser visto na Figura 8.

Figura 8 – Modelo quadrático em formato \*.bug já implementando no pacote BioBayes.

## Simulation specifications

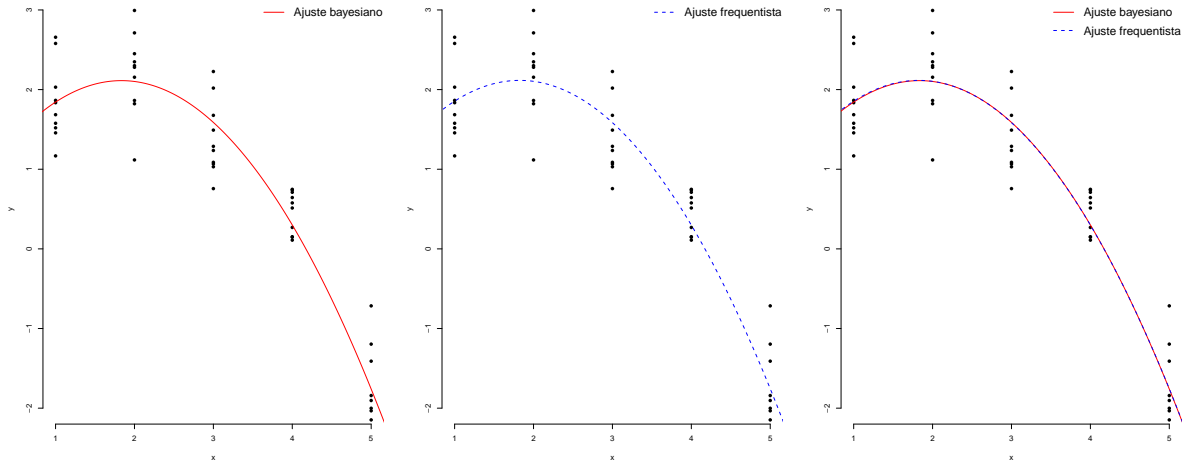
```
## Priors:
## mu_beta = 0
## tau_beta = 1e-06
## a_tau = 0.001
## b_tau = 0.001
##
## BUG Model:
## function (n, beta0, beta1, beta2, x, x0, tau)
## {
##   for (i in 1:n) {
##     y[i] ~ dnorm(mu[i], tau)
##     mu[i] <- beta0 + beta1 * x[i] + beta2 * pow(x[i], 2)
##   }
##   beta0 ~ dnorm(mu_beta, tau_beta)
##   beta1 ~ dnorm(mu_beta, tau_beta)
##   beta2 ~ dnorm(mu_beta, tau_beta)
##   tau ~ dgamma(a_tau, b_tau)
##   variance <- 1/tau
##   y.pred <- beta0 + beta1 * x0 + beta2 * pow(x0, 2)
##   CP.x <- -(beta1/(2 * beta2))
##   CP.y <- -((pow(beta1, 2) - 4 * beta2 * beta0)/(4 * beta2))
## }
##
## specifications used:
##
## Interactions: 1e+05
## Burn: 10000
## Jump: 100
```

Apesar das oscilações que são comuns nas cadeias de Markov, há no geral, uma estabilização em torno de um valor central, para todos os parâmetros (Figura 5). Já para a autocorrelação, nota-se que esta não mostra-se significativa, pois todos os pontos se encontram dentro dos limites considerados aceitáveis.

Pelos gráficos das densidades *a posteriori* (Figura 6), é possível perceber que para os parâmetros  $\beta_k$ , a *posteriori* possui simetria considerável, comum em casos em que estes foram gerados de uma distribuição normal. Isto indica que a média será um bom estimador, coincidindo com a mediana. Observa-se que todos os parâmetros convergiram para um valor, segundo os critérios de Heidelberg e Welch, pois passaram nos testes de convergência (Figura 6). O valor para DIC (71, 99) do modelo ajustado também é apresentado (Tabela 4), porém este não tem utilização neste caso, pois não há comparação entre modelos, como por exemplo, comparação entre o modelo quadrático e o linear. No caso de uma comparação, o melhor modelo possuiria o menor valor de DIC.

O ajuste gráfico aos dados e a comparação ao método frequentista obtido pelo modelo de regressão quadrática pode ver verificado pela Figura 9.

Figura 9 – Gráfico dos ajustes do modelo quadrático pelo método bayesiano, frequentista, e a comparação simultânea entre ambos.



Pelos gráficos da Figura 9, verifica-se que o ajuste bayesiano e frequentista são equivalentes, muito comum nos casos em que a *priori* é não-informativa, e que, aparentemente há um bom ajuste aos dados. Dado que os ajustes são equivalentes, podemos nos basear no  $R^2$  do modelo frequentista, que resultou em  $R^2 = 0,9030$ , indicando que o modelo é capaz de explicar em torno de 90% da variabilidade dos dados, considerado um valor eficiente para um modelo.

#### 1.3.1.2 Exemplo 2: O modelo para testar médias de populações normais com variâncias heterogêneas

Havendo a necessidade de uma comparação de médias entre  $k > 2$  tratamentos, como no caso dos dados apresentados na Tabela 5 e apresentados graficamente pelo *boxplot* na Figura 10, bem como, não havendo informação sobre a igualdade de variâncias, pode-se utilizar um procedimento para comparar populações normais com variâncias heterogêneas, que irá testar além da igualdade de médias *a posteriori*, a heterocedasticidade. Escolhido o modelo, define-se como argumento na função principal o `modelo_anova_hetero`, assim como as especificações sobre os valores dos parâmetros das distribuições *a priori* (neste caso será usada novamente uma *priori* não-informativa), o número desejado de iterações, fixado neste caso em 10.000, *burn* fixado em 1.000 e salto definido como 1. As distribuições *a priori* não-informativas foram determinadas como:

$$\begin{array}{ll} \text{Normal}(0, 10^{-6}) & \text{para as médias de cada tratamento } \mu_k, \ k = 1, 2, 3 \text{ e} \\ \text{Gama}(10^{-3}, 10^{-3}) & \text{para as precisões } \tau_k. \end{array}$$

Tabela 5 – Dados hipotéticos para comparação de médias *a posteriori* de três populações normais.

id	grupo	y	id	grupo	y
1	1	-0,56	20	2	2,53
2	1	-0,23	21	2	1,93
3	1	1,56	22	2	2,78
4	1	0,07	23	2	1,97
5	1	0,13	24	2	2,27
6	1	1,72	25	2	2,37
7	1	0,46	26	3	-0,06
8	1	-1,27	27	3	7,51
9	1	-0,69	28	3	5,46
10	1	-0,45	29	3	1,59
11	2	4,22	30	3	8,76
12	2	3,36	31	3	6,28
13	2	3,40	32	3	4,11
14	2	3,11	33	3	7,69
15	2	2,44	34	3	7,63
16	2	4,79	35	3	7,46
17	2	3,50	36	3	7,07
18	2	1,03	37	3	6,66
19	2	3,70			

Os valores iniciais foram obtidos pelo método frequentista, utilizando a média de  $y$  dentro de cada grupo, isto é,  $\mu_1 = 0,7462$  ( $\sigma_1 = 0,9537$ ),  $\mu_2 = 2,8946$  ( $\sigma_2 = 0,9684$ ) e  $\mu_3 = 5,8472$  ( $\sigma_3 = 2,6799$ ).

Após as simulações, obteve-se um relatório HTML com informações a respeito da análise na forma mostrada pelas Figuras 11 a 14 e Tabela 6.

As estimativas pontuais e intervalares para os parâmetros geradas pelo método bayesiano aparecem na mesma forma que o exemplo anterior, porém neste caso, acrescidas das comparações dois-a-dois (contrastes) entre as médias e desvios-padrão *a posteriori* dos grupos de tratamentos.

Figura 10 – *Boxplot* dos dados de três populações normais com variâncias heterogêneas.

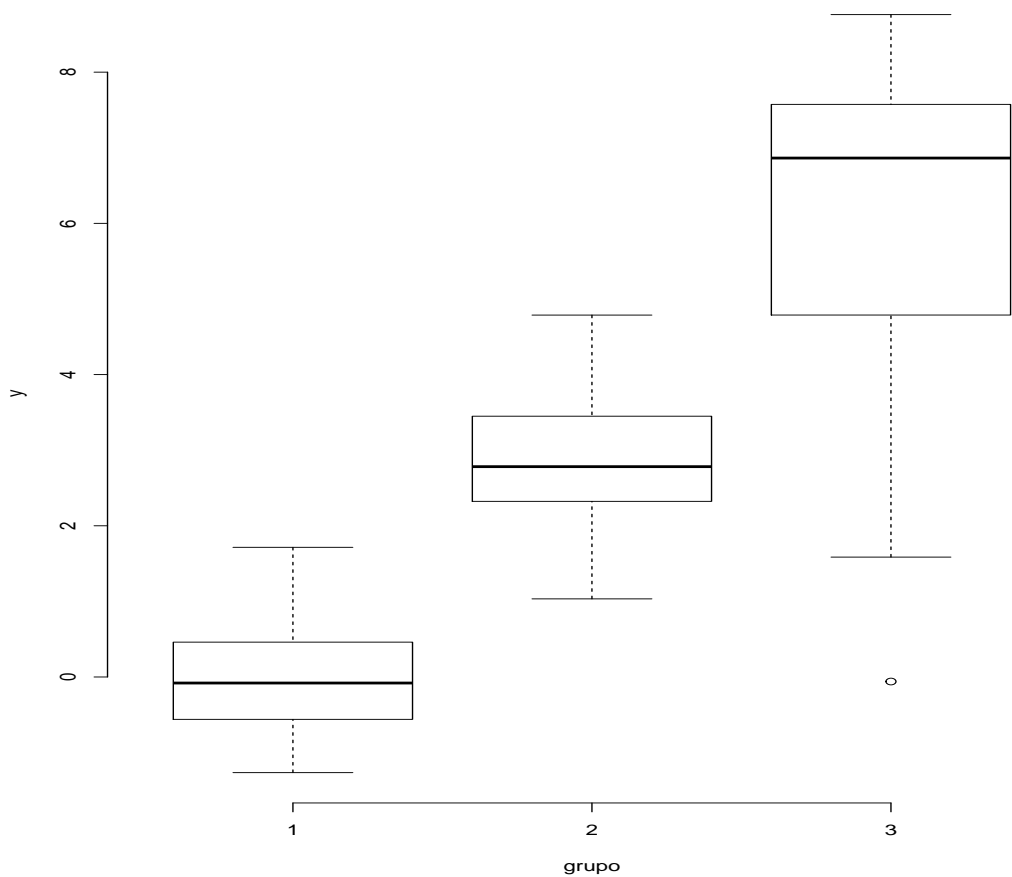
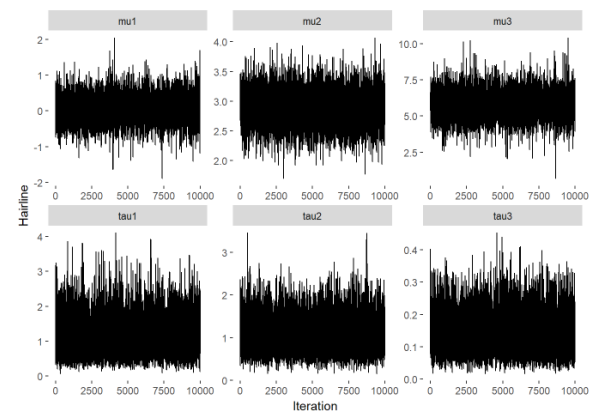


Figura 11 – Gráfico das cadeias de Markov geradas a partir da simulação e da autocorrelação para cada parâmetro, respectivamente.

Simulation Results

Bayesian ANOVA Results

Chain graphics



Autocorrelation

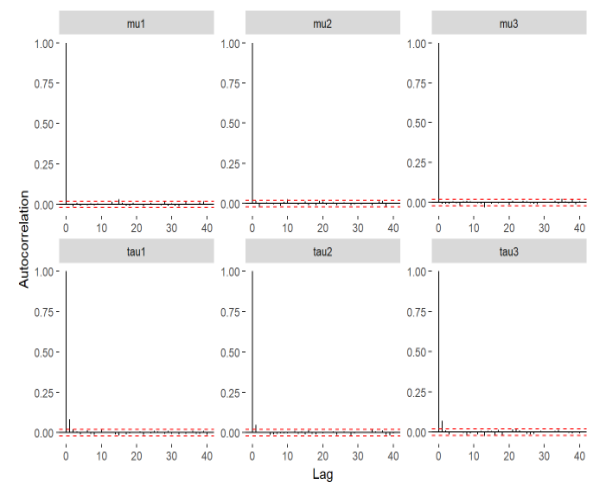




Figura 12 – Gráfico das densidades *a posteriori* geradas a partir da simulação e dos testes de convergência segundo os critérios de Heidelberg e Welch, respectivamente.

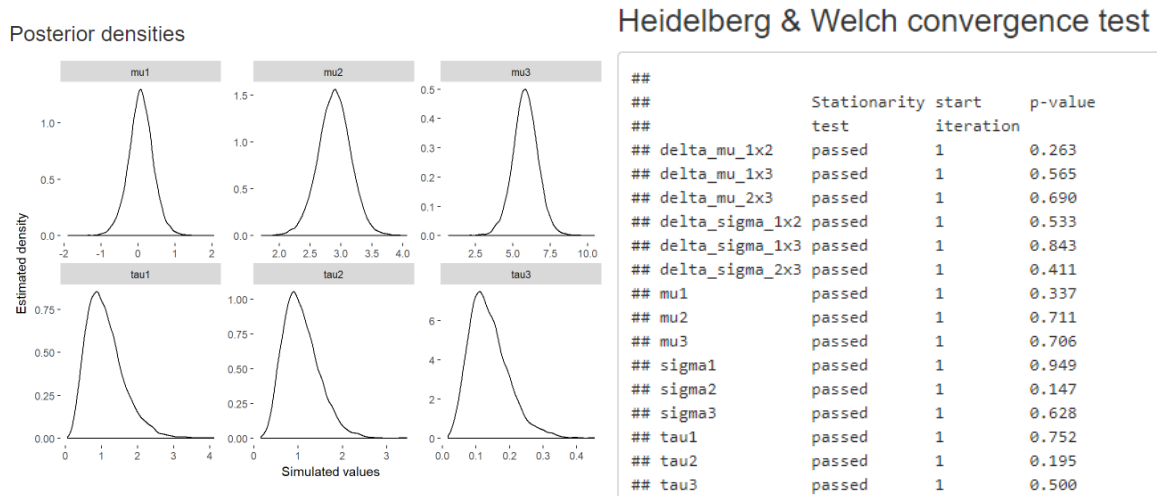


Figura 13 – Estimativas *a posteriori* geradas do modelo para populações normais com variâncias heterogêneas.

Summary of posterior distributions

	Mean	Std. Error	Median	ICr 95%	HPD 95%
mu1	0.074	0.34	0.074	(-0.616; 0.757)	(-0.64; 0.72)
mu2	2.892	0.269	2.893	(2.349; 3.412)	(2.349; 3.411)
mu3	5.831	0.856	5.828	(4.103; 7.499)	(4.102; 7.499)
tau1	1.1	0.518	1.02	(0.338; 2.343)	(0.245; 2.142)
tau2	1.071	0.403	1.018	(0.436; 1.983)	(0.373; 1.869)
tau3	0.139	0.059	0.13	(0.048; 0.28)	(0.033; 0.251)
sigma1	1.042	0.278	0.99	(0.653; 1.721)	(0.587; 1.577)
sigma2	1.022	0.211	0.991	(0.71; 1.514)	(0.664; 1.429)
sigma3	2.887	0.687	2.776	(1.889; 4.578)	(1.731; 4.189)
delta_mu_1x2	-2.817	0.434	-2.817	(-3.686; -1.962)	(-3.692; -1.97)
delta_mu_1x3	-5.757	0.918	-5.751	(-7.587; -3.939)	(-7.625; -4.005)
delta_mu_2x3	-2.94	0.898	-2.935	(-4.717; -1.152)	(-4.786; -1.227)
delta_sigma_1x2	0.019	0.351	0	(-0.622; 0.779)	(-0.685; 0.703)
delta_sigma_1x3	-1.845	0.739	-1.763	(-3.58; -0.648)	(-3.381; -0.549)
delta_sigma_2x3	-1.865	0.718	-1.757	(-3.578; -0.757)	(-3.328; -0.643)

Tabela 6 – DIC do modelo para populações normais com variâncias heterogêneas.

DIC	Dbar	Dhat	DIC	pD
y	130,1	123,8	136,4	6,338
total	130,1	123,8	136,4	6,338

Figura 14 – Modelo para populações normais heterocedásticas em formato \*.bug já implementando no pacote BioBayes.

### Simulation specifications

```
## Priors:
## mu_mu = 0
## tau_mu = 1e-06
## a_tau = 0.001
## b_tau = 0.001
##
## BUG Model:
## model;
## {
##   for (k in n1_l:n1_u) {
##     y[k] ~ dnorm(mu1, tau1)
##   }
##   for (k in n2_l:n2_u) {
##     y[k] ~ dnorm(mu2, tau2)
##   }
##   for (k in n3_l:n3_u) {
##     y[k] ~ dnorm(mu3, tau3)
##   }
##   mu1 ~ dnorm(mu_mu, tau_mu)
##   mu2 ~ dnorm(mu_mu, tau_mu)
##   mu3 ~ dnorm(mu_mu, tau_mu)
##   tau1 ~ dgamma(a_tau, b_tau)
##   tau2 ~ dgamma(a_tau, b_tau)
##   tau3 ~ dgamma(a_tau, b_tau)
##   sigma1 <- 1/sqrt(tau1)
##   sigma2 <- 1/sqrt(tau2)
##   sigma3 <- 1/sqrt(tau3)
##   delta_mu_1x2 <- mu1 - mu2
##   delta_mu_1x3 <- mu1 - mu3
##   delta_mu_2x3 <- mu2 - mu3
##   delta_sigma_1x2 <- sigma1 - sigma2
##   delta_sigma_1x3 <- sigma1 - sigma3
##   delta_sigma_2x3 <- sigma2 - sigma3
## }
##
## specifications used:
##
## Interactions: 10000
## Burn: 1000
## Jump: 1
```

As interpretações dos gráficos das cadeias de Markov, autocorrelação, densidades *a posteriori* e DIC são análogas ao Exemplo 1.

Dado que foi utilizado um modelo para comparar três populações normais com variâncias heterogêneas, estas devem ser analisadas primeiro para que se verifique se o modelo heterocedástico é o ideal. Dessa forma, ao se comparar os pares  $\Delta_{\sigma_{12}} = \sigma_1 - \sigma_2$ ,  $\Delta_{\sigma_{13}} = \sigma_1 - \sigma_3$  e  $\Delta_{\sigma_{23}} = \sigma_2 - \sigma_3$  a partir das estimativas intervalares (ICr e/ou HPD), podemos inferir com 95% de credibilidade que as variâncias dos grupos 1 e 3 (ICr :  $(-3, 58; -0, 648)$ ) e 2 e 3 (ICr :  $(-3, 578; -0, 757)$ ) são diferentes entre si, porém não temos evidências de que haja diferença entre as variâncias do grupo 1 e 2 (ICr :  $(-0, 622; 0, 7798)$ ), já que o zero encontra-se contido neste intervalo.

Considerando que houve a confirmação de variâncias heterogêneas, a comparação de médias pode ser realizada, também por meio da verificação dos limites inferiores e superiores dos intervalos de credibilidade e/ou HPD. Considerando um nível de credibilidade  $1 - \alpha = 0,95$ , observa-se que no caso da comparação entre os pares  $\Delta_{\mu_{12}} = \mu_1 - \mu_2$ ,  $\Delta_{\mu_{13}} = \mu_1 - \mu_3$  e  $\Delta_{\mu_{23}} = \mu_2 - \mu_3$ , todos podem ser considerados significativamente diferentes, pois o valor zero não encontra-se contido no intervalo de seus respectivos  $\Delta$ .

Todos os parâmetros passaram no teste de convergência segundo os critérios de Heidelberg e Welch, indicando que todos convergiram para um determinado valor (Figura 12).

Se compararmos as estimativas bayesianas e frequentistas para as médias, por exemplo, temos que estas são consideravelmente iguais, pois diferem entre si apenas na terceira casa decimal, indicando que as distribuições *a priori* funcionaram como o esperado, por serem não-informativas, resultando na equivalência de resultados entre o método bayesiano e o método frequentista.

Apesar das facilidades advindas do pacote BioBayes, cabe dizer que é de extrema necessidade que o usuário tenha, no mínimo, conhecimento estatístico básico, capaz de julgar a escolha dos modelos, fazer interpretações adequadas e lidar com demais problemas provenientes dos dados. A simplicidade de uso do pacote não exime o usuário de seu conhecimento prévio sobre estatística!

## 1.4 Considerações finais

Pôde observar-se que o pacote desenvolvido BioBayes supriu o esperado em relação à sua implementação, às análises estatísticas e à execução conjunta ao programa OpenBUGS.

O relatório gerado em formato HTML pelo rmarkdown contém as informações principais a respeito da análise, propiciando ao usuário fazer inferência a respeito dos dados.

---

## Seção 2

---

# Capítulo 2

---

BioBayes Package for Bayesian Analysis

# BioBayes Package for Bayesian Analysis

by Stéfane Lele, Robson Marcelo Rossi

**Abstract** This article shows details about idealization, development and operation of a package for bayesian-focused data analysis, called **BioBayes**. The main goal is to facilitate the bayesian analysis jointly to OpenBUGS, allowing the user to have more flexibility in practical applications like unconcern about the sample size, prior knowledge insertion and others. Through the program R, were implemented models for statistical analysis using both the R features (packages previously developed) and the OpenBUGS program. The collection of available distribution functions includes from basic models of probability distribution, in addition to means comparisons models; analysis of variance; linear, non-linear and log-linear regression models (including for dose-response); growth and lactation curves. After the analysis, a technical report in HTML is automatically generated, containing graphics about the chains, estimates, posterior densities for the parameters, convergence analysis, DIC (Deviance Information Criterion), and the internally used model in the language *bug*.

## Introduction

The different types of data found daily, united to a technological advanced and in-depth studies about statistical methods, it allows the academic community to explore newer techniques for analysis of results with different points of view, improving the accuracy and the quality of results.

The bayesian statistic gained strength from the computational advancement, occurred in the 80's due to the need for simulations with innumerable iterations for parameters estimating. This technique is ideal mostly for the cases where the sample size is insufficient for assume asymptotic distribution, or even when there is a need to aggregate prior information to the data obtained by the researcher through previous analysis, experience in the field, literature reviews or meta-analysis. Those advantages are possible due to the bayesian concept of considering the parameter vector as random quantities.

It is known that in the classical method, it is possible to obtain information about a parameter of interest from the sample, by the likelihood function, allowing the researcher to make estimates about a parameter  $\theta$  only using sample information. However, according to [Paulino et al. \(2003\)](#), the information already known about the parameter of interest  $\theta$  is of great importance, although the true  $\theta$  is still unknown, making mention of the importance of the prior distribution. The authors also claims that if  $\theta$  is unknown, this is uncertain, and all uncertainty must be quantified in terms of probability.

By the fact that the parameter is considered random in the Bayesian concept (unlike the frequentist, where the estimator is the one that has distribution), its information is given by the greater probability of this parameter assuming a certain value. The use of this probability in the inference decreases the dependence of the estimates in a relation to the number of observations, requiring a smaller amount of these ([Silva, 2006](#)).

Another vantage is that even working with a small number of observations, do are not generated infinities confidence intervals, like occurs in classic inference, getting them by iterative methods ([Nogueira et al., 2003](#)). The same can be done with the estimates for parameters when obtaining the marginal distributions are too complex or non-feasible, can make use of intensive computational methods of simulation like Metropolis-Hastings ([Metropolis et al., 1953](#); [Hastings, 1970](#)), Gibbs sampling in Markov Chains ([Geman and Geman, 1984](#)), Hibrid Monte Carlo ([Neal et al., 2011](#)), R packages like **MCMCglmm** ([Hadfield et al., 2017](#)), **BRugs** ([Thomas et al., 2017](#)), **INLA**<sup>1</sup> ([Martino and Rue, 2009](#)) and others.

There are several vantages in the using of bayesian methodology, such as accept the reality of the data, without assuming that they come from a normal distribution ([Rossi, 2011](#)); incorporate informations previously acquired about the parameter, enriching the process of inference [Nogueira et al. \(2003\)](#), among many others that can be found in the literature.

After the technological advancement and the increase in demand for the bayesian method, a new tool emerged for analysis: a software called BUGS (Bayesian inference Using Gibbs Sampling) ([Lunn et al., 2009](#)), such that in 1997 had its version available for Windows, the WinBUGS. In 2005, OpenBUGS<sup>2</sup> was created, as open source variant of WinBUGS. Also, in the same year, was developed a R<sup>3</sup> package, denominated BRugs, and that works interactively with the OpenBUGS, making use of a MCMC sampling for bayesian analysis. According to [Lunn et al. \(2009\)](#), one of the goals that motivated the creation of OpenBUGS was

“[...]uncouple the principal functionality and and the user interface, for facilitate the use of software from another environments. The most significant contribution to this, was the

<sup>1</sup><http://www.r-inla.org/>

<sup>2</sup><http://www.openbugs.net/>

<sup>3</sup><http://www.r-project.org/>

BRugs, a interface that enable the user to use BUGS interactively within the R.”

A disadvantage observed when using OpenBUGS, is that it needs a extra file containing a statistical model described, in \*.bug. This means that, for each new change, the original file must be modified or be created a copy. In addition, there is a big difficulty about reading large data and the click-point process.

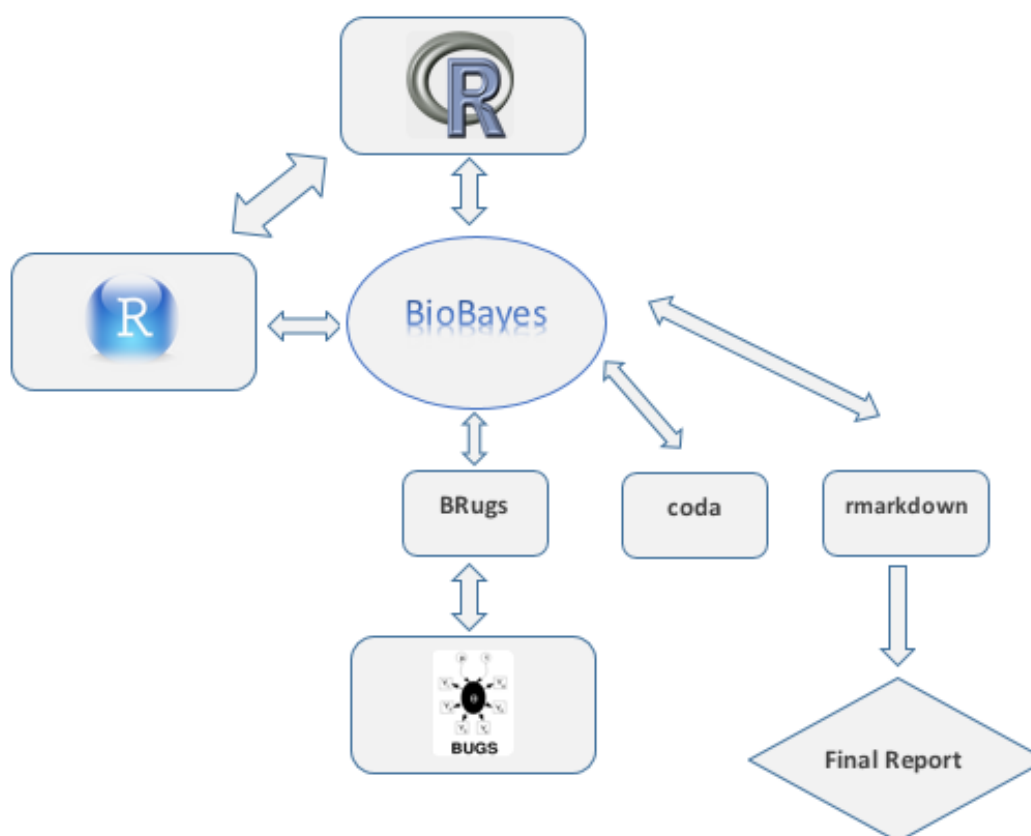
The **BioBayes** package for R, was created precisely to counteract these disadvantages in addition to making use of the advantages already developed so far.

As an example of the operation of BioBayes, a credit analysis was performed using logistic regression with a bayesian approach, to fit a model that explains the chances of a customer becoming a defaulter. After estimation of the parameters, it was verified that all passed in the convergence tests, being possible to infer, that some factors such as age, number of dependents, profession, education and type of housing are significant for a customer to become defaulter with 95% credibility.

## Materials and methods

### The BioBayes

From the bond created between the OpenBUGS and R, through the BRugs (Figure 1 ), it was possible to develop an algorithm for the bayesian analysis, called **BioBayes**, currently available on <https://github.com/rmrossidesuem/BioBayes>, which, in addition to generating Markov chains for the interesting parameter, also generates, in HTML format, a complete report that contains the information obtained through the analysis. This feature is possible due to the functionality of an existing package in R, called **rmarkdown**, ideal for creating reports in Word, PDF or HTML formats.



**Figure 1:** Operating diagram of the **BioBayes** package.

Font: Own authorship.

To use the **BioBayes**, it is necessary to install the OpenBUGS and the RStudio<sup>4</sup>, as well, to make use of the already existing packages in it, as follows:

<sup>4</sup><https://www.rstudio.com/>

- **BRugs** to generate the Markov Chains Monte Carlo (MCMC), in addition to calculations of the bayesian estimates ;
- **fitdistplus** for the estimation of initial values, used to generate the MCMC chains;
- **data.table** for organization of the generated chains;
- **coda** for convergence analysis of MCMC chains;
- **rmarkdown** to create a HTML report, which contains the most important informations obtained in the analisys;
- **ggplot2** to create the report graphics;
- **RColorBrewer** for the colors used in report.

Once the principal function was created, called `BioBayes.fit`, it will be used in all the statistical models posteriorly created for the data analysis. The `BioBayes.fit` bring the necessary arguments for the user to determine their preferences about the simulation, like:

- `bug_model`: name of the model desired for the analysis;
- `database`: data that is entering the model, in format list;
- `priors`: hyperparameters for priors, in format list;
- `parameters`: vector with the names of the output values after the bayesian analysis (described in `*.bug model`);
- `initials`: initial values for the parameters that will be estimated, in list format;
- `iter`: number of iterations to generate the MCMC chains;
- `burn`: number of observations to be discarded in the beginning (adaptation period) of the MCMC process;
- `jump`: jump to eliminate possible autocorrelation in the simulated chain;
- `Round`: number of decimals values after the comma to be showed;
- `seed`: fixed seed to generate random values;
- `title`: title for the results, example: "Bayesian Inferential Results";
- `save_html`: For not to save, use NULL. Any different word, causes the report to be saved;
- `render`: To generate the HTML report, use TRUE.

Behind the user's choices, the `BioBayes.fit` still contains the implementation of algorithms to generate tables showing the estimates of the parameters in the final report, with their respective names in each column (which may vary, depending on the model chosen), informations about the convergence criterion and others details that may be needed.

In the same way, there is the implementation of all graphic resources for the final report, so that to be showed the MCMC chains, autocorrelation and posterior distribution densities, obtained by the **ggplot2** package.

After the creation of the principal algorithm, the statistical models for data analysis were implemented, each one with their own specifications. The priors distribution is already intrinsic and immutable for each one. If necessary, the user can only change the parameter values for informative and non-informative priors, but it will not to be possible modify the type of distribution.

The choices of a prior distributions to build the package is based, in general, on the conjugate families found in the literature. However, some of them are defined according to parametric restriction, as for example, of the non-linear regression models.

For each statistical model implemented, there is a theoretical description, available in the "help" of **BioBayes** package, able to support the user to understand the use and the parametrization in each case. In addition, for all implemented models, there is at least one example with the form of data input and another necessary specifications, along with the theoretical description. Also, there are the algorithms in *bug* language (in the same way as they are used in OpenBUGS), but these are no longer the responsibility of the user anymore, because they already have been implemented within those functions. The user will only have access to the *bug* model used at the end of the analysis, where it is shown along with the another specifications in the final report.

After the insertion of the necessary codes for the implementation of package, the **BioBayes** has resulted into a project with basic functions for analysis, which can be divided into four modules:

1. Basic probability distributions functions;
2. Models for testing means of two populations;
3. Models for testing  $k > 2$  means of normal populations;
4. Regression models.

In this version of the **BioBayes**, a total of 38 functions were implemented, for many types of data, as shown in Table 1.

Module	Function name	Description
Probability distributions	binomial_model poisson_model beta_model exponential_model gamma_model normal_model	Model for binomial distribution. Model for Poisson distribution. Model beta distribution. Model for exponential distribution. Model for gamma distribution. Model for normal distribution.
Models for testing two means	binomial_diff_model  poisson_diff_model  beta_diff_model  gamma_diff_model  exponential_diff_model  normal_homo_diff_model  normal_hetero_diff_model  normal_paired_model	Model for difference between proportional of two binomials independent populations. Model for difference between means of two Poisson independent populations. Model for difference between means of two beta independent populations. Model for difference between means of two gamma independent populations. Model for difference between means of two exponential independent populations. Model for difference between means of two normal independent populations with homoscedasticity. Model for difference between means of two normal independent populations with heteroskedasticity. Model for difference between means of two normal dependent populations (paired).
Models for testing $k > 2$ means of normal populations	anova_homo_model anova_hetero_model	Model for analysis of variance for normal populations with homoscedasticity. Model for analysis of variance for normal populations with heteroskedasticity.
Regression models	linear_reg_model linear_multi_reg_model quadratic_reg_model quadratic_control_reg_model  bin_reg_model_logit  bin_multi_reg_model_logit  bin_multi_reg_model_probit  bin_multi_reg_model_loglog  bin_multi_reg_model_cloglog  bin_multi_reg_model_cauchy  bin_reg_model_logit_DR  bin_reg_model_probit_DR  bin_reg_model_cloglog_DR  bin_reg_model_loglog_DR  bin_reg_model_cauchy_DR  linear_reg_model_poisson  linear_multi_reg_model_poisson	Simple linear regression model. Multiple linear regression model. Quadratic regression model. Quadratic regression model versus control group. Binomial regression model with link logit. Multiple binomial regression model with link logit. Multiple binomial regression model with link probit. Multiple binomial regression model with link loglog. Multiple binomial regression model with link cloglog. Multiple binomial regression model with link Cauchy. Simple binomial regression model with link logit for dose-response. Simple binomial regression model with link probit for dose-response. Simple binomial regression model with link cloglog for dose-response. Simple binomial regression model with link loglog for dose-response. Simple binomial regression model with link Cauchy for dose-response. Simple linear regression model with Poisson errors. Multiple regression model with Poisson errors.

continue

**Table 1:** Available functions in the **BioBayes** package with their respective descriptions.



Module	Function name	Description
Regression models	logistic_normal_model	Non-linear logistic growth model.
	gompertz_normal_model	Non-linear Gompertz growth model.
	brody_normal_model	Non-linear Brody growth model.
	vonbertalanffy_normal_model	Non-linear Von Bertalanffy growth model.
	wood_normal_model	Wood lactation curve model.

conclusion

	Dbar	Dhat	DIC	pD
y	117.6	115.6	119.6	1.996
total	117.6	115.6	119.6	1.996

**Table 2:** Values of DIC of a normal distribution.

### Estimation by BioBayes

To present the data entry, a simple simulation example will be used, like generate the data by a normal distribution and put them an ideal format for the analysis using **BioBayes**, using non-informative prior.

```
#generating data by a normal distribution
y <- rnorm(n = 50, mean = 0, sd = 1)

#putting the data in a list format for entry into the model
data_input <- list('n'= length(y), 'y'= y)

#the names of the hyperparameters and parameters can not be changed
#because they are already implemented in this way

#choosing the hyperparameters values for non-informative prior
priors_parm <- list('a' = 0, 'b' = 0.000001, 'c' = 0.001, 'd' = 0.001)

parms <- c("mu", "tau", "variance")

#initials values, based in frequentist inference
#and tau = 1/variance, according parametrization of OpenBUGS
start_kick <- function() list(mu = mean(y), tau = 1)

fit <- BioBayes.fit(bug_model = normal_model,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 10000,
  burn = 1000,
  jump = 1,
  Round = 3,
  seed = 123,
  title = 'Bayesian Inferential Results',
  save_html = NULL,
  render = TRUE)
```

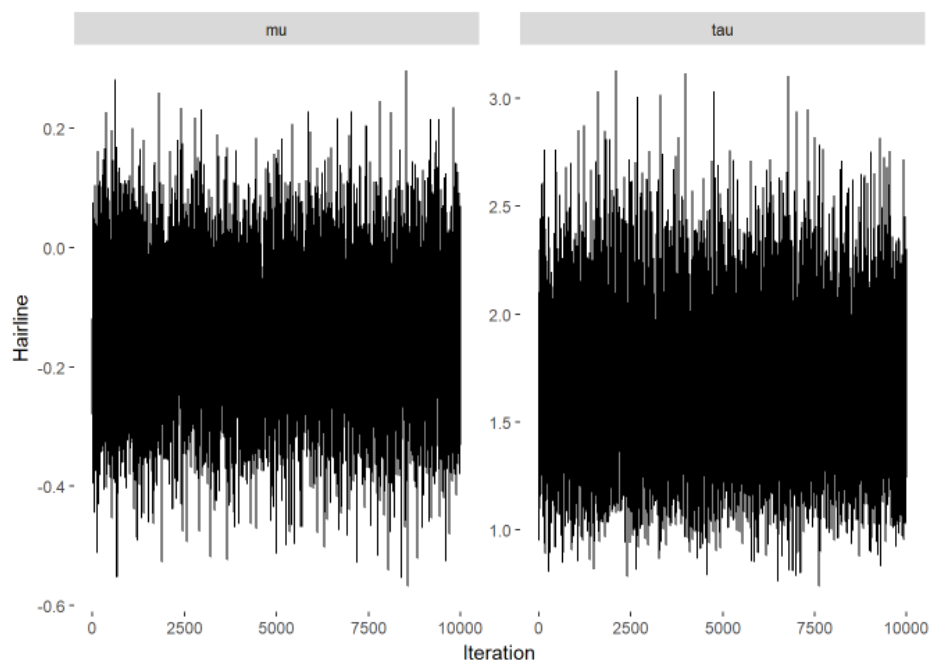
The final report is generated with the template shown in Figures 2, to 6 and Table 2.

As can be seen, the output contains graphics about the chains, autocorrelation, and densities. Also, a table with estimated values of parameters (mean, median, standart deviation, and intervals estimates like credibility and HPD (Highest Posterior Density) interval for 95%). In addition, is presented the DIC (Deviance Information Criterion) values ([Spiegelhalter et al., 2002](#)), convergence analysis ([Heidelberger and Welch, 1983](#)) and the chosen model implemented in *bug* language.

# Simulation Results

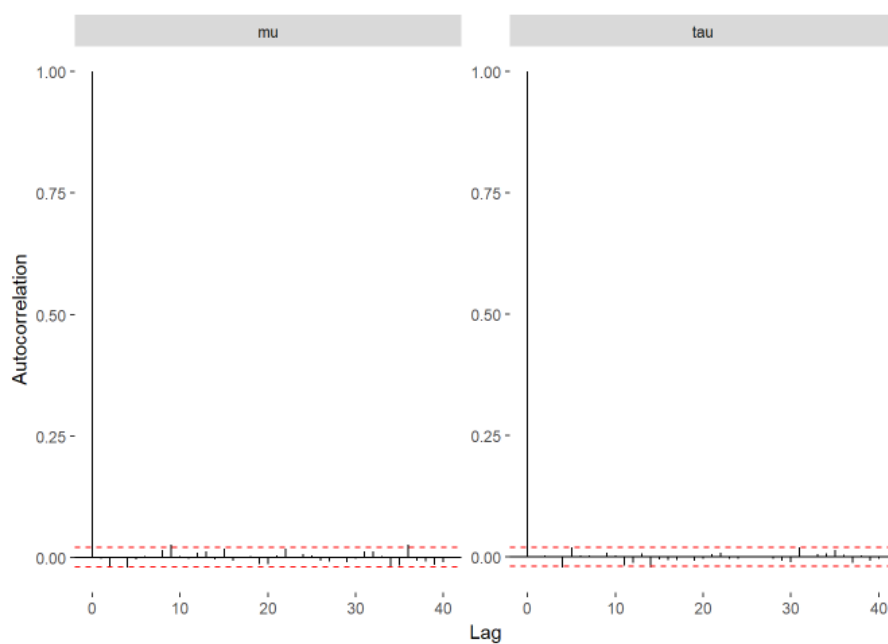
## Bayesian Inferential Results

Chain graphics



**Figure 2:** Graph of the chains generated via MCMC simulation.

Autocorrelation



**Figure 3:** Graph of autocorrelation for each parameter.

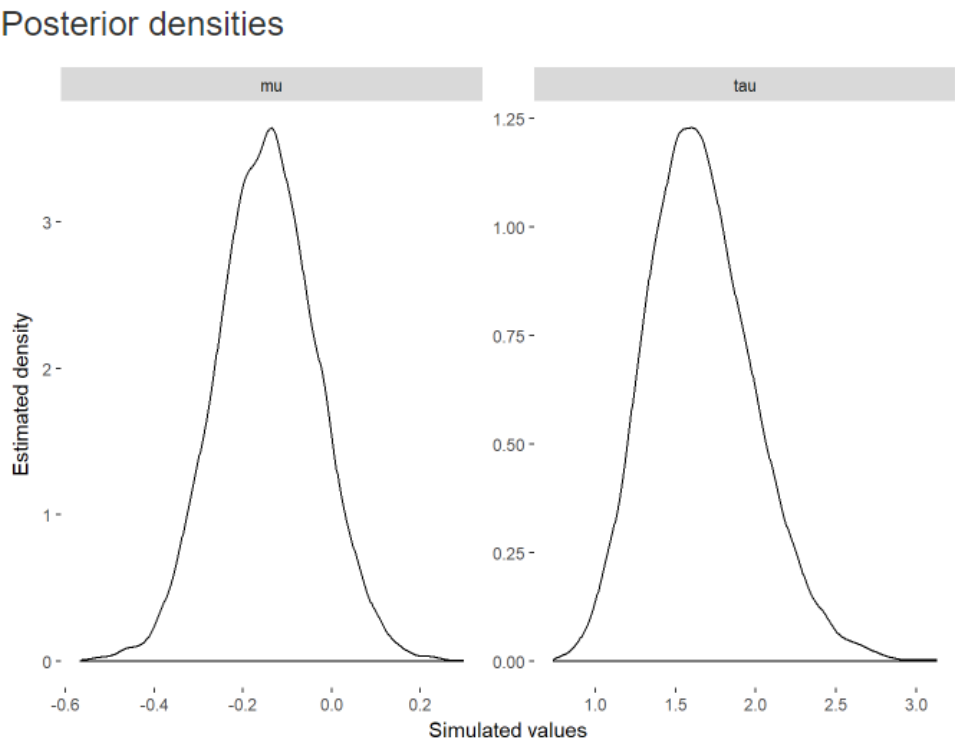


Figure 4: Posterior densities.

Summary of posterior distributions

	Mean	Std. Error	Median	ICr 95%	HPD 95%
mu	-0.146	0.112	-0.146	(-0.363; 0.074)	(-0.36; 0.075)
tau	1.658	0.331	1.633	(1.077; 2.378)	(1.02; 2.293)
variance	0.628	0.129	0.612	(0.421; 0.929)	(0.407; 0.899)

Heidelberg & Welch convergence test

##	Stationarity	start	p-value
##	test	iteration	
## mu	passed	1	0.322
## tau	passed	1	0.933
## variance	passed	1	0.504
##	Halfwidth	Mean	Halfwidth
##	test		
## mu	passed	-0.146	0.00209
## tau	passed	1.658	0.00650
## variance	passed	0.628	0.00254

Figure 5: Posterior estimates about the parameters and convergence tests according Heidelberg & Welch criterion.

## Simulation specifications

```
## Priors:
## a = 0
## b = 1e-06
## c = 0.001
## d = 0.001
##
## BUG Model:
## function (n, tau)
## {
##   for (i in 1:n) {
##     y[i] ~ dnorm(mu, tau)
##   }
##   mu ~ dnorm(a, b)
##   tau ~ dgamma(c, d)
##   variance <- 1/tau
## }
##
## specifications used:
##
## Interactions: 10000
## Burn: 1000
## Jump: 1
```

Figure 6: Normal model in *bug* language.

## Application : Logistic regression for credit analysis

In the moment that a client request a credit from a company, the toll known as credit analysis is used to make a diagnosis of the client history through documents (public or not), that goes far beyond checking the defaults in credit protection agencies but also, make a whole observation and consideration of the client's profile.

Keeping payments on track, having a stable income, age and marital status are some of the many points that deserve attention at the time of analysis.

This concern was established from the Basel treaty, which are sets of banking agreements signed between several central banks around the world to prevent credit risk, agreed upon after the Great Depression of the 1930's, creating a minimum capital reserve requirements, as well as measure to ensure their stability (di Basilea, 2004). Of which:

- Measures capable to increase financial solvency;
- Extra prudence in relation to the liquidity of the banking system;
- Increase customer confidence to ensure deposits;
- Forms of refinancing, that, in extreme cases, financial institutions would lose the customer confidence but would continue to operate.

Currently, the basic idea still applies in relation to ensuring the security of the system, for example, the caution with the concession of credit. There are several ways of achieving the same goal, of measuring the probability of a particular customer becomes defaulter or a bad payer.

This process can be performed in two ways: through judgment (subjective form), verifying the cadastral account of the client, in which the credit agent captures positive and negative aspects such as customer character, management capacity, among others, or through quantitative valuation models capable of numerically express the credit risk.

A widely used tool capable of measuring credit risk is logistic regression, which emerged in 1845, initially linked to demographic growth. In the 1930's, its use went through problems related to biology and only later extended to the economic and social issues, currently becoming one of the greatest tools of statistical modeling, due to the ease of interpretation of the parameters and the possibility of being used in discriminant analysis (Paula, 2004). A very positive point attached to the choice of logistics is the fact that it does not require the assumption of multivariate normality and equality between the variance-covariance matrices (Hair et al., 2009).

## The logistic model

Taking  $k$  as the number of explanatory variables belonging to the model, we have, according to Hosmer Jr et al. (2013)

$$p_i = \frac{e^{X_i' \beta_k}}{1 + e^{X_i' \beta_k}} \quad (1)$$

where  $X_i = (1, X_{i1}, X_{i2}, \dots, X_{ik})'$  is the vector of predictor variables for the  $i$ th observation and  $\beta = (\beta_0, \beta_1, \dots, \beta_k)'$  is the vector of coefficients of regression.

The logistic model, according to McCullagh (1980), for  $k$  explanatory variables with the logit function will be written as:

$$\ln \left( \frac{p_i}{1 - p_i} \right) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_k X_{ik} \quad (2)$$

## Parameters interpretation

The interpretation for the logistic model is given when comparing two probabilities ( $0 < p < 1$ ), the success ( $P(Y = 1)$ ) and failure ( $P(Y = 0)$ ) probabilities, via the method called odds or chance, given by:

$$\frac{p_i}{1 - p_i} = e^{X_i' \beta_k} \quad (3)$$

However, when we want to compare two odds, we use the equation called Odds Ratio (OR).

Therefore, the odds ratio for distinct values of a continuous independent covariant can be written as

$$OR_k = \frac{e^{X_{i(j+1)} \beta_k}}{e^{X_{i(j)} \beta_k}} = e^{\beta_k} \quad (4)$$

for any  $k$  variable.

Consequently, if the slope coefficient is positive, the interpretation of the odds ratio for the presence of a explanatory variable, in cases where  $x$  is a continuous covariant, is that the increase of one unit in  $x$ , increases the probability of default by a multiplicative factor equal to the correspondent  $\beta$ . However, if  $x$  is a dichotomous covariant, we interpret that, in the presence of  $x$ , the chance of default is  $\beta$  times the chance of someone in the absence of  $x$ .

## Credit Analysis: A empirical example

### The dataset

The dataset used in this analysis refers to customers who obtained credit from the brazilian company Cooper Card. The objective is to identify the variables that contribute to the default so that a new credit analysis model will be implemented. For this article, a small percentage of the total database was ceded, resulting in approximately 10 thousand observations, containing dichotomous responses (0 or 1) to indicate if a particular customer came to become defaulter ( $Y = 1$ ) or not ( $Y = 0$ ). The other variables are gender, age, number of dependents, occupation, scholarity, type of habitation and time of admission in the current employment, as showed, in detail, in Table 3.

### The bayesian analysis using BioBayes

To prepare the data that will enter the model, is necessary put that the response variable be in a numeric format and that the others explanatory variables be in a matrix, however, it is necessary to manually assign the levels of qualitative covariants, designating each level of  $\beta$  coefficient to be estimated. To do this in an easy way, one can use the `model.matrix` command of the frequentist regression obtained by the `glm` function.

```
y <- datadefaulting
fit_glm <- glm(y ~ ., family = binomial(link = "logit"), data = data)
M <- model.matrix(fit_glm)
M <- M[1:length(y), 2:ncol(M)]
```

Variable	Type of variable	Description
Default (response Y)	Dichotomous	0 - non-defaulter 1 - defaulter
Gender	Dichotomous	0 - female 1 - male
Age	Quantitative	$\in \mathbb{N}$
Number of dependents	Integer	$\in \mathbb{N}$
Profession	Qualitative	0 - retiree 1 - homemaker 2 - registered employee 3 - businessman 4 - government employee 5 - private employee
Scholarity	Qualitative	0 - postgraduate 1 - university graduate 2 - high school 3 - basic education 4 - did not attend school
Type of habitation	Qualitative	0 - own home (full payment) 1 - own home (incomplete payment) 2 - living with parents 3 - rented house 4 - living with familiar or friends
Time of admission in the current employment (in months)	Integer	$\in \mathbb{N}$

**Table 3:** Description of variables contained in the dataset.

or to manually indicate the presence and absence of each level, so that

```
x <- matrix(c(gender, age, n_dependents, profession1, profession2, profession3,
              profession4, profession5, scholarity1, scholarity2, scholarity3,
              habitation1, habitation2, habitation3, habitation4, t_admission ),
            byrow = F, ncol = 16)
data2 <- cbind(y, x)
k <- ncol(data2)-1
n <- rep(1, length(y)) #indicating that for each observation, there is a dichotomous possibility
```

Following the example of help added to the package, we have:

```
data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)
```

In this case, non-informative priors were chosen, as mean  $\mu = 0$  and precision parameter  $\tau = 10^{-6}$  for the  $\beta$ 's coefficients:

```
priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)
```

As previously mentioned, the parameter names can not be changed since they have already been implemented in this way, so they will be kept as:

```
parms <- c('beta0', 'beta', 'OR0', 'OR')
```

For initial kicks, the  $\beta$ 's from the frequentist estimation will be used:

```
fit_freq <- glm(y ~ x, family = binomial(link = "logit"))
summary(fit_freq)

beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
```

```

}
beta.aux = beta.aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta.aux)

```

Finally, the arguments of `BioBayes.fit` function can be determined. For this analysis, 10.000 iterations were chosen, with `burn = 1.000` and resampling period `jump = 10`.

```

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_logit,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 10000,
  burn = 1000,
  jump = 10,
  Round = 3,
  seed = 123,
  title = 'Bayesian Inferential Results',
  save_html = TRUE,
  render = TRUE)

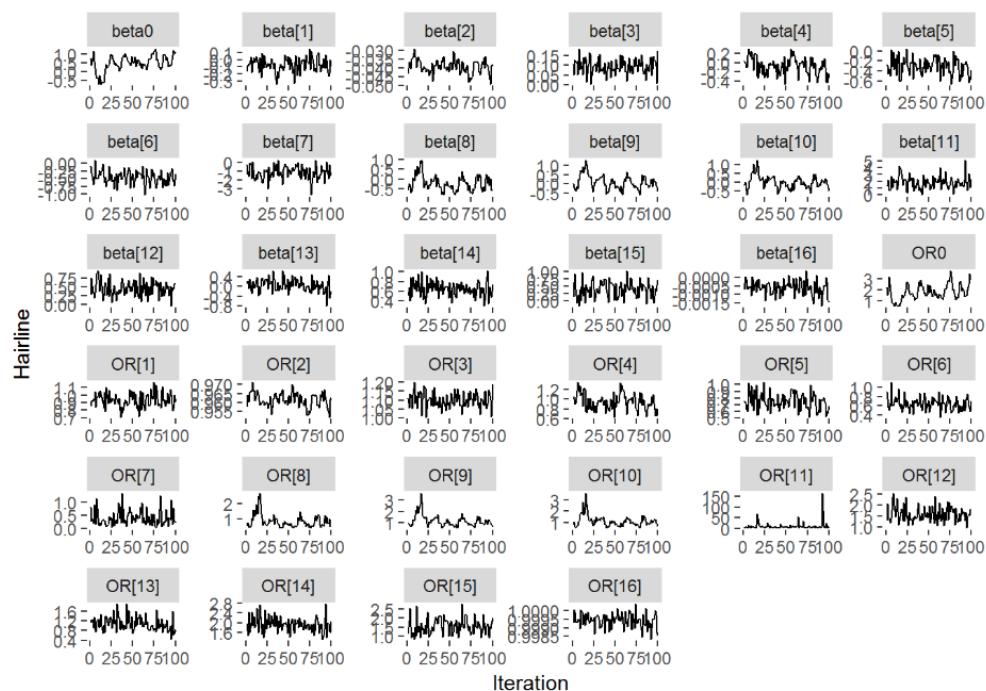
```

After that, the final report was generated, containing the graphical information, parameter estimates, convergence tests, among others, as can be seen by the Figure 7 to 13 and Table 4.

## Simulation Results

### Bayesian Inferential Results

#### Chain graphics



**Figure 7:** Graph of the chains generated via MCMC simulation.

The Figure 7 showed, for each parameter, the MCMC chains, which the user can see graphically the stabilization to a value of parameter. For  $\beta$ 's, although oscillations, it is possible to see a convergence around a value.

Observing the autocorrelation graph of Figure 8, all lags are found to be within acceptable limits, indicating that the jump size chosen was sufficient to eliminate the dependence between the samples.

## Autocorrelation

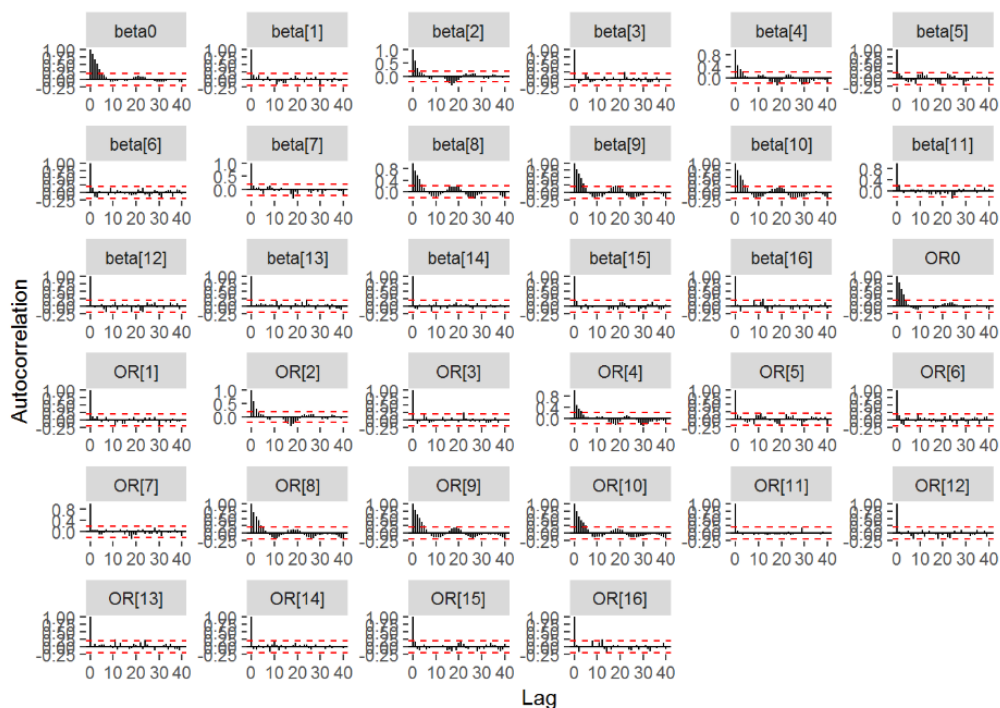


Figure 8: Graph of autocorrelation for each parameter.

## Posterior densities

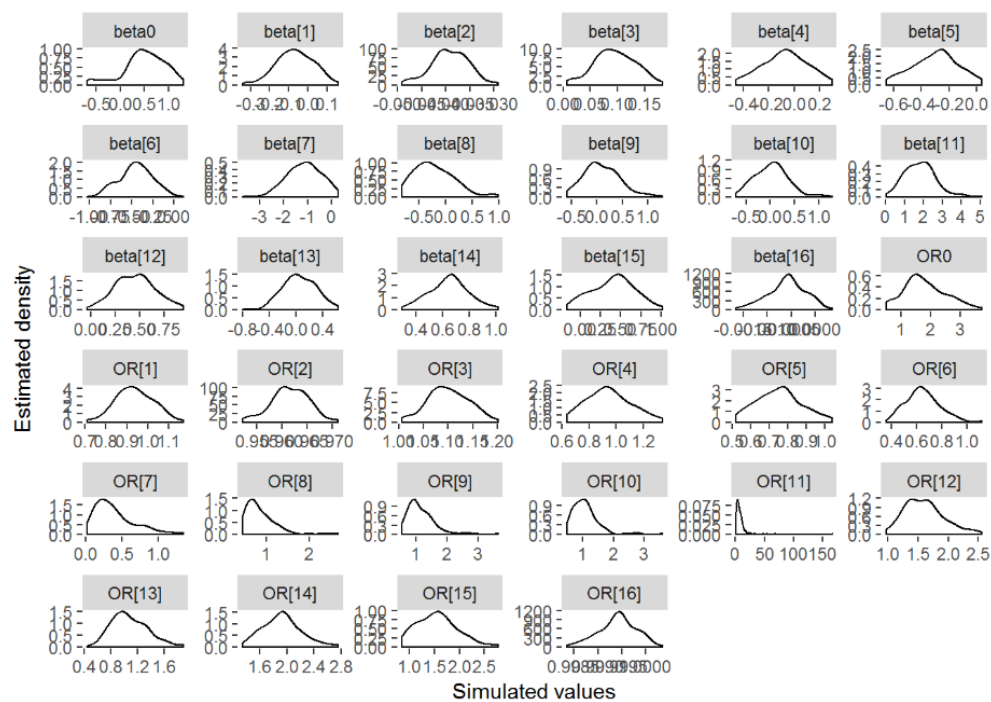


Figure 9: Posterior densities.



	Dbar	Dhat	DIC	pD
y	2,541	2,525	2,558	16.66
total	2,541	2,525	2,558	16.66

**Table 4:** Values of DIC.

The Figure 9 shown the posterior densities for each interest parameter, allow the user get a sense about the symmetry of the distribution.

### Summary of posterior distributions

	Mean	Std. Error	Median	ICr 95%	HPD 95%
beta0	0.496	0.444	0.518	(-0.636; 1.177)	(-0.642; 1.19)
beta[1]	-0.069	0.097	-0.067	(-0.257; 0.099)	(-0.265; 0.102)
beta[2]	-0.039	0.004	-0.039	(-0.048; -0.032)	(-0.048; -0.033)
beta[3]	0.095	0.037	0.095	(0.018; 0.162)	(0.01; 0.158)
beta[4]	-0.073	0.176	-0.075	(-0.422; 0.247)	(-0.426; 0.25)
beta[5]	-0.298	0.162	-0.274	(-0.605; -0.002)	(-0.608; -0.002)
beta[6]	-0.453	0.208	-0.449	(-0.828; -0.089)	(-0.84; -0.085)
beta[7]	-1.161	0.75	-1.101	(-2.493; 0.103)	(-2.522; 0.129)
beta[8]	-0.212	0.387	-0.28	(-0.779; 0.757)	(-0.825; 0.463)
beta[9]	0.099	0.366	0.042	(-0.53; 0.945)	(-0.588; 0.799)
beta[10]	0.034	0.367	0.047	(-0.604; 0.924)	(-0.705; 0.603)
beta[11]	1.837	0.876	1.863	(0.366; 3.904)	(0.015; 3.468)
beta[12]	0.452	0.203	0.469	(0.083; 0.873)	(0.078; 0.873)
beta[13]	0.049	0.255	0.035	(-0.376; 0.498)	(-0.389; 0.499)
beta[14]	0.647	0.149	0.658	(0.354; 0.944)	(0.343; 0.933)
beta[15]	0.407	0.264	0.429	(-0.067; 0.892)	(-0.087; 0.852)
beta[16]	-0.001	0	-0.001	(-0.001; 0)	(-0.001; 0)

**Figure 10:** Posterior estimates about the parameters.

Analyzing the estimates of the parameters, we have that  $\beta_2$ ,  $\beta_3$ ,  $\beta_5$ ,  $\beta_6$ ,  $\beta_{11}$ ,  $\beta_{12}$  and  $\beta_{14}$  were significant, since zero is not within the credibility and HPD interval. These variables represent age, number of dependents, registered employed, businessman, basic education, did not attend school and living with parents, respectively.

By median of Odds Ratio results showed in Figure 11, it is possible to infer that

- $\beta_2$ : with increase of 1 unit in age, the chance of the customer becoming defaulter decreases by 4%;
- $\beta_3$ : with increase of 1 dependent, the chance of the customer becoming defaulter increases in 10%;
- $\beta_5$ : the chance of a retiree becoming defaulter is 1.31 times greater than a registered employed;
- $\beta_6$ : the chance of a retiree becoming defaulter is 1.57 times greater than a businessman;
- $\beta_{11}$ : the chance of a customer with basic education becoming defaulter is 6.4 times greater than a individual with postgraduate;
- $\beta_{12}$ : the chance of a customer that did no attend school becoming defaulter is 1.6 times greater than a individual with post graduate;
- $\beta_{14}$ : the chance of a customer that lives with parents becoming defaulter is 1.93 times greater than a individual with own home (total payment).

By the Heidelberg & Welch tests (Heidelberger and Welch, 1983), all parameters passed the chain convergence test, that is, they reached stabilization, as can be seen in Figure 12.

In Figure 13, we see the model in *bug* language that was previously implemented in the package and works internally, without requiring the user to use this type of language.

The Table 4 shows the DIC values, which in this case there is no interpretation since no model comparison was performed.

## Summary of posterior distributions

	Mean	Std. Error	Median	ICr 95%	HPD 95%
OR0	1.793	0.709	1.679	(0.529; 3.247)	(0.501; 3.163)
OR[1]	0.937	0.09	0.935	(0.774; 1.104)	(0.767; 1.108)
OR[2]	0.962	0.004	0.961	(0.953; 0.968)	(0.953; 0.967)
OR[3]	1.1	0.041	1.1	(1.018; 1.175)	(1.01; 1.171)
OR[4]	0.943	0.165	0.928	(0.656; 1.28)	(0.634; 1.251)
OR[5]	0.752	0.121	0.76	(0.546; 0.998)	(0.545; 0.998)
OR[6]	0.649	0.135	0.638	(0.437; 0.915)	(0.432; 0.919)
OR[7]	0.402	0.283	0.333	(0.083; 1.109)	(0.024; 0.982)
OR[8]	0.878	0.41	0.756	(0.459; 2.138)	(0.438; 1.588)
OR[9]	1.186	0.504	1.043	(0.589; 2.572)	(0.555; 2.223)
OR[10]	1.112	0.482	1.048	(0.547; 2.524)	(0.494; 1.828)
OR[11]	10.338	18.602	6.444	(1.452; 49.685)	(1.015; 32.08)
OR[12]	1.604	0.332	1.599	(1.086; 2.394)	(1.082; 2.395)
OR[13]	1.084	0.275	1.035	(0.686; 1.645)	(0.678; 1.648)
OR[14]	1.93	0.289	1.932	(1.425; 2.57)	(1.345; 2.451)
OR[15]	1.554	0.41	1.536	(0.935; 2.441)	(0.849; 2.256)
OR[16]	0.999	0	0.999	(0.999; 1)	(0.999; 1)

Figure 11: Posterior estimates for Odds Ratio.

## Heidelberg &amp; Welch convergence test

```
##
##      Stationarity start  p-value
##      test      iteration
## OR[1] passed      1      0.3551
## OR[2] passed      1      0.6832
## OR[3] passed      1      0.8013
## OR[4] passed      1      0.1661
## OR[5] passed      1      0.1703
## OR[6] passed      1      0.2196
## OR[7] passed      1      0.9648
## OR[8] passed      1      0.6320
## OR[9] passed      1      0.8797
## OR[10] passed     1      0.4725
## OR[11] passed     1      0.5824
## OR[12] passed     1      0.3823
## OR[13] passed     1      0.6961
## OR[14] passed     1      0.8160
## OR[15] passed     1      0.4205
## OR[16] passed     11     0.1599
## OR0 passed      1      0.0862
## beta[1] passed   1      0.2865
## beta[2] passed   1      0.6786
## beta[3] passed   1      0.8061
## beta[4] passed   1      0.0849
## beta[5] passed   1      0.0564
## beta[6] passed   1      0.0705
## beta[7] passed   1      0.9855
## beta[8] passed   1      0.5240
## beta[9] passed   1      0.9623
## beta[10] passed  1      0.9660
## beta[11] passed  1      0.7239
## beta[12] passed  1      0.6995
## beta[13] passed  1      0.9751
## beta[14] passed  1      0.8125
## beta[15] passed  1      0.2700
## beta[16] passed  11     0.1598
```

Figure 12: Convergence test according Heidelberg &amp; Welch criterion.

## Simulation specifications

```
## Priors:
## mu_beta = 0
## tau_beta = 1e-06
##
## BUG Model:
## function (N, beta0, x, k)
## {
##   for (i in 1:N) {
##     y[i] ~ dbin(p[i], n[i])
##     logit(p[i]) <- beta0 + inprod(x[i, ], beta[])
##   }
##   beta0 ~ dnorm(mu_beta, tau_beta)
##   for (j in 1:k) {
##     beta[j] ~ dnorm(mu_beta, tau_beta)
##   }
##   OR0 <- exp(beta0)
##   for (j in 1:k) {
##     OR[j] <- exp(beta[j])
##   }
## }
##
## specifications used:
##
## Interactions: 1000
## Burn: 100
## Jump: 10
```

**Figure 13:** Utilized model in *bug* language.

Some another researches bring the topic about credit analysis or credit card analysis using logistic regression, as in [Laitinen \(1999\)](#), which predicts a corporate credit analyst's risk by logistic regression for dichotomous response and also by a linear regression, in which, both models give a classification accuracy of 90% in the sample estimate and 96% in the test sample. Another paper about credit card using logistic model, is seen in [Nie et al. \(2011\)](#), which approach a study of a prediction model for describe when a customer will stop doing business or end the relationship with a company. The paper of [Van Gestel et al. \(2005\)](#) which makes use of logistic regression jointly with additive models built by non-linear transformations, and also, a technique based in kernel known like a "support vectors machines" (SVM), is introduced to construct a advanced non-linear model, capable to captures the multivariate non-linear relationships in the data.

## Discussion and Conclusions

Bayesian inference has come to be another alternative to the researcher when faced with the diversity of data existing in the phenomena. This method, widely used in several areas, still does not have as many simple analysis tools, making it difficult to include researchers from other areas that need use this method.

The **BioBayes** package was implemented with the intention of facilitating the handling of Bayesian models for statisticians and other researchers, providing a simple implementation beyond all theoretical description regarding mathematical models, examples and generation of graphs automatically to assist the user.

Currently, only the most basic models used have been approached in the construction of the package, but the intention is to make it more complete in the future.

## Acknowledgements

The authors gratefully acknowledge financial support from State of Parana Research Foundation and CAPES.

## Bibliography

- C. di Basilea. Convergenza internazionale della misurazione del capitale e dei coefficienti patrimoniali. *Nuovo schema di regolamentazione*, Giugno, 2004. [p8]
- S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, 6(6):721–741, 1984. [p1]
- J. Hadfield, M. J. Hadfield, and C. SystemRequirements. Package ‘mcmcglmm’, 2017. [p1]
- J. F. Hair, W. C. Black, B. J. Babin, R. E. Anderson, and R. L. Tatham. *Análise multivariada de dados*. Bookman Editora, 2009. [p8]
- W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. 1970. [p1]
- P. Heidelberger and P. D. Welch. Simulation run length control in the presence of an initial transient. *Operations Research*, 31(6):1109–1144, 1983. [p5, 13]
- D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013. [p9]
- E. K. Laitinen. Predicting a corporate credit analyst’s risk estimate by logistic and linear models. *International review of financial analysis*, 8(2):97–121, 1999. [p15]
- D. Lunn, D. Spiegelhalter, A. Thomas, and N. Best. The bugs project: Evolution, critique and future directions. *Statistics in medicine*, 28(25):3049–3067, 2009. [p1]
- S. Martino and H. Rue. Implementing approximate bayesian inference using integrated nested laplace approximation: A manual for the inla program. *Department of Mathematical Sciences, NTNU, Norway*, 2009. [p1]
- P. McCullagh. Regression models for ordinal data. *Journal of the royal statistical society. Series B (Methodological)*, pages 109–142, 1980. [p9]
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. [p1]
- R. M. Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011. [p1]
- G. Nie, W. Rowe, L. Zhang, Y. Tian, and Y. Shi. Credit card churn forecasting by logistic regression and decision tree. *Expert Systems with Applications*, 38(12):15273–15285, 2011. [p15]
- D. A. Nogueira, T. Sáfiadi, E. Bearzoti, and J. Bueno-Filho. Análises clássica e bayesiana de um modelo misto aplicado ao melhoramento animal: uma ilustração. *Ciência e Agrotecnologia, Lavras*, 27: 1614–1624, 2003. [p1]
- G. A. Paula. *Modelos de regressão: com apoio computacional*. IME-USP São Paulo, 2004. [p8]
- C. D. M. Paulino, M. A. A. Turkman, and B. Murteira. *Estatística bayesiana*. Fundação Calouste Gulbenkian, Lisboa, 2003. [p1]
- R. M. Rossi. Introdução aos métodos bayesianos na análise de dados zootécnicos com uso do winbugs e r. *Eduem: Maringá*, 2011. [p1]
- F. Silva. *Análise bayesiana do modelo auto-regressivo para dados em painel: aplicação na avaliação genética de touros da raça Nelore*. PhD thesis, Federal University of Lavras, Lavras, MG (Brazil), 2006. [p1]
- D. J. Spiegelhalter, N. G. Best, B. P. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the royal statistical society: Series b (statistical methodology)*, 64(4):583–639, 2002. [p5]
- A. Thomas, D. Lunn, D. Spiegelhalter, and B. Nicky. Package ‘brugs’, 2017. [p1]
- T. Van Gestel, B. Baesens, P. Van Dijke, J. Suykens, J. Garcia, and T. Alderweireld. Linear and nonlinear credit scoring by combining logistic regression and support vector machines. *Journal of credit Risk*, 1 (4), 2005. [p15]

*Stéfane Lele*  
*Department of Statistics*  
*State University of Maringá*  
*Brazil*  
[ste.tistics@gmail.com](mailto:ste.tistics@gmail.com)

*Robson Marcelo Rossi*  
*Department of Statistics*  
*State University of Maringá*  
*Brazil*  
[rmrossi@uem.br](mailto:rmrossi@uem.br)

---

## Referências

---

BAYES, T.; PRICE, R.; CANTON, J. **An essay towards solving a problem in the doctrine of chances**. 1763. C. Davis, Printer to the Royal Society of London. London, U.K, 1763.

BERNARDO, J. M. A Bayesian analysis of classical hypothesis testing. **Trabajos de estadística y de investigación operativa**, v.31, n.1, p.605-647, 1980.

BLASCO, A. The Bayesian controversy in animal breeding. **Journal of animal science**, v.79, n.8, p.2023-2046, 2001.

COUTINHO, L. A terceira revolução industrial e tecnológica. As grandes tendências das mudanças. **Economia e sociedade**, v.1, n.1, p.69-87, 1992.

DE MOIVRE, A. **The doctrine of chances: or, A method of calculating the probabilities of events in play**. Chelsea Publishing Company, 1756.

EHLERS, R. S. **Introdução à Inferência Bayesiana**. 2005. Disponível em: <[http://leg.ufpr.br/~edson/Artigos/Riccardo\\_Heller-bayes.pdf](http://leg.ufpr.br/~edson/Artigos/Riccardo_Heller-bayes.pdf)>.

FARIA, C. U. D., MAGNABOSCO, C. D. U., REYES, A. D. L., LÔBO, R. B.; BEZERRA, L. A. F. **Inferência bayesiana e sua aplicação na avaliação genética de bovinos da raça nelore: revisão bibliográfica**. 2007.

GAMERMAN, D. Markov chain Monte Carlo: stochastic simulation for Bayesian inference. **Taylor & Francis**, v.1, 1997.

GEMAN, S.; GEMAN, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. **IEEE Transactions on pattern analysis and machine intelligence**, n.6, p.721-741, 1984.

GEWEKE, J. **Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments**. Minneapolis, MN, USA: Federal Reserve Bank of Minneapolis, Research Department, 1991.

HASTINGS, W. K. **Monte Carlo sampling methods using Markov chains and their applications**. 1970.

HEIDELBERGER, P; WELCH, P. D. Simulation run length control in the presence of an initial transient. **Operations Research**, v.31, n.6, p.1109-1144, 1983.

HOFFMANN-JORGENSEN, J. **Probability with a view toward statistics**. Chapman & Hall, New York, v.2, 1994.

JEFFREYS, S. H. **Theory of Probability: 3d Ed**. Clarendon Press, 1967.

LINDLEY, D. V. Approximate bayesian methods. **Trabajos de estadística y de investigación operativa**, v.31, n.1, p.223-245, 1980.

LUNN, D.; SPIEGELHALTER, D.; THOMAS, A.; BEST, N.. The BUGS project: Evolution, critique and future directions. **Statistics in medicine**, v.28, n.25, p.3049-3067, 2009.

LYKOU, A.; NTZOUFRAS, I. WinBUGS: a tutorial. **Wiley Interdisciplinary Reviews: Computational Statistics**, v.3, n.5, p.385-396, 2011.

MACEDO, L. R. de. **Abordagens frequentista e bayesiana para descrição das curvas de acúmulo de matéria seca de plantas de alho**. 2015. Tese (Doutorado em Estatística Aplicada e Biometria) - Universidade Federal de Viçosa, Viçosa, 2015.

METROPOLIS, N.; ROSENBLUTH, A. W.; ROSENBLUTH, M. N.; TELLER, A. H.; Teller, E. Equation of state calculations by fast computing machines. **The journal of chemical physics**, v.21, n.6, p.1087-1092, 1953.

MILONE, G. **Estatística: geral e aplicada**. Pioneira Thomson Learning, 2004.

NEAL, R. M. MCMC using Hamiltonian dynamics. **Handbook of Markov Chain Monte Carlo**, v.2, n.11, p.2, 2011.

NOGUEIRA, D. A.; SÁFADI, T.; BEARZOTI, E.; BUENO-FILHO, J. S. S. Análises clás-

sica e Bayesiana de um modelo misto aplicado ao melhoramento animal: uma ilustração. **Ciência e Agrotecnologia**, Lavras, v.27, p.1614-1624, 2003.

NTZOUFRAS, I. **Bayesian modeling using WinBUGS**. John Wiley & Sons, 2011.

PAGANO, M.; GAUVREAU, K. **Princípios de Bioestatística**. Cengage learning, 2004.

PAULINO, C. D. M.; TURKMAN, M. A. A.; MURTEIRA, B. **Estatística bayesiana**. Fundação Calouste Gulbenkian, Lisboa, 2003.

RAFTERY, A. E.; LEWIS, S. **How many iterations in the Gibbs sampler?**. Washington University Seattle, Department of Statistics, 1991.

ROSSI, R. M. **Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R**. Eduem, Maringá, 2011.

RUBIN, D. B. The bayesian bootstrap. **The annals of statistics**, JSTOR, p.130-134, 1981.

RUE, H.; MARTINO, S.; CHOPIN, N. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. **Journal of the royal statistical society: Series b (statistical methodology)**, v.71, n.2, p.319-392, 2009.

SILVA, F. F. **Análise bayesiana do modelo auto-regressivo para dados em painel: aplicação na avaliação genética de touros da raça Nelore**. 2006. Tese (Doutorado) - Universidade Federal de Lavras, Lavras, 2006.

SMITH, A. F.; GELFAND, A. E. Bayesian statistics without tears: a sampling–resampling perspective. **The American Statistician**, v.46, n.2, p.84-88, 1992.

SPIEGELHALTER, D.; BEST, N. G.; CARLIN, B. P.; VAN DER LINDE, A. Bayesian measures of model complexity and fit. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, v.64, n.4, p.583-639, 2002.

SPIEGELHALTER, D.; THOMAS, A.; BEST, N.; GILKS, W. BUGS 0.5: Bayesian inference using Gibbs sampling manual (version ii). **MRC Biostatistics Unit**, Institute of Public Health, Cambridge, UK, p.1-59, 1996.

WEST, M. Outlier models and prior distributions in Bayesian linear regression. **Journal of the Royal Statistical Society. Series B (Methodological)**, JSTOR, p.431-439, 1984.



YANG, F.; ZOU, J.; GE, L.; TIAN, J.; LEE, M. S.; JUN, J. H.; ZHANG, J. Chinese herbal injections for heart failure: A protocol for systematic review and network meta-analyses. **Medicine**, Wolters Kluwer Health, v.97, n.8, 2018.

# Anexos

---

ANEXO A

---

Manual do pacote BioBayes

---

# Package ‘BioBayes’

January 15, 2019

**Type** Package

**Title** Bayesian Models

**Version** 0.1.0

**Author** Robson Marcelo Rossi, Stefane Lele

**Maintainer** Stefane Lele <ste.tistics@gmail.com>

**Description** The idea of constructing this package is to facilitate the Bayesian analysis using RStudio jointly with the OpenBUGS program (installation is required on the computer), in which some basic models in the bug format are already implemented in the BioBayes package. The collection of available distributions functions is listed in help of principal function BioBayes.fit. After the analysis, a report is automatically generated containing graphs about the chains, estimates, posterior densities for the parameters, convergence analysis, DIC, and the model used in the bug format.

**License** GPL (>= 2)

**Encoding** UTF-8

**RoxygenNote** 6.0.1

**Depends** coda, BRugs, ggplot2, rmarkdown, fitdistrplus

**Imports** data.table, RColorBrewer, grDevices, stats, utils

## R topics documented:

anova_hetero_model . . . . .	2
anova_homo_model . . . . .	3
beta_diff_model . . . . .	5
beta_model . . . . .	7
binomial_diff_model . . . . .	8
binomial_model . . . . .	10
bin_multi_reg_model_cauchy . . . . .	11
bin_multi_reg_model_cloglog . . . . .	13
bin_multi_reg_model_logit . . . . .	16
bin_multi_reg_model_loglog . . . . .	19
bin_multi_reg_model_probit . . . . .	21
bin_reg_model_cauchy_DR . . . . .	24
bin_reg_model_cloglog_DR . . . . .	27
bin_reg_model_logit . . . . .	29
bin_reg_model_logit_DR . . . . .	31
bin_reg_model_loglog_DR . . . . .	34

bin_reg_model_probit_DR . . . . .	36
BioBayes.fit . . . . .	38
brody_normal_model . . . . .	40
exponential_diff_model . . . . .	42
exponential_model . . . . .	44
gamma_diff_model . . . . .	45
gamma_model . . . . .	47
gompertz_normal_model . . . . .	48
linear_multi_reg_model . . . . .	50
linear_multi_reg_model_poisson . . . . .	52
linear_reg_model . . . . .	54
linear_reg_model_poisson . . . . .	56
logistic_normal_model . . . . .	57
normal_hetero_diff_model . . . . .	59
normal_homo_diff_model . . . . .	61
normal_model . . . . .	63
normal_paired_model . . . . .	64
poisson_diff_model . . . . .	66
poisson_model . . . . .	67
quadratic_control_reg_model . . . . .	68
quadratic_reg_model . . . . .	71
vonbertalanffy_normal_model . . . . .	73
wood_normal_model . . . . .	75

<b>Index</b>	<b>79</b>
--------------	-----------

---

anova_hetero_model	<i>Model for an analysis of variance for normal populations with heteroskedasticity.</i>
--------------------	--

---

## Description

Bayesian model for analysis of variance (ANOVA) considering normal populations with heterogeneous variances - Adjusts the difference of means for  $K$  treatments in which:

$$Y_k \sim \text{Normal}(\mu_k, \tau_k), \text{ with } Y_k \text{ and } \mu_k \text{ real and } \tau_k > 0, k = 1, 2, \dots, K$$

with priors distributions:

$$\mu_k \sim \text{Normal}(\mu, \tau\mu_k)$$

$$\tau_k \sim \text{Gamma}(a\tau_k, b\tau_k)$$

## Usage

```
anova_hetero_model(y, group)
```

## Arguments

y	vector of response variable of all groups.
group	vector of numbers or factors that designate each group.

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\mu_k$ ” and “ $\tau_k$ ” as specified in the above function.

## References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y <- c(rnorm(10, 0, 1), rnorm(15, 3, 1), rnorm(12, 5, 3))

group <- rep(c(1, 2, 3), c(10, 15, 12))

priors_parm <- list('mu_mu' = 0, 'tau_mu' = 0.000001,
                   'a_tau' = 0.001, 'b_tau' = 0.001)

get_fit_data <- anova_hetero_model(y, group)

fit <- BioBayes.fit(bug_model = get_fit_data$model,
                   database = get_fit_data$dados,
                   priors = priors_parm,
                   parameters = get_fit_data$parametros,
                   initials = get_fit_data$kicks,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian ANOVA Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)
```



```

initials = get_fit_data$kicks,
iter = 10000,
burn = 1000,
jump = 1,
Round = 3,
seed = 123,
title = 'Bayesian ANOVA Results',
save_html = NULL,
render = TRUE)

## End(Not run)

```

---

beta_diff_model	<i>Model for difference of means between two independent beta populations.</i>
-----------------	--

---

## Description

Bayesian model for adjusting the difference of means between two independent beta populations.

$Y_k \sim \text{Beta}(a_k, b_k)$ ,  $0 < Y_k < 1$ ,  $a_k > 0$ ,  $b_k > 0$ ,  $k = 1, 2$ ,

with priors distributions:

$a_1 \sim \text{Gamma}(c_1, d_1)$

$b_1 \sim \text{Gamma}(e_1, f_1)$

$a_2 \sim \text{Gamma}(c_2, d_2)$

$b_2 \sim \text{Gamma}(e_2, f_2)$

## Usage

```
beta_diff_model(n1, n2, a1, b1, a2, b2)
```

## Arguments

n1	length of vector y1
n2	length of vector y2
a1	inicial kick for shape1 of y1
b1	inicial kick for shape2 of y1
a2	inicial kick for shape1 of y2
b2	inicial kick for shape2 of y2

## Details

To change the parameters of the priors distributions, is just change the values in object “ $a_1, b_1, a_2, b_2$ ” as specified in the above function.



## References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y1 <- rbeta(n = 50, shape1 = 1, shape2 = 1)

y2 <- rbeta(n = 50, shape1 = 2, shape2 = 0.5)

data_input <- list('n1' = length(y1), 'y1' = y1,
                  'n2' = length(y2), 'y2' = y2)

priors_parm <- list('c1' = 0.001, 'd1' = 0.001, 'e1' = 0.001, 'f1' = 0.001,
                  'c2' = 0.001, 'd2' = 0.001, 'e2' = 0.001, 'f2' = 0.001)

parms <- c('a1', 'b1', 'a2', 'b2', 'mean1', 'variance1',
          'mean2', 'variance2', 'delta_mu_1x2')

fit1 <- fitdist(data = y1, dist = 'beta',
               start <- list('shape1' = 1, 'shape2' = 1), method = 'mme')

fit2 <- fitdist(data = y2, dist = 'beta',
               start <- list('shape1' = 1, 'shape2' = 2), method = 'mme')

start_kick <- function() list('a1' = fit1$estimate[1], 'b1' = fit1$estimate[2],
                             'a2' = fit2$estimate[1], 'b2' = fit2$estimate[2])

fit <- BioBayes.fit(bug_model = beta_diff_model,
                  database = data_input,
                  priors = priors_parm,
                  parameters = parms,
                  initials = start_kick,
                  iter = 100000,
                  burn = 10000,
                  jump = 100,
                  Round = 3,
                  seed = 123,
                  title = 'Bayesian Inferential Results',
                  save_html = NULL,
                  render = TRUE)

## End(Not run)
```

---

beta_model	<i>Model for beta distribution.</i>
------------	-------------------------------------

---

### Description

Bayesian model to fit  
 $Y \sim \text{Beta}(a, b), 0 < Y < 1,$   
 such that  $a > 0$  and  $b > 0$ ,  
 with priors distributions:  
 $a \sim \text{Gamma}(a1, b1)$  with  $a1 > 0$  and  $b1 > 0$ ;  
 $b \sim \text{Gamma}(a2, b2)$  with  $a2 > 0$  and  $b2 > 0$ .

### Usage

```
beta_model(n, a, b)
```

### Arguments

n	length of vector y
a	inicial quick for shape1 of y
b	inicial quick for shape2 of y

### Details

To change the parameters of the priors distributions, is just change the values in object “a” and “b” as specified in the above function.

### References

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.  
 McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.  
 NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.  
 PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.  
 ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

y <- rbeta(n = 30, shape1 = 1, shape2 = 5)

data_input <- list('n' = length(y), 'y' = y)

priors_parm <- list('a1' = 0.001, 'b1' = 0.001, 'a2' = 0.001, 'b2' = 0.001)

parms <- c('a', 'b', 'mean', 'variance')
```

```

fit_beta <- fitdist(data = y, dist = 'beta', start = list('shape1' = 2,
  'shape2' = 4), method = 'mme')

start_kick <- function() list('a' = fit_beta$estimate[1], 'b' = fit_beta$estimate[2])

fit <- BioBayes.fit(bug_model = beta_model,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 100000,
  burn = 10000,
  jump = 100,
  Round = 3,
  seed = 123,
  title = 'Bayesian Inferential Results',
  save_html = NULL,
  render = TRUE)

## End(Not run)

```

---

binomial_diff_model	<i>Model for difference of proportions between two independent binomial populations.</i>
---------------------	--

---

### Description

Bayesian model for adjusting the difference of proportion between two independent binomial populations.

$Y_k \sim \text{Binomial}(n_k, p_k)$  such that  $0 < p_k < 1$  and  $k = 1, 2$

with priors distributions:

$p_1 \sim \text{Beta}(a_1, b_1)$  with  $a_1 > 0$  and  $b_1 > 0$ ;

$p_2 \sim \text{Beta}(a_2, b_2)$  with  $a_2 > 0$  and  $b_2 > 0$ .

### Usage

```
binomial_diff_model(N1, N2, n1, n2, p1, p2)
```

### Arguments

N1	length of vector y1
N2	length of vector y2
n1	size of binomial random variable y1
n2	size of binomial random variable y2
p1	probability of binomial random variable y1
p2	probability of binomial random variable y2

## Details

To change the parameters of the priors distributions, is just change the values in object “*p1*” and “*p2*” as specified in the above function.

## References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y1 <- rbinom(20, 5, 0.2)

y2 <- rbinom(30, 7, 0.5)

data_input <- list('N1' = length(y1), 'y1' = y1, 'n1' = 5, 'N2' = length(y2), 'y2' = y2, 'n2' = 7)

priors_parm <- list('a1' = 1, 'b1' = 1, 'a2' = 1, 'b2' = 1)

parms <- c('p1', 'p2', 'mean1', 'mean2', 'variance1', 'variance2', 'delta_p_1x2')

start_kick <- function() list('p1' = mean(y1)/data_input$n1, 'p2' = mean(y2)/data_input$n2)

fit <- BioBayes.fit(bug_model = binomial_diff_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)
```

---

binomial_model	<i>Model for binomial distribution.</i>
----------------	---

---

### Description

Bayesian model to fit  
 $Y \sim \text{Binomial}(n, p)$ ,  
 such that  $0 < p < 1$ ,  
 with prior distribution:  
 $p \sim \text{Beta}(a, b)$  with  $a > 0$  and  $b > 0$ .

### Usage

```
binomial_model(N, n, p)
```

### Arguments

N	length of vector Y
n	size of binomial random variable Y
p	probability of binomial random variable Y

### Details

To change the parameters of the priors distributions, is just change the values in object “p” as specified in the above function.

### References

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.

GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

y <- rbinom(n = 100, size = 10, prob = 0.2)

data_input <- list('N' = length(y), 'y' = y, 'n' = 10)
```

```

priors_parm <- list('a' = 1, 'b' = 1)

parms <- c('p', 'mean', 'variance')

start_kick <- function() list('p' = mean(y)/data_input$n)

fit <- BioBayes.fit(bug_model = binomial_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

```

---

bin\_multi\_reg\_model\_cauchy

*Binomial multiple regression model with cauchy link.*

---

## Description

Bayesian binomial multiple regression model with cauchy link.

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with cauchy link:

$$p = 0.5 + (\arctan(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k)) / \pi$$

wherein  $X$  is the covariates matrix of the model,  $X = x_1, x_2, \dots, x_K$ , and

$\beta_k$  is the vector of beta coefficients,  $k = 1, \dots, K$ .

Priors distributions for the “ $\beta_0$ ” and “ $\beta_k$ ”:

$\beta_0, \beta_k \sim \text{Normal}(\mu\beta, \tau\beta)$ , with  $\mu\beta$  real and  $\tau\beta > 0$ .

Odds Ratio:  $OR = \exp(\beta)$ .

## Usage

```
bin_multi_reg_model_cauchy(N, beta0, x, k)
```

## Arguments

N	length of vector y
beta0	inicial kick for beta0
x	matrix that contains all covariates
k	number of covariates included in model

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_0$ ” and “ $\beta_k$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
#(n > 1)

set.seed(123)
y <- c(78, 101, 68, 40, 54, 34)
n <- c(106, 112, 114, 45, 59, 40)
y0 <- n - y
x1 <- factor(c(1, 2, 3, 1, 2, 3))
x2 <- factor(c(1, 1, 1, 2, 2, 2))

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
colnames(x)[1:2] <- c("x1", "x2")
resp <- cbind(y, y0)
data <- cbind(y, n, x)
k <- ncol(data)-2

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(resp ~ x, family = binomial(link = "cauchit"))

beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
beta_aux = beta_aux[2:(k+1)]
```

```

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta.aux)

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_cauchy,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 100000,
  burn = 10000,
  jump = 10,
  Round = 3,
  seed = 123,
  title = 'Bayesian Inferential Results',
  save_html = NULL,
  render = TRUE)

## End(Not run)

```

---

bin\_multi\_reg\_model\_cloglog

*Binomial multiple regression model with complementary log-log link.*

---

## Description

Bayesian binomial multiple regression model with complementary log-log link (Gumbel of minimum value).

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with clog-log link:

$$p = 1 - \exp(-\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k))$$

wherein  $X$  is the covariates matrix of the model,  $X = x_1, x_2, \dots, x_K$ , and

$\beta_k$  is the vector of beta coefficients,  $k = 1, \dots, K$ .

Priors distributions for the “ $\beta_0$ ” and “ $\beta_k$ ”:

$\beta_0, \beta_k \sim \text{Normal}(\mu\beta, \tau\beta)$ , with  $\mu\beta$  real and  $\tau\beta > 0$ .

Odds Ratio:  $OR = \exp(\beta)$ .

## Usage

```
bin_multi_reg_model_cloglog(N, beta0, x, k)
```

## Arguments

N	length of vector y
beta0	inicial kick for beta0
x	matrix that contains all covariates
k	number of covariates included in model



## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_0$ ” and “ $\beta_k$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
#(n = 1)

set.seed(123)

y <- c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1)
x1 <- c(4, 4, 5, 6, 6, 8, 9, 11, 12, 13, 13, 14, 18, 18, 19, 20, 22, 22, 24,
        25, 28, 29, 30, 30, 32)
x2 <- c(1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3)

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
n <- rep(1, length(y))
data <- cbind(y, x)
k <- ncol(data)-1

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(y ~ x, family = binomial(link = "cloglog"))
beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
beta_aux = beta_aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta_aux)
```

```

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_cloglog,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

## Not run:
#(n > 1)

set.seed(123)
y <- c(78, 101, 68, 40, 54, 34)
n <- c(106, 112, 114, 45, 59, 40)
y0 <- n - y
x1 <- factor(c(1, 2, 3, 1, 2, 3))
x2 <- factor(c(1, 1, 1, 2, 2, 2))

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
colnames(x)[1:2] <- c("x1", "x2")
resp <- cbind(y, y0)
data <- cbind(y, n, x)
k <- ncol(data)-2

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(resp ~ x, family = binomial(link = "cloglog"))

beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
beta_aux = beta_aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta_aux)

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_cloglog,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,

```

```

        jump = 10,
        Round = 3,
        seed = 123,
        title = 'Bayesian Inferential Results',
        save_html = NULL,
        render = TRUE)

## End(Not run)

```

---

```
bin_multi_reg_model_logit
```

*Binomial multiple regression model with logit link.*

---

## Description

Bayesian binomial multiple regression model with logit link.

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with logit link:

$$p = \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k) / (1 + \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k))$$

wherein  $X$  is the covariates matrix of the model,  $X = x_1, x_2, \dots, x_K$ , and

$\beta_k$  is the vector of beta coefficients,  $k = 1, \dots, K$ .

Priors distributions for the “ $\beta_0$ ” and “ $\beta_k$ ”:

$$\beta_0, \beta_k \sim \text{Normal}(\mu\beta, \tau\beta), \text{ with } \mu\beta \text{ real and } \tau\beta > 0.$$

Odds Ratio:  $OR = \exp(\beta)$ .

## Usage

```
bin_multi_reg_model_logit(N, beta0, x, k)
```

## Arguments

N	length of vector y
beta0	inicial kick for beta0
x	matrix that contains all covariates
k	number of covariates included in model

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_0$ ” and “ $\beta_k$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
#(n = 1)

set.seed(123)

y <- c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1)
x1 <- c(4, 4, 5, 6, 6, 8, 9, 11, 12, 13, 13, 14, 18, 18, 19, 20, 22, 22, 24,
        25, 28, 29, 30, 30, 32)
x2 <- c(1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3)

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
n <- rep(1, length(y))
data <- cbind(y, x)
k <- ncol(data)-1

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(y ~ x, family = binomial(link = "logit"))

beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
beta_aux = beta_aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta_aux)

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_logit,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
```

```

        initials = start_kick,
        iter = 10000,
        burn = 1000,
        jump = 10,
        Round = 3,
        seed = 123,
        title = 'Bayesian Inferential Results',
        save_html = NULL,
        render = TRUE)

## End(Not run)

## Not run:
#(n > 1)

set.seed(123)
y <- c(78, 101, 68, 40, 54, 34)
n <- c(106, 112, 114, 45, 59, 40)
y0 <- n - y
x1 <- factor(c(1, 2, 3, 1, 2, 3))
x2 <- factor(c(1, 1, 1, 2, 2, 2))

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
colnames(x)[1:2] <- c("x1", "x2")
resp <- cbind(y, y0)
data <- cbind(y, n, x)
k <- ncol(data)-2

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(resp ~ x, family = binomial(link = "logit"))

beta.aux <- numeric()
for (j in 1:(k+1))
{
  beta.aux[j] = fit_freq$coef[[j]]
}
beta.aux = beta.aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta.aux)

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_logit,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 10000,
  burn = 1000,
  jump = 10,
  Round = 3,
  seed = 123,
  title = 'Bayesian Inferential Results',
  save_html = NULL,

```

```

render = TRUE)

## End(Not run)

```

---

bin\_multi\_reg\_model\_loglog

*Binomial multiple regression model with log-log link.*

---

## Description

Bayesian binomial multiple regression model with log-log link.

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with log-log link:

$p = \exp(-\exp(-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k)))$

wherein  $X$  is the covariates matrix of the model,  $X = x_1, x_2, \dots, x_K$ , and

$\beta_k$  is the vector of beta coefficients,  $k = 1, \dots, K$ .

Priors distributions for the “ $\beta_0$ ” and “ $\beta_k$ ”:

$\beta_0, \beta_k \sim \text{Normal}(\mu\beta, \tau\beta)$ , with  $\mu\beta$  real and  $\tau\beta > 0$ .

Odds Ratio:  $OR = \exp(\beta)$ .

## Usage

```
bin_multi_reg_model_loglog(N, beta0, x, k)
```

## Arguments

N	length of vector y
beta0	inicial kick for beta0
x	matrix that contains all covariates
k	number of covariates included in model

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_0$ ” and “ $\beta_k$ ” as specified in the above function.

## References

CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.

GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
#(n = 1)

set.seed(123)

y <- c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1)
x1 <- c(4, 4, 5, 6, 6, 8, 9, 11, 12, 13, 13, 14, 18, 18, 19, 20, 22, 22, 24,
        25, 28, 29, 30, 30, 32)
x2 <- c(1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3)

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
n <- rep(1, length(y))
data <- cbind(y, x)
k <- ncol(data)-1

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(y ~ x, family = binomial(link = "cloglog")) #because there is no log-log link
#in the glm family, clog-log will be used.
beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
beta_aux = beta_aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta_aux)

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_loglog,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

## Not run:
#(n > 1)
```

```

set.seed(123)
y <- c(78, 101, 68, 40, 54, 34)
n <- c(106, 112, 114, 45, 59, 40)
y0 <- n - y
x1 <- factor(c(1, 2, 3, 1, 2, 3))
x2 <- factor(c(1, 1, 1, 2, 2, 2))

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
colnames(x)[1:2] <- c("x1", "x2")
resp <- cbind(y, y0)
data <- cbind(y, n, x)
k <- ncol(data)-2

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(resp ~ x, family = binomial(link = "cloglog")) #because there is no log-log link
#in the glm family, clog-log will be used.

beta.aux <- numeric()
for (j in 1:(k+1))
{
  beta.aux[j] = fit_freq$coef[[j]]
}
beta.aux = beta.aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta.aux)

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_loglog,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 10000,
  burn = 1000,
  jump = 10,
  Round = 3,
  seed = 123,
  title = 'Bayesian Inferential Results',
  save_html = NULL,
  render = TRUE)

## End(Not run)

```

---

bin\_multi\_reg\_model\_probit

*Binomial multiple regression model with probit link.*


---

## Description

Bayesian binomial multiple regression model with probit link.



$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with probit link:

$$p = \Phi(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k)$$

wherein  $\Phi$  is the c.d.f of the normal distribution,

$X$  is the covariates matrix of the model,  $X = x_1, x_2, \dots, x_K$ , and

$\beta_k$  is the vector of beta coefficients,  $k = 1, \dots, K$ .

Priors distributions for the “ $\beta_0$ ” and “ $\beta_k$ ”:

$\beta_0, \beta_k \sim \text{Normal}(\mu\beta, \tau\beta)$ , with  $\mu\beta$  real and  $\tau\beta > 0$ .

Odds Ratio:  $OR = \exp(\beta)$ .

### Usage

```
bin_multi_reg_model_probit(N, beta0, x, k)
```

### Arguments

N	length of vector y
beta0	inicial kick for beta0
x	matrix that contains all covariates
k	number of covariates included in model

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_0$ ” and “ $\beta_k$ ” as specified in the above function.

### References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

**Examples**

```

## Not run:
#(n = 1)

set.seed(123)

y <- c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1)
x1 <- c(4, 4, 5, 6, 6, 8, 9, 11, 12, 13, 13, 14, 18, 18, 19, 20, 22, 22, 24,
        25, 28, 29, 30, 30, 32)
x2 <- c(1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3)

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
n <- rep(1, length(y))
data <- cbind(y, x)
k <- ncol(data)-1

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(y ~ x, family = binomial(link = "probit"))
beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
beta_aux = beta_aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta_aux)

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_probit,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

## Not run:
#(n > 1)

set.seed(123)
y <- c(78, 101, 68, 40, 54, 34)
n <- c(106, 112, 114, 45, 59, 40)
y0 <- n - y
x1 <- factor(c(1, 2, 3, 1, 2, 3))

```

```

x2 <- factor(c(1, 1, 1, 2, 2, 2))

x <- matrix(c(x1, x2), byrow = F, ncol = 2)
colnames(x)[1:2] <- c("x1", "x2")
resp <- cbind(y, y0)
data <- cbind(y, n, x)
k <- ncol(data)-2

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(resp ~ x, family = binomial(link = "probit"))

beta.aux <- numeric()
for (j in 1:(k+1))
{
  beta.aux[j] = fit_freq$coef[[j]]
}
beta.aux = beta.aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta.aux)

fit <- BioBayes.fit(bug_model = bin_multi_reg_model_probit,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 10000,
  burn = 1000,
  jump = 10,
  Round = 3,
  seed = 123,
  title = 'Bayesian Inferential Results',
  save_html = NULL,
  render = TRUE)

## End(Not run)

```

---

bin\_reg\_model\_cauchy\_DR

*Binomial regression model with logit cauchy for dose-response.*

---

## Description

Bayesian binomial regression model with cauchy link for dose-response that fit:

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with cauchy link:

$$p = 0.5 + \arctan(\beta_0 + \beta_1 x) / \pi$$

wherein  $x$  is the explanatory variable (usually, values of experimental doses) of model and

$\beta_k$  is the vector of beta coefficients,  $k = 0, 1$ .

Prior distribution:

$\beta_k \sim Normal(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ .

### Usage

```
bin_reg_model_cauchy_DR(N, beta0, beta1, x)
```

### Arguments

N	number of levels in independent variable x
beta0	inicial kick for beta0
beta1	inicial kick for beta1
x	vector of independent variable

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” as specified in the above function.

### References

CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.

GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

y    <- c(0, 1, 4, 15, 28, 29)
yn   <- c(30, 29, 26, 15, 2, 1)
n    <- y + yn
x    <- c(0, 9.21, 16.12, 18.42, 20.72, 23.02)
resp <- cbind(y, yn)

data_input <- list('N' = length(levels(factor(x))), 'n' = n, 'y' = y, 'x' = x)

priors_parm <- list('mu_beta0' = 0, 'tau_beta0' = 0.000001,
```

```

'mu_beta1' = 0, 'tau_beta1' = 0.000001)

parms <- c('beta0', 'beta1', 'DL50')

fit_freq <- glm(resp ~ x, family = binomial(link = cauchit))

start_kick <- function() c('beta0' = fit_freq$coefficients[[1]],
                           'beta1' = fit_freq$coefficients[[2]])

fit <- BioBayes.fit(bug_model = bin_reg_model_cauchy_DR,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 500000,
                   burn = 50000,
                   jump = 500,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y/n, bty = "n", pch = 16)

p.cauchy = function(beta,x)
{
  k = length(x)
  beta0 = beta[1]
  beta.aux = beta[2:(k+1)]
  z = beta0 + beta.aux%%x
  p = 0.5+(atan(z)/pi)
  p
}

y.aux <- numeric()
pars <- apply(X = fit$MCMC[[1]][, 2:3], FUN = median, MARGIN = 2)
x0 <- seq(0, 25, 0.01)
for (i in 1:length(x0))
{
  y.aux[i] = p.cauchy(beta = pars, x = x0[i])
}
lines(x0, y.aux, type="l", col = "red", lwd = 2)

y.freq <- numeric()
pars_freq <- fit_freq$coefficients[1:2]
for (i in 1:length(x0))
{
  y.freq[i] = p.cauchy(beta = pars_freq, x = x0[i])
}
lines(x0, y.freq, type="l", col = "blue", lwd = 2, lty = 2)

legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

bin\_reg\_model\_cloglog\_DR

*Binomial regression model with complementary log-log link for dose-response.*


---

## Description

Bayesian binomial regression model with complementary log-log link for dose-response that fit:

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with clog-log link:

$$p = 1 - \exp(-\exp(\beta_0 + \beta_1 x))$$

wherein  $x$  is the explanatory variable (usually, values of experimental doses) of model and

$\beta_k$  is the vector of beta coefficients,  $k = 0, 1$ .

Prior distribution:

$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ .

## Usage

```
bin_reg_model_cloglog_DR(N, beta0, beta1, x)
```

## Arguments

N	number of levels in independent variable x
beta0	inicial kick for beta0
beta1	inicial kick for beta1
x	vector of independent variable

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

y    <- c(0, 1, 4, 15, 28, 29)
yn   <- c(30, 29, 26, 15, 2, 1)
n    <- y + yn
x    <- c(0, 9.21, 16.12, 18.42, 20.72, 23.02)
resp <- cbind(y, yn)

data_input <- list('N' = length(levels(factor(x))), 'n' = n, 'y' = y, 'x' = x)

priors_parm <- list('mu_beta0' = 0, 'tau_beta0' = 0.000001,
                   'mu_beta1' = 0, 'tau_beta1' = 0.000001)

parms <- c('beta0', 'beta1', 'DL50')

fit_freq <- glm(resp ~ x, family = binomial(link = "cloglog"))

start_kick <- function() c('beta0' = fit_freq$coefficients[[1]],
                           'beta1' = fit_freq$coefficients[[2]])

fit <- BioBayes.fit(bug_model = bin_reg_model_cloglog_DR,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 500000,
                   burn = 50000,
                   jump = 100,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y/n, bty = "n", pch = 16)

p.cloglog = function(beta,x)
{
  k = length(x)
  beta0 = beta[1]
  beta.aux = beta[2:(k+1)]
  z = beta0 + beta.aux%%x
  p = 1-exp(-exp(z))
  p
}

y.aux <- numeric()
pars <- apply(X = fit$MCMC[[1]][, 2:3], FUN = median, MARGIN = 2)
x0 <- seq(0, 25, 0.01)
```

```

for (i in 1:length(x0))
{
  y.aux[i] = p.cloglog(beta = pars, x = x0[i])
}
lines(x0, y.aux, type="l", col = "red", lwd = 2)

y.freq <- numeric()
pars_freq <- fit_freq$coefficients[1:2]
for (i in 1:length(x0))
{
  y.freq[i] = p.cloglog(beta = pars_freq, x = x0[i])
}
lines(x0, y.freq, type="l", col = "blue", lwd = 2, lty = 2)

legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

bin\_reg\_model\_logit      *Binomial regression model with logit link.*

---

## Description

Bayesian model to fit a binomial regression model with logit link.

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with logit link:

$p = \exp(\beta_0 + \beta_1 x) / (1 + \exp(\beta_0 + \beta_1 x))$ ;

wherein  $x$  is the explanatory variable (independent).

Priors distributions for the  $\beta$ 's coefficients:

$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real,  $\tau\beta_k > 0$  and  $k = 0, 1$ .

Odds Ratio:  $OR_k = \exp(\beta_k)$

## Usage

```
bin_reg_model_logit(N, beta0, beta1, x)
```

## Arguments

N	length of vector y
beta0	inicial kick for beta0
beta1	inicial kick for beta1
x	vector of independent variable

## Details

To change the parameters of the priors distributions, is just change the values in object " $\beta_k$ " as specified in the above function.



## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
#(n = 1)

x <- c(0, 0, 0, 0, 250, 250, 250, 250, 500, 500, 500, 500,
       750, 750, 750, 750, 1000, 1000, 1000, 1000)
y <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0)
n <- c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)

data_input <- list('N' = length(y), 'n' = n, 'y' = y, 'x' = x)

priors_parm <- list('mu_beta0' = 0, 'tau_beta0' = 0.000001,
                   'mu_beta1' = 0, 'tau_beta1' = 0.000001)

parms <- c('beta0', 'beta1', 'OR0', 'OR1')

fit_freq <- glm(y~x, family = binomial(link = logit))

start_kick <- function() c('beta0' = fit_freq$coefficients[[1]],
                           'beta1' = fit_freq$coefficients[[2]])

fit <- BioBayes.fit(bug_model = bin_reg_model_logit,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 100000,
                   burn = 10000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)
```

```

## Not run:
#(n > 1)

x <- c(0, 250, 500, 750, 1000)
y <- c(0, 0, 2, 2, 3)
n <- c(4, 4, 4, 4, 4)

data_input <- list('N' = length(levels(factor(x))), 'n' = n, 'y' = y, 'x' = x)

priors_parm <- list('mu_beta0' = 0, 'tau_beta0' = 0.000001,
                   'mu_beta1' = 0, 'tau_beta1' = 0.000001)

parms <- c('beta0', 'beta1', 'OR0', 'OR1')

resp <- cbind(y,n)

fit_freq <- glm(resp~x, family = binomial(link = logit))

start_kick <- function() c('beta0' = fit_freq$coefficients[[1]],
                           'beta1' = fit_freq$coefficients[[2]])

fit <- BioBayes.fit(bug_model = bin_reg_model_logit,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 100000,
                   burn = 10000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

```

---

bin\_reg\_model\_logit\_DR

*Binomial regression model with logit link for dose-response.*


---

## Description

Bayesian binomial regression model with logit link for dose-response that fit:

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with logit link

$$p = \exp(\beta_0 + \beta_1 x) / (1 + \exp(\beta_0 + \beta_1 x))$$

wherein  $x$  is the explanatory variable (usually, values of experimental doses) of model and

$\beta_k$  is the vector of beta coefficients,  $k = 0, 1$ .

Prior distribution:

$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ .

## Usage

```
bin_reg_model_logit_DR(N, beta0, beta1, x)
```

## Arguments

N	number of levels in independent variable x
beta0	inicial kick for beta0
beta1	inicial kick for beta1
x	vector of independent variable

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” as specified in the above function.

## References

CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.

GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y    <- c(0, 1, 4, 15, 28, 29)
yn   <- c(30, 29, 26, 15, 2, 1)
n    <- y + yn
x    <- c(0, 9.21, 16.12, 18.42, 20.72, 23.02)
resp <- cbind(y, yn)

data_input <- list('N' = length(levels(factor(x))), 'n' = n, 'y' = y, 'x' = x)

priors_parm <- list('mu_beta0' = 0, 'tau_beta0' = 0.000001,
                  'mu_beta1' = 0, 'tau_beta1' = 0.000001)
```

```

parms <- c('beta0', 'beta1', 'DL50')

fit_freq <- glm(resp ~ x, family = binomial(link = logit))

start_kick <- function() c('beta0' = fit_freq$coefficients[[1]],
                           'beta1' = fit_freq$coefficients[[2]])

fit <- BioBayes.fit(bug_model = bin_reg_model_logit_DR,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 500000,
                   burn = 50000,
                   jump = 500,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y/n, bty = "n", pch = 16)

p.logit = function(beta,x)
{
  k = length(x)
  beta0 = beta[1]
  beta.aux = beta[2:(k+1)]
  z = beta0 + beta.aux*%*%x
  p = exp(z)/(1+exp(z))
  p
}

y.aux <- numeric()
pars <- apply(X = fit$MCMC[[1]][, 2:3], FUN = median, MARGIN = 2)
x0 <- seq(0, 25, 0.01)
for (i in 1:length(x0))
{
  y.aux[i] = p.logit(beta = pars, x = x0[i])
}
lines(x0, y.aux, type="l", col = "red", lwd = 2)

y.freq <- numeric()
pars_freq <- fit_freq$coefficients[1:2]
for (i in 1:length(x0))
{
  y.freq[i] = p.logit(beta = pars_freq, x = x0[i])
}
lines(x0, y.freq, type="l", col = "blue", lwd = 2, lty = 2)

legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

bin\_reg\_model\_loglog\_DR

*Binomial regression model with log-log link for dose-response.*


---

### Description

Bayesian binomial regression model with log-log link for dose-response that fit:

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with log-log link:

$p = \exp(-\exp(-(\beta_0 + \beta_1 x)))$

wherein  $x$  is the explanatory variable (usually, values of experimental doses) of model and

$\beta_k$  is the vector of beta coefficients,  $k = 0, 1$ .

Prior distribution:

$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ .

### Usage

```
bin_reg_model_loglog_DR(N, beta0, beta1, x)
```

### Arguments

N	number of levels in independent variable x
beta0	inicial kick for beta0
beta1	inicial kick for beta1
x	vector of independent variable

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” as specified in the above function.

### References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

**Examples**

```
## Not run:
set.seed(123)

y    <- c(0, 1, 4, 15, 28, 29)
yn   <- c(30, 29, 26, 15, 2, 1)
n    <- y + yn
x    <- c(0, 9.21, 16.12, 18.42, 20.72, 23.02)
resp <- cbind(y, yn)

data_input <- list('N' = length(levels(factor(x))), 'n' = n, 'y' = y, 'x' = x)

priors_parm <- list('mu_beta0' = 0, 'tau_beta0' = 0.000001,
                   'mu_beta1' = 0, 'tau_beta1' = 0.000001)

parms <- c('beta0', 'beta1', 'DL50')

fit_freq <- glm(resp ~ x, family = binomial(link = "cloglog")) #because there is no log-log link
#in the glm family, clog-log will be used.

start_kick <- function() c('beta0' = fit_freq$coefficients[[1]],
                           'beta1' = fit_freq$coefficients[[2]])

fit <- BioBayes.fit(bug_model = bin_reg_model_loglog_DR,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 500000,
                   burn = 50000,
                   jump = 100,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y/n, bty = "n", pch = 16, xlim = c(0, 30))

p.loglog = function(beta,x)
{
  k = length(x)
  beta0 = beta[1]
  beta.aux = beta[2:(k+1)]
  z = beta0 + beta.aux%*%x
  p = exp(-exp(-z))
  p
}

y.aux <- numeric()
pars <- apply(X = fit$MCMC[[1]][, 2:3], FUN = median, MARGIN = 2)
x0 <- seq(0, 25, 0.01)
for (i in 1:length(x0))
{
  y.aux[i] = p.loglog(beta = pars, x = x0[i])
}
```

```

lines(x0, y.aux, type="l", col = "red", lwd = 2)

y.freq <- numeric()
pars_freq <- fit_freq$coefficients[1:2]
for (i in 1:length(x0))
{
  y.freq[i] = p.loglog(beta = pars_freq, x = x0[i])
}
lines(x0, y.freq, type="l", col = "blue", lwd = 2, lty = 2)

legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

bin\_reg\_model\_probit\_DR

*Binomial regression model with probit link for dose-response.*

---

## Description

Bayesian binomial regression model with probit link for dose-response that fit:

$Y_i \sim \text{Binomial}(n, p)$ , such that  $0 < p < 1$

with probit link:

$$p = \Phi(\beta_0 + \beta_1 x)$$

wherein  $\Phi$  is the c.d.f of normal distribution,

$x$  is the explanatory variable (usually, values of experimental doses) of model and

$\beta_k$  is the vector of beta coefficients,  $k = 0, 1$ .

Prior distribution:

$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ .

## Usage

```
bin_reg_model_probit_DR(N, beta0, beta1, x)
```

## Arguments

N	number of levels in independent variable x
beta0	inicial kick for beta0
beta1	inicial kick for beta1
x	vector of independent variable

## Details

To change the parameters of the priors distributions, is just change the values in object " $\beta_k$ " as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Disponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y    <- c(0, 1, 4, 15, 28, 29)
yn   <- c(30, 29, 26, 15, 2, 1)
n    <- y + yn
x    <- c(0, 9.21, 16.12, 18.42, 20.72, 23.02)
resp <- cbind(y, yn)

data_input <- list('N' = length(levels(factor(x))), 'n' = n, 'y' = y, 'x' = x)

priors_parm <- list('mu_beta0' = 0, 'tau_beta0' = 0.000001,
                   'mu_beta1' = 0, 'tau_beta1' = 0.000001)

parms <- c('beta0', 'beta1', 'DL50')

fit_freq <- glm(resp ~ x, family = binomial(link = "probit"))

start_kick <- function() c('beta0' = fit_freq$coefficients[[1]],
                           'beta1' = fit_freq$coefficients[[2]])

fit <- BioBayes.fit(bug_model = bin_reg_model_probit_DR,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 500000,
                   burn = 50000,
                   jump = 100,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y/n, bty = "n", pch = 16)
```



```

p.probit = function(beta,x)
{
  k = length(x)
  beta0 = beta[1]
  beta.aux = beta[2:(k+1)]
  z = beta0 + beta.aux%%x
  p = pnorm(z)
  p
}

y.aux <- numeric()
pars <- apply(X = fit$MCMC[[1]][, 2:3], FUN = median, MARGIN = 2)
x0 <- seq(0, 25, 0.01)
for (i in 1:length(x0))
{
  y.aux[i] = p.probit(beta = pars, x = x0[i])
}
lines(x0, y.aux, type="l", col = "red", lwd = 2)

y.freq <- numeric()
pars_freq <- fit_freq$coefficients[1:2]
for (i in 1:length(x0))
{
  y.freq[i] = p.probit(beta = pars_freq, x = x0[i])
}
lines(x0, y.freq, type="l", col = "blue", lwd = 2, lty = 2)

legend("topleft", legend = c("Bayesian adjstment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

BioBayes.fit

*Bayesian function to fit the available models.*


---

## Description

Bayesian function to fit the available models based on Markov Chains Monte Carlo simulation (MCMC).

## Usage

```
BioBayes.fit(bug_model, database, priors, parameters, initials, iter, burn,
             jump, Round, seed, title, save_html, render)
```

## Arguments

bug_model	Appropriate model contained in the BioBayes package to analyze the data types. The following models are available: <a href="#">anova_hetero_model</a> <a href="#">anova_homo_model</a>
-----------	--

```

beta_diff_model
beta_model
binomial_diff_model
binomial_model
bin_multi_reg_model_cauchy
bin_multi_reg_model_cloglog
bin_multi_reg_model_logit
bin_multi_reg_model_loglog
bin_multi_reg_model_probit
bin_reg_model_cauchy_DR
bin_reg_model_cloglog_DR
bin_reg_model_logit
bin_reg_model_logit_DR
bin_reg_model_loglog_DR
bin_reg_model_probit_DR
brody_normal_model
exponential_diff_model
exponential_model
gamma_diff_model
gamma_model
gompertz_normal_model
linear_multi_reg_model
linear_multi_reg_model_poisson
linear_reg_model
linear_reg_model_poisson
logistic_normal_model
normal_hetero_diff_model
normal_homo_diff_model
normal_model
normal_paired_model
poisson_diff_model
poisson_model
quadratic_control_reg_model
quadratic_reg_model
vonbertalanffy_normal_model
wood_normal_model

```

database	Data that will enter to the model, in list format.
priors	Values for the parameters of the prior distributions, in list format.
parameters	Vector with the output names of the values after the Bayesian analysis (defined in the bug model).
initials	Initial kicks for the parameters to be estimated, in list format.
iter	Number of iterations to generate the MCMC chains.
burn	Number of observations to be discarded at the beginning (adaptation period) of the MCMC process.
jump	Jump to eliminate possible autocorrelation in the simulated chain.

Round	Rounds the values to the specified number of decimal places.
seed	Fixed seed.
title	Title for results. Sugestion: 'Bayesian Inferential Results'.
save_html	Not save: NULL.
render	Generate html: TRUE.

### Details

The function uses the arguments defined with the same names shown in the exemples for each type of data.

### References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- GIOLO, S. R. Introdução à análise de dados categóricos com aplicações. Diponível em: [www.ufpr.br/~giolo](http://www.ufpr.br/~giolo), 2012.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.
- WOOD, P.D.P. Breed variation in the shape of the lactation curve of cattle and their implications for efficiency. Animal Production, v.34, p.133-41, 1980.

---

brody_normal_model	<i>Brody's nonlinear growth model.</i>
--------------------	--

---

### Description

Brody's nonlinear growth bayesian model with normal errors that adjusts

$$Y_i \sim Normal(FNL, \tau)$$

$$\text{such that } FNL = \beta_1(1 - \beta_2 \exp(-\beta_3 x))$$

where  $x$  is the vector of the explanatory variable (usually observations in time) of the model and

$\beta_k$  is the vector of beta coefficients,  $k = 1, 2, 3$  such that:

$\beta_1$  represents the asymptotic value of the response variable,  $\beta_2$  is a constant that indicates the proportion of the asymptotic value to be gained and  $\beta_3$  is the growth rate of the response variable that determines the efficiency of growth.

Priors distributions:

$$\beta_k \sim Normal(\mu\beta_k, \tau\beta_k), \text{ with } \mu\beta_k \text{ real and } \tau\beta_k > 0;$$

$$\tau \sim Gamma(a, b), \text{ with } a > 0 \text{ and } b > 0.$$

**Usage**

```
brody_normal_model(n, beta1, beta2, beta3, x, tau)
```

**Arguments**

n	length of vector y
beta1	inicial kick for beta1
beta2	inicial kick for beta2
beta3	inicial kick for beta3
x	vector of independent variable
tau	inicial kick for tau

**Details**

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” and “ $\tau$ ” as specified in the above function.

**References**

CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

**Examples**

```
## Not run:
set.seed(123)

y <- c(1.3, 1.3, 1.9, 3.4, 5.3, 7.1, 10.6, 16, 16.4, 18.3, 20.9, 20.5, 21.3, 21.2, 20.9)
x <- c(0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5, 11.5, 12.5, 13.5, 14.5)

data_input <- list('n' = length(y), 'y' = y, 'x' = x)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001,
                  'mu_tau' = 0.001, 'tau_tau' = 0.001)

parms <- c('beta1', 'beta2', 'beta3', 'tau', 'variance')

brody = function(pars,x)
{
  beta1 = pars[1]
  beta2 = pars[2]
  beta3 = pars[3]
```

```

y = beta1*(1 - beta2*exp(-beta3*x))
y
}

fit_freq <- nls(y ~ brody(pars, x), start = list(pars = c(22, 4, 0.5)))

start_kick <- function() c('beta1' = 45.7,
                           'beta2' = 1.06,
                           'beta3' = 0.05,
                           'tau'   = 1)

fit <- BioBayes.fit(bug_model = brody_normal_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 100000,
                   burn = 10000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian inferential results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y, bty = "n", pch = 16)
pars = apply(X = fit$MCMC[[1]][, 1:3], FUN = median, MARGIN = 2)
y.est = brody(pars = pars, sort(x))
lines(sort(x), y.est, type = "l", col = "red", lwd = 2)
pars_freq = c(45.77544, 1.06236, 0.05389)
y.freq = brody(pars = pars_freq, sort(x))
lines(sort(x), y.freq, type = "l", col = "blue", lwd = 2, lty = 2)
legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

exponential\_diff\_model

*Model for difference of means between two independent exponential populations.*

---

## Description

Bayesian model for adjusting the difference of means between two independent exponential populations.

$Y_k \sim \text{Exponential}(\lambda_k)$ , with  $Y_k > 0$ , and  $k = 1, 2$

with priors distribution:

$\lambda_1 \sim \text{Gamma}(a, b)$

$\lambda_2 \sim \text{Gamma}(c, d)$

**Usage**

```
exponential_diff_model(n1, n2, lambda1, lambda2)
```

**Arguments**

n1	length of vector y1
n2	length of vector y2
lambda1	inicial kick for lambda in y1
lambda2	inicial kick for lambda in y2

**Details**

To change the parameters of the priors distributions, is just change the values in object “ $\lambda_1, \lambda_2$ ” as specified in the above function.

**References**

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

**Examples**

```
## Not run:
set.seed(123)

y1 <- rexp(n = 50, rate = 5)

y2 <- rexp(n = 60, rate = 7)

data_input <- list('n1' = length(y1), 'y1' = y1,
                  'n2' = length(y2), 'y2' = y2)

priors_parm <- list('a' = 0.001, 'b' = 0.001, 'c' = 0.001, 'd' = 0.001)

parms <- c('lambda1', 'lambda2', 'mu1', 'mu2', 'delta_mu_1x2')

start_kick <- function() list('lambda1' = 1/mean(y1), 'lambda2' = 1/mean(y2))

fit <- BioBayes.fit(bug_model = exponential_diff_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
```

```

title = 'Bayesian Inferential Results',
save_html = NULL,
render = TRUE)

## End(Not run)

```

---

exponential_model	<i>Model for exponential distribution.</i>
-------------------	--

---

### Description

Bayesian model to fit  
 $Y \sim \text{Exponential}(\lambda)$ , with  $Y > 0$   
 with prior distribution:  
 $\lambda \sim \text{Gamma}(a, b)$  with  $a > 0, b > 0$ .

### Usage

```
exponential_model(n, lambda)
```

### Arguments

n	length of vector y
lambda	inicial kick for lambda

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\lambda$ ” as specified in the above function.

### References

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.  
 McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.  
 NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.  
 PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.  
 ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```

## Not run:
set.seed(123)

y <- rexp(n = 50, rate = 5)

data_input <- list('n' = length(y), 'y' = y)

```

```

priors_parm <- list('a' = 0.001, 'b' = 0.001)

parms <- c('lambda', 'mean', 'variance')

start_kick <- function() list('lambda' = 1/mean(y))

fit <- BioBayes.fit(bug_model = exponential_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

```

---

gamma_diff_model	<i>Model for difference of means between two independent gamma populations.</i>
------------------	---

---

## Description

Bayesian model for adjusting the difference of means between two independent gamma populations.

$Y_k \sim \text{Gamma}(a_k, b_k)$ , such that  $Y_k > 0, a_k > 0, b_k > 0, k = 1, 2$

with priors distributions:

$a_1 \sim \text{Gamma}(c_1, d_1)$

$b_1 \sim \text{Gamma}(e_1, f_1)$

$a_2 \sim \text{Gamma}(c_2, d_2)$

$b_2 \sim \text{Gamma}(e_2, f_2)$

## Usage

```
gamma_diff_model(n1, n2, a1, b1, a2, b2)
```

## Arguments

n1	length of vector y1
n2	length of vector y2
a1	inicial kick for the shape parameter in the gamma distribution for y1
b1	inicial kick for the rate parameter in the gamma distribution for y1
a2	inicial kick for the shape parameter in the gamma distribution for y2
b2	inicial kick for the rate parameter in the gamma distribution for y2



## Details

To change the parameters of the priors distributions, is just change the values in object “ $a_1, b_1, a_2, b_2$ ” as specified in the above function.

## References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y1 <- rgamma(30, 2, 4)

y2 <- rgamma(20, 51, 10)

data_input <- list('n1' = length(y1), 'y1' = y1,
                  'n2' = length(y2), 'y2' = y2)

priors_parm <- list('c1' = 0.001, 'd1' = 0.001, 'e1' = 0.001, 'f1' = 0.001,
                  'c2' = 0.001, 'd2' = 0.001, 'e2' = 0.001, 'f2' = 0.001)

parms <- c('a1', 'a2', 'b1', 'b2', 'mu1', 'mu2', 'delta_mu_1x2')

fit1 <- fitdist(data = y1, distr = 'gamma', method = 'mme')

fit2 <- fitdist(data = y2, distr = 'gamma', method = 'mme')

start_kick <- function() list('a1' = fit1$estimate[1], 'a2' = fit2$estimate[1],
                              'b1' = fit1$estimate[2], 'b2' = fit2$estimate[2])

fit <- BioBayes.fit(bug_model = gamma_diff_model,
                  database = data_input,
                  priors = priors_parm,
                  parameters = parms,
                  initials = start_kick,
                  iter = 100000,
                  burn = 10000,
                  jump = 50,
                  Round = 3,
                  seed = 123,
                  title = 'Bayesian Inferential Results',
                  save_html = NULL,
                  render = TRUE)
```

```
## End(Not run)
```

---

gamma_model	<i>Model for gamma distribution.</i>
-------------	--------------------------------------

---

## Description

Bayesian model to fit

$Y \sim \text{Gamma}(a, b)$ , such that  $Y > 0$

with priors distributions:

$a \sim \text{Gamma}(a_1, b_1)$  with  $a_1 > 0, b_1 > 0$ ;

$b \sim \text{Gamma}(a_2, b_2)$  with  $a_2 > 0, b_2 > 0$ .

## Usage

```
gamma_model(n, a, b)
```

## Arguments

n	length of vector y
a	inicial kick for the shape parameter in the gamma distribution for y
b	inicial kick for the rate parameter in the gamma distribution for y

## Details

To change the parameters of the priors distributions, is just change the values in object “a” and “b” as specified in the above function.

## References

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y <- rgamma(n = 50, shape = 2, rate = 5)

data_input <- list('n' = length(y), 'y' = y)

priors_parm <- list('a1' = 0.001, 'b1' = 0.001, 'a2' = 0.001, 'b2' = 0.001)
```

```

parms <- c('a', 'b', 'mean', 'variance')

fit_gamma <- fitdist(data = y, distr = 'gamma', method = 'mme')

start_kick <- function() list('a' = fit_gamma$estimate[1], 'b' = fit_gamma$estimate[2])

fit <- BioBayes.fit(bug_model = gamma_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inference Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

```

---

gompertz\_normal\_model *Gompertz nonlinear growth model.*

---

## Description

Gompertz nonlinear growth bayesian model with normal errors that adjusts

$$Y_i \sim \text{Normal}(FNL, \tau)$$

$$\text{such that } FNL = \beta_1 \exp(-\beta_2 \exp(-\beta_3 x))$$

where  $x$  is the vector of the explanatory variable (usually observations in time) of the model and

$\beta_k$  is the vector of beta coefficients,  $k = 1, 2, 3$  such that

$\beta_1$  represents the asymptotic value of the response variable,  $\beta_2$  is a constant that indicates the proportion of the asymptotic value to be gained and  $\beta_3$  is the growth rate of the response variable that determines the efficiency of growth.

Priors distributions:

$$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k), \text{ with } \mu\beta_k \text{ real and } \tau\beta_k > 0;$$

$$\tau \sim \text{Gamma}(a, b), \text{ with } a > 0 \text{ and } b > 0.$$

## Usage

```
gompertz_normal_model(n, beta1, beta2, beta3, x, tau)
```

## Arguments

n	length of vector y
beta1	inicial kick for beta1
beta2	inicial kick for beta2

beta3	inicial kick for beta3
x	vector of independent variable
tau	inicial kick for tau

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” and “ $\tau$ ” as specified in the above function.

### References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

y <- c(1.3, 1.3, 1.9, 3.4, 5.3, 7.1, 10.6, 16, 16.4, 18.3, 20.9, 20.5, 21.3, 21.2, 20.9)

x <- c(0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5, 11.5, 12.5, 13.5, 14.5)

data_input <- list('n' = length(y), 'y' = y, 'x' = x)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001,
                  'mu_tau' = 0.001, 'tau_tau' = 0.001)

parms <- c('beta1', 'beta2', 'beta3', 'tau', 'variance')

gompertz = function(pars, x)
{
  beta1 = pars[1]
  beta2 = pars[2]
  beta3 = pars[3]
  y = beta1*exp(-beta2*exp(-beta3*x))
  y
}

fit_freq <- nls(y ~ gompertz(pars, x), start = list(pars = c(20, 4, 0.5)))

start_kick <- function() c('beta1' = 22.5,
                          'beta2' = 8.2,
                          'beta3' = 0.38,
                          'tau' = 1)
```

```

fit <- BioBayes.fit(bug_model = gompertz_normal_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 100000,
                   burn = 10000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y, bty = "n", pch = 16)
pars = apply(X = fit$MCMC[[1]][, 1:3], FUN = median, MARGIN = 2)
y.est = gompertz(pars = pars, sort(x))
lines(sort(x), y.est, type = "l", col = "red", lwd = 2)
pars_freq = c(22.5066, 8.2175, 0.3881)
y.freq = gompertz(pars = pars_freq, sort(x))
lines(sort(x), y.freq, type = "l", col = "blue", lwd = 2, lty = 2)
legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

linear\_multi\_reg\_model

*Linear multiple regression model.*

---

## Description

Bayesian linear multiple regression model for  $P$  covariates that fit:

$$Y \sim \text{Normal}(\mu, \tau)$$

such that  $\mu$  is the multiple linear predictor, where

$$\mu = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$$

and  $\tau > 0$ . Also,

$X$  is the covariates matrix of the model,  $X = x_1, x_2, \dots, x_K$  and

$\beta_k$  is the vector of beta coefficients,  $k = 1, \dots, K$ .

Priors distributions for the “ $\beta_0$ ” and “ $\beta_k$ ”:

$$\beta_0, \beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k), \text{ with } \mu\beta_k \text{ real and } \tau\beta_k > 0.$$

$$\tau \sim \text{Gamma}(a, b), \text{ with } a > 0 \text{ and } b > 0.$$

## Usage

```
linear_multi_reg_model(n, beta0, x, k, tau)
```

## Arguments

n	length of vector y
beta0	initial kick for beta0
x	vector of independent variable
k	number of covariates included in model
tau	initial kick for tau

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” and “ $\tau$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y  <- rnorm(20, 9, 3)
x1 <- rnorm(20, 3, 1)
x2 <- rnorm(20, 1, 1/3)

x  <- matrix(c(x1, x2), byrow = F, ncol = 2)
data <- cbind(y, x)
k  <- ncol(data)-1

data_input <- list('n' = length(y), 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001, 'a_tau' = 0.001, 'b_tau' = 0.001)

parms <- c('beta0', 'beta', 'tau', 'variance')

fit_freq <- lm(y ~ x)

beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
beta_aux <- beta_aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta_aux, 'tau' = 1)
```

```

fit <- BioBayes.fit(bug_model = linear_multi_reg_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 1000000,
                   burn = 50000,
                   jump = 100,
                   Round = 3,
                   seed = 123,
                   title= 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

```

---

```
linear_multi_reg_model_poisson
```

*Linear multiple regression model with poisson errors.*

---

## Description

Bayesian linear multiple regression model with poisson erros that fit:

$Y \sim \text{Poisson}(\mu)$ ,

such that  $Y$  is the integer response variable (dependent),

$\mu$  is the multiple linear predictor where

$\log(\mu) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$ . Also,

$X$  is the covariates matrix of the model,  $X = x_1, x_2, \dots, x_K$  and

$\beta_k$  is the vector of beta coefficients,  $k = 1, \dots, K$ .

Priors distributions for the regression coefficients “ $\beta_0$ ” and “ $\beta_k$ ”:

$\beta_0, \beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ .

## Usage

```
linear_multi_reg_model_poisson(n, beta0, x, k)
```

## Arguments

n	length of vector y
beta0	initial kick for beta0
x	vector of independent variable
k	number of covariates included in model

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y <- c(0, 1, 0, 0, 0, 0, 1, 0, 0, 2, 1, 1, 1, 1, 2, 3, 1, 1, 1, 2, 0, 1, 1,
      2, 5, 1, 1, 5, 5, 7)

x1 <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 1, 1)

x2 <- c(4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8, 7, 7, 7, 10, 10, 10, 12,
      12, 12, 8, 8, 8, 14, 14, 14)

x3 <- c(91.5, 84, 76.5, 69, 61.5, 80, 72.5, 65, 57.5, 50, 103, 95.5, 88, 80.5,
      73, 116.1, 100.6, 85, 69.4, 53.9, 112.3, 96.7, 81.1, 65.6, 50, 120,
      104.4, 88.9, 73.7, 57.8)

x <- matrix(c(x1, x2, x3), byrow = F, ncol = 3)
data <- cbind(y, x)
k <- ncol(data)-1

data_input <- list('n' = length(y), 'y' = y, 'x' = x, 'k' = k)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta', 'OR0', 'OR')

fit_freq <- glm(y ~ x, family = poisson())
beta_aux <- numeric()
for (j in 1:(k+1))
{
  beta_aux[j] = fit_freq$coef[[j]]
}
beta_aux <- beta_aux[2:(k+1)]

start_kick <- function() list('beta0' = fit_freq$coef[[1]], 'beta' = beta_aux)

fit <- BioBayes.fit(bug_model = linear_multi_reg_model_poisson,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 10000,
```



```

        burn = 1000,
        jump = 10,
        Round = 3,
        seed = 123,
        title = 'Bayesian Inferential Results',
        save_html = NULL,
        render = TRUE)

## End(Not run)

```

---

linear_reg_model	<i>Linear regression model.</i>
------------------	---------------------------------

---

### Description

Bayesian linear regression model to fit:

$$Y \sim \text{Normal}(\mu, \tau),$$

such that  $Y$  is a response variable (dependent) in real numbers,

$\mu$  is the linear predictor, where

$$\mu = \beta_0 + \beta_1 x \text{ and}$$

$\tau > 0$ . Also,

$x$  is the explanatory variable (independent), real and

$\beta_k$  is the vector of regression coefficients,  $k = 0, 1$ .

Priors distributions for the “ $\beta_k$ ”:

$$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k), \text{ with } \mu\beta_k \text{ real and } \tau\beta_k > 0,$$

$$\tau \sim \text{Gamma}(a, b), \text{ with } a > 0 \text{ and } b > 0.$$

### Usage

```
linear_reg_model(n, beta0, beta1, x, tau, x0)
```

### Arguments

n	length of vector y
beta0	initial kick for beta0
beta1	initial kick for beta1
x	vector of independent variable
tau	initial kick for tau
x0	point to be predicted

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” and “ $\tau$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y <- c(21.28, 31.6, 32.42, 41.74, 42.06, 53.38)

x <- c(0.184, 0.35, 0.516, 0.683, 0.849, 1.015)

x0 <- 0.5 #point to be predicted (y.pred)

data_input <- list('n' = length(y), 'y' = y, 'x' = x, 'x0' = x0)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001, 'a_tau' = 0.001, 'b_tau' = 0.001)

parms <- c('beta0', 'beta1', 'tau', 'variance', 'y.pred')

fit_freq <- lm(y ~ x)

start_kick <- function() list('beta0' = fit_freq$coef[1], 'beta1' = fit_freq$coef[2], 'tau' = 1)

fit <- BioBayes.fit(bug_model = linear_reg_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y, bty = "n", pch = 16)
x0 = seq(0, 2, by = 0.01)
pars = apply(X = fit$MCMC[[1]][, 1:2], FUN = median, MARGIN = 2)
y.est = pars[1] + pars[2]*x0
lines(sort(x0), y.est, type = "l", col = "red", lwd = 2)
y.freq = fit_freq$coef[1] + fit_freq$coef[2]*x0
lines(sort(x0), y.freq, type = "l", col = "blue", lwd = 2, lty = 2)
legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")
```

```
## End(Not run)
```

---

```
linear_reg_model_poisson
```

*Linear regression model with poisson errors.*

---

## Description

Bayesian linear regression model with poisson errors to fit:

$$Y \sim \text{Poisson}(\mu),$$

such that  $Y$  is the response variable (dependent) in integers,

$\mu$  is the linear predictor, where

$$\log(\mu) = \beta_0 + \beta_1 x. \text{ Also,}$$

$x$  is the explanatory variable (independent), real and

$\beta_k$  is the vector of regression coefficients,  $k = 0, 1$ .

Priors distributions for the “ $\beta_k$ ”:

$$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k), \text{ with } \mu\beta_k \text{ real and } \tau\beta_k > 0.$$

## Usage

```
linear_reg_model_poisson(beta0, beta1, x, n)
```

## Arguments

beta0	initial kick for beta0
beta1	initial kick for beta1
x	vector of independent variable
n	length of vector y

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

**Examples**

```
## Not run:
set.seed(123)

y <- c(0, 1, 0, 0, 0, 0, 1, 0, 0, 2, 1, 1, 1, 1, 2, 3, 1, 1, 1, 2, 0, 1, 1,
      2, 5, 1, 1, 5, 5, 7)

x <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 1, 1)

data_input <- list('n' = length(y), 'y' = y, 'x' = x)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001)

parms <- c('beta0', 'beta1', 'OR0', 'OR1')

fit_freq <- glm(y ~ x, family = poisson())

start_kick <- function() list('beta0' = fit_freq$coef[1], 'beta1' = fit_freq$coef[2])

fit <- BioBayes.fit(bug_model = linear_reg_model_poisson,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)
```

---

logistic\_normal\_model *Logistic nonlinear growth model.*


---

**Description**

Logistic nonlinear growth bayesian model with normal errors that adjusts

$$Y_i \sim \text{Normal}(FNL, \tau)$$

such that  $FNL = \beta_1 / (1 + \exp(\beta_2 - \beta_3 x))$

where  $x$  is the vector of the explanatory variable (usually observations in time) of the model and

$\beta_k$  is the vector of beta coefficients,  $k = 1, 2, 3$  such that

$\beta_1$  represents the asymptotic value of the response variable,  $\beta_2$  is a constant that indicates the proportion of the asymptotic value to be gained and  $\beta_3$  is the growth rate of the response variable that determines the efficiency of growth.

Priors distributions:

$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ ;

$\tau \sim \text{Gamma}(a, b)$ , with  $a > 0$  and  $b > 0$ .

### Usage

```
logistic_normal_model(n, beta1, beta2, beta3, x, tau)
```

### Arguments

n	length of vector y
beta1	inicial kick for beta1
beta2	inicial kick for beta2
beta3	inicial kick for beta3
x	vector of independent variable
tau	inicial kick for tau

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” and “ $\tau$ ” as specified in the above function.

### References

CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

y <- c(1.3, 1.3, 1.9, 3.4, 5.3, 7.1, 10.6, 16, 16.4, 18.3, 20.9, 20.5, 21.3, 21.2, 20.9)

x <- c(0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5, 11.5, 12.5, 13.5, 14.5)

data_input <- list('n' = length(y), 'y' = y, 'x' = x)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001,
                  'mu_tau' = 0.001, 'tau_tau' = 0.001)

parms <- c('beta1', 'beta2', 'beta3', 'tau', 'variance')
```

```

logistic = function(pars,x)
{
  beta1 = pars[1]
  beta2 = pars[2]
  beta3 = pars[3]
  y = beta1/(1+exp(beta2-beta3*x))
  y
}

fit_freq <- nls(y ~ logistic(pars, x), start = list(pars = c(20, 4, 0.5)))

start_kick <- function() c('beta1' = 21.5,
                           'beta2' = 3.9,
                           'beta3' = 0.62,
                           'tau'   = 1)

fit <- BioBayes.fit(bug_model = logistic_normal_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 100000,
                   burn = 10000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y, bty = "n", pch = 16)
pars = apply(X = fit$MCMC[[1]][, 1:3], FUN = median, MARGIN = 2)
y.est = logistic(pars = pars, sort(x))
lines(sort(x), y.est, type = "l", col = "red", lwd = 2)
pars_freq = c(21.5089, 3.9573, 0.6222)
y.freq = logistic(pars = pars_freq, sort(x))
lines(sort(x), y.freq, type = "l", col = "blue", lwd = 2, lty = 2)
legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

normal\_hetero\_diff\_model

*Model for difference of means between two independent normal populations with heterogeneous variances.*

---

## Description

Bayesian model for adjusting the difference of means and variances between two independent normal populations with heterogeneous variances.

$Y_k \sim \text{Normal}(\mu_k, \tau_k)$ , with  $Y_k$  and  $\mu_k$  reals  $\tau_k > 0$  and  $k = 1, 2$

with priors distributions:

$$\mu_1 \sim \text{Normal}(\mu\mu_1, \tau\mu_1)$$

$$\mu_2 \sim \text{Normal}(\mu\mu_2, \tau\mu_2)$$

$$\tau_1 \sim \text{Gamma}(a\tau_1, b\tau_1)$$

$$\tau_2 \sim \text{Gamma}(a\tau_2, b\tau_2)$$

## Usage

```
normal_hetero_diff_model(n1, n2, tau1, tau2, mu1, mu2)
```

## Arguments

n1	length of vector y1
n2	length of vector y2
tau1	initial kick for tau of y1
tau2	initial kick for tau of y2
mu1	initial kick for mu of y1
mu2	initial kick for mu of y2

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\mu_1, \mu_2, \tau_1, \tau_2$ ” as specified in the above function.

## References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y1 <- rnorm(n = 30, mean = 4, sd = 1)

y2 <- rnorm(n = 50, mean = 5, sd = 0.5)

data_input <- list('n1'=length(y1), 'y1'=y1,
                  'n2'=length(y2), 'y2'=y2)

priors_parm <- list('mu_mu1' = 0, 'tau_mu1' = 0.000001,
                  'mu_mu2' = 0, 'tau_mu2' = 0.000001,
                  'a_tau1' = 0.001, 'b_tau1' = 0.001,
```

```

'a_tau2' = 0.001, 'b_tau2' = 0.001)

fit1 <- fitdist(data = y1, distr = 'norm', method = "mme")

fit2 <- fitdist(data = y2, distr = 'norm', method = "mme")

parms <- c('mu1', 'mu2', 'tau1', 'tau2', 'variance1', 'variance2',
           'delta_mu_1x2', 'delta_var_1x2')

start_kick <- function() list('mu1'= fit1$estimate[1], 'tau1' = 1/(fit1$estimate[2]^2),
                              'mu2'= fit2$estimate[1], 'tau2' = 1/(fit2$estimate[2]^2))

fit <- BioBayes.fit(bug_model = normal_hetero_diff_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

```

---

normal\_homo\_diff\_model

*Model for difference of means between two independent normal populations with homogeneous variances.*

---

## Description

Bayesian model for adjusting the difference of means and variances between two independent normal populations with homogeneous variances.

$Y_k \sim \text{Normal}(\mu_k, \tau)$ , with  $Y_k$  and  $\mu_k$  reals,  $\tau > 0$  and  $k = 1, 2$

with priors distributions:

$\mu_1 \sim \text{Normal}(\mu\mu_1, \tau\mu_1)$

$\mu_2 \sim \text{Normal}(\mu\mu_2, \tau\mu_2)$

$\tau \sim \text{Gamma}(a\tau, b\tau)$

## Usage

```
normal_homo_diff_model(n1, n2, tau, mu1, mu2)
```

## Arguments

n1	length of vector y1
n2	length of vector y2



tau	initial kick for tau
mu1	initial kick for mu of y1
mu2	initial kick for mu of y2

### Details

To change the parameters of the priors distributions, is just change the values in object " $\mu_1, \mu_2, \tau$ " as specified in the above function.

### References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

y1 <- rnorm(n = 30, mean = 4, sd = 0.5)

y2 <- rnorm(n = 50, mean = 5, sd = 0.5)

data_input <- list('n1' = length(y1), 'y1' = y1,
                  'n2' = length(y2), 'y2' = y2)

priors_parm <- list('mu_mu1' = 0, 'tau_mu1' = 0.000001,
                  'mu_mu2' = 0, 'tau_mu2' = 0.000001,
                  'a_tau' = 0.001, 'b_tau' = 0.001)

fit1 <- fitdist(data = y1, distr = 'norm', method = 'mme')

fit2 <- fitdist(data = y2, distr = 'norm', method = 'mme')

parms <- c('mu1', 'mu2', 'tau', 'variance', 'delta_mu_1x2')

start_kick <- function() list('mu1' = fit1$estimate[1], 'tau' = 1/(fit1$estimate[2]^2),
                             'mu2' = fit2$estimate[1])

fit <- BioBayes.fit(bug_model = normal_homo_diff_model,
                  database = data_input,
                  priors = priors_parm,
                  parameters = parms,
                  initials = start_kick,
                  iter = 10000,
                  burn = 1000,
                  jump = 1,
                  Round = 3,
```

```

seed = 123,
title = 'Bayesian Inferential Results',
save_html = NULL,
render = TRUE)

## End(Not run)

```

---

normal_model	<i>Model for normal distribution.</i>
--------------	---------------------------------------

---

## Description

Bayesian model to fit

$Y \sim \text{Normal}(\mu, \tau)$ , such that  $\mu$  is real and  $\tau > 0$ ,

with priors distributions:

$\mu \sim \text{Normal}(a, b)$ , with  $a$  real and  $b > 0$ ;

$\tau \sim \text{Gamma}(c, d)$ , with  $c > 0$  and  $d > 0$ .

## Usage

```
normal_model(n, tau)
```

## Arguments

n	length of vector y
tau	initial kick for tau

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\mu$ ” and “ $\tau$ ” as specified in the above function.

## References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

**Examples**

```
## Not run:
set.seed(123)

y <- rnorm(n = 50, mean = 0, sd = 1)

data_input <- list('n' = length(y), 'y' = y)

priors_parm <- list('a' = 0, 'b' = 0.000001, 'c' = 0.001, 'd' = 0.001)

parms <- c("mu", "tau", "variance")

start_kick <- function() list(mu = mean(y), tau = 1)

fit <- BioBayes.fit(bug_model = normal_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)
```

---

normal_paired_model	<i>Model for difference of means between two dependent normal populations (paired).</i>
---------------------	---

---

**Description**

Bayesian model for adjusting the difference of means and variances between two dependent normal populations (paired).

$diff = Y_1 - Y_2 \sim Normal(\mu, \tau)$ , with  $Y_1, Y_2$  and  $\mu$  reals and  $\tau > 0$ ,

with priors distributions:

$\mu \sim Normal(a, b)$  with  $a$  real and  $b > 0$ .

$\tau \sim Gamma(c, d)$  with  $c > 0$  and  $d > 0$ .

**Usage**

```
normal_paired_model(n, tau)
```

**Arguments**

n	length of vector y1 or y2
tau	initial kick for tau

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\mu$ ” and “ $\tau$ ” as specified in the above function.

## References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y1 <- rnorm(10, 10, 1)

y2 <- rnorm(10, 8, 0.8)

diff <- y2-y1

data_input <- list('diff' = diff, 'n' = length(diff))

priors_parm <- list('a' = 0, 'b' = 0.000001,
                   'c' = 0.001, 'd' = 0.001)

fit_freq <- fitdist(data = diff, distr = 'norm')

parms <- c('mu', 'tau', 'variance')

start_kick <- function() list('mu' = fit_freq$estimate[1], 'tau' = 1/(fit_freq$estimate[2]^2))

fit <- BioBayes.fit(bug_model = normal_paired_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render= TRUE)

## End(Not run)
```

---

poisson_diff_model	<i>Model for difference of means between two independent poisson populations.</i>
--------------------	---

---

### Description

Bayesian model for adjusting the difference of means between two independent poisson populations.

$Y_k \sim \text{Poisson}(\lambda_k)$ , with integers  $Y_k$ ,  $\lambda_k > 0$  and  $k = 1, 2$

with priors distributions:

$\lambda_1 \sim \text{Gamma}(a_1, b_1)$ ,  $a_1 > 0$ ,  $b_1 > 0$ ;

$\lambda_2 \sim \text{Gamma}(a_2, b_2)$ ,  $a_2 > 0$ ,  $b_2 > 0$ .

### Usage

```
poisson_diff_model(n1, n2, lambda1, lambda2)
```

### Arguments

n1	length of vector y1
n2	length of vector y2
lambda1	initial kick for lambda of y1
lambda2	initial kick for lambda of y2

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\lambda_1$ ” and “ $\lambda_2$ ” as specified in the above function.

### References

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

y1 <- c(45, 26, 9, 3, 32)

y2 <- c(46, 1, 94, 13, 22)

data_input <- list('n1'= length(y1), 'y1'= y1, 'n2'= length(y2), 'y2'= y2)
```

```

priors_parm <- list('a1' = 0.001, 'b1' = 0.001, 'a2' = 0.001, 'b2' = 0.001)

parms <- c('lambda1', 'lambda2', 'delta_mu_1x2')

start_kick <- function() list('lambda1' = mean(y1), 'lambda2' = mean(y2))

fit <- BioBayes.fit(bug_model = poisson_diff_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)

```

---

poisson_model	<i>Model for poisson distribution.</i>
---------------	--

---

## Description

Bayesian model to fit

$Y \sim \text{Poisson}(\lambda)$ , such that  $Y$  are positive integer numbers, and  $\lambda > 0$ ,

with prior distribution:

$\lambda \sim \text{Gamma}(a, b)$  with  $a > 0$  and  $b > 0$ .

## Usage

```
poisson_model(n, lambda)
```

## Arguments

n	length of vector y
lambda	initial kick for lambda of y

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\lambda$ ” as specified in the above function.

## References

- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
set.seed(123)

y <- rpois(n = 50, lambda = 3)

data_input <- list('n' = length(y), 'y' = y)

priors_parm <- list('a' = 0.001, 'b' = 0.001)

parms <- c('lambda', 'variance')

start_kick <- function() list('lambda' = mean(y))

fit <- BioBayes.fit(bug_model = poisson_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 10000,
                   burn = 1000,
                   jump = 1,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

## End(Not run)
```

---

quadratic\_control\_reg\_model

*Quadratic versus control regression model.*

---

## Description

Bayesian quadratic versus control regression model with normal erros to fit:

$$Y \sim \text{Normal}(\mu, \tau)$$

such that  $\mu$  is the linear predictor, where

$$\mu = \beta_0 + \beta_1 x + \beta_2 x^2 \text{ and}$$

$\tau > 0$ . Also,

$x$  is the explanatory variable (independent), real and

$\beta_k$  is the vector of regression coefficients,  $k = 0, 1, 2$ .

Priors distributions for the “ $\beta_k$ ”:

$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ ;

$\tau \sim \text{Gamma}(a, b)$ , with  $a, b > 0$ .

Coordinates of the critical points of  $x$  and  $y$ :

$CP.x = -\beta_1/2\beta_2$

$CP.y = -(\beta_1^2 - 4\beta_0\beta_2)/4\beta_2$

$\Delta = CP.y - \mu_0$ .

Control data specifications:

$Y0 \sim \text{Normal}(\mu_0, \tau_0)$

with priors

$\mu_0 \sim \text{Normal}(\mu\mu_0, \tau\mu_0)$

$\tau_0 \sim \text{Gamma}(\mu\tau_0, \tau\tau_0)$ .

## Usage

```
quadratic_control_reg_model(n0, tau0, n, beta0, beta1, beta2, x, tau, mu0)
```

## Arguments

n0	length of vector y0
tau0	initial kick for tau0
n	length of vector y
beta0	initial kick for beta0
beta1	initial kick for beta1
beta2	initial kick for beta2
x	vector of independent variable
tau	initial kick for tau
mu0	initial kick for mu0

## Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” and “ $\tau$ ” as specified in the above function.

## References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.



PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

## Examples

```
## Not run:
#simulated values

set.seed(123)
trat = 5                                # number of levels (treatment)
r = 10                                  # number of replicates of Y
n = trat*r                              # sample size of Y
beta0 = 0.7                             # intercept
beta1 = 1.5                             # linear coefficient
beta2 = -0.4                             # quadratic coefficient
sigma = 0.5                             # standart deviation for error
x = rep(1:trat, each=r)                 # regressive variable
y = numeric()
e = numeric()
for(i in 1:n)
{
  e[i] = rnorm(1)                        # Normal error
  y[i] = beta0 + beta1*x[i] + beta2*(x[i]^2) + sigma*e[i]
}
data = cbind(x,y)

#Options for simulate control subjects with fixed mean
mu.0 = 1.6                              #hypothetically
y0 = rnorm(r, mu.0, sigma)
mu.0 = mean(y0)

data_input <- list('n' = length(y), 'y' = y, 'x' = x, 'n0' = length(y0), 'y0' = y0)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001, 'a_tau' = 0.001, 'b_tau' = 0.001,
  'mu_mu0' = 0, 'tau_mu0' = 0.000001, 'a_tau0' = 0.001, 'b_tau0' = 0.001)

parms <- c('beta0', 'beta1', 'beta2', 'tau', 'variance', 'variance0', 'PC.x', 'PC.y', 'delta',
  'mu0', 'tau0')

fit_freq <- lm(y~x+I(x^2))

start_kick <- function() list('beta0' = fit_freq$coef[1], 'beta1' = fit_freq$coef[2],
  'beta2' = fit_freq$coef[3], 'tau' = 1,
  'mu0' = mu.0, 'tau0' = 1)

fit <- BioBayes.fit(bug_model = quadratic_control_reg_model,
  database = data_input,
  priors = priors_parm,
  parameters = parms,
  initials = start_kick,
  iter = 100000,
  burn = 10000,
  jump = 100,
  Round = 3,
  seed = 123,
```

```

        title = 'Bayesian Inferential Results',
        save_html = NULL,
        render = TRUE)

plot(x, y, xlim = c(min(c(x0, x)), max(c(x0, x))), pch = 16, bty = "n")
x0 = rep(0, length(y0))
points(x0, y0, col = 1, pch = 4)                                #controls
pars = apply(fit$MCMC[[1]][,3:5], 2, median)
x0.aux = seq(min(x), max(x), 0.01)
y.bayes = pars[1] + pars[2]*x0.aux + pars[3]*x0.aux^2
lines(x0.aux, y.bayes, col = "red", lwd = 2)
y.freq = fit_freq$coef[1] + fit_freq$coef[2]*x0.aux + fit_freq$coef[3]*x0.aux^2
lines(x0.aux, y.freq, col = "blue", lwd = 2, lty = 2)
abline(h = median(fit$MCMC[[1]][,7]), lty = 2)
points(x0.aux[which.max(y.bayes)], y.bayes[which.max(y.bayes)], col = "red", pch = 8,
       cex = 2, lwd = 2)
legend("topright", legend = c("Bayesian adjustment", "Frequentist adjustment",
                             paste0("mu0 = ", round(median(fit$MCMC[[1]][,7]),3)),
                             paste0("PC: (", round(x0.aux[which.max(y.bayes)], 3), "; ",
                                     round(y.bayes[which.max(y.bayes)], 3), ")")),
      lwd = c(2, 2, 1, NA), lty = c(1, 2, 2, NA), pch = c(NA, NA, NA, 8),
      col = c("red", "blue", "black", "red"), bty = "n")

## End(Not run)

```

---

quadratic\_reg\_model      *Quadratic regression model.*


---

## Description

Bayesian quadratic regression model with normal erros to fit:

$$Y \sim Normal(\mu, \tau)$$

such that  $\mu$  is the linear predictor, where

$$\mu = \beta_0 + \beta_1 x + \beta_2 x^2 \text{ and}$$

$\tau > 0$ . Also,

$x$  is the explanatory variable (independent), real and

$\beta_k$  is the vector of regression coefficients,  $k = 0, 1, 2$ .

Priors distributions for the " $\beta_k$ ":

$$\beta_k \sim Normal(\mu\beta_k, \tau\beta_k), \text{ with } \mu\beta_k \text{ real and } \tau\beta_k > 0;$$

$$\tau \sim Gamma(a, b), \text{ with } a, b > 0.$$

Coordinates of the critical points of  $x$  and  $y$ :

$$CP.x = -\beta_1/2\beta_2$$

$$CP.y = -(\beta_1^2 - 4\beta_0\beta_2)/4\beta_2$$

## Usage

```
quadratic_reg_model(n, beta0, beta1, beta2, x, x0, tau)
```

### Arguments

n	length of vector y
beta0	initial kick for beta0
beta1	initial kick for beta1
beta2	initial kick for beta2
x	vector of independent variable
x0	point to be predicted
tau	initial kick for tau

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” and “ $\tau$ ” as specified in the above function.

### References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- MONTGOMERY, D. C. Design and Analysis of Experiments. Wiley, 8th Edition, 2012.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
#simulated values

set.seed(123)

trat = 5                                # number of levels (treatment)
r = 10                                  # number of replicates of Y
n = trat*r                              # sample size of Y
beta0 = 0.7                             # intercept
beta1 = 1.5                             # linear coefficient
beta2 = -0.4                             # quadratic coefficient
sigma = 0.5                             # standart deviation for error
x = as.numeric(rep(1:trat, each=r))     # regressive variable
y = numeric()
e = numeric()
for(i in 1:n)
{
  e[i] = rnorm(1)                        # Normal error
  y[i] = beta0 + beta1*x[i] + beta2*(x[i]^2) + sigma*e[i]
}

x0 <- 2.5 #point to be predicted (y.pred)
```

```

data_input <- list('n' = length(y), 'y' = y, 'x' = x, 'x0' = x0)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001, 'a_tau' = 0.001, 'b_tau' = 0.001)

parms <- c('beta0', 'beta1', 'beta2', 'tau', 'variance', 'y.pred', 'CP.x', 'CP.y')

fit_freq <- lm(y~x+I(x^2))

start_kick <- function() list('beta0' = fit_freq$coef[1], 'beta1' = fit_freq$coef[2],
                             'beta2' = fit_freq$coef[3], 'tau' = 1)

fit <- BioBayes.fit(bug_model = quadratic_reg_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 100000,
                   burn = 10000,
                   jump = 100,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y, pch = 16, ylim = c(-2, 3), xlim = c(1,5), bty = "n")
x0 = seq(0, 6, by = 0.01)
pars = apply(X = fit$MCMC[[1]][, 3:5], FUN = median, MARGIN = 2)
y.est = pars[1] + pars[2]*x0 + pars[3]*x0^2
lines(sort(x0), y.est, type = "l", col = "red", lwd = 2)
y.freq = fit_freq$coef[1] + fit_freq$coef[2]*x0 + fit_freq$coef[3]*x0^2
lines(sort(x0), y.freq, type = "l", col = "blue", lwd = 2, lty = 2)
legend("topright", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

vonbertalanffy\_normal\_model

*Von Bertalanffy's nonlinear growth model.*

---

## Description

Von Bertalanffy's nonlinear growth bayesian model with normal errors that adjusts

$$Y_i \sim \text{Normal}(FNL, \tau)$$

$$\text{such that } FNL = \beta_1(1 - \beta_2 \exp(-\beta_3 x)^3)$$

where  $x$  is the vector of the explanatory variable (usually observations in time) of the model and

$\beta_k$  is the vector of beta coefficients,  $k = 1, 2, 3$  such that

$\beta_1$  represents the asymptotic value of the response variable,  $\beta_2$  is a constant that indicates the proportion of the asymptotic value to be gained and  $\beta_3$  is the growth rate of the response variable that determines the efficiency of growth.

Priors distributions:

$\beta_k \sim \text{Normal}(\mu\beta_k, \tau\beta_k)$ , with  $\mu\beta_k$  real and  $\tau\beta_k > 0$ ;

$\tau \sim \text{Gamma}(a, b)$ , with  $a > 0$  and  $b > 0$ .

### Usage

```
vonbertalanffy_normal_model(n, beta1, beta2, beta3, x, tau)
```

### Arguments

n	length of vector y
beta1	inicial kick for beta1
beta2	inicial kick for beta2
beta3	inicial kick for beta3
x	vector of independent variable
tau	inicial kick for tau

### Details

To change the parameters of the priors distributions, is just change the values in object “ $\beta_k$ ” and “ $\tau$ ” as specified in the above function.

### References

CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.

DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2<sup>a</sup> Ed., 1989.

McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.

NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.

PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.

RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.

ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.

### Examples

```
## Not run:
set.seed(123)

x <- c(9,12,14,21,28,35,42)

y <- c(1,45,82,200,205,221,223)

data_input <- list('n' = length(y), 'y' = y, 'x' = x)

priors_parm <- list('mu_beta' = 0, 'tau_beta' = 0.000001,
                  'mu_tau' = 0.001, 'tau_tau' = 0.001)

parms <- c('beta1', 'beta2', 'beta3', 'tau', 'variance')
```

```

bertalanffy = function(pars, x)
{
  beta1 = pars[1]
  beta2 = pars[2]
  beta3 = pars[3]
  y = beta1*(1 - beta2*exp(-beta3*x))^3
  y
}

fit_freq <- nls(y ~ bertalanffy(pars, x), start = list(pars = c(220, 9, 0.25)))

start_kick <- function() c('beta1' = 220,
                           'beta2' = 9,
                           'beta3' = 0.25,
                           'tau' = 1)

fit <- BioBayes.fit(bug_model = vonbertalanffy_normal_model,
                   database = data_input,
                   priors = priors_parm,
                   parameters = parms,
                   initials = start_kick,
                   iter = 100000,
                   burn = 10000,
                   jump = 10,
                   Round = 3,
                   seed = 123,
                   title = 'Bayesian Inferential Results',
                   save_html = NULL,
                   render = TRUE)

plot(x, y, bty = "n", pch = 16)
pars = apply(X = fit$MCMC[[1]][, 1:3], FUN = median, MARGIN = 2)
y.est = bertalanffy(pars = pars, sort(x))
lines(sort(x), y.est, type = "l", col = "red", lwd = 2)
pars_freq = c(220.27, 8.79, 0.251)
y.freq = bertalanffy(pars = pars_freq, sort(x))
lines(sort(x), y.freq, type = "l", col = "blue", lwd = 2, lty = 2)
legend("topleft", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = c(1, 2), lwd = 2, bty = "n")

## End(Not run)

```

---

wood\_normal\_model

Wood's lactation curve model.

---

## Description

Wood's lactation curve bayesian model with normal erros that adjusts

$Y_i \sim Normal(\mu, \tau)$  com  $Y$  representing a milk production in  $t$  lactation days,

where  $FNL = x^b \exp(a + cx)$

such that  $x$  is the vector of the explanatory variable (lactation day) of model,

$a$  which represents the initial milk production,  
 $b$  which indicates the rate of production increase up to the peak and  
 $c$  which is the rate of decline in production after the peak (persistence factor).

Priors distributions:

$a \sim \text{Gamma}(a_1, a_2)$ , with  $a_1$  real and  $a_2 > 0$ ;

$b \sim \text{Uniform}(b_1, b_2)$ , with  $b_1$  real and  $b_2 > 0$ ;

$c \sim \text{Uniform}(c_1, c_2)$ , with  $c_1$  real and  $c_2 > 0$ ;

$\tau \sim \text{Gamma}(t_1, t_2)$ , with  $t_1$  real and  $t_2 > 0$ .

The model also obtains the values for:

$pp$  : production in peak

$dpp$  : peak production day

$s$  : persistence

### Usage

```
wood_normal_model(n, x, a, b, tau, x0)
```

### Arguments

$n$	length of vector $y$
$x$	vector of independent variable
$a$	initial kick for the initial milk production
$b$	initial kick for the rate of production
$\tau$	initial kick for $\tau$
$x_0$	value to be predicted

### Details

To change the parameters of the priors distributions, is just change the values in object “ $a, b, c$ ” and “ $\tau$ ” as specified in the above function.

### References

- CONGDON, P. Bayesian statistical modelling. John Wiley & Sons, 2007.
- DeGROOT, M. H. Probability and Statistics. Addison Wesley, 2ª Ed., 1989.
- McCULLOGH, C. E.; SEARLE, S. R. Generalized, linear and mixed model. John Wiley, 2011.
- NTZOUFRAS, I. Bayesian modeling using WinBUGS. John Wiley & Sons, 2011.
- PAULINO, C. D., TURKMAN, M. A. A., MURTEIRA, B. Estatística Bayesiana. Fundação Calouste Gulbenkian, Lisboa, 2003.
- RATKOWSKY, D.A. Nonlinear regression modeling: a unified practical approach. New York: Marcel Dekker, 1983. 276p.
- ROSSI, R. M. Introdução aos métodos Bayesianos na análise de dados zootécnicos com uso do WinBUGS e R. Eduem: Maringá, 2011.
- WOOD, P.D.P. Breed variation in the shape of the lactation curve of cattle and their implications for efficiency. Animal Production, v.34, p.133-41, 1980.

**Examples**

```

## Not run:
set.seed(123)

x <- c(15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150,
      160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300)

y <- c(3.5, 4, 4.5, 5.5, 7, 7.3, 6.7, 6.7, 6.6, 5.9, 5.6, 5.5, 5, 5, 4.7, 4.7,
      4.5, 4.4, 4, 4, 3.9, 3.8, 3.5, 3.3, 3.2, 3, 3, 2.9, 2.8, 2.5)

x0 <- 125 #value to be predicted (y.pred)

data_input <- list('n' = length(y), 'y' = y, 'x' = x, 'x0' = x0)

priors_parm <- list('a1' = 1, 'a2' = 1,
                  'b1' = 0, 'b2' = 1,
                  'c1' = -1, 'c2' = 0,
                  't1' = 0.001, 't2' = 0.001)

parms <- c('a', 'b', 'c', 'tau', 'variance', 'pp', 'dpp', 's', 'y.pred')

#Generating values for the initial kicks
a0 = rgamma(1, 1) # Gamma(1,1)
b0 = runif(1) # U(0,1)
c0 = runif(1, -1, 0) # U(-1,0)

start_kick <- function() c('a' = a0,
                          'b' = b0,
                          'c' = c0,
                          'tau' = 1)

fit <- BioBayes.fit(bug_model = wood_normal_model,
                  database = data_input,
                  priors = priors_parm,
                  parameters = parms,
                  initials = start_kick,
                  iter = 500000,
                  burn = 50000,
                  jump = 500,
                  Round = 3,
                  seed = 123,
                  title= 'Bayesian Inferential Results',
                  save_html = NULL,
                  render = TRUE)

plot(x, y, bty = "n", pch = 16)
pars = apply(X = fit$MCMC[[1]][, 1:3], FUN = median, MARGIN = 2)
x = sort(x)
curve(x^pars[2]*exp(pars[1] + pars[3]*x), lwd = 2, add = T, col = "red")
pars0 = c(0.101, 0.5319, -0.007581)
curve(x^pars0[2]*exp(pars0[1] + pars0[3]*x), lwd = 2, lty = 2, add = T, col = "blue")
legend("topright", legend = c("Bayesian adjustment", "Frequentist adjustment"),
      col = c("red", "blue"), lty = 1, lwd = 2, bty = "n")

## End(Not run)

```





# Index

anova\_hetero\_model, 2, 38  
anova\_homo\_model, 3, 38  
  
beta\_diff\_model, 5, 39  
beta\_model, 7, 39  
bin\_multi\_reg\_model\_cauchy, 11, 39  
bin\_multi\_reg\_model\_cloglog, 13, 39  
bin\_multi\_reg\_model\_logit, 16, 39  
bin\_multi\_reg\_model\_loglog, 19, 39  
bin\_multi\_reg\_model\_probit, 21, 39  
bin\_reg\_model\_cauchy\_DR, 24, 39  
bin\_reg\_model\_cloglog\_DR, 27, 39  
bin\_reg\_model\_logit, 29, 39  
bin\_reg\_model\_logit\_DR, 31, 39  
bin\_reg\_model\_loglog\_DR, 34, 39  
bin\_reg\_model\_probit\_DR, 36, 39  
binomial\_diff\_model, 8, 39  
binomial\_model, 10, 39  
BioBayes.fit, 38  
brody\_normal\_model, 39, 40  
  
exponential\_diff\_model, 39, 42  
exponential\_model, 39, 44  
  
gamma\_diff\_model, 39, 45  
gamma\_model, 39, 47  
gompertz\_normal\_model, 39, 48  
  
linear\_multi\_reg\_model, 39, 50  
linear\_multi\_reg\_model\_poisson, 39, 52  
linear\_reg\_model, 39, 54  
linear\_reg\_model\_poisson, 39, 56  
logistic\_normal\_model, 39, 57  
  
normal\_hetero\_diff\_model, 39, 59  
normal\_homo\_diff\_model, 39, 61  
normal\_model, 39, 63  
normal\_paired\_model, 39, 64  
  
poisson\_diff\_model, 39, 66  
poisson\_model, 39, 67  
  
quadratic\_control\_reg\_model, 39, 68  
quadratic\_reg\_model, 39, 71  
  
vonbertalanffy\_normal\_model, 39, 73  
wood\_normal\_model, 39, 75