Universidade de Brasília
Departamento de Estatística

Distribuições logísticas generalizadas: Pacote para o R

Eduardo Ochetski Hellas

Projeto apresentado para obtenção do título de Bacharel em Estatística.

Brasília
2017

Eduardo Ochetski Hellas

Distribuições logísticas generalizadas: Pacote para o R

Orientador:
Prof. Dr. **Eduardo Monteiro de Castro Gomes**

Projeto apresentado para obtenção do título de Bacharel em Estatística.

Brasília
2017

AGRADECIMENTOS

Aos meus pais Júlio Hellas† e Katia Hellas, aos meus irmãos Henrique e William e cunhadas Clarice e Ana Flávia por serem o melhor incentivo e apoio que poderia ter.

A minha companheira e amiga Luciana por me acompanhar nessa jornada.

Ao professor Dr. Eduardo Monteiro de Castro Gomes pela atenção e orientação sem precedentes.

Aos docentes e colegas do Departamento de Estatística da UnB, em especial a Danilo Ramos, Matheus Ferreira, Pedro Brom, Allan Vieira e Caio Balena.

Aos amigos que deixaram essa carta para mim, Giovanna, Gabriel e Alberto.

Por fim, aos amigos da Taverna, Bruno, Juliano, Tauãnara, Jordy e a todos os que contribuíram de alguma forma na conclusão desta etapa.

"Why is that when one man builds a wall,
the next man immediately needs to know
what's on the other side.
(George R.R. Martin, A Game of Thrones)

RESUMO

As distribuições logísticas generalizas são modelos estatísticos que dependem de suporte computacional para serem utilizadas de forma eficiente. Neste trabalho apresentaremos uma proposta de distribuição logística generalizada que é inversível analiticamente porém, seus resultados envolvem somas infinitas. Foram utilizados métodos numéricos para aproximações e estimação de parâmetros e apresentamos também o uso de ferramentas visuais para auxiliar a utilização destes métodos. Tudo será apresentado como guia de criação de pacotes para o programa R de forma que possa ser utilizado futuramente para quem tenha interesse.

Palavras-chave: Distribuição Rathie-Swamee, distribuição logística, R, Pacote

ABSTRACT

Generalized logistic distributions are statistical models that rely on computational support to be used efficiently. This paper presents a generalized logistic distribution proposal which is analytically invertible but, its results involve infinite sums. Numerical methods were used for approximations and estimation of parameters, as well as the use of visual tools to aid the use of these methods. Everything will be presented as a guide to the creation of packages for the program R in order to be used in the future for those who are interested.

Keywords: Rathie-Swamee distribution, Logistic distribution, R, Pacakge

SUMÁRIO

1 INTRODUÇÃO

Os problemas existentes que envolvem distribuições logísticas generalizadas sofrem, normalmente, da necessidade de processos computacionais intensivos para sua resolução. Além disso, existem poucos pacotes em R voltados para estas distribuições e, seus recursos são limitados ao se tratar de funções logísticas de generalizações mais complexas.

Será apresentada uma implementação computacional de uma distribuição logística generalizada proposta por Rathie e Swamee (2006) como também uma implementação assimétrica desta mesma distribuição utilizando um método proposto por Azzalini (1985). Esta implementação é um roteiro de criação de pacotes em R desde a ambientação inicial até a publicação no CRAN[1] (repositório oficial de bibliotecas para o R).

O modelo Rathie-Swamee é uma distribuição logística generalizada que permite trabalhar com dados bimodais e multimodais de forma flexível, sendo útil em diversas situações de multimodalidade. Será apresentado um conjunto de dados que é possível ser modelado segundo a distribuição Rathie-Swamee para explicar seus efeitos.

Este trabalho está organizado de forma que há primeiro a seção 2, para revisão de literatura. Em seguida, a seção 3 apresenta a metodologia utilizada para alcançar o objetivo desejado, que é a criação de um pacote para o software R. Por último, a seção 4 mostra aplicações e usos para as ferramentas criadas em situações reais e simuladas.

[1] https://cran.r-project.org/

2 REVISÃO DE LITERATURA

2.1 Definição da distribuição

O pacote que é construído neste trabalho utiliza uma distribuição proposta por Rathie e Samwee (2006), sua função distribuição de probabilidade (F.D.P) é definida da maneira a seguir:

$$f(z) = \frac{[a + b(1+p)|z-\mu|^p]e^{-(z-\mu)(a+b|z-\mu|^p)}}{(e^{-z(a+b|z-\mu|^p)}+1)^2}, z, \mu \in \mathbb{R}, a, b, p \geq 0 \quad (1)$$

Sua função distribuição acumulada (F.D.A) é bem definida, dada por:

$$F(z) = \frac{1}{e^{-(z-\mu)(a+b|z-\mu|^p)}+1}, z, \mu \in \mathbb{R}, a, b, p \geq 0 \quad (2)$$

Apesar de existir uma função inversa $F^{-1}(z)$ fechada para o cálculo dos quantis, esta envolve somas infinitas, que não são facilmente operadas computacionalmente. Assim, mais adiante será mostrada uma forma computacional utilizada para fazer a aproximação da distribuição.

Além disso, a distribuição possui restrições para os parâmetros a, b e p:

1. Os parâmetros a, b e p não podem ser iguais a 0 simultaneamente ou a distribuição não será definida;

2. No caso do parâmetro p ser igual a 1 o parâmetro a ou o parâmetro b deve ser igual a zero, ou existirá um problema de identificabilidade no modelo.

3. O modelo foi escrito utilizando o parâmetro de locação μ porém não foi possível utilizar o parâmetro de escala pois ele é não identificável.

2.1.1 Análise da distribuição

A distribuição proposta possui alguns casos particulares como mostraremos utilizando a ferramenta criada.

A figura 1 mostra uma das possíveis formas da distribuição logística generalizada apresentada, que é multimodal. Utilizando os parâmetros $a = \sqrt{\frac{2}{\pi}}$, $b = 0.5$ e $p = 2$ a distribuição possui a média 0 e a variância próxima a 1.

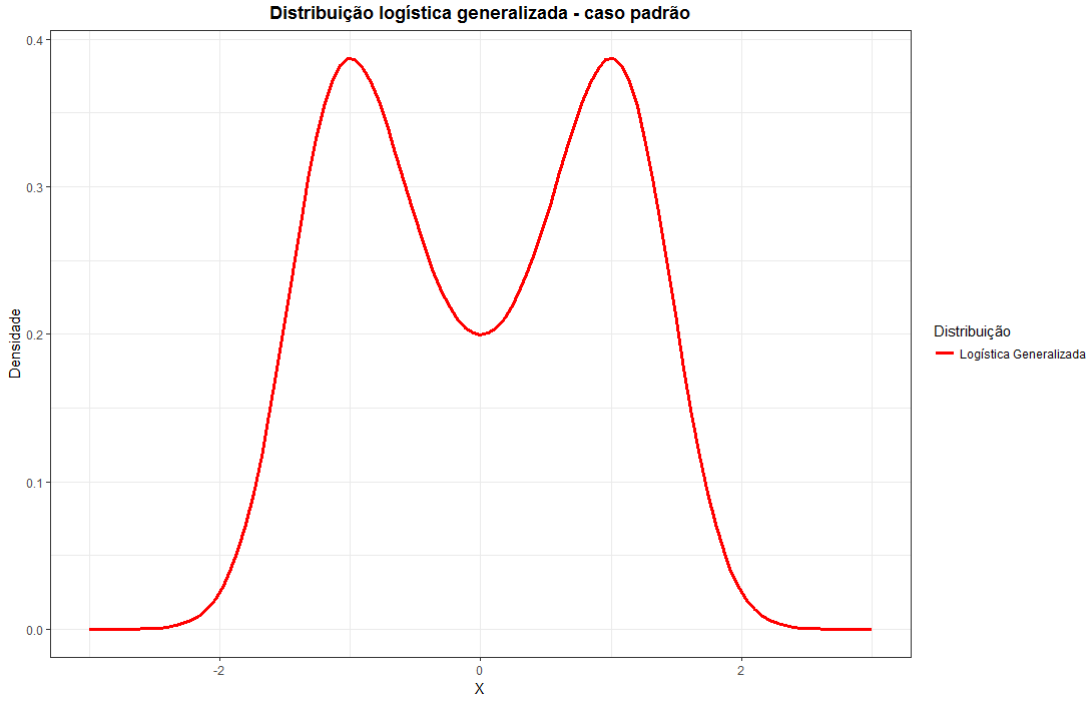

Figura 1 – Distribuição logística - formato genérico

2.1.2 Casos particulares

Ao manipularmos os parâmetros conseguimos fazer duas aproximações interessantes, para as distribuições Normal Padrão e logística. A aproximação para a distribuição logística era esperada por ser uma das propriedades necessárias para uma distribuição generalizada.

2.1.2.1 Distribuição Normal Padrão

Utilizando os parâmetros $a = 1.595768$, $b = 0.0727$ e $p = 1.962$ a distribuição se aproxima de uma distribuição Normal (0,1) como se vê na figura 3.

Figura 2 – Aproximação normal

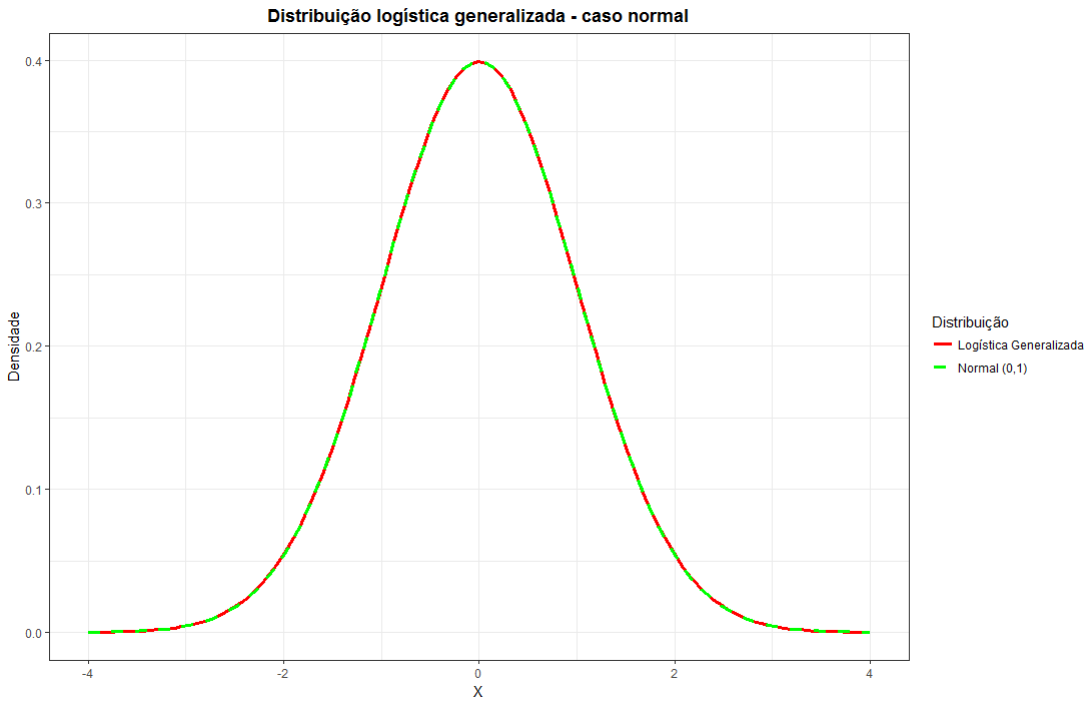

Figura 3 – Aproximação normal

2.1.2.2 Distribuição Logística

Ao fazer o parâmetro $b = 0$, a distribuição se aproxima de uma distribuição logística onde o parâmetro de escala é dado por $\frac{1}{a}$, como mostram a equação 3 e a figura 4.

$$F(z) = \frac{1}{e^{-(z-\mu)a} + 1}, z \in \mathbb{R}, a > 0. \tag{3}$$

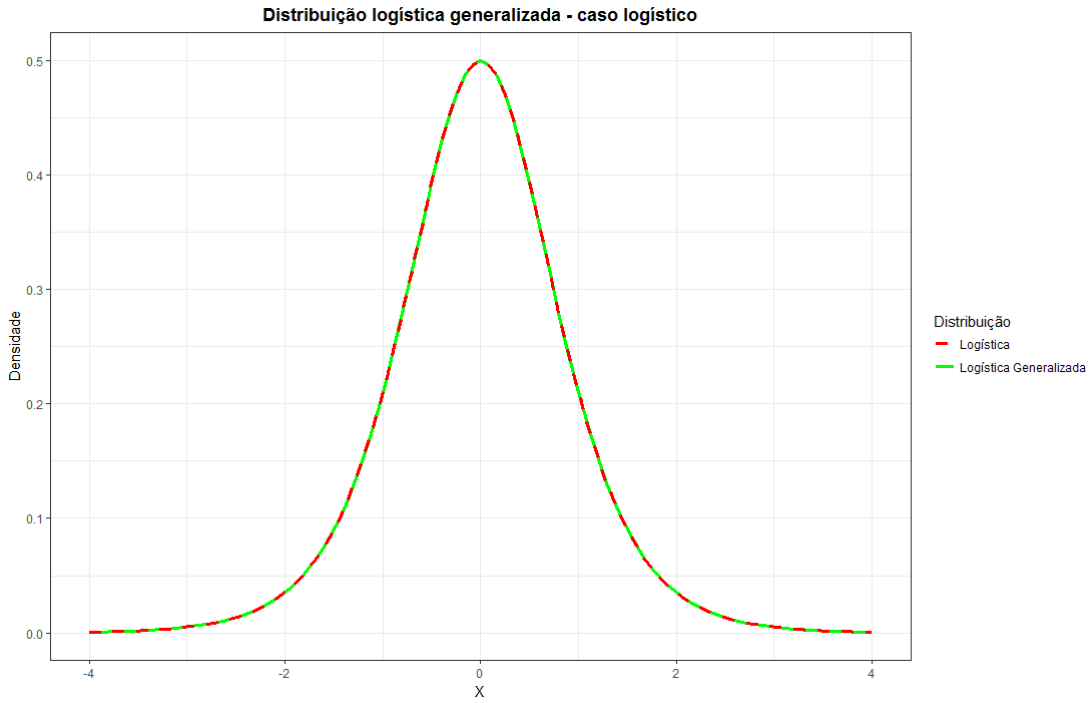

Figura 4 – Aproximação logística, $a = 2$ e $b = 0$

Essa aproximação também pode ser obtida no caso dos parâmetros serem $a = 0$, $b > 0$ e $p = 0$. Agora, o parâmetro de escala é dado por $\frac{1}{b}$ como mostra a equação 4.

$$F(z) = \frac{1}{e^{-(z-\mu)b} + 1}, z \in \mathbb{R}, b > 0. \tag{4}$$

2.1.3 Casos de interesse

A distribuição também possui diversas situações onde criam um ponto de descontinuidade, como apresentados a seguir na figura 5 quando $a = 1$, $b = 1$ e $p = 1$ e na figura 6 quando $a = 0$, $b = 1$ e $p = 1$, essa descontinuidade existe para diversos valores de a e b desde que o parâmetro p não mude muito. A distribuição associada está escrita na equação 5.

$$F(z) = \frac{1}{e^{-(z-\mu)(a+b|z-\mu|)} + 1}, z \in \mathbb{R}, b > 0. \tag{5}$$

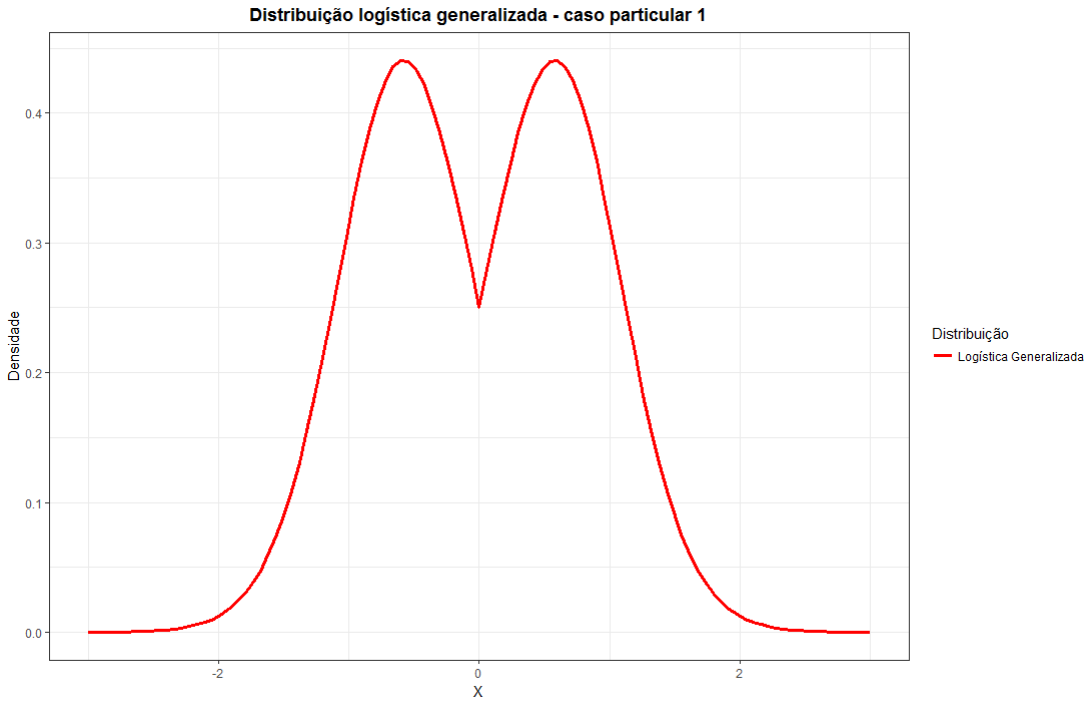

Figura 5 – Caso descontínuo 1

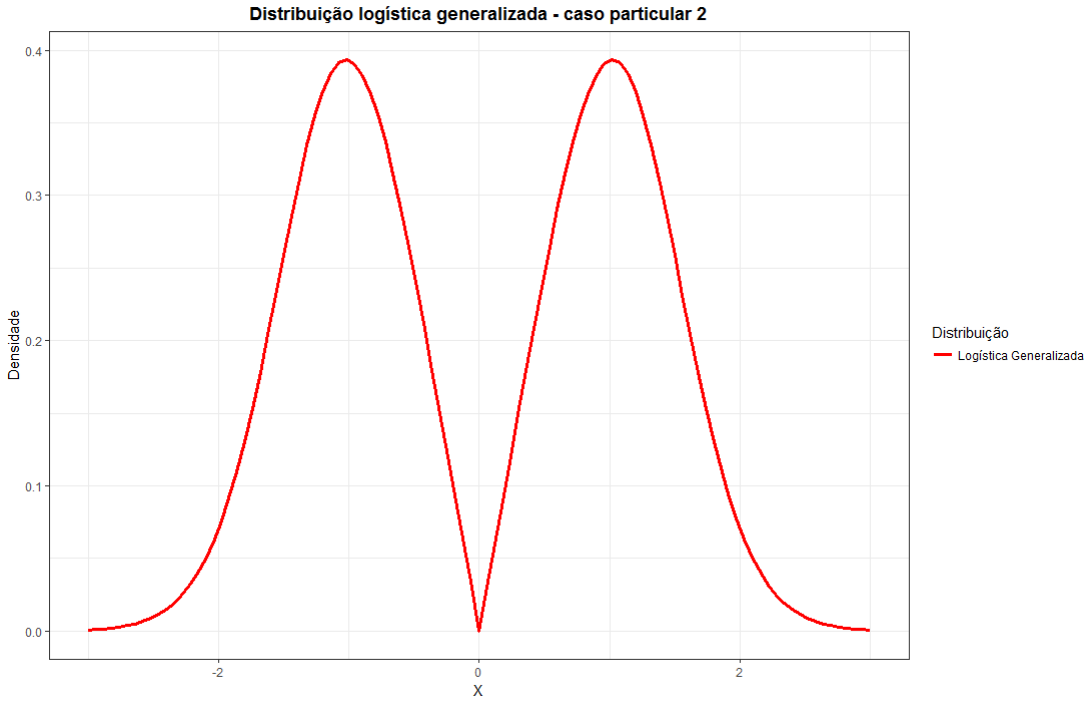

Figura 6 – Caso descontínuo 2

2.1.4 Efeitos dos parâmetros

Ao se tratar de um modelo de probabilidade multiparamétrico, diferentes combinações entre os parâmetros geram diversos modelos de distribuições que podem ser identificáveis ou não, como citado anteriormente. Os efeitos dos parâmetros observados por Monteiro[2] forem reproduzidos e estão representados nas figuras 7,8 e 9. Cada uma das figuras possuem os efeitos isolados dos parâmetros a, b e p respectivamente, com os valores crescentes do parâmetro de interesse no sentido da esquerda para a direita com todos os outros mantidos constantes.

A Figura 7 ilustra os efeitos que o parâmetro a na função de densidade de probabilidade. Nota-se que o efeito de a refere-se principalmente à modalidade da função, quando se aproxima de zero a função de densidade se aproxima de zero, causando a bimodalidade. Quando o valor de a aumenta causa um efeito de unimodalidade na distribuição.

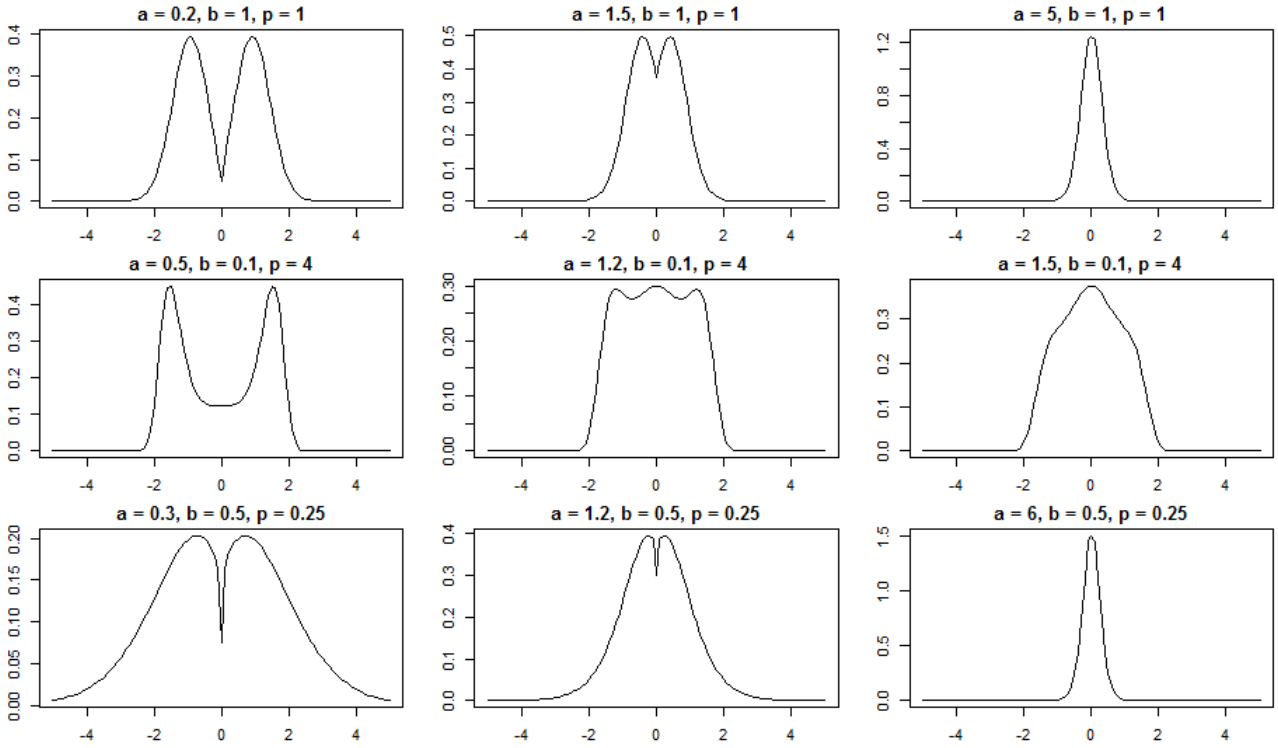

Figura 7 – Efeito do parâmetro a na distribuição

A Figura 8 ilustra os efeitos que o parâmetro b na função de densidade de

[2]Monteiro, E. Modelo Rathie-Swamee: aplicações e extensão para modelo de regressão, 2013

probabilidade. O parâmetro é responsável, principalmente, pelo efeito da variabilidade da distribuição. Ao crescer, o parâmetro b aproxima as concavidades da distribuição, causando queda na variabilidade.

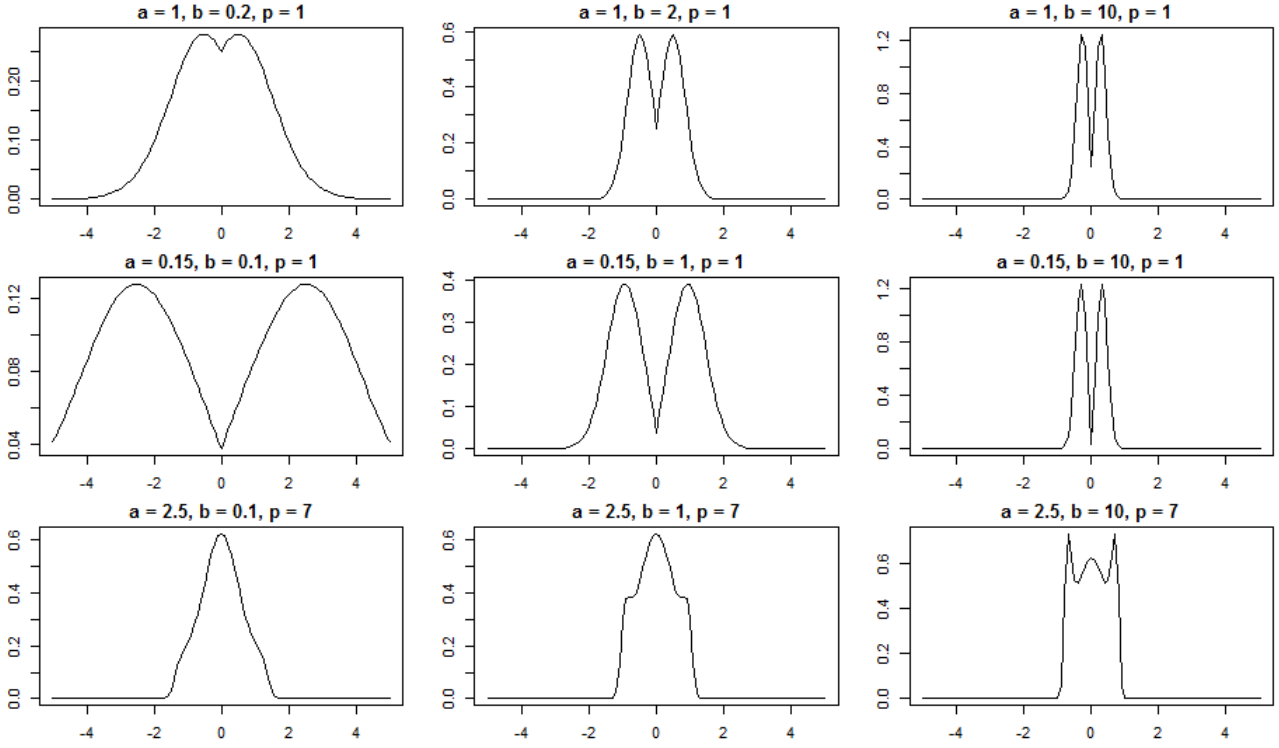

Figura 8 – Efeito do parâmetro b na distribuição

A Figura 9 ilustra os efeitos que o parâmetro p na função de densidade de probabilidade. O parâmetro é responsável pela distância entre as modas das distribuições. Um aumento do parâmetro p afasta as modas em casos bimodais, resultando, em alguns casos, com distribuições com três modas.

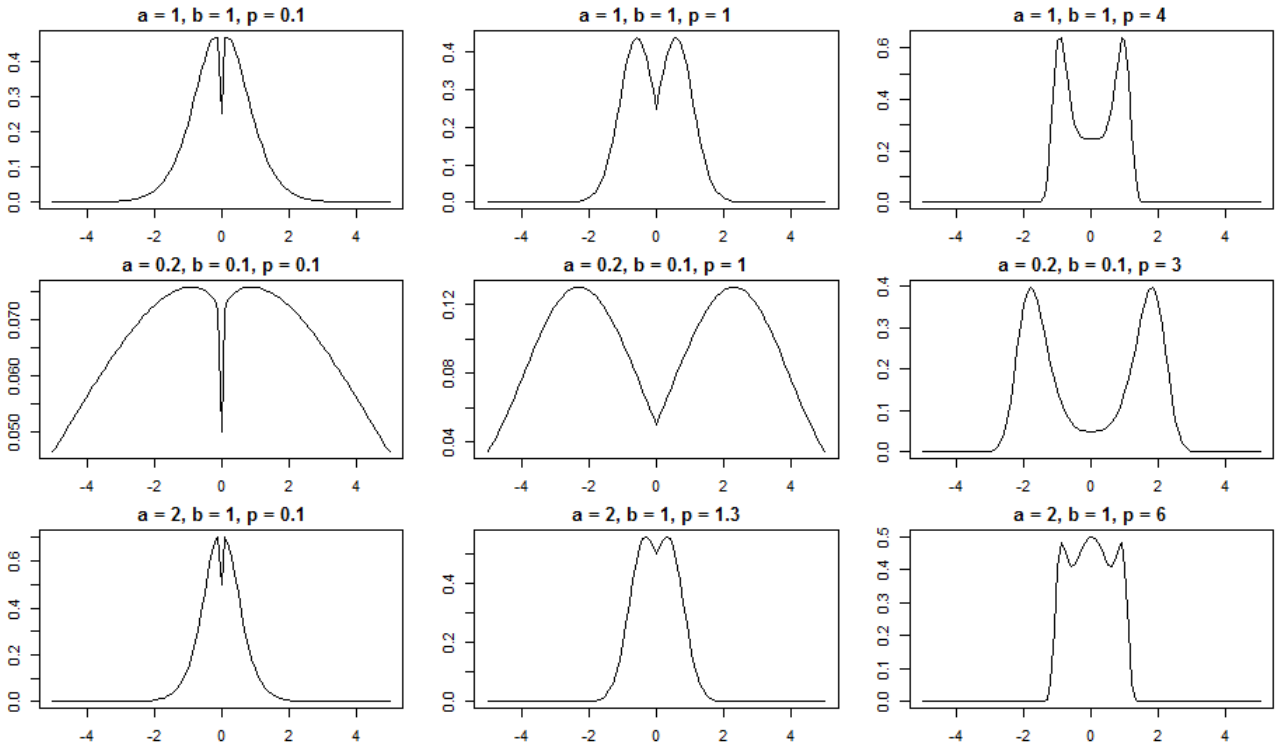

Figura 9 – Efeito do parâmetro p na distribuição

2.1.5 Estimação de parâmetros por máxima verossimilhança

O estimador de máxima verossimilhança (EMV) é obtido maximizando a equação 6.

$$log\, L(a,b,p,\mu;x_1,\ldots,x_n) = \sum_{i=1}^{n} \Big(log[a+b(1+p)|x_i-\mu|^p] e^{-x_i(a+b|x_i-\mu|^p)} \\ -2log(e^{-x_i(a+b|x_i-\mu|^p)}+1)\Big), \\ x,\mu \in \mathbb{R}, a,b,p \geq 0 \tag{6}$$

Como a função de máxima verossimilhança será otimizada em diversos parâmetros, é necessário utilizar um método quase-Newton proposto por Byrd (1994) para a estimação destes diversos parâmetros com restrições, o "*Broyden-Fletcher-Goldfarb-Shanno*" (BFGS). Os métodos quase-Newton são generalizações do método da secante, utilizados para encontrar máximos locais. É importante ressaltar que o método não garante convergência a não ser que a função tenha uma expansão quadrática de Taylor próxima do ponto ótimo.

Para melhor convergência do algorítimo ao otimizar os 4 parâmetros que maximizam a função de máxima verossimilhança utilizaremos o gradiente analítico da função que é dado pela equação 7.

$$\nabla l(a,b,p,\mu,\alpha;x_1,\ldots,x_n)=\left\langle \frac{\partial l}{\partial a},\frac{\partial l}{\partial b},\frac{\partial l}{\partial p},\frac{\partial l}{\partial \mu}\right\rangle, \tag{7}$$

$$\frac{\partial l}{\partial a}=\sum_{i=1}^{n}\left(e^{(x_i-\mu)(b|x_i-\mu|^p+a)}\times \frac{\left((\mu-x_i)\,e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}\left(b\left(p+1\right)|x_i-\mu|^p+a\right)+e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}\right)}{b\left(p+1\right)|x_i-\mu|^p+a}- \frac{2\left(\mu-x_i\right)e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}}{e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}+1}\right),$$

$$\frac{\partial l}{\partial b}=\sum_{i=1}^{n}\left(e^{(x_i-\mu)(b|x_i-\mu|^p+a)}\times \frac{\left((p+1)\,e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}|x_i-\mu|^p|x_i-\mu|^p\left(b\left(p+1\right)|x_i-\mu|^p+a\right)\right)}{b\left(p+1\right)|x_i-\mu|^p+a}- \frac{\left(x_i-\mu\right)e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}}{b\left(p+1\right)|x_i-\mu|^p+a}+ \frac{2\left(x_i-\mu\right)e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}|x_i-\mu|^p}{e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}+1}\right),$$

$$\frac{\partial l}{\partial p}=\sum_{i=1}^{n}\left(e^{(x_i-\mu)(b|x_i-\mu|^p+a)}\times \left(\frac{e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}\left(b\left(p+1\right)|x_i-\mu|^p\log\left(|x_i-\mu|\right)+b|x_i-\mu|^p\right)}{b\left(p+1\right)|x_i-\mu|^p+a}- \frac{b\left(x_i-\mu\right)e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}|x_i-\mu|^p\left(b\left(p+1\right)|x_i-\mu|^p+a\right)\log\left(|x_i-\mu|\right)}{b\left(p+1\right)|x_i-\mu|^p+a}\right)+ \frac{2b\left(x_i-\mu\right)e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}|x_i-\mu|^p\log\left(|x_i-\mu|\right)}{e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}+1}\right),$$

$$\frac{\partial l}{\partial \mu} = \sum_{i=1}^{n} \Bigg(e^{(x_i-\mu)(b|x_i-\mu|^p+a)} \times \Bigg(\frac{e^{-(x_i-\mu)(b|x_i-\mu|^p+a)} \left(bp|x_i-\mu|^p + b|x_i-\mu|^p + a\right)\left(b\left(p+1\right)|x_i-\mu|^p + a\right)}{b\left(p+1\right)|x_i-\mu|^p + a} - \frac{\frac{bp(p+1)e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}|x_i-\mu|^p}{x_i-\mu}}{b\left(p+1\right)|x_i-\mu|^p + a} \Bigg) - \frac{2e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}\left(bp|x_i-\mu|^p + b|x_i-\mu|^p + a\right)}{e^{-(x_i-\mu)(b|x_i-\mu|^p+a)}+1} \Bigg).$$

Para garantir que as derivadas apresentadas nesta sessão e na sessão 2.2.2 a seguir estão corretas, o valor do gradiente analítico foi comparado ao gradiente numérico da função.

2.2 Modelo Assimétrico

Também foi implementado um modelo assimétrico da distribuição de probabilidade apresentado na seção 2.1 utilizando uma classe de distribuições de probabilidades apresentada por Azzalini (1985) que possuem a forma $f(x) = 2\,\phi(x)\,\Phi(\alpha x)$ na qual $\phi(x)$ é uma F.D.P simétrica em torno do zero e $\phi(x)$ é uma F.D.A com sua F.D.P também simétrica em torno do zero. O parâmetro α é o parâmetro de assimetria que é limitado ao intervalo $(-1, 1)$.

A F.D.P modelo assimétrico apresentado na equação 8 possui as mesmas restrições para os parâmetros a, b e p apresentados na seção 2.1.

$$f(x|a,b,p,\mu,\alpha) = 2\frac{[a+b(1+p)|x-\mu|^p]e^{-(x-\mu)(a+b|x-\mu|^p)}}{(e^{-(x-\mu)(a+b|x-\mu|^p)}+1)^2} \times \frac{1}{e^{-(\alpha(x-\mu))(a+b(|\alpha(x-\mu)|^p))}+1}, \quad \begin{matrix} x, \mu \in \mathbb{R}, \\ a, b, p \geq 0, \\ \alpha \in (-1, 1). \end{matrix} \tag{8}$$

Tanto F.D.A e a função inversa $F^{-1}(x)$ serão calculadas computacionalmente utilizando o R pois suas forma analíticas não são obtidas trivialmente.

2.2.1 Efeitos do parâmetro de assimetria

O parâmetro de assimetria α indica qual a direção da assimetria, a figura 10 mostra algumas funções densidade de probabilidade possíveis de serem obtidas a partir de um valor de α mantendo todos os outros parâmetros constantes com valores de α decrescendo em cada linha. Se $\alpha > 0$ então a distribuição é assimétrica a direita, se $\alpha > 0$ é simétrica a esquerda e, se $\alpha = 0$ a função distribuição de probabilidade se iguala ao modelo simétrico.

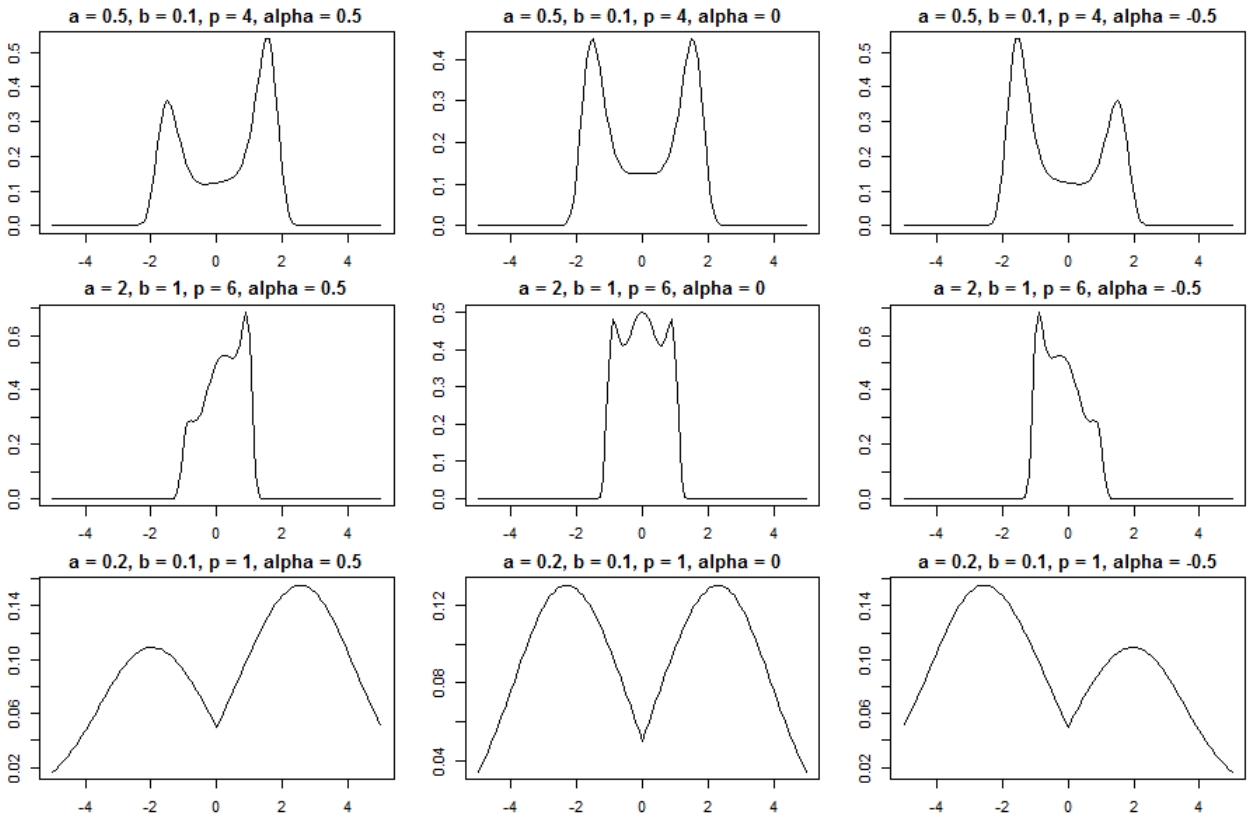

Figura 10 – Efeito do parâmetro α na distribuição

2.2.2 Estimação de parâmetros por máxima verossimilhança

O estimador de máxima verossimilhança (EMV) é obtido maximizando a equação 9 utilizando o método BFGS, citado na seção 2.1.5.

$$
\begin{aligned}
log\, L(a,b,p,\mu,\alpha;x_1,\ldots,x_n) = \sum_{i=1}^{n} \Bigg(& log(2) + log(a + (b(1+p)|x_i-\mu|^p)) - \\
& ((x_i-\mu)(a+(b|x_i-\mu|^p))) - 2log(e^{-(x_i-\mu)(a+(b|x_i-\mu|^p))}+1) - \\
& log(e-(\alpha(x_i-\mu))(a+(b(|\alpha(x_i-\mu)|^p)))+1)\Bigg), \qquad (9) \\
& x,\mu \in \mathbb{R}, \\
& a,b,p \geq 0, \\
& \alpha \in (-1,1).
\end{aligned}
$$

Para melhor convergência do algorítimo ao otimizar os 5 parâmetros que maximizam a função de máxima verossimilhança utilizaremos o gradiente analítico da função que é dado pela equação 10.

$$
\nabla l(a,b,p,\mu,\alpha;x_1,\ldots,x_n) = \left\langle \frac{\partial l}{\partial a}, \frac{\partial l}{\partial b}, \frac{\partial l}{\partial p}, \frac{\partial l}{\partial \mu}, \frac{\partial l}{\partial \alpha} \right\rangle, \qquad (10)
$$

$$
\begin{aligned}
\frac{\partial l}{\partial a} = \sum_{i=1}^{n} \Bigg(& \frac{1}{b(p+1)|x_i-\mu|^p + a} + \frac{\alpha(x_i-\mu)e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}}{e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}+1} - \\
& \frac{2(\mu-x_i)e^{(\mu-x_i)(b|x_i-\mu|^p+a)}}{(e^{(\mu-x_i)(b|x_i-\mu|^p+a)}+1)} - (x_i-\mu)\Bigg),
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial l}{\partial b} = \sum_{i=1}^{n} \Bigg(& \frac{(p+1)|x-\mu|^p}{b(p+1)|x-\mu|^p + a} + \\
& \frac{\alpha|\alpha|^p(x-\mu)e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}|x_i-\mu|^p}{e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}+1} - \\
& \frac{2(\mu-x_i)e^{(\mu-x_i)(b|x_i-\mu|^p+a)}|x_i-\mu|^p}{e^{(\mu-x_i)(b|x_i-\mu|^p+a)}+1} - (x_i-\mu)|x_i-\mu|^p\Bigg),
\end{aligned}
$$

$$\frac{\partial l}{\partial p} = \sum_{i=1}^{n} \left(\frac{b(p+1)|x_i-\mu|^p log(|x_i-\mu|) + b|x_i-\mu|^p}{b(p+1)|x_i-\mu|^p + a} + \right.$$
$$\frac{\alpha(x_i-\mu)e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}\Big(|\alpha|^p b|x_i-\mu|^p log(|x_i-\mu|) + |\alpha|^p log(|\alpha|) b|x_i-\mu|^p\Big)}{e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}+1} -$$
$$\frac{2b(\mu-x_i)e^{(\mu-x_i)(b|x_i-\mu|^p+a)}|x_i-\mu|^p log(|x_i-\mu|)}{e^{(\mu-x_i)(b|x_i-\mu|^p+a)}+1)} -$$
$$\left. b(x_i-\mu)|x_i-\mu|^p log(|x_i-\mu|) \right),$$

$$\frac{\partial l}{\partial \mu} = \sum_{i=1}^{n} \left(-\frac{bp(p+1)|x_i-\mu|^p}{(x_i-\mu)(b(p+1)|x_i-\mu|^p+a)} - \right.$$
$$\frac{e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}\Big(\alpha(|\alpha|^p b|x_i-\mu|^p+a) + \alpha|\alpha|^p bp|x_i-\mu|^p\Big)}{e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}+1} -$$
$$\frac{2e^{(\mu-x_i)(b|x_i-\mu|^p+a)}\Big(-\frac{(bp(\mu-x_i)|x_i-\mu|^p)}{(x_i-\mu)} + b|x_i-\mu|^p + a\Big)}{e^{(\mu-x_i)(b|x_i-\mu|^p+a)}+1} +$$
$$\left. bp|x_i-\mu|^p + b|x_i-\mu|^p + a \right),$$

$$\frac{\partial l}{\partial \alpha} = -\sum_{i=1}^{n} \left(e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)} \times \right.$$
$$\left. \frac{-(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a) - |\alpha|^p bp(x_i-\mu)|x_i-\mu|^p}{e^{-\alpha(x_i-\mu)(|\alpha|^p b|x_i-\mu|^p+a)}+1} \right).$$

2.3 Critério de Cramér-von Mises

O critério de Cramér-von Mises é um teste utilizado para medir a aderência de uma função distribuição acumulada comparada com uma função distribuição empírica obtida a partir dos dados, Cramér (1928). Uma forma geral proposta por Anderson (1962) permite, também, a comparação entre dois modelos empíricos. Este critério é uma alternativa ao teste de Kolmogorov-Smirnov pois permite a existência de empates dos dados.

A hipótese nula é que os dados seguem o modelo da distribuição de probabilidade proposta.

3 METODOLOGIA

3.1 Criação de pacotes

A criação de um pacote para a linguagem R deve seguir alguns princípios de qualidade e normas para que seja publicada no CRAN, o principal repositório de pacotes para o R. Essas normas tem como objetivo permitir que o pacote seja estruturado de maneira uniforme e possa ser compreendido e utilizada por terceiros sem dificuldades.

3.1.1 Pré-requisitos

Primeiramente, existem algumas ferramentas que devem ser de conhecimento do usuário para criar um pacote, seja por necessidade ou facilidade. Obviamente é necessário saber programar na linguagem desejada, neste trabalho será abordada a linguagem R e não será utilizado linguagens complementares. Os pacotes mais importantes na criação de pacotes são "devtools", utilizado para construir e compilar o seu pacote, e "roxygen2" que auxilia na montagem da documentação necessária para o pacote.

O conteúdo de um pacote deve ser definido antes de iniciar sua produção. Ao criar pacotes aconselha-se fortemente ter objetivos de tarefas que o pacote deve cumprir para evitar desperdício de tempo em funções desnecessárias e desvio do tema. O objetivo do pacote apresentado neste trabalho é o facilitar o uso e a estimação dos parâmetros de uma distribuição logística generalizada apresentada na seção 2.1. O nome dado ao pacote aqui abordado é "*genlogis*".

3.1.2 Preparando o ambiente

Todos os arquivos (documentação e códigos) devem estar agrupados em um mesmo diretório com certa estrutura ele pode ser criado da seguinte forma:

Código 1 – Criação de pasta

```
library('devtools')
setwd('C:/diretorio/para/o/pacote')
create('genlogis')
```

E a pasta deve ficar no formato da figura 11.

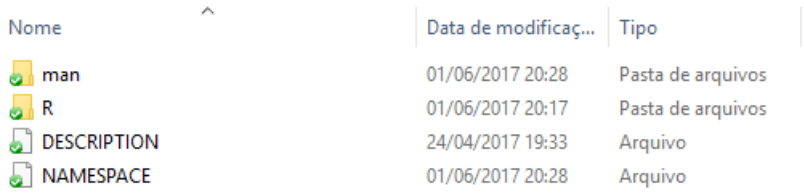

Figura 11 – Estrutura da pasta

A finalidade de cada arquivo dentro do diretório são:

1. A pasta "R" é onde ficam guardados os códigos dessas funções, os nomes devem ser de fácil identificação:

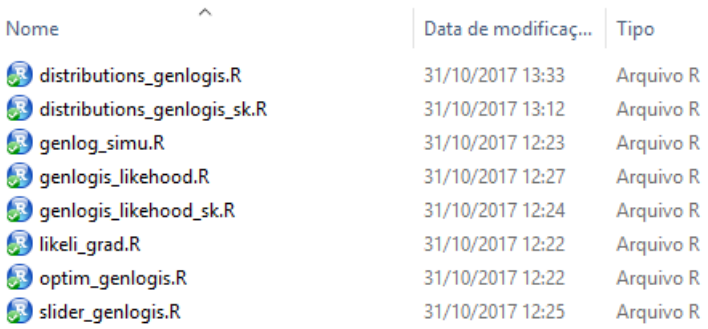

Figura 12 – Pasta "R"

2. A pasta "man" é onde ficam as documentações das funções criadas, cada função terá sua documentação.

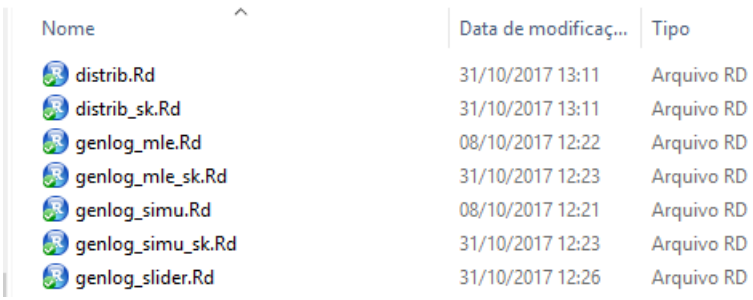

Figura 13 – Pasta "man"

3. O arquivo "DESCRIPTION" são as informações básicas sobre o pacote como nome, autor, contato, dependências etc.

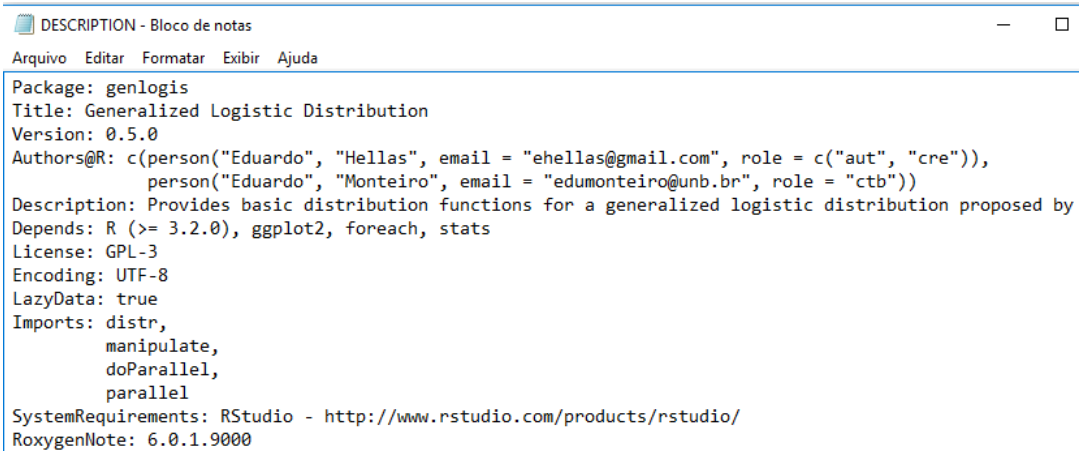

Figura 14 – Aquivo "DESCRIPTION"

(a) *Package*: nome do pacote;

(b) *Title*: título do pacote;

(c) *Version*: versão atual;

(d) *Authors@R*: listar os autores, funções (autor, criador, mantenedor...) e contato;

(e) *Description*: rápida descrição do pacote;

(f) *Depends*: pacotes os quais seu pacote depende para funcionar, eles devem ser devidamente importados no NAMESPACE;

(g) *License*: licença de distribuição do seu pacote;

(h) *Encoding*: codificação dos caracteres utilizado no pacote;

(i) *LazyData*: variável lógica que controla o uso de dados prontos, mas hoje em dia é ignorado;

(j) *Imports*: pacotes os quais são utilizadas funções porém não há necessidade de importar o pacote totalmente;

(k) *Suggests*: Não está listado mas é uma opção utilizada para listar pacotes não obrigatórios (não serão instalados junto com o pacote) porém são utilizados em exemplos ou melhoram as funcionalidades de alguma forma.

4. O arquivo "NAMESPACE" é a lista das funções exportadas pelo novo pacote e, listar

os pacotes importados, que serão carregados junto com o pacote criado. Este arquivo não deve ser editado manualmente, ele é controlado pelo pacote "roxygen2".

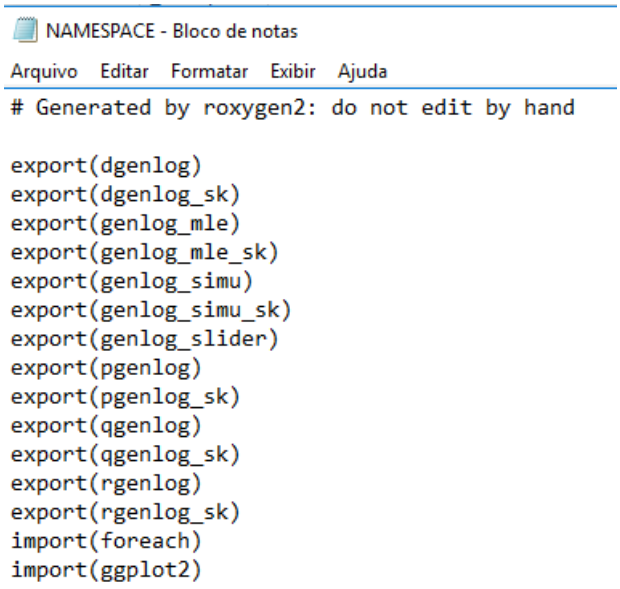

Figura 15 – Aquivo "NAMESPACE"

3.1.3 Repositório de "*backup*"

Por último, é importante ter um local onde se possam controlar as versões do seu pacote, manter "*backups*" para caso de acidentes e reverter atualizações problemáticas. Não entraremos em detalhes sobre como criar um repositório porém para o pacote deste programa foi utilizado o "*GitHub*"[3], onde todas as funções que serão apresentadas estão disponíveis para uso.

O repositório serve também para deixar o pacote disponível na internet sem estar devidamente publicado no CRAN. Assim, o pacote pode ser utilizado instalado em qualquer máquina utilizando o código 2. Como o pacote já foi publicado oficialmente no CRAN é possível instalá-lo diretamente.

Código 2 – Instalação do pacote em repositório

```
devtools::install_github('pinduzera/genlogis') ## GitHub
install.packages('genlogis') ## CRAN
```

[3] https://github.com/pinduzera/genlogis

3.2 Criando funções

Após todos os procedimentos iniciais foram executados inicia-se o processo de criação das funções do pacote. Todo o processo é apresentado utilizando o pacote utilizado neste trabalho.

3.2.1 Função de probabilidade e distribuição

Utilizando equações da seção 2.1 criamos as 4 funções básicas de uma distribuição no R, a função de probabilidade ("*pgenlogis*"), distribuição ("*dgenlogis*"), quantil ("*qgenlogis*") e o gerador de valores aleatórios ("*rgenlogis*").

Todas 4 funções básicas foram todas escritas em um mesmo arquivo, "distributions_genlogis.R" para reduzir a quantidade de documentos e por serem relativamente simples e terem um objetivo similar.

1. A função "*pgenlogis*" foi programada de forma direta e implementada com as restrições da distribuição como mostra o código 3;

Código 3 – Função "pgenlogis( )"

```
pgenlog <- function(q, a = sqrt(2/pi),
                          b = 0.5, p = 2, mu = 0){
if(!missing(a)){
if(a < 0){
stop('The argument "a" must be positive.')
        }
}
if(!missing(b)){
if(b < 0){
stop('The argument "b" must be positive.')
        }
}
if(!missing(p)){
if(p < 0){
stop('The argument "p" must be positive.')
        }
```

```
}
if(p == 0 && b > 0 && a > 0){
stop('If "p" equals to 0, "b" or "a" must be 0
          otherwise there is identifiability problem.')
}
if(b == 0 && a == 0){
stop('The distribution is not defined for "a" and b
          equal to 0 simultaneously.')
}
z <- (exp(-(q-mu) *
          (a+b*(abs(q-mu)^p)))+1)^(-1)

return(z)
}
```

2. A função "*dgenlogis*" é feita de maneira semelhante, mudando apenas a equação no fim do código;

3. A função "*qgenlogis*", como dito anteriormente, não pode ser feita de forma direta uma vez que sua forma fechada envolve somas infinitas. Assim, existem dois caminhos possíveis a serem tomados: primeiro, implementar um algoritmo de inversão de distribuições ou utilizar algum pacote existente (criando a primeira dependência) que faça isso. Neste trabalho foi tomada a segunda opção por dois motivos, primeiramente para facilitar a implementação e, segundo, pois esse pacote utiliza métodos emprestados de outras linguagens de programação que são mais eficientes e confiáveis. O nome do pacote utilizado é "distr", o código 4 mostra a implementação.

Código 4 – Função "qgenlogis( )"

```
qgenlog <- function(k, a = sqrt(2/pi), b = 0.5, p = 2,
                                              mu = 0){
if(!missing(a)){
if(a < 0){
stop('The argument "a" must be positive.')
        }
```

```
}
if(!missing(b)){
if(b < 0){
stop('The argument "b" must be positive.')
        }
}
if(!missing(p)){
if(p < 0){
stop('The argument "p" must be positive.')
        }
}
if(p == 0 && b > 0 && a > 0){
stop('If "p" equals to 0, "b" or "a" must be 0 otherwise
          there is identifiability problem.')
}
if(b == 0 && a == 0){
stop('The distribution is not defined for "a" and "b"
          equal to 0 simultaneously.')
}

dgen_log <- function(x, a1 = a, b1 = b, p1 = p){
d <- ((a1 + b1*(1+p1)*(abs(x-mu)^p1)) *
exp(-(x-mu)*(a1+b1*(abs(x-mu)^p1)))) /
((exp(-(x-mu)*(a1 + b1* (abs(x-mu)^p1)))+1)^2)

d <- ifelse(is.nan(d), 0, d)

return(d)
}

cont_dist <- distr::AbscontDistribution(d = dgen_log)
qdist <- distr::q(cont_dist)

return(qdist(k))
}
```

Observação: sempre que for utilizar funções de outros pacotes dentro de suas fun-

ções, utilizamos o formato "pacote_de_origem::função" pois ao voltar futuramente para revisar a programação é mais fácil saber o que está sendo utilizado e evita erros de ambiente das funções caso existam funções homônimas carregadas. Além disso os pacotes não precisam ser inteiramente carregados quando se vai usar poucas funções dele, não precisando ser listado no "NAMESPACE", mas deve ser listado no "DESCRIPTION" que funções deste pacote são importadas, utilizando o campo *Imports*.

4. A função "*rgenlogis*" foi feito analogamente à função anterior, utilizando o mesmo pacote.

3.2.2 Documentando funções

A maneira mais fácil para documentar as funções criadas anteriormente é utilizando o pacote "*roxygen2*", que funciona em conjunto com o "*devtools*". Existe toda uma sintaxe que facilita como apresentado no código 5.

Código 5 – Documentação das funções de distribuição

```
############################

#' The Generalized logistic distribution
#'
#' Density, distribution function, quantile function and
#' random generation a generalized logistic distribution.
#' @param x,q vector of quantiles.
#' @param k vector of probabilities.
#' @param n number of observations. If length(n) > 1,
#' the length is taken to be the number required
#' @param a,b,p  parameters >= 0, with restrictions.*
#' @param mu mu parameter
#' @keywords genlogis
#'
#' @export
#' @examples
#' pgenlog(0.5)
#' curve(dgenlog(x), xlim = c(-3,3))
```

```
#'
#' rgenlog(100)
#'
#' qgenlog(0.95)
#'
#' @usage
#' pgenlog(q, a = sqrt(2/pi), b = 0.5, p = 2, mu = 0)
#'
#' @name distrib
#'
#' @return
#' \code{dgenlog} gives the density, \code{pgenlog} gives
#' the distribution function,
#' \code{qgenlog} gives the quantile function, and \code{rgenlog}
#' generates random deviates.\cr
#'
#'  The length of the result is determined by \code{n} for
#' \code{rgenlog}, and is the maximum of the lengths
#'  of the numerical arguments for the other functions.
#'
#' @details
#'
#' The used distribution for this package is given by:
#' \deqn{f(x) = ((a + b*(1+p)*(abs(x-mu)^p))*
#' exp(-(x-mu)*(a+b*(|x-mu|^p)))) /
#' ((exp(-(x-mu)*(a + b* (|x-mu|^p)))+1)^2)}
#'
#' The \code{qgenlog()} returns values for P(X < x).\cr
#'
#' The default values for \code{a, b, p and mu} produces a function
#' with mean 0 and variance close to 1.\cr
#'
#' *Restrictions:\cr
#'
#' If \code{p} equals to 0, \code{b} or \code{a} must be 0
#' otherwise there is identifiability problem.\cr
```

```
#'
#' The distribution is not defined for \code{a} and \code{b} equal
#'  to 0 simultaneously.\cr
#'
#' @references
#' Rathie, P. N. and Swamee, P. K (2006) \emph{On a new invertible
#' generalized logistic distribution
#' approximation to normal distribution},
#' Technical Research Report in Statistics, 07/2006,
#' Dept. of Statistics, Univ. of Brasilia, Brasilia, Brazil.
```

Seguem os diversos comandos utilizados nesta sintaxe e suas funções:

1. "*@param* utilizado para citar os parâmetros utilizados";
2. "*@keywords*": palavras-chave para facilitar a busca na ajuda do programa, aconselhável utilizar o nome do pacote como uma palavra-chave pois ela cria uma indexação do pacote nas buscas;
3. "*@export*": exporta as funções para o "NAMESPACE";
4. "*@import*": caso a função possua algum pacote que seja estritamente dependente deve ser listado aqui para que possa ser transcrito para o "NAMESPACE";
5. "*@examples*": são os exemplos que podem ser copiados e testados para ver o código em funcionamento;
6. "*@usage*": é a forma que a função é utilizada e, também mostrar os valores padrão, caso existam;
7. "*@name*": atribui um nome para o arquivo de documentação e é util quando deseja documentar várias funções juntas;
8. "*@details*": área reservada para escrever qualquer detalhe necessário sobre o funcionamento da função;
9. "*@author*": área reservada para listar os autores das funções nos casos de múltiplos autores;

10. "*@return*": área utilizada para explicar o resultado que a função retorna, é de extrema importância saber o tipo de objeto retornado pela função;

11. "*@references*": área reservada para listar referências bibliográficas caso necessário;

12. "*@rdname*": adiciona a função abaixo a uma documentação anterior com nome definido por "*@name*".

O código 5 deve ser colocado antes da função que for documentar e exportar do pacote, após isso feito a documentação pode ser criada de uma vez com os comandos listados no código 6, lembrando que seu R deve estar com o "*working directory*" definido na pasta que guarda a pasta do pacote.

Código 6 – Criando a documentação

```
setwd('E:/diretorio/onde/esta/o/pacote/') ## windows
devtools::document('genlogis') #cria a documentação
devtools::install('genlogis') #instala o pacote

library('genlogis') #carrega o pacote

?rgenlog() #testa a ajuda do pacote
```

Caso queira criar várias funções num mesmo arquivo documentadas conjuntamente não é necessário copiar totalmente a documentação, antes de cada função, basta utilizar o código 7 antes desta função e ela será incluída no arquivo principal com seus valores padrão.

Código 7 – Documentação conjunta

```
#' @rdname distrib
#' @export
```

Uma vez que o pacote está criado, com documentação correta, pode ser instalado direto do computador utilizando o código 6 e depois a documentação pode ser conferida, produzindo o resultado da figura 16.

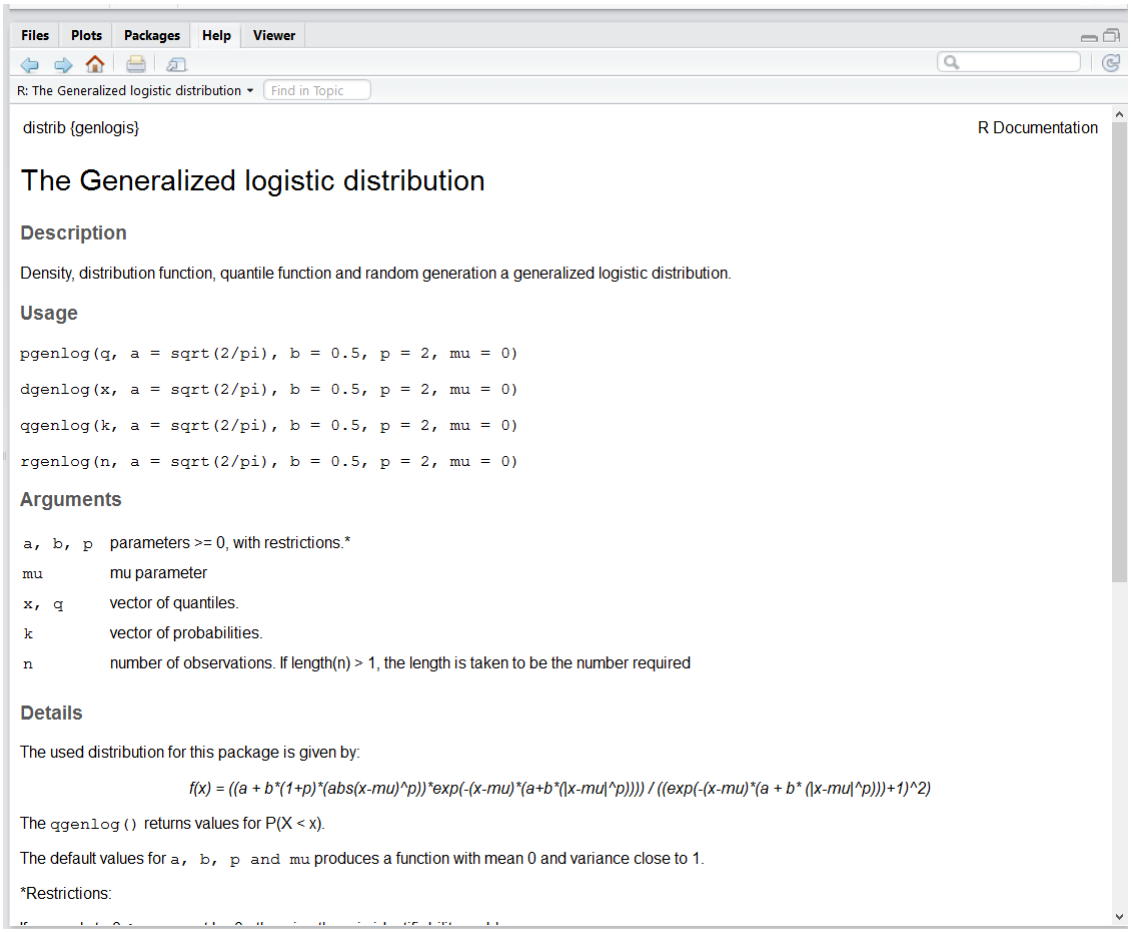

Figura 16 – Documentação

3.2.3 Teste de construção e publicação

De forma geral é simples fazer um pacote para uso pessoal porém, quando se tem a intenção de publicá-lo certos padrões devem ser mantidos e o pacote não deve possuir erros. Para fazer a verificação basta utilizar a função "*check*" (código 8) do pacote "*devtools*". Essa verificação pode levar alguns minutos, ela vai testar todos os exemplos do pacote e checar se a documentação e outros arquivos não possuem divergências e se estão funcionando corretamente. A rigor esta função não deve retornar qualquer erro, aviso ou notas (vide figura 17) mas existem exceções porém dificultam o processo de publicação.

Código 8 – Teste e construção do pacote

```
setwd('C:/diretorio/onde/esta/o/pacote/') ## windows
```

```
devtools::check('genlogis') # verifica o pacote
devtools::build('genlogis') # .tar.gz para publicar
```

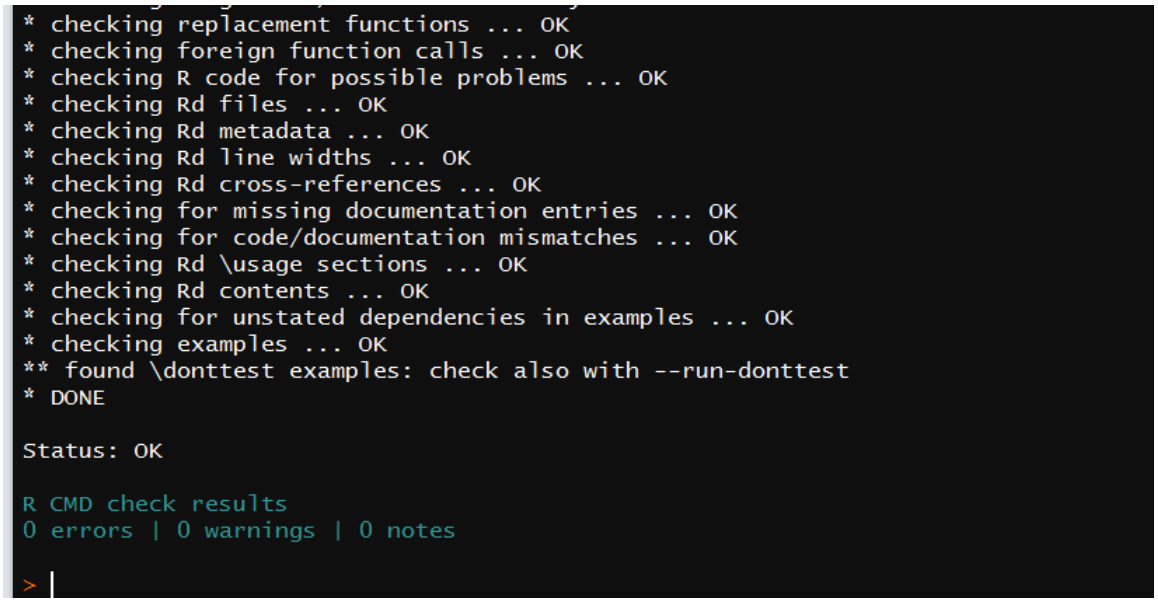

Figura 17 – Verificação do pacote

Para publicar o pacote é necessário que o pacote esteja em um arquivo compacto único, a função "*build*" cria este arquivo com a extensão ".tar.gz" para esta finalidade. Para publicar acesse a página[4] de submissão do CRAN e envie o arquivo. Tenha certeza de ter um pacote funcional e sem problemas pois ele passa por uma verificação automática e depois manual os quais podem haver "*feedbacks*" para problemas que devem ser consertados. Também é aconselhável ler as políticas de publicação do site[5].

3.3 Função de estimação de parâmetros por máxima verossimilhança

Como visto na seção 2.1.5 pode-se, através função de máxima verossimilhança, estimar múltiplos parâmetros simultaneamente. A função "*constrOptim*" utiliza o método numérico quase-Newton BFGS para a otimizar a função com múltiplos parâmetros e restrições no espaço paramétrico, que não possui convergência garantida. Para melhorar a taxa de convergência o programa foi implementado utilizando o gradiente analítico. Outro fator que

[4] https://cran.r-project.org/submit.html
[5] https://cran.r-project.org/web/packages/policies.html

auxilia a convergência é um bom chute dos valores iniciais dos parâmetros para o algoritmo, para isso foi criado um recurso visual para comparar um conjunto de dados contra a distribuição logística empírica, é a função "*genlog_slider*". Utilizando o código 9 podemos ver as funções em funcionamento.

Código 9 – Estimação por MV e chute inicial

```
library(genlogis)
## simulação de dados da distribuição
datas <- rgenlog(10000, 1.5,2,2, 0)
## visualização e seleção de chutes iniciais
genlog_slider(datas, return_var = 'parameters')
## estimação por MV
genlog_mle(parameters,datas)
```

Dentro do Ambiente de Desenvolvimento Integrado (IDE) "*RStudio*" existe uma interface de visualização gráfica que facilita a exibição, principalmente quando se utiliza recursos interativos. A função "*genlog_slider*" depende desta interface para funcionar, a vantagem é que é possível manipular os parâmetros da distribuição para tentar encaixar no histograma de dados para achar um bom valor do chute inicial sem ter que rodar o código múltiplas vezes e depois guardar os valores em um objeto para ser utilizado futuramente, como vemos na figura 18.

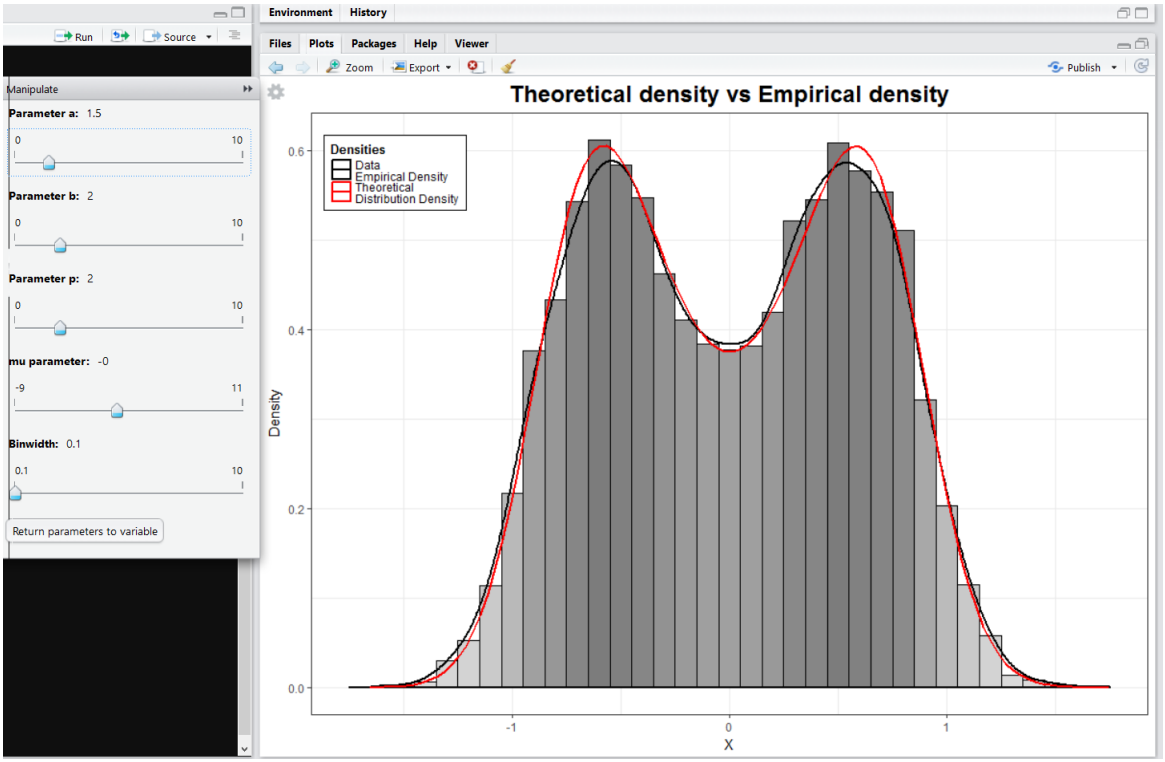

Figura 18 – Interface interativa para chute inicial

A função de estimação de máxima verossimilhança foi implementada utilizando a função "*constrOptim*" pois ela permite criar limites para os parâmetros de forma consistente. Ao utilizar a função "*genlog_mle(parameters, datas, hessian = F)*" obtemos a saída da figura 19, que nos fornece na primeira linha os valores estimados dos parâmetros a, b, p, e μ, que chegam razoavelmente próximos dos reais simulados, 1.5, 2, 2 e 0, respectivamente.

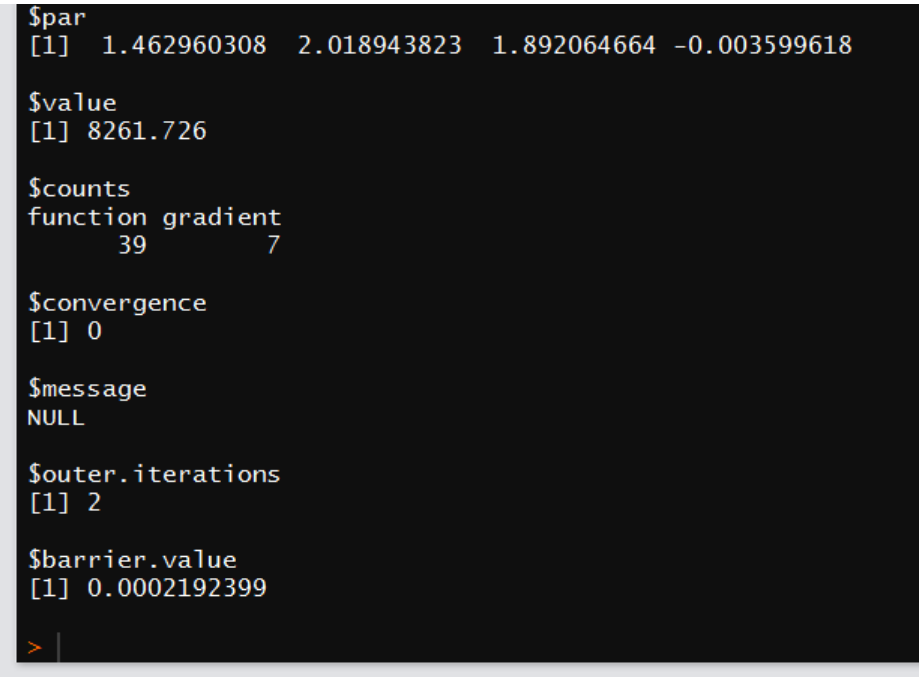

Figura 19 – Resultado de estimação por máxima verossimilhança

3.4 Função de simulação

Ao trabalhar com novos pacotes e distribuições de probabilidade é de interesse fazer simulações controladas para testar a capacidade e eficiência dos métodos utilizados. Para isso foi implementada uma função que facilita fazer inúmeras simulações e estimações por máxima verossimilhança apresentada na seção 3.3. Além disso, a forma padrão de execução do R utiliza apenas com um núcleo de processadores, essa função foi implementada de forma que "força" o *software* a utilizar múltiplos núcleos, se disponíveis, acelerando o tempo de processamento consideravelmente.

O código 10 faz k simulações de amostras de tamanho n e faz seus EMVs utilizando um mesmo chute inicial e retorna um "*data.frame*" de tamanho k com o resultado de todas as estimações.

Código 10 – Simulações

```
genlog_simu(real.par = c(0.3, 0.9, 1.5, 0.0),
            init.par = c(0.9, 0.3, 0.2, 0.0),
            sample.size = 100, k = 50, threads = 2,
            seed = 200)
```

3.5 Glossário de funções

Todas as funções apresentadas anteriormente utilizam como base o modelo Rathie-Swamee apresentado na seção 2.1. Como este modelo é estritamente simétrico, não é o ideal para todos os casos. Deste modo, todas as funções mostradas foram refeitas utilizando como referência o modelo da seção 2.2 que é uma modificação que permite transformar modelos simétricos em assimétricos. Todas as implementações que envolvem o modelo assimétrico incluem o argumento (e parâmetro) "*skew*" que permite controlar a assimetria, com exceção da função interativa "*genlog_slider*" que foi adicionado um argumento que permite a troca da distribuição teórica utilizada. A tabela 1 é uma referência com todas as funções criadas neste trabalho.

Tabela 1 – Glossário de funções

=

Função	Objetivo	Distribuição utilizada
dgenlog	Calcular o valor da densidade.	Rathie-Swamee
pgenlog	Calcular o valor da distribuição acumulada.	Rathie-Swamee
qgenlog	Calcular o quantil da distribuição.	Rathie-Swamee
rgenlog	Gerar valores aleatórios da distribuição.	Rathie-Swamee
dgenlog_sk	Calcular o valor da densidade.	Rathie-Swamee assimétrico
pgenlog_sk	Calcular o valor da distribuição acumulada.	Rathie-Swamee assimétrico
qgenlog_sk	Calcular o quantil da distribuição.	Rathie-Swamee assimétrico
rgenlog_sk	Gerar valores aleatórios da distribuição.	Rathie-Swamee assimétrico
genlog_mle	Calcular o EMV.	Rathie-Swamee
genlog_mle_sk	Calcular o EMV.	Rathie-Swamee assimétrico
genlog_simu	Simular o EMV controladamente.	Rathie-Swamee
genlog_simu_sk	Simular o EMV controladamente.	Rathie-Swamee assimétrico
genlog_slider	Auxiliar escolha do valor inicial para o EMV.	Rathie-Swamee e Rathie-Swamee assimétrico

4 RESULTADOS E APLICAÇÕES

4.1 Gêiseres

Boas aplicação para o modelo Rathie-Swamee devem envolver dados bimodais, um bom conjunto de dados é sobre as erupções de gêiseres. O Parque Nacional de Yellowstone, Estados Unidos da América, possui a maior parte dos gêiseres existentes. Gêiseres são fontes termais vulcânicas que entram em erupção periodicamente expelindo água e vapor, em casos mais extremos alcançando entre 90 e 120 metros de altura[6]. As erupções dos gêiseres estão entre as principais atrações do parque.

O gêiser *Old Faithful* é o mais famoso e é o mais estudado devido à frequência de suas erupções. Existem diversos registros sobre a frequência de erupções do *Old Faithful*, aqui será utilizado o conjunto de dados *faithful* disponível no R que possui 272 observações com registro do tempo de cada erupção em minutos e o tempo de espera até a próxima erupção, também em minutos.

As variáveis apresentadas serão ajustadas e comparadas entre três distribuições de probabilidade, Rathie-Swamme, Normal e Rathia-swamee assimétrica.

Figura 20 – Gêiser *Old Faithful* - Fonte: site do serviço nacional de parques do EUA

[6]https://www.nps.gov/yell/learn/nature/geysers.htm

4.1.1 Duração das erupções

Fazendo uma breve análise exploratória do tempo de duração temos que o tempo médio de duração de cada erupção é de 3.49 minutos com um desvio padrão de 1.14. Ao ajustar pelo modelo normal usaremos estes valores como chutes iniciais para a estimação de verossimilhança. O ajuste de máxima verossimilhança obtido foi $\mu = 3.487670$ e $\sigma = 1.139239$, apresentado figura 21. Para comparar a qualidade do ajuste com a distribuição empírica utilizaremos o teste Cramér-von Mises apresentado na seção 2.3. Neste caso, o p-valor obtido foi menor que 0.001, rejeitando fortemente o ajuste. Esse resultado era esperado pois a distribuição normal não é capaz de modelar a dados bimodais.

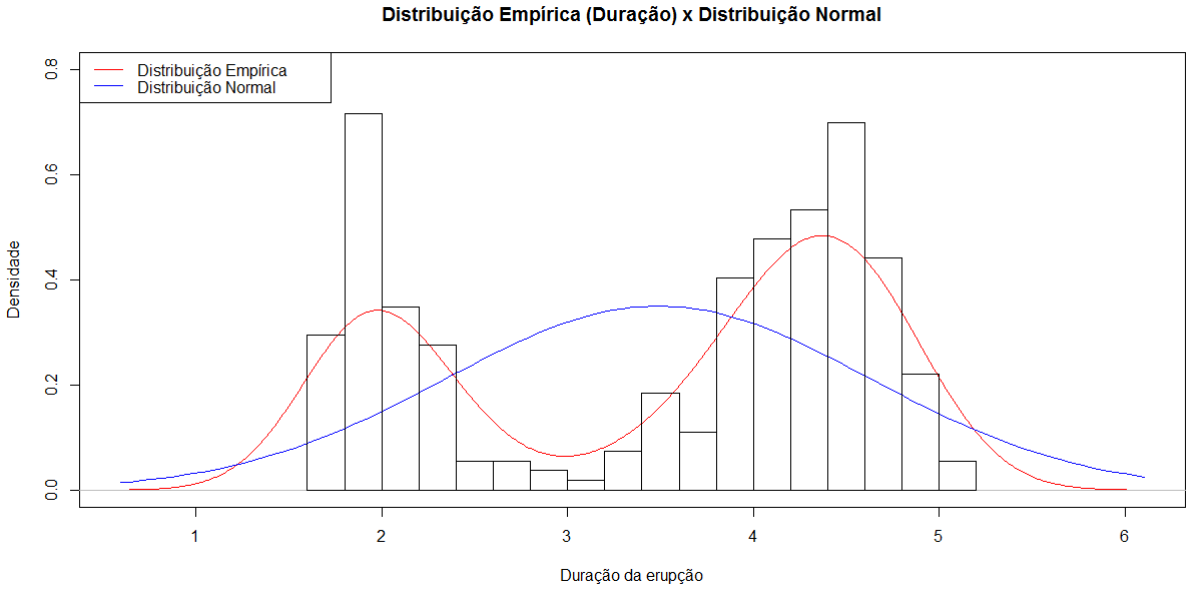

Figura 21 – Duração de erupção - Ajuste para a distribuição Normal

Para o modelo de Rathie-Swamee utilizamos a ferramenta apresentada na seção 3.3 para definir o chute inicial. Os valores iniciais selecionados para os parâmetros foram: $a = 0.3$, $b = 1.01$, $p = 2.11$ e $\mu = 3.5$. O ajuste obtido está representado na figura 22 e os valores dos parâmetros são $a = 0.27$, $b = 0.40$, $p = 3.34$ e $\mu = 3.23$. Pelo teste de Cramér-von Mises obtemos um p-valor de 0.002, rejeitando o ajuste. O ajuste obtido foi melhor para captar a bimodalidade que o modelo anterior porém, não é capaz de lidar com a diferença entre as modas devido à simetria do modelo.

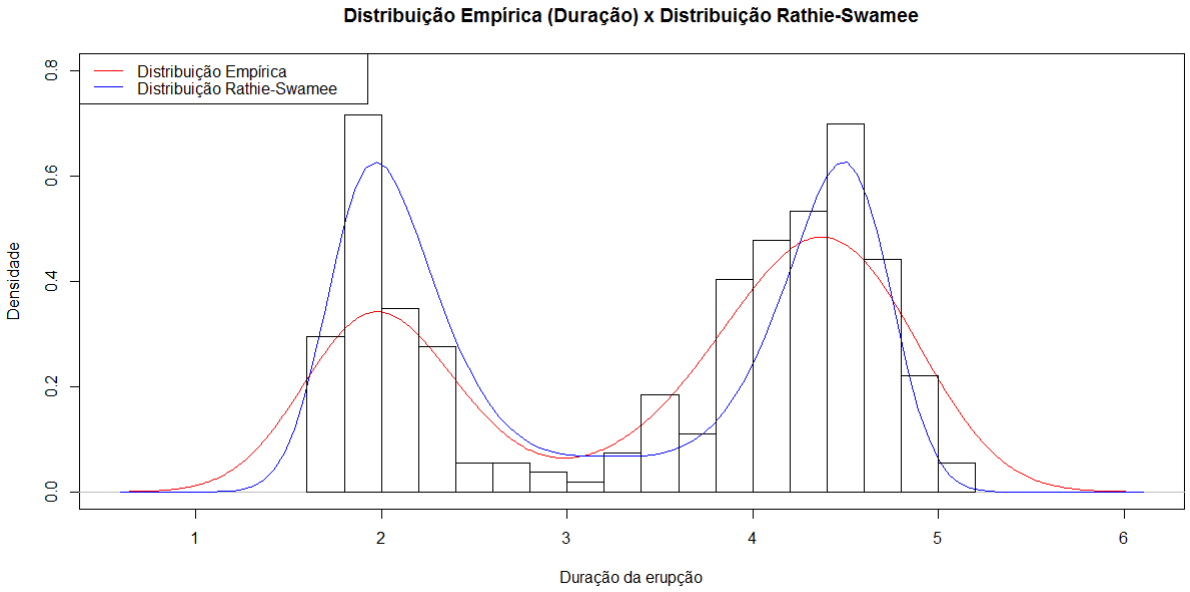

Figura 22 – Duração de erupção - Ajuste para a distribuição Rathie-Swamee

Por último, o modelo Rathie-Swamee assimétrico, com chutes iniciais $a = 0.3$, $b = 0.5$, $p = 1.98$, $\mu = 3.11$ e $\alpha = 0.52$ obtemos os ajuste da figura 23. Os valores ótimos obtidos para os parâmetros foram $a = 0.17$, $b = 0.48$, $p = 2.85$, $\mu = 3.15$ e $\alpha = 0.78$. O teste de Cramér-von Mises teve um p-valor menor 0.0001, o que nos leva a rejeição do modelo proposto. Apesar da rejeição, vemos no gráfico que o modelo capta bem a bimodalidade e a diferença entre elas, apesar de não seguir muito o modelo empírico.

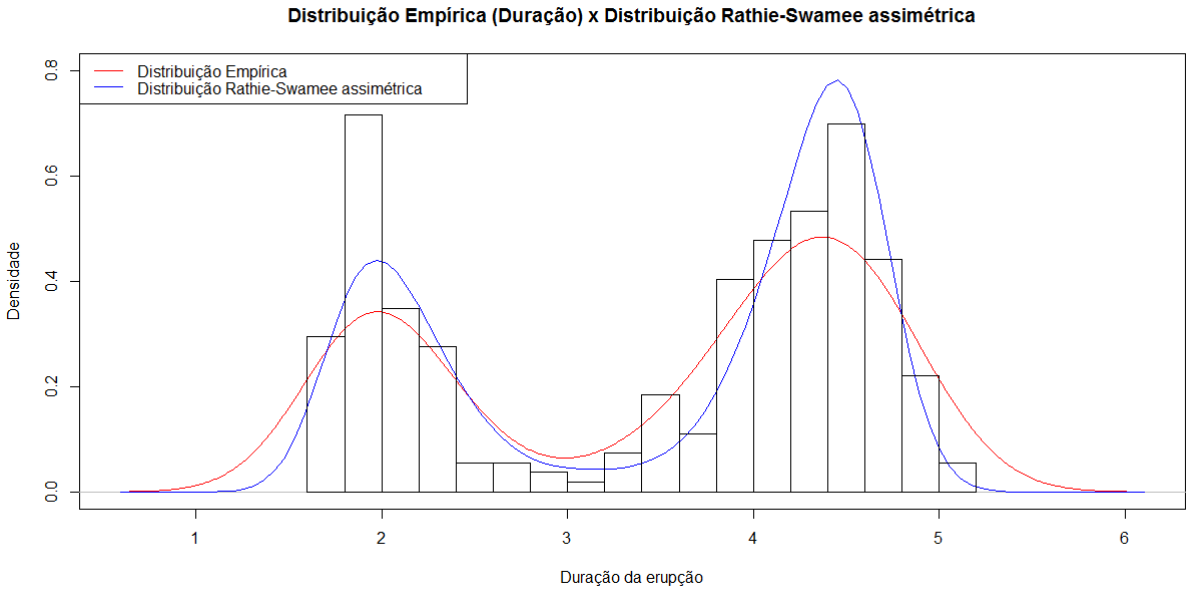

Figura 23 – Duração de erupção - Ajuste para a distribuição Rathie-Swamee assimétrica

4.1.2 Tempo entre as erupções

Ao seguir o mesmo procedimento da variável anterior, foi feita uma análise exploratória do tempo de espera entre as erupção e temos que o tempo médio de espera entre cada erupção é de 70.89 minutos com um desvio padrão de 13.59. Ao ajustar pelo modelo normal obtemos $\mu = 70.89$ e $\sigma = 13.57$, apresentado figura 24. O teste Cramér-von Mises forneceu um p-valor menor que 0.001, rejeitando fortemente o ajuste. Esse resultado, novamente, era esperado pois a distribuição normal não é capaz de modelar dados bimodais.

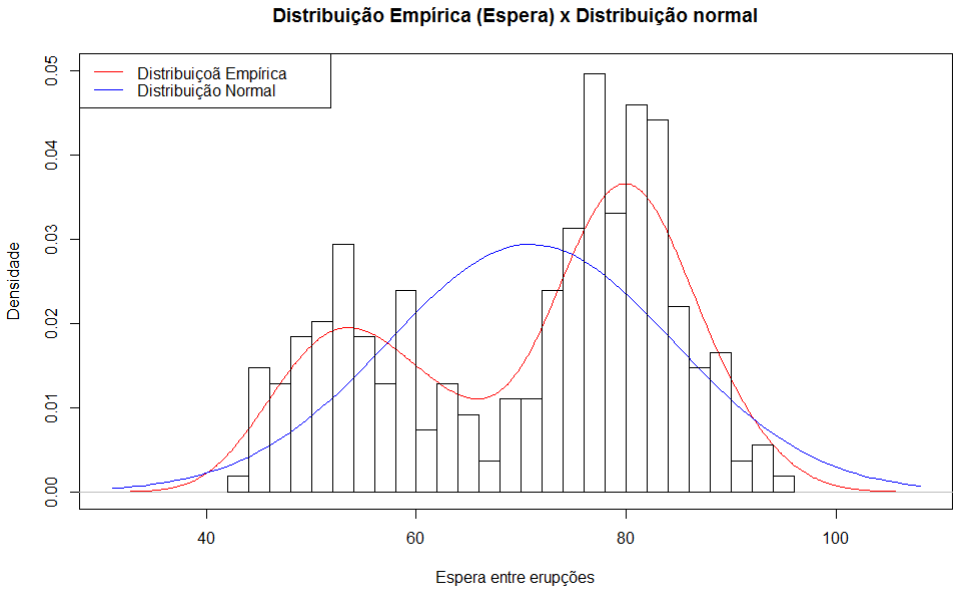

Figura 24 – Tempo de espera - Ajuste para a distribuição Normal

Para o modelo de Rathie-Swamee os valores iniciais utilizados para os parâmetros foram: $a = 0.1$, $b = 0.1$, $p = 0.7$ e $\mu = 71$. O ajuste obtido está representado na figura 25 e os valores dos parâmetros são $a = 0.022$, $b = 0.001$, $p = 1.471$ e $\mu = 67.56$. Pelo teste de Cramér-von Mises obtemos um p-valor menor que 0.001, rejeitando o ajuste. O ajuste obtido, novamente, foi melhor para captar a bimodalidade que o modelo normal porém, não é capaz de lidar com a diferença entre as modas devido à simetrida do modelo.

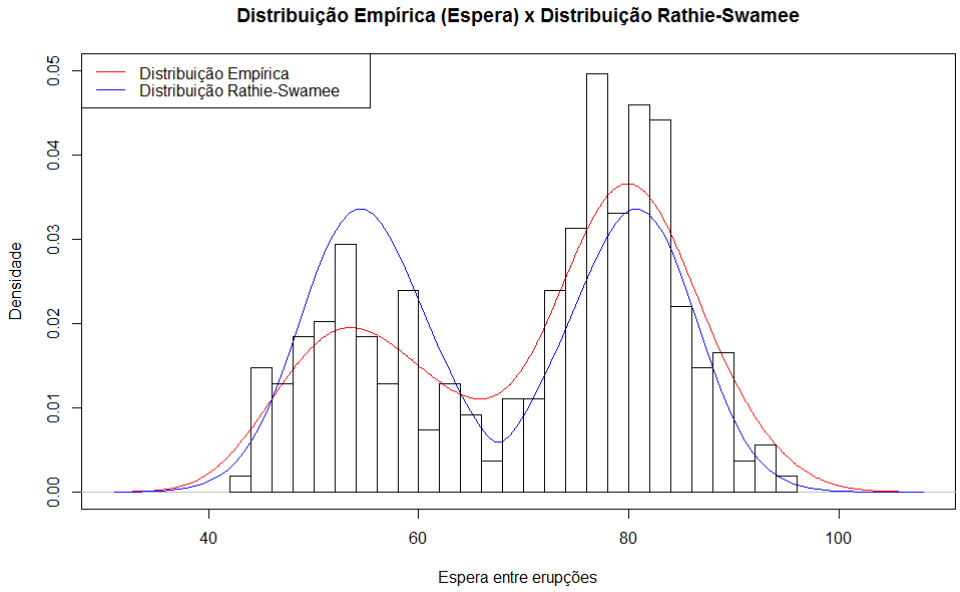

Figura 25 – Tempo de espera - Ajuste para a distribuição Rathie-Swamee

Por último, o modelo Rathie-Swamee assimétrico, com chutes iniciais $a = 0.01$, $b = 0.01$, $p = .8$, $\mu = 65.5$ e $\alpha = 0.7$ obtemos os ajuste da figura 26. Os valores ótimos obtidos para os parâmetros foram $a = 0.0276$, $b = 0.0007$, $p = 1.6605$, $\mu = 65.9347$ e $\alpha = 0.6906$. O teste de Cramér-von Mises teve um p-valor menor que 0.001, o que nos leva a rejeição do modelo proposto. Apesar do teste dar um resultado de rejeição, é possível ver pelo gráfico que o modelo capta bem a bimodalidade e a diferença entre elas, e se ajusta bem ao modelo empírico. Seria necessário fazer um estudo mais cuidadoso das limitações do teste.

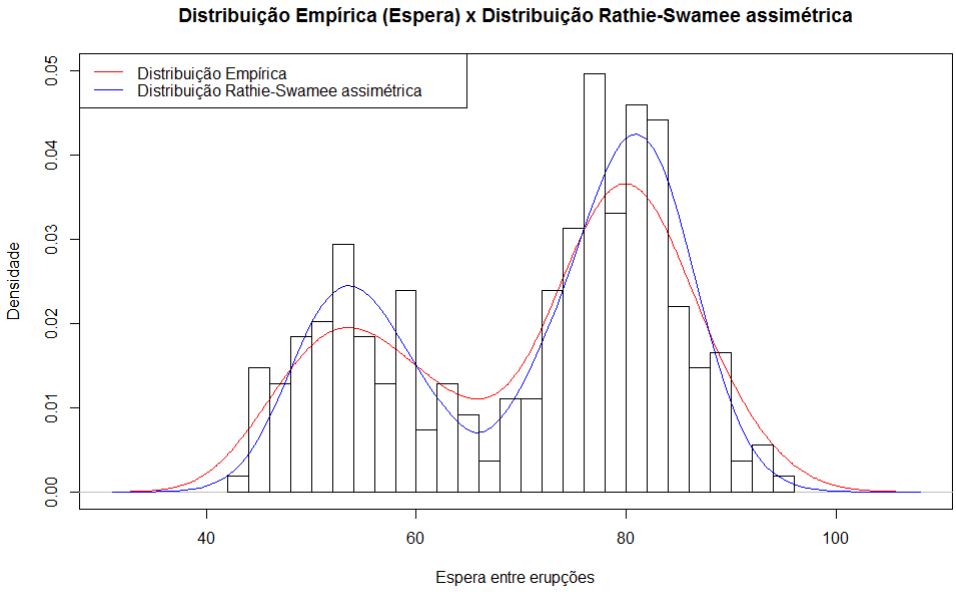

Figura 26 – Tempo de espera - Ajuste para a distribuição Rathie-Swamee assimétrica

4.2 Simulações

Utilizando a função apresentada na sessão 3.4 foi possível testar a capacidade de estimação do pacote para diversas distribuições apresentadas na seção 2.1, o caso genérico, aproximação para a distribuição normal, aproximação para a distribuição logística e dois casos de descontinuidade. A tabela 2 mostra os valores reais dos parâmetros da distribuição logística generalizada utilizados na simulações e a indicação para que distribuição ela se aproxima, caso se aconteça.

Tabela 2 – Simulações do EMV parâmetro a

Distribuição	a	b	p	μ
Genérica	0.3	0.9	1.5	0.0
Normal	1.5977	0.0727	1.9620	0.0
Logística	2.0	0.0	5.0	0.0
Descontinuidade 1	1.0	1.0	1.0	0.0
Descontinuidade 2	0.0	1.0	1.0	0.0

As tabelas 3, 4, 5 e 6 mostram as médias e os desvios padrão das estimações de

máxima verossimilhança de cada parâmetro, a, b, p e μ, respectivamente. Foram feitas 1000 simulações para cada tamanho de amostra e distribuição. Para todos as estimações o chute inicial dos parâmetros foi o mesmo: $a = 0.9$, $b = 0.3$, $p = 0.2$ e $\mu = 0$.

Em todos os casos, é importante notar que os estimadores são não viciados e consistentes pois conforme o tamanho da amostra aumenta as estimações convergem para o valor real e sua variância diminui. Além disso, nas estimações dos parâmetros b e p em alguns casos extrapolam muito os valores no caso de amostras pequenas, iguais a 10 ou, em alguns casos, iguais a 50, é necessário investigação mais profunda do motivo das estimações não serem ideais, um provável motivo pode ser o chute inicial (fixo) ser muito ruim para as amostras utilizadas pois a estimação por máxima verossimilhança do modelo Rathie-Swamee é sensível ao chute inicial.

O caso da distribuição logística deve ser analisada com mais cuidado pois como o parâmetro b é igual a zero o valor de p é irrelevante mas, neste caso, é importante notar que existe um problema de identificabilidade da distribuição. Se p se aproxima de 0 a estimação dos valores dos parâmetros a e b se confundem e, a soma deles é o valor original do parâmetro a.

Tabela 3 – Simulações do EMV parâmetro a

Tamanho da amostra	Distribuição	Param. a	Média(a)	Var(a)	DP(a)
10	Descontínuo 1	1.0	0.7520	0.6991	0.8361
50	Descontínuo 1	1.0	0.7801	0.5357	0.7319
100	Descontínuo 1	1.0	0.7757	0.4045	0.6360
500	Descontínuo 1	1.0	0.9182	0.1044	0.3232
1000	Descontínuo 1	1.0	0.9716	0.0385	0.1963
10000	Descontínuo 1	1.0	1.0012	0.0028	0.0527
10	Descontínuo 2	0.0	0.2970	0.2229	0.4721
50	Descontínuo 2	0.0	0.1090	0.0466	0.2160
100	Descontínuo 2	0.0	0.0575	0.0150	0.1225
500	Descontínuo 2	0.0	0.0221	0.0018	0.0430
1000	Descontínuo 2	0.0	0.0156	0.0008	0.0290
10000	Descontínuo 2	0.0	0.0039	0.0001	0.0073
10	Genérica	0.3	0.3952	0.2880	0.5366
50	Genérica	0.3	0.2741	0.0887	0.2979
100	Genérica	0.3	0.2733	0.0517	0.2273
500	Genérica	0.3	0.3006	0.0099	0.0995
1000	Genérica	0.3	0.2981	0.0051	0.0716
10000	Genérica	0.3	0.3004	0.0005	0.0217
10	Logística	2.0	0.9082	0.6873	0.8290
50	Logística	2.0	0.9404	0.6172	0.7856
100	Logística	2.0	0.8993	0.4925	0.7018
500	Logística	2.0	1.0502	0.3389	0.5821
1000	Logística	2.0	0.9635	0.5285	0.7270
10000	Logística	2.0	1.4129	0.1754	0.4188
10	Normal	1.5977	0.7072	0.5805	0.7619
50	Normal	1.5977	0.8127	0.5416	0.7360
100	Normal	1.5977	0.9544	0.5104	0.7144
500	Normal	1.5977	1.3784	0.2556	0.5056
1000	Normal	1.5977	1.5259	0.0848	0.2911
10000	Normal	1.5977	1.5961	0.0007	0.0260

Tabela 4 – Simulações do EMV parâmetro b

Tamanho da amostra	Distribuição	Param. b	Média(b)	Var(b)	DP(b)
10	Descontínuo 1	1.0	9.0893	1440.4322	37.9530
50	Descontínuo 1	1.0	1.2292	0.6874	0.8291
100	Descontínuo 1	1.0	1.2343	0.4977	0.7054
500	Descontínuo 1	1.0	1.0858	0.1200	0.3464
1000	Descontínuo 1	1.0	1.0288	0.0440	0.2098
10000	Descontínuo 1	1.0	0.9991	0.0031	0.0557
10	Descontínuo 2	1.0	0.7501	0.3138	0.5602
50	Descontínuo 2	1.0	0.8629	0.0726	0.2695
100	Descontínuo 2	1.0	0.9285	0.0294	0.1714
500	Descontínuo 2	1.0	0.9732	0.0045	0.0674
1000	Descontínuo 2	1.0	0.9812	0.0023	0.0476
10000	Descontínuo 2	1.0	0.9953	0.0002	0.0134
10	Genérica	0.9	1.3961	149.8797	12.2425
50	Genérica	0.9	0.9181	0.1240	0.3522
100	Genérica	0.9	0.9267	0.0729	0.2700
500	Genérica	0.9	0.9001	0.0128	0.1131
1000	Genérica	0.9	0.9024	0.0071	0.0842
10000	Genérica	0.9	0.8998	0.0006	0.0247
10	Logística	0.0	11.4015	2322.9593	48.1971
50	Logística	0.0	1.0841	0.6428	0.8017
100	Logística	0.0	1.0992	0.5065	0.7117
500	Logística	0.0	0.9502	0.3382	0.5816
1000	Logística	0.0	1.0348	0.5253	0.7248
10000	Logística	0.0	0.5864	0.1757	0.4191
10	Normal	0.0727	5.0754	704.7159	26.5465
50	Normal	0.0727	0.9066	0.5917	0.7692
100	Normal	0.0727	0.7426	0.5543	0.7445
500	Normal	0.0727	0.2999	0.2710	0.5205
1000	Normal	0.0727	0.1503	0.0880	0.2967
10000	Normal	0.0727	0.0745	0.0004	0.0198

Tabela 5 – Simulações do EMV parâmetro p

Tamanho da amostra	Distribuição	Param. p	Média(p)	Var(p)	DP(p)
10	Descontínuo 1	1.0	33.4423	10605.7145	102.9840
50	Descontínuo 1	1.0	3.4596	65.3355	8.0830
100	Descontínuo 1	1.0	1.5840	6.6964	2.5877
500	Descontínuo 1	1.0	1.0213	0.1116	0.3341
1000	Descontínuo 1	1.0	1.0139	0.0458	0.2141
10000	Descontínuo 1	1.0	1.0048	0.0039	0.0625
10	Descontínuo 2	1.0	7.6801	329.6395	18.1560
50	Descontínuo 2	1.0	1.3583	0.6658	0.8160
100	Descontínuo 2	1.0	1.1481	0.0936	0.3059
500	Descontínuo 2	1.0	1.0344	0.0089	0.0943
1000	Descontínuo 2	1.0	1.0238	0.0040	0.0631
10000	Descontínuo 2	1.0	1.0056	0.0003	0.0182
10	Genérica	1.5	12.7112	1629.0326	40.3613
50	Genérica	1.5	1.8537	1.0314	1.0156
100	Genérica	1.5	1.6130	0.2495	0.4995
500	Genérica	1.5	1.5294	0.0344	0.1854
1000	Genérica	1.5	1.5130	0.0181	0.1347
10000	Genérica	1.5	1.5025	0.0015	0.0390
10	Logística	5	22.4122	4231.8644	65.0528
50	Logística	5	2.9907	98.5709	9.9283
100	Logística	5	1.1636	21.6441	4.6523
500	Logística	5	0.2454	1.3187	1.1483
1000	Logística	5	0.3284	1.4993	1.2245
10000	Logística	5	0.1168	0.1966	0.4434
10	Normal	1.962	19.3039	3320.0851	57.6202
50	Normal	1.962	4.3823	86.6199	9.3070
100	Normal	1.962	3.2237	38.1029	6.1728
500	Normal	1.962	2.2986	4.2591	2.0638
1000	Normal	1.962	2.1550	1.4629	1.2095
10000	Normal	1.962	1.9780	0.0612	0.2475

Tabela 6 – Simulações do EMV parâmetro μ

Tamanho da amostra	Distribuição	Param. μ	Média(μ)	Var(μ)	DP(μ)
10	Descontínuo 1	0	-0.0142	0.0711	0.2666
50	Descontínuo 1	0	-0.0047	0.0131	0.1142
100	Descontínuo 1	0	0.0001	0.0052	0.0722
500	Descontínuo 1	0	-0.0008	0.0009	0.0298
1000	Descontínuo 1	0	0.0013	0.0004	0.0200
10000	Descontínuo 1	0	0.0002	0.0000	0.0062
10	Descontínuo 2	0	0.0041	0.0901	0.3002
50	Descontínuo 2	0	0.0014	0.0071	0.0842
100	Descontínuo 2	0	-0.0003	0.0027	0.0519
500	Descontínuo 2	0	-0.0004	0.0004	0.0203
1000	Descontínuo 2	0	-0.0002	0.0002	0.0141
10000	Descontínuo 2	0	0.0001	0.0000	0.0041
10	Genérica	0	-0.0089	0.0743	0.2726
50	Genérica	0	0.0020	0.0052	0.0718
100	Genérica	0	0.0026	0.0026	0.0509
500	Genérica	0	-0.0002	0.0005	0.0223
1000	Genérica	0	0.0004	0.0003	0.0158
10000	Genérica	0	0.0001	0.0000	0.0049
10	Logística	0	-0.0099	0.1058	0.3252
50	Logística	0	0.0041	0.0191	0.1383
100	Logística	0	-0.0009	0.0086	0.0927
500	Logística	0	-0.0009	0.0016	0.0404
1000	Logística	0	-0.0002	0.0008	0.0285
10000	Logística	0	-0.0001	0.0001	0.0095
10	Normal	0	-0.0111	0.1275	0.3570
50	Normal	0	-0.0018	0.0266	0.1631
100	Normal	0	-0.0023	0.0133	0.1155
500	Normal	0	-0.0006	0.0023	0.0485
1000	Normal	0	-0.0008	0.0011	0.0330
10000	Normal	0	-0.0002	0.0001	0.0100

4.2.1 *Benchmark* de processamento em múltiplos núcleos

A função de simulação foi implementada de forma em que pode aproveitar a capacidade de processamento simultâneo dos processadores modernos. Por padrão o R utiliza somente um núcleo do processador. A tabela 7 mostra o tempo de execução da simulação completa feita na seção 3.4 utilizado 1, 2, 3 e 4 núcleos para processamento, sob as mesmas condições[7].

A forma como o processo de simulação foi construído faz com que os núcleos possam trabalhar independentemente sem a necessidade de esperar o processo de outro núcleo para passar para a próxima estimação. Isso faz com que cada a adição de um núcleo aumente cerca 90% a velocidade processamento da simulação relação a uso de um único núcleo.

Tabela 7 – Tempo de execução por núcleo

Núcleos	Tempo de execução (minutos)	Ganho de velocidade
1	228.82	0.00%
2	116.51	96.40%
3	80.32	184.90%
4	61.90	269.64%

[7] *Hardware* utilizado: processador Intel Core i5-3470 (3.2 GHz, quad-core), 8 Gb de memória RAM DDR3 (1333 MHz).

5 CONCLUSÃO

As distribuições logísticas generalizadas, apesar de serem difíceis de se trabalhar analiticamente, existem formas computacionais para usá-las de forma eficiente, como foi apresentado. Este trabalho tem como a objetivo de ser um passo inicial para facilitar a analisar a distribuição Rathie-Swamme e criar análises mais robustas.

Neste trabalho foi apresentado o processo de criação e publicação de um pacote para o R passando pelas etapas de preparação, planejamento e execução apontando cuidados que se deve ter para que tudo seja apresentado da melhor forma possível. Existe certa carência de uma documentação formal do processo de criação de bibliotecas no R de forma em que esta publicação possa auxiliar os interessados a criar conteúdos relevantes para o desenvolvimento acadêmico, profissional e intelectual.

As simulações apresentadas são processos computacionalmente intensivos e, puderam ter seu tempo de execução melhorado devido à implementação de processamento paralelo, utilizando a capacidade total de processadores com múltiplos núcleos. Além disso, as simulações mostram que o método de estimação por máxima verossimilhança utilizado é consistente e não viesado para o modelo Rathie-Swamee, que são propriedades de interesse quando se deseja estimar parâmetros.

As ferramentas criadas se mostraram eficientes para fazer as estimações de parâmetros, apresentando um bom resultado para modelar as modalidades e assimetria de fenômenos reais, como os gêiseres, sendo uma ótima opção para situações com dados de comportamentos semelhantes.

Por último, tudo que foi apresentado neste trabalho está disponível no repositório oficial de bibliotecas para o R, o CRAN, e pode ser facilmente utilizado por qualquer usuário do R para aplicar em seus modelos utilizando uma linha de código.

REFERÊNCIAS

RATHIE, P. N., SWAMEE, P. K. *On a new invertible generalized logistic distribution approximation to normal distribution*, Technical Research Report in Statistics, 07/2006, Dept. of Statistics, Univ. of Brasilia, Brasilia, Brasil. 2006.

AZZALINI, A. *A class of distributions which includes the normal ones*. Scandinavian Journal of Statistics. 12: 171-178, 1985.

MONTEIRO, E. Modelo Rathie-Swamee: aplicações e extensão para modelo de regressão, Universidade de São Paulo, Piracicaba, 2013.

JOHNSON, N.L.; SAMUEL, K.; BALAKRISHNAN, N. *Continuous Univariate Distributions*, 2ª ed. Nova York. Wiley, 1995, 2º vol.

BYRD, R. H., Lu, P., Nocedal, J. and Zhu, C. *A limited memory algorithm for bound constrained optimization*, Technical Report NAM-08, Dept. of Elecrtrical Engineering and Computer Science, Northwestern University, Estados Unidos da América, mar. 1994.

FRIEDRICH, L. *Creating R Packages: A Tutorial*. Disponível em: https://cran.r-project.org/doc/contrib/Leisch-CreatingPackages.pdf. Acesso em: 18 de mar. 2017

RUCKDESCHEL, P. KOHL, M. *How to generate new distributions in packages "distr", "distrEx"*. Disponível em: https://cran.r-project.org/web/packages/distr/vignettes/newDistributions.pdf. Acesso em: 18 de mar. 2017

ANDERSON, T. W. *On the Distribution of the Two-Sample Cramér-von Mises Criterion*. Annals of Mathematical Statistics. Institute of Mathematical Statistics. 33 (3): 1148-1159, 1962.

CRAMÉR, H. *On the Composition of Elementary Errors*. Scandinavian Actuarial Journal. 1928 (1): 13-74. 1928.

VON MISES, R. E. *Wahrscheinlichkeit, Statistik und Wahrheit*. Julius Springer. 1928.

Linguagem de programação R. (Versão 3.4.1). Disponível em: https://cran.r-project.org/.

RStudio. (Versão 1.0.143). Disponível em: https://www.rstudio.com/.