Instituto de Ciências Exatas

Departamento de Estatística

Jéssica Franco Cançado Richard

Divergência de Kullback-Leibler:

Uma aplicação à Modelagem

Brasília

03 de dezembro de 2013

Jéssica Franco Cançado Richard

Divergência de Kullback-Leibler: Uma aplicação à Modelagem

Trabalho apresentado como requisito parcial para a obtenção do título de Bacharela em Estatística pela Universidade de Brasília sob orientação do professor Dr. Antônio Eduardo Gomes.

BANCA EXAMINADORA

_____

Antônio Eduardo Gomes, PhD em Estatística
University of Washington, 1999
(Orientador)

_____

Eduardo Yoshio Nakano, Doutor em Estatística,
Universidade de São Paulo, 2010

_____

Lucas Moreira, Doutor em Estatística,
Universidade Estadual de Campinas, 2012

Brasília

03 de dezembro de 2013

RESUMO

Este trabalho tem como objetivo apresentar uma nova proposta de teste de adequabilidade de modelos utilizando a Divergência de Kullback-Leibler entre o núcleo-estimador (kernel) da amostra e o modelo estimado pelo método de máxima verossimilhança para análise de dados completos e censurados.

Através de simulações, analisa-se o comportamento do teste para as distribuições Normal, Gamma e Weibull. Ao final do trabalho, é apresentada uma aplicação prática a bancos de dados implementados no *software* R.

O teste em questão apresentou excelentes resultados em simulações para a detecção do modelo normal, não tendo rejeitado nenhuma vez o modelo correto simulado. No entanto, devido a problemas de fronteira na implementação de núcleo-estimadores, o teste proposto mostrou-se muito sensível a distribuições com grande assimetria e limites no domínio da função densidade de probabilidade, como a distribuição χ^2 ou exponencial, casos particulares da distribuição Gamma.

Ressalta-se as várias possibilidades de extensão deste estudo, seja no desenvolvimento de algoritmos melhorados, aplicação do teste a outras funções densidade de probabilidade, comparação com demais testes, estudo de seu poder ou utilização de técnicas que diminuam o efeito de fronteira apresentado pelo kernel.

Palavras-chave: Modelagem, Núcleo-estimadores, Análise de sobrevivência, Testes de Hipóteses, Testes de adequabilidade

ABSTRACT

This paper proposes a new goodness-of-fit test using Kullback-Leiber divergence between the sample kernel estimator and the estimated model via the maximum likelihood method for complete and censored data.

Through simulations, the test is analyzed in terms of the model's goodness-of-fit for the Normal, Gamma and Weibull distributions. Finally, an application is given using R software datasets.

The test presented excellent results in the normal simulations, with no rejection of the right simulated model. However, the proposed test has showed considerable sensitivity for given high asymmetric distributions or density functions with limited dominium as χ^2 or exponential, Gamma's particular cases, caused by the kernel's boundary problems.

Given the present work, it's possible to develop various other studies as better algorithms, application in other density functions, comparison with other tests and its power or the implementation of techniques to lessen the boundary problems.

Keywords: Modeling, Nucleo-estimators, Survival Analysis, Hypothesis Tests, Goodness-of fit Tests.

AGRADECIMENTOS

A Deus, por ter me dado tantas pessoas importantes que sempre me apoiaram.

LISTA DE GRÁFICOS E FIGURAS

LISTA DE TABELAS

Sumário

INTRODUÇÃO

Sabe-se da importância, em inferência estatística, de se definir um modelo para o estudo das características de uma amostra, ou seja, definir qual o modelo de melhor ajuste para os dados. Para tanto, há vários tipos de testes de ajuste de modelos.

Os modelos podem ser ajustados parametricamente. Isso implica em definir previamente um modelo para a população e estimar os parâmetros deste. Essa estimação pode ser feita pelo método dos momentos ou pela função de máxima verossimilhança, por exemplo.

Há várias formas de ajuste não paramétricas, dentre as quais se destacam os núcleo-estimadores (Kernel). Estes estimadores resultam em uma curva suavizada da função densidade de probabilidade que segue o comportamento da amostra.

O ajuste do modelo é analisado a partir de testes de hipóteses para verificação de adequabilidade que podem ser derivados de probabilidades empíricas da função de probabilidade, das discrepâncias entre frequências observadas e esperadas ou dos conceitos de entropia.

Em 1951, Kullback e Leibler generalizaram a definição de entropia dada por Shannon em 1948. Essa generalização, conhecida como divergência de Kullback-Leibler, permite a comparação de informação dada por duas funções. Por esse motivo, essa é uma medida amplamente utilizada em testes de adequabilidade.

Este trabalho propõe o uso da divergência de Kullback-Leibler em um teste de adequabilidade considerando essa medida entre o estimador de máxima verossimilhança e o núcleo-estimador kernel.

1 METODOLOGIA

1.1. Testes de hipóteses e entropia

Os parâmetros de um modelo podem ser estimados por vários métodos, como o de máxima verossimilhança ou o método dos momentos.

Após a estimação dos parâmetros, é importante verificar se há a devida adequabilidade para a população estudada. Para tanto, há vários testes que ajudam a prever se o modelo estimado pode ser aceito para fins de inferência.

Testes de hipóteses são testes para a validação de uma afirmação chamada comumente de hipótese nula (H_0). Em adequabilidade, estes testes são da forma

$$\begin{cases} H_0: \text{A amostra segue o modelo estipulado} \\ H_1: \text{A amostra não segue o modelo estipulado} \end{cases}$$

ou seja, H_0 assume que os parâmetros estimados do modelo são adequados. Caso haja evidências do contrário, rejeita-se essa hipótese.

Nesses testes, é definida uma estatística (conhecida na literatura como estatística do teste), ou seja, uma função que não depende de parâmetros desconhecidos, para que se possa construir um intervalo de confiança ou resulte em um p-valor que mostre se o modelo ajustado é ou não aceitável para a amostra considerada. Cada teste utiliza uma estatística para avaliar se a hipótese nula deve ou não ser rejeitada.

Segundo Evren & Tuna (2012), pode-se classificar os testes de hipóteses, em relação à adequabilidade de ajustes de modelos, em 3 tipos:

i. Derivados de probabilidades empíricas da função de probabilidade (Kolmogorov-Smirnov);

ii. Derivados das discrepâncias entre frequências observadas e esperadas (Teste da razão da verossimilhança, χ^2,…);

iii. Derivados dos conceitos de entropia (Divergência de Kullback-Leibler, Divergência de Jeffreys, ...).

Em 1948, Shannon sugeriu uma forma de medida de informação associada a uma função densidade *f*(*x*) que ficou conhecida como entropia de Shannon:

Definição 1.1.1: A entropia de Shannon é definida como:

$$E_S = -\int_{-\infty}^{\infty} f(x)\ln(f(x))\,dx \quad (1)$$

Em 1971, Papaioannu & Kempthorne definiram um conceito de entropia relacionado à teoria estatística.

> *"Entropia, em teoria estatística, pode ser definida como a quantidade de incerteza sobre uma distribuição ou sobre o parâmetro desta. O seu conceito está diretamente relacionado com a quantidade de informação sobre os parâmetros."*

Este trabalho concentra-se nos testes do tipo (iii) acima, especificamente no estudo da divergência de Kullback-Leibler (DKL), proposta como estatística de teste para funções a serem definidas posteriormente.

1.2. Divergência de Kullback-Leibler

Kullback e Leibler generalizaram essa entropia em 1951 da seguinte forma

Definição 1.2.1: Seja Θ o espaço paramétrico em que as funções $f_1(\mathrm{x})$ e $f_2(\mathrm{x})$ são definidas, a divergência de Kullback-Leibler é expressa como:

$$D_{KL}\big(f_1(x), f_2(x)\big) = E_{f_1} \log\left(\frac{f_1}{f_2}\right) = \int_{\Theta} f_1(x) log\left(\frac{f_1(x)}{f_2(x)}\right) dx. \qquad (2)$$

Definição 1.2.2: Pode-se definir a divergência de Kullback-Leibler também como a diferença entre a entropia de Shannon para $f_1(x)$ e a entropia cruzada de $f_1(x)$ e $f_2(x)$, ou seja,

$$D_{KL} = \int_{\Theta} f_1(x) log(f_1(x)) dx - \int_{\Theta} f_1(x) log(f_2(x)) dx. \qquad (3)$$

Essa divergência permite comparar duas funções, sendo muito utilizada em testes de adequabilidade.

Este trabalho propõe a utilização das estimativas via núcleo estimador e do modelo ajustado com parâmetros estimados por máxima verossimilhança para avaliar o ajuste de um modelo.

A estimativa via núcleo estimador foi escolhida por ser um estimador não paramétrico e será definida na próxima seção. Dessa forma, a estatística do teste proposto será a D_{KL} da função densidade obtida via núcleo estimador gaussiano (com o tamanho ótimo da janela já estimada pelo *software* estatístico R através da regra de Silverman, 1986) e a função densidade do modelo ajustado via estimadores de máxima verossimilhança de seus parâmetros.

1.3. Estimadores de máxima verossimilhança e núcleo-estimadores (Kernel)

Há duas formas gerais de estimadores de funções densidade: paramétricos e não paramétricos. Os estimadores paramétricos são obtidos a partir da suposição de que a amostra tem origem em uma população com função de distribuição conhecida, mas parâmetros desconhecidos. A partir de métodos como o estimador de máxima verossimilhança (EMV), que será utilizado no teste proposto neste trabalho, os parâmetros dessa função são estimados.

No caso de estimadores não paramétricos, não se define previamente uma fora funcional conhecida para a função densidade, mas apenas analisa-se a amostra. Um estimador não paramétrico amplamente utilizado é conhecido como núcleo-estimador ou kernel.

Kernel é um método não paramétrico para estimação de densidades em que cada observação é ponderada pela distância em relação a um valor central (núcleo). Nesse método, centra-se cada observação *x* onde se queira estimar a densidade. A partir de uma janela de tamanho ótimo, define-se a vizinhança de *x* e os pontos a serem utilizados na estimação.

Definição 1.3.1: O estimador Kernel é dado por:

$$\tilde{f}(x,h) = \frac{1}{nh}\sum_{i=1}^{n} K\left\{\frac{x-X_i}{h}\right\} \qquad (2)$$

em que *K* é uma função não negativa que satisfaz $\int_{-\infty}^{\infty} K(x)dx = 1$, é chamada kernel e *h* é o tamanho da janela tomado para definir-se a vizinhança de um ponto *x*. Neste estudo, *K* é escolhida como gaussiana (distribuição normal) e h é obtido através da regra de Silverman (1986):

$$0{,}9\ \min\left\{s, \frac{IQ}{1{,}34}\right\} n^{-1/5} \qquad (3)$$

em que n e *s* são o tamanho e o desvio padrão da amostra, respectivamente, e IQ é a amplitude interquartílica

A princípio, não se espera que o resultado do teste proposto seja muito afetado pelo tamanho de janela aleatório, já que compara-se o kernel da amostra com simulações e a função estimada pelo método de máxima verossimilhança.

1.4. Análise de sobrevivência

1.4.1. Características de dados de sobrevivência

O objetivo de uma análise de sobrevivência é observar indivíduos de uma população até que estes apresentem determinado fenômeno de interesse (falha) e estimar sua função de sobrevivência ($S(t) = 1 - F(t)$), que fornece a probabilidade de um "indivíduo" não falhar até o tempo *t*. Observa-se, então, o tempo de falha desses indivíduos, ou seja, o tempo até que ocorra o evento de interesse.

Frequentemente, no entanto, esses indivíduos não permanecem no estudo até que ocorra a falha desejada ou que o tempo de estudo acabe antes destes falharem. A essa perda de informação não controlada dá-se o nome de *censura*.

Mesmo havendo censura, o tempo em que o indivíduo permaneceu no estudo fornece informações que não podem ser descartadas. Dessa forma, várias técnicas para a análise de sobrevivência estão sendo desenvolvidas continuamente.

1.4.2. Tipos de censura

Quando não se observa censura nos dados, diz-se que os dados são completos. Caso contrário, tem-se dados censurados. Há vários tipos de censura:

i. Censura à esquerda: ocorre quando o indivíduo falhou antes do início do estudo
ii. Censura à direita do tipo I: após um período pré-definido de estudo, não se observou falhas em todos os indivíduos.
iii. Censura à direita do tipo II: define-se no estudo o número de falhas que devem ser observadas. O indivíduo em que a falha não foi observada recebe a classificação de censura à direita do tipo II.
iv. Censura intervalar: quando o estudo é feito analisando-se se ocorreu falha em certo intervalo de tempo, as censuras ocorridas em cada intervalo são chamadas intervalares.
v. Censura aleatória: ocorre se o indivíduo sair do estudo antes deste terminar sem ter apresentado a falha de interesse por quaisquer motivos exteriores.

Um caso particular de censura intervalar ocorre quando se está interessado em estudar, para cada indivíduo, se o tempo de falha irá ultrapassar ou não um determinado tempo *T*, ou seja, há apenas um intervalo de tempo para detecção de falha ou não. Os dados são, então, classificados como dados de *status corrente*.

1.4.3. Estimador de Kaplan-Meier

O estimador para a curva de sobrevivência mais utilizado na literatura é o Estimador de Kaplan-Meier (EKM) (1958) devido a sua propriedade de estimador de máxima verossimilhança para dados censurados. O EKM define a curva de sobrevivência estimada no tempo *t* como a razão do número de sobreviventes até este tempo e o número total de observações em risco. Ou seja,

Definição 1.4.3.1: Sejam $t_1 < t_2 < \cdots < t_k$ tempos de falha distintos e ordenados, seja d_j o número de falhas em t_j e n_j o número de indivíduos sob risco em t_j, o estimador de Kaplan-Meyer é dado por

$$EKM = \hat{S}(t) = \prod_{j:t_j<t}\left(\frac{n_j - d_j}{n_j}\right). \qquad (4)x$$

Seja $L(\theta|t_i)$ a função de verossimilhança para uma função com vetor de parâmetros θ, t_i os tempos de falha observados das x_i observações, *n* o número de observações no estudo e δ_i a variável indicadora de falha, ou seja,

$$\delta_i = \begin{cases} 1, & se\ houver\ falha\ no\ tempo\ t_i, \\ 0, & se\ não\ houver\ falha\ no\ tempo\ t_i. \end{cases}$$

A função de verossimilhança para os tempos de falha difere para cada classificação de dados, assim como o núcleo estimador da mesma. Seja r o número de censuras observadas, δ_i e γ_i indicadoras de falha nos tempos U_i, V_i respectivamente, *f(t)* e *F(t)* as funções de densidade e de probabilidade dos tempos de falha; as verossimilhanças para cada tipo de dado são dadas abaixo.

Tabela 1. Funções de Verossimilhança

Dados	Função de Verossimilhança
Sem censura	$\prod_{i=1}^{n} f(u_i;\theta)$
Censura do tipo I	$\prod_{i=1}^{n}[f(t_i;\theta)]^{\delta_i}\,[S(t_i;\theta)]^{1-\delta_i}$
Censura do tipo II	$\prod_{i=1}^{n}\frac{n!}{(n-r)!}[f(t_i;\theta)]^{\delta_i}\,[S(t_i;\theta)]^{1-\delta_i}$
Censura intervalar	$\prod_{i=1}^{n} F(U_i)^{\delta_i}[F(V_i)-F(U_i)]^{\gamma_i}[F(V_i)]^{1-\delta_i-\gamma_i}$
Status Corrente	$\prod_{i=1}^{n} F(U_i)^{\delta_i}[F(U_i)]^{1-\delta_i}$

O Kernel para os três primeiros tipos de censura acima é calculado da mesma forma. Não obstante, no caso de censura intervalar a estimação via núcleo-estimador não é trivial. Neste estudo serão considerados apenas dados de *status* corrente e o seu estimador não paramétrico será calculado a partir de um algoritmo proposto por Groeneboom e Wellner(1992) que consiste em um cálculo de regressão isotônica.

1.5. Regressão isotônica

Na ocorrência de dados de *status* corrente, para a obtenção do estimador não paramétrico de máxima verossimilhança da função $F(T_i)$, definida como a função de distribuição dos tempos de falha T_i, deve-se obter $\tilde{F}(T_i)$, $i=1, .., n$, que maximize a função de verossimilhança dada na sessão 1.4.

Para encontrar tais estimativas, utiliza-se um algoritmo proposto por Groeneboom e Wellner (1992) para o cálculo de regressão isotônica.

Definição 1.5.1: Uma função real h em D é isotônica se, para x ≤ y, tem-se

$$h(x) \leq h(y) \ \forall x, y \in D.$$

Definição 1.5.2: Seja s uma função qualquer em *D* e w uma função positiva em *D*. Uma função isotônica s* em *D* é uma regressão isotônica de *s* com pesos *w* se s* minimiza, na classe de funções isotônicas *h* em *D*, a soma

$$\sum_{x \in D} [s(x) - h(x)]^2 w(x). \ (5)$$

Teorema 1.5.1: Se

i. h é isotônica em D;

ii. a imagem de h está em $I \subset \mathbb{R}$;

iii. ϕ é uma função estritamente convexa;

então s* é a única função que maximiza

$$\sum_{x} \{\phi(h(x)) + (s(x) - h(x))\varphi(h(x))\} w(x),$$

$$onde \ \varphi(x) = \frac{d\phi(x)}{dx}. (6)$$

A log-verossimilhança para o caso de dados de status corrente pode ser escrita como

$$\sum_{i=1}^{n} \{\delta_i log F(T_i) + (1 - \delta_i) \log(1 - F(T_i))\}. (7)$$

Considere g(T_i)= $\delta_i, w(T_i) = 1$,

$$\phi(x) = xlogx + (1-x)\log(1-x),$$

$$\varphi(x) = logx - log(1-x).$$

Pelo **Teorema 1.5.1**, conclui-se que o estimador não paramétrico de máxima verossimilhança de *F* é dado pela regressão isotônica de g* e obtido por

$$\tilde{F}(T_m) = max_{i\leq m} min_{k\geq m} \frac{\sum_{i\leq j\leq k}\delta_j}{k-i+1}, \quad (8)$$

$$m = 1, \dots, n,\ i \geq 0.$$

Há também uma interpretação gráfica para essa solução que consiste em plotar os pontos (0,0) e $\left(i, \sum_{j\leq i}\delta_j\right)$, *i=1,...,n*, e calcular o máximo minorante convexo dos pontos no intervalo [*0,n*].

Definição 1.5.3: O máximo minorante convexo é definido como sendo a função H*:[0,n]→ $\mathbb{R}$ tal que

$$H^*(t) = \sup\{H(t): H(i) \leq \sum_{j\leq i}\delta_j\,, 0 \leq i \leq n, H(0) = 0, H\ convexa\}.$$

O valor de $\tilde{F}(T_m)$ é dado pela derivada à esquerda de H* em *n* para n=*1,...,n.*

Se $\delta_i = 0, 1 \leq i \leq k_1$ e $\delta_i = 1, k_2 \leq i \leq n$ para $0 < k_1 < k_2 < n$, então $\tilde{F}(T_m) = 0, 1 \leq i \leq k_1$, e $\tilde{F}(T_m) = 1, k_2 \leq i \leq n$.

Prova Ver Barlow *et al.* (1972), p. 42.

2 PROPOSTA DE TESTE

Este trabalho propõe a utilização da D_{KL} como estatística do teste para análise da adequabilidade do modelo ajustado pelos parâmetros estimados por máxima verossimilhança. O teste de hipóteses consiste em considerar

$$\begin{cases} H_0: O\ modelo\ estimado\ foi\ bem\ ajustado \\ H_1: Não\ houve\ bom\ ajuste\ do\ modelo \end{cases}$$

A partir do núcleo-estimador obtido da amostra de tamanho *n*, calcula-se a D_{KL} entre as curvas estimadas pelo Kernel, $\tilde{f}(.)$, e pelo estimador de máxima verossimilhança, $\hat{f}$. Quanto mais próxima de zero, essa medida evidencia mais fortemente a semelhança entre as funções densidade comparadas. Quando esta medida cresce, no entanto, não se sabe ao certo quando rejeitar a hipótese nula do teste, pois a distribuição exata (ou mesmo assintótica) da estatística do teste sob H_0. Por esse motivo, é proposta a aplicação de *bootstrap* paramétrico para o cálculo do p-valor através dos seguintes passos:

1) Obtém-se as funções densidade pelo núcleo estimador, $\tilde{f}(.)$, da amostra e pelo estimador de máxima verossimilhança, $\hat{f}$.
2) Para j = 1,..., *m*, gera-se a j-ésima amostra de tamanho *n* sob o modelo ajustado $f_j(.|\hat{\theta})$, $\hat{\theta}$ o EMV de θ.
3) Calcula-se a D_{KL} entre o kernel $\tilde{f}(.)$ e a função densidade gerada $f_j(.|\hat{\theta})$
4) O p-valor é dado por:

$$\mathrm{p} = \frac{\#\{j\epsilon\{1,\dots,m\}:\ D_{KL}\left(f_j(.|\hat{\theta}), \tilde{f}(.)\right) > D_{KL}\left(\hat{f},\ \widetilde{f(.)}\ \right)\}}{m}$$

Rejeita-se a hipótese nula caso, para um nível de significância de 100(1-α)%, o *p*-valor encontrado for menor que α, α a probabilidade de erro tipo I.

3 SIMULAÇÕES

Para avaliar o poder do teste, primeiro aplicou-se a teoria proposta a dados simulados utilizando-se o software R. O script se encontra no **Apêndice A**.

3.1. Dados completos

A simulação para dados completos consistiu em comparar o kernel de uma amostra gerada da distribuição com parâmetros especificados previamente e seu estimador de máxima verossimilhança. Espera-se, portanto, que não haja rejeição da hipótese nula.

Esse estudo leva em conta apenas as distribuições Normal, Gamma e Weibbul, por serem mais amplamente utilizadas.

3.1.1. Distribuição Normal

Também conhecida como distribuição gaussiana, essa distribuição é amplamente utilizada para o estudo de fenômenos naturais, em inferência estatística, além de ser a distribuição exata ou assintótica da média da amostra.

Definição 3.1.1.1: Seja X uma variável aleatória contínua, X tem função de distribuição normal se sua função densidade de probabilidade é dada por: $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}}$

A média e a variância são dadas por $E(x) = \mu$ e $Var(x) = \sigma^2$, respectivamente.

Definição 3.1.1.2: Uma variável aleatória Y tem função de distribuição normal padrão se sua função densidade de probabilidade é normal com média $\mu = 0$ e variância $Var = 1$, ou seja:

$$f(y)=\frac{1}{\sqrt{2\pi}} e^{\left\{-\frac{1}{2}y^2\right\}}, \quad (9)$$

ou seja, se segue uma distribuição normal com média $\mu = 0$ e variância $Var = 1$.

Considere uma amostra aleatória de tamanho *n* gerada a partir de uma distribuição normal padrão. Os resultados das simulações estão dispostas na **tabela 2**.

Tabela 2. Resultados para Normal(0,1)

Tamanho da amostra	Divergência	Média	Variância	P-valor	Conclusão
10	0,1149	0,0789	0,8162	0,4749	Não significativo
20	0,3384	-0,2806	0,8206	0,3231	Não significativo
50	0,1458	0,0693	1,1192	0,2455	Não significativo

Note que em todas as simulações com amostras provindas da distribuição normal padrão o teste não rejeitou o modelo, como esperado e ilustrado na **Figura 1.**

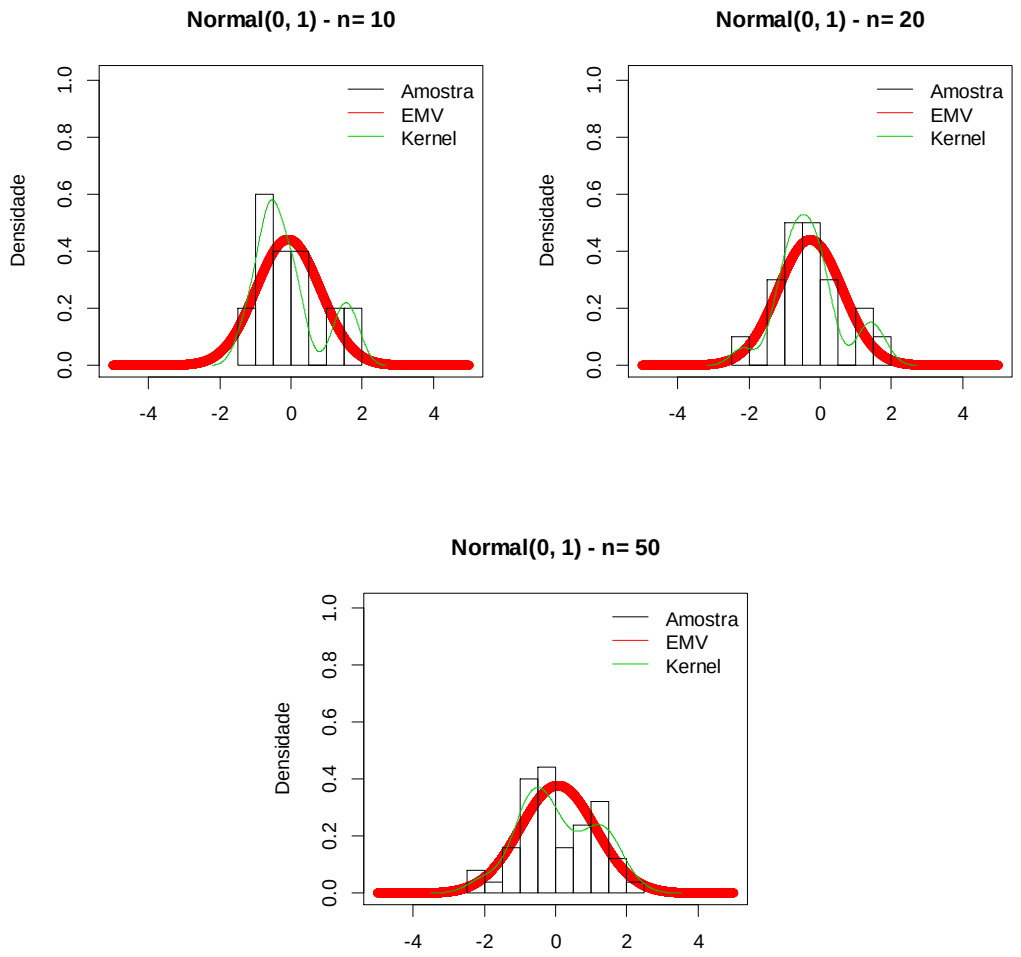

Figura 1. Simulações para distribuição Normal(0,1)

3.1.2. Distribuição Gamma

Definição 3.1.2.1: Seja X uma variável aleatória contínua não negativa, X tem função de densidade Gamma (α,β) se:

$$f(x) = \frac{\beta^{\alpha} x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)},\ \alpha,\ \beta \geq 0, x \geq 0.$$

Em que $\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx$.

Dependendo de seus parâmetros, a função Gamma define a distribuição χ^2 ou a distribuição exponencial, que não são bem ajustadas pelo núcleo estimador devido a problemas de fronteira. Portanto, para as simulações do teste proposto, utilizou-se o par de parâmetros (3,2) para evitar o problema de fronteira.

Tabela 3. Resultados para Gamma(3,2)

Tamanho da amostra	Divergência	Forma	Escala	P-valor	Conclusão
10	0,4453	3,5396	1,7477	0	Significativo
20	0,6493	2,6004	1,5612	0	Significativo
50	0,3166	3,0509	1,9320	0	Significativo

É importante salientar que, mesmo a amostra gerada sendo originalmente de uma distribuição Gamma (3,2) e, pelos gráficos da **Figura 2**, o modelo pareça ser bem ajustado para todos os tamanhos de amostra testados, porém, o kernel prolonga sua cauda além dos limites da função, forçando a hipótese nula a ser rejeitada pelo problema de fronteira. Além disso, o teste pode ter rejeitado o modelo devido ao ajuste da amostra gerada.

Quando o teste foi aplicado à distribuição χ^2, por exemplo, todas as simulações também resultaram em rejeição da hipótese nula com divergência estimada tendendo a infinito.

Conclui-se, portanto, que este teste não tem boa aplicabilidade a funções com limites no domínio de *x* pois, nesses casos, o problema de fronteira é mais acentuado. Uma solução para tal problema é a aplicação de técnicas que diminuam o peso da cauda do kernel onde a distribuição não é definida.

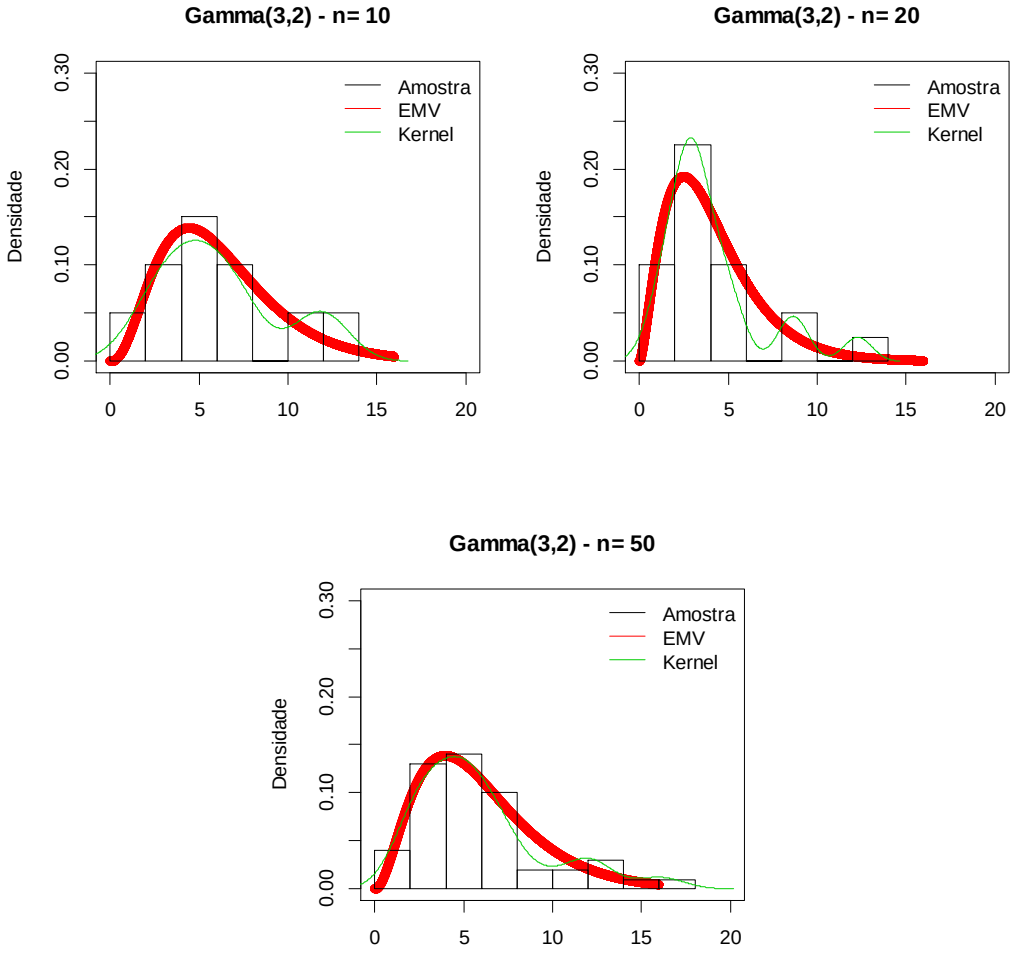

Figura 2. Simulações para Gamma(3,2)

3.1.3. Distribuição Weibull

A função de distribuição Weibull é amplamente utilizada em aplicações práticas devido ao fato de poder apresentar vários tipos de forma, mas função taxa de falha monótona (crescente, decrescente ou constante).

> **Definição 3.1.3.1:** Seja X uma variável aleatória contínua não negativa, X tem função densidade Weibull(α, β) se sua função de distribuição é dada por:
>
> $$f(x) = \frac{\alpha}{\beta^{\alpha}} x^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^{\alpha}}, \alpha, \ \beta \geq 0.$$

Note que, para $\alpha = 1$, a distribuição Weibull também se torna a distribuição exponencial com média β e, portanto, o problema de fronteira aconteceria. Fez-se simulações para uma distribuição Weibull (5,5) para três tamanhos de amostra esperando que não houvesse rejeição do modelo proposto já que os dados não se encontram perto do limite de domínio da função.

Tabela 4. Resultados da simulação para X~Weibull (5,5)

Tamanho da amostra	**Divergência**	**Forma**	**Escala**	**P-valor**	**Conclusão**
10	0,0636	4,8112	4,8068	1,0000	Não significativo
20	0,0362	4,7582	5,0414	0,9996	Não significativo
50	0,2226	4,8868	5,2006	0,9999	Não significativo

A **Figura 3.** ilustra os resultados dos modelos estimados.

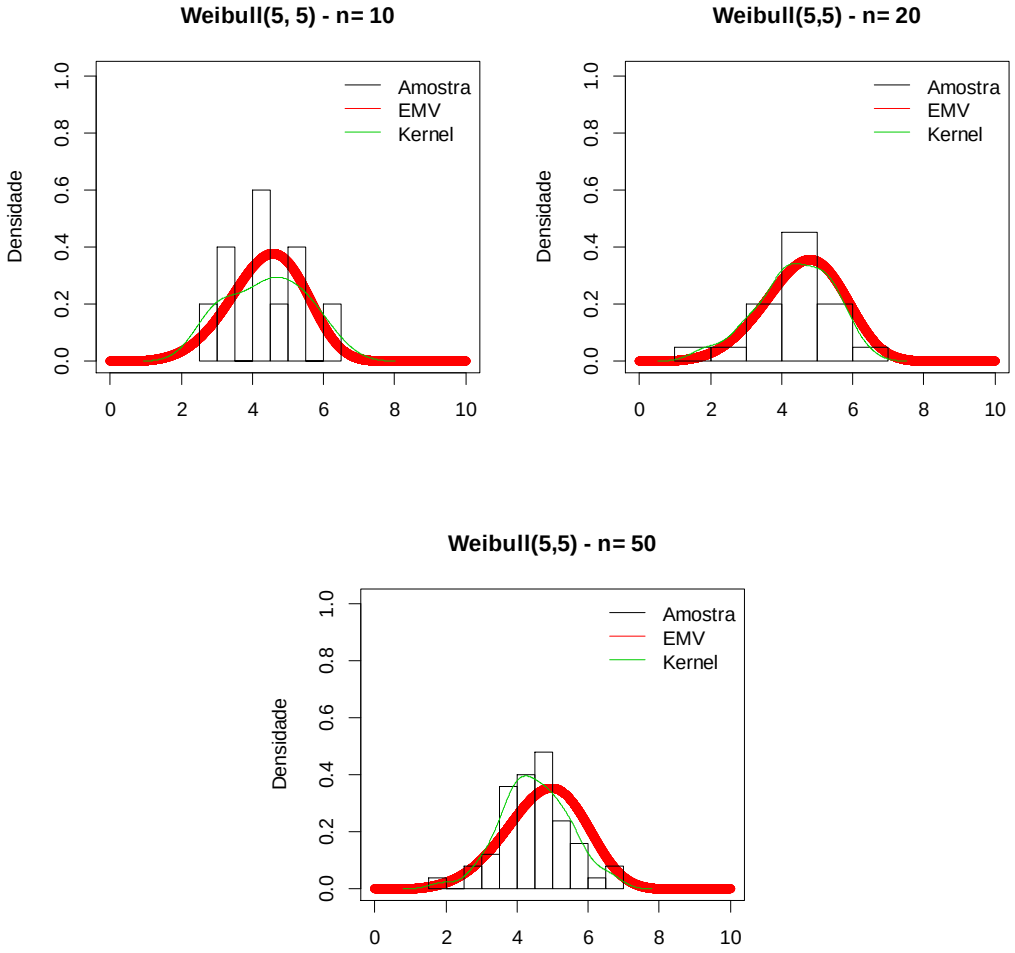

Figura 3. Simulações para a distribuição Weibull(5,5)

3.2. Dados censurados

Para a análise de dados censurados, simulou-se apenas a distribuição Weibull, já que é uma das mais utilizadas para análise de sobrevivência. Os parâmetros escolhidos para essa análise foram (5, 5), para se evitar o problema de fronteira apresentado anteriormente.

Os modelos ajustados pelo método de máxima verossimilhança são derivados das funções de máxima verossimilhança apresentadas na **Tabela 1.** A janela do kernel foi definida como 1, já que não se espera que o tamanho da janela faça grande diferença nas conclusões.

Para as simulações, gerou-se três amostras a partir de uma distribuição Weibull com tamanhos 10, 20 e 50 respectivamente. Neste trabalho, não se estudou o comportamento do teste para dados com fração de cura, caso em que a função de sobrevivência não tende a zero.

3.2.1. Simulação para dados com censura à direita

Para a simulação com dados com censura à direita, o teste apresentou os seguintes resultados:

Tabela 5. Resultados da simulação para X~Weibull (5,5) com censura à direita

Tamanho da amostra	**Divergência**	**Forma**	**Escala**	**P-valor**	**Conclusão**
10	0,1216	5,0370	4,7945	0,5935	Não significativo
20	0.1915	6,9165	4,8862	0,2104	Não significativo
50	0,1573	6,5647	5,1423	0,1951	Não significativo

Ou seja, o teste aceitou o modelo Weibull (5, 5) em todos os casos. O que é de se esperar já que as amostras foram geradas a partir desse modelo.

Neste trabalho não serão feitas simulações para os demais tipos de censura. Porém, recomenda-se para estudos futuros a melhor análise da aplicabilidade do teste em dados censurados.

4 APLICAÇÃO A BANCOS DE DADOS REAIS

Os bancos de dados utilizados para a aplicação do teste proposto se encontram disponibilizados no software R.

Para dados correntes, os bancos de dados escolhidos foram *Motor Trend Car Road Tests* (mtcars) e *Speed and Stoping Distances of Cars*(cars). O primeiro contém 32 observações e 11 variáveis disponibilizadas por Henderson e Velleman (1981). Para a aplicação do teste proposto, no entanto, apenas uma variável é analisada: potência (em cavalos-força).

No segundo banco de dados, a variável modelada é a velocidade em carros dos anos 1920s. Analisa-se, então, a adequabilidade de três modelos para os dados: normal pardrão, Normal (15,1) e Normal(15, 30)

A análise de dados censurados com o banco de dados foi aplicada ao banco *Acute Myelogenous Leukemia* (aml), que contém observações para análise de sobrevivência de 23 pacientes com leucemia aguda disponibilizados por Rupert G. Miller (1997).

4.1. Análise de dados completos

4.1.1. Modelagem do Logaritmo de Cavalos força (*horse power*)

Aplicando-se o teste ao logaritmo para testar as hipóteses:

$$\begin{cases} H_0: \text{A amostra segue o modelo Normal } (0,1) \\ H_1: \text{A amostra segue outro modelo} \end{cases}$$

Obtêm-se, pelo EMV, um modelo Normal(4,88; 0,23). O ajuste do modelo com parâmetros estimados pelo EMV, o kernel e o histograma da amostra comparado com o modelo Normal(0,1) estão representados na **Figura 4** Vê-se claramente que a média não está em torno de zero como se espera da distribuição normal padrão. O valor da DKL apresentado foi de 0,382.

Pelo resultado do teste e considerando-se um nível de significância de 95%, rejeita-se a hipótese de que a amostra segue um modelo Normal(0,1) com p-valor 0, ou

seja, em nenhuma ou quase nenhuma das simulações encontrou-se maior DKL que a gerada pelo modelo estimado por EMV e o kernel da amostra. Dessa forma, rejeita-se o modelo proposto.

Já para as hipóteses:

$$\begin{cases} H_0: \text{A amostra segue o modelo Normal } (5,1) \\ H_1: \text{A amostra segue outro modelo} \end{cases}$$

Obtêm-se uma DKL de 0,13 e, com um p-valor de 0,83, não se rejeita o modelo Normal(5,1), que se ajusta muito bem aos dados como ilustrado na **Figura 4.**

Observa-se que o teste não rejeitou, pelo modelo, o valor 1 para a variância, ainda que tal parâmetro estimado pelo método EMV seja 0,23. Isso pode se ocorrer pelo fato de que, por problemas de fronteira, a cauda da distribuição no núcleo-estimador é mais pesada do que a cauda da amostra.

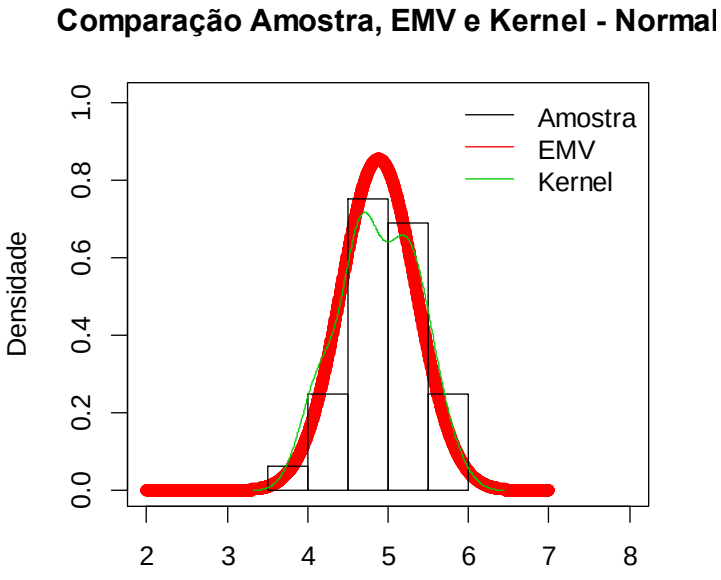

Figura 4. Estimadores para a amostra

4.1.2. Velocidade em carros dos anos 1920s

Com o presente banco de dados, quis-se avaliar se o teste rejeita a hipótese nula quando esta é falsa (poder do teste) empiricamente.

Testou-se 4 modelos para a amostra: Normal(0,1), Normal(15,1), Normal(15,30) e Normal(7,30), assumindo dados com essa distribuição clássica por conveniência, aplicando-se a transformação logarítmica. Os EMV para média e variância da amostra são 15,4 e 27,4, respectivamente. Importante notar a alta variância dos dados.

Os resultados estão apresentados na tabela abaixo.

Tabela 6. Comparação de Modelos

Modelo	Conclusão do Teste	P-valor
Normal(0,1)	Significativo	0
Normal(15,1)	Significativo	0
Normal(7,30)	Significativo	0
Normal(15,30)	Não significativo	1

Ou seja, o teste aceitaria apenas o modelo Normal(15,30) cujos parâmetros estão mais próximos dos estimados pelo EMV.

5 DISCUSSÃO

O teste proposto mostrou-se uma boa alternativa à validação de modelagens para a distribuição normal, além de apresentar aplicabilidade também à análise de sobrevivência.

É importante salientar que, pelo problema de fronteira que o kernel apresenta, distribuições com limite no domínio poderão apresentar maior DKL e diferenças entre o modelo tomado como certo e a amostra serão detectadas mais facilmente, como ocorrido nos casos simulados.

Neste trabalho não foi possível tratar mais profundamente de características deste teste ou mesmo de sua aplicabilidade à análise de sobrevivência, mas recomenda-se que, em estudos futuros, sejam feitas simulações para outros tipos de censura, estudo do poder do teste, implementação de técnicas para reduzir o problema de fronteira encontrado, comparação com outros testes, entre outros.

Em relação à parte computacional para a aplicação do teste, encontrou-se problemas na implementação do algoritmo no software R. Tais dificuldades provém das limitações da função para o cálculo da divergência de Kullback-Leibler no pacote *modeltools* (*KLdiv*) e da função *integrate*. Estas funções não geram a divergência em questão para determinados valores em que a função kernel ou a função estimada pelo EMV tenham valor muito pequeno, pois, pela definição da DKL, obtêm-se a diferença entre dois números que tendem a infinito ($\lim_{x\to 0} \log(x) \to -\infty$). Dessa forma, após detectado o problema, foi necessário a implementação do cálculo da DKL por métodos numéricos e inserida a condição de que, caso os valores não pudessem ser calculados por esse problema, o valor da diferença $\log\left(\hat{f}(x)\right) - \log\left(\tilde{f}(x)\right) = 0$.

Outro ponto interessante a ser posteriormente estudado é o de melhoria do algoritmo implementado através da definição de matrizes em contraponto ao *loop* utilizado neste trabalho ou mesmo a menor quantidade de simulações necessárias no *bootstrap* para se chegar a bons resultados de modelagem.

6 REFERÊNCIAS

Barlow, R. E., Bartholomew, D. J., Bremmer, J. M. e Brunk, H. D. Statistical Inference Under Order Restrictions". John Wiley & Sons, New York, 1972.

Casella, G. Berger, L. R. Inferência Estatística. São Paulo: Cengage Learning, 2010.

Colosimo, E. A. Giolo, S, R. Análise de sobrevivência Aplicada. São Paulo: ABE – Projeto Fisher, 2006.

Evren, A, Tuna, E. (2012) *On some properties based on goodness of fit measures based on statistical entropy.* IJRRAS *Vol. 13 (1), 192-205.*

Fisher, R.A. (1922) *On the mathematical foundations of theoretical statistics.* Phil. Trans., A, 222, 309—368. CP 18.

Groeneboom, P. Wellner, J.A. (1992) *Information Bounds and Non-Parametric Maximum Likelihood Estimation.* Birkhauser Verlag, Berlim.

Henderson and Velleman (1981), *Building multiple regression models interactively.* Biometrics, 37, 391–411.

Kaplan, E.L., Meier, P. (1958) *Nonparametric estimation from incomplete observations.* Journal of the American Statistical Association, 53, 457-481.

Kullback, S.; Leibler, R. A. (1951) *On information and sufficiency.* The Annals of Mathematical Statistics*, 22*, 79-86.

Lopes, E.R. (2005) *Intervalos de confiança para a função de distribuição na presença de censura intervalar, Caso 1*. Universidade Federal de Minas Gerais, Belo Horizonte.

Papaioannou, P. C. & Kempthorne, O. (1971) *On Statistical information theory and related measures of information.* Air Force Systems Command, United States Air Force. Wright-Patterson Air Base, Ohio.

Rupert G. Miller (1997), Survival Analysis. John Wiley & Sons. ISBN: 0-471-25218-2.

Shannon, C. E. (1948) *A Mathematical Theory of Communication.* Bell System Technical Journal*, 27*, 379-423; 623-656.

Silverman, B. W. (1986) *Density Estimation*. London: Chapman and Hall.

Wang C., Sun J., Sun L., Zhou J., Wang D. (2012) *Nonparametric estimation of current status data with dependent censoring.* Lifetime Data Anal. 18(4):434-45. doi: 10.1007/s10985-012-9223-7. Epub 2012 Jun 27.

APÊNDICE A

PROGRAMAÇÃO PARA O SOFTWARE R

```
#SCRIPT PARA DADOS COMPLETOS

#DISTRIBUIÇÃO WEIBULL
wei.test<-function(amostra, alpha){
                lik <- function(par) {
 -
length(amostra)*log(par[2])+length(amostra)*par[2]*log(par[1])-
        (par[2]-
1)*sum(log(amostra))+(1/(par[1]^par[2]))*sum(amostra^par[2])
}

## função pelo estimador de máxima verossimilhança
emv <- nlminb(c(2,2),lik)
amp <- 0.001
pontos <- seq(0.001,10.001,amp)
f.hat <- dweibull(pontos,emv$par[1],emv$par[2])

#Estimador não paramétrico
f.til<-density(amostra, n=length(pontos))$y

#Divergência de Kullback-Leibler
```

```
dkl <- -amp*sum(f.hat*(log(f.hat)-log(f.til)))


#Gráfico para comparação:

plot(pontos,f.hat, col=2, main=paste("Weibull(5, 5) -
n=",length(amostra)),

ylim=c(0,1.01), ylab="Densidade", xlab="")

hist(amostra, freq=F,add=T)

lines(density(amostra), col=3)

legend('topright', col=c(1,2,3), c('Amostra','EMV','Kernel'), lty=1,
bty="n")


#Bootstrap

i<-j<-0

m<-10000

while(i<m){

        sim<-rweibull(length(amostra), 5, 5)

        lik2 <- function(par) {

          -length(sim)*log(par[2])+length(sim)*par[2]*log(par[1])-

                (par[2]-
1)*sum(log(sim))+(1/(par[1]^par[2]))*sum(sim^par[1])

        }


        emv2 <- nlminb(c(4,4),lik2)

        f.hat2 <- dweibull(pontos,emv2$par[1],emv2$par[2])
```

```
        dklsim <- -amp*sum(f.hat2*(log(f.hat2)-log(f.til)))
        if(!is.na(dklsim)){
        if(dklsim>dkl){j<-j+1}else{j<-j+0}}
        i<-i+1}
pvalor<-j/m
if(pvalor>alpha){conclusao<-"Não significativo"}else{conclusao<-
"Significativo!"}

resultado<-data.frame(cbind(dkl, pvalor, conclusao, emv$par[1],
emv$par[2]))
names(resultado)<-c("DKL","p-valor","Conclusão", "Forma",
"Escala")

return(resultado)
}

#Dados simulados:
set.seed(1)
amostra<-rweibull(20, 5, 5)
teste1<-wei.test(amostra, .05)

set.seed(1)
amostra<-rweibull(50, 5, 5)
teste2<-wei.test(amostra, .05)
```

```
set.seed(1)
amostra<-rweibull(10, 5, 5)
teste3<-wei.test(amostra, .05)

#DISTRIBUIÇÃO NORMAL
norm.test<-function(amostra, alpha){
            lik <- function(par) {

      (length(amostra)/2)*log(2*pi)+(length(amostra)/2)*log(par
[2])+
            (1/2)*sum((amostra-par[1])^2)/par[2]
            }

## função pelo estimador de máxima verossimilhança
emv <- nlminb(c(1,1),lik)
amp <- 0.001
pontos <- seq(-5,5,amp)
f.hat <- dnorm(pontos,emv$par[1],sqrt(emv$par[2]))

#Estimador não paramétrico
f.til<-density(amostra, n=length(pontos))$y


#Divergência da amostra com seu kernel
```

```
dkl <- -amp*sum(f.hat*(log(f.hat)-log(f.til)))


#Gráfico para comparação:

plot(pontos,f.hat, col=2, main=paste("Normal(0, 1) -
n=",length(amostra)),

ylim=c(0,1.01), xlim=c(-5,5), ylab="Densidade", xlab="")

hist(amostra, freq=F,add=T)

lines(density(amostra), col=3)

legend('topright', col=c(1,2,3), c('Amostra','EMV','Kernel'), lty=1,
bty="n")


#Bootstrap

i<-j<-0

m<-10000

while(i<m){

        sim<-rnorm(length(amostra))

                lik2 <- function(par) {

                (length(sim)/2)*log(2*pi)+(length(sim)/2)*log(par[2])+

                (1/2)*sum((sim-par[1])^2)/par[2]

}

        emv2 <- nlminb(c(1,1),lik2)

        f.hat2 <- dnorm(pontos,emv2$par[1],sqrt(emv2$par[2]))


        dklsim <- -amp*sum(f.hat2*(log(f.hat2)-log(f.til)))
```

```
        if(!is.na(dklsim)){
        if(dklsim>dkl){j<-j+1}}
        i<-i+1}
pvalor<-j/m
if(pvalor>alpha){conclusao<-"Não significativo"}else{conclusao<-
"Significativo!"}

resultado<-data.frame(cbind(dkl, pvalor, conclusao, emv$par[1],
emv$par[2]))
names(resultado)<-c("DKL","p-valor","Conclusão", "Média",
"Variância")

return(resultado)
}

#Dados simulados:
set.seed(1)
amostra<-rnorm(10)
teste1n<-norm.test(amostra, .05)

set.seed(1)
amostra<-rnorm(20)
teste2n<-norm.test(amostra, .05)

set.seed(1)
```

```
amostra<-rnorm(50)
teste3n<-norm.test(amostra, .05)



rbind(teste1n, teste2n, teste3n)



#DISTRIBUIÇÃO GAMMA
gamma.test<-function(amostra, alpha){
            lik <- function(par) {

      length(amostra)*log(gamma(par[1]))+length(amostra)*par[
1]*log(par[2])-
            (par[1]-
1)*sum(log(amostra))+(1/par[2])*sum(amostra)
            }


## função pelo estimador de máxima verossimilhança
emv <- nlminb(c(1,1),lik)
amp <- 0.001
pontos <- seq(0,16,amp)
f.hat <- dgamma(pontos,emv$par[1],scale=emv$par[2])

#Estimador não paramétrico
f.til<-density(amostra, n=length(pontos))$y
```

```
#Divergência da amostra com seu kernel

soma<-(f.hat*(log(f.hat)-log(f.til)))

for (i in 1:length(soma))if(is.na(soma[i])){soma[i]<-0}

dkl <- -amp*sum(f.hat*(log(f.hat)-log(f.til)))


#Gráfico para comparação:

plot(pontos,f.hat, col=2, main=paste("Gamma(3,2) - n=",
length(amostra)),

ylab="Densidade", xlab="", xlim=c(0,20), ylim=c(0,.3))

hist(amostra, freq=F,add=T)

lines(density(amostra), col=3)

legend('topright', col=c(1,2,3), c('Amostra','EMV','Kernel'), lty=1,
bty="n")


#Bootstrap

i<-j<-0

m<-10000

while(i<m){

        sim<-rgamma(length(amostra), 3, 2)

                lik2 <- function(par) {


        length(amostra)*log(gamma(par[1]))+length(amostra)*par[
1]*log(par[2])-
```

```
                (par[1]-
1)*sum(log(amostra))+(1/par[2])*sum(amostra)
                }


        emv2 <- nlminb(c(1,1),lik2)
        f.hat2 <- dgamma(pontos,emv2$par[1],scale=emv2$par[2])


        dklsim <- -amp*sum(f.hat2*(log(f.hat2)-log(f.til)))
        if(!is.na(dklsim)){
        if(dklsim>dkl){j<-j+1}}
        i<-i+1}
pvalor<-j/m
if(pvalor>alpha){conclusao<-"Não significativo"}else{conclusao<-
"Significativo!"}


resultado<-data.frame(cbind(dkl, pvalor, conclusao, emv$par[1],
emv$par[2]))
names(resultado)<-c("DKL","p-valor","Conclusão", "Forma",
"Escala")


return(resultado)
}


#Dados simulados:
set.seed(4)
```

```
amostra<-rgamma(10, 3, scale=2)
teste1g<-gamma.test(amostra, .05)

set.seed(4)
amostra<-rgamma(20, 2, scale=2)
teste2g<-gamma.test(amostra,  .05)

set.seed(4)
amostra<-rgamma(50, 3, scale=2)
teste3g<-gamma.test(amostra,  .05)



#DISTRIBUIÇÃO GAMMA
chi2.test<-function(amostra, alpha){
            lik <- function(par) {

      length(amostra)*log(gamma(par[1]))+length(amostra)*par[
1]*log(par[2])-
            (par[1]-
1)*sum(log(amostra))+(1/par[2])*sum(amostra)
            }

## função pelo estimador de máxima verossimilhança
emv <- nlminb(c(1,1),lik)
amp <- 0.001
```

```
pontos <- seq(0,16,amp)

f.hat <- dgamma(pontos,emv$par[1],scale=emv$par[2])


#Estimador não paramétrico

f.til<-density(amostra, n=length(pontos))$y


#Divergência da amostra com seu kernel

soma<-(f.hat*(log(f.hat)-log(f.til)))

for (i in 1:length(soma))if(is.na(soma[i])){soma[i]<-0}

dkl <- -amp*sum(f.hat*(log(f.hat)-log(f.til)))


#Gráfico para comparação:

plot(pontos,f.hat, col=2, main=paste("Gamma(3,2) - n=",
length(amostra)),

ylab="Densidade", xlab="", xlim=c(0,20), ylim=c(0,.3))

hist(amostra, freq=F,add=T)

lines(density(amostra), col=3)

legend('topright', col=c(1,2,3), c('Amostra','EMV','Kernel'), lty=1,
bty="n")


#Bootstrap

i<-j<-0

m<-10000

while(i<m){
```

```
        sim<-rgamma(length(amostra), .5, .5)
                lik2 <- function(par) {

        length(amostra)*log(gamma(par[1]))+length(amostra)*par[
1]*log(par[2])-
                (par[1]-
1)*sum(log(amostra))+(1/par[2])*sum(amostra)
                }

        emv2 <- nlminb(c(1,1),lik2)
        f.hat2 <- dgamma(pontos,emv2$par[1],scale=emv2$par[2])

        dklsim <- -amp*sum(f.hat2*(log(f.hat2)-log(f.til)))
        if(!is.na(dklsim)){
        if(dklsim>dkl){j<-j+1}}
        i<-i+1}
pvalor<-j/m
if(pvalor>alpha){conclusao<-"Não significativo"}else{conclusao<-
"Significativo!"}

resultado<-data.frame(cbind(dkl, pvalor, conclusao, emv$par[1],
emv$par[2]))
names(resultado)<-c("DKL","p-valor","Conclusão", "Forma",
"Escala")

return(resultado)
```

```
}

#Dados simulados:
set.seed(4)
amostra<-rgamma(10, 3, scale=2)
teste1g<-chi2.test(amostra, .05)

set.seed(4)
amostra<-rgamma(20, 2, scale=2)
teste2g<-chi2.test(amostra,  .05)

set.seed(4)
amostra<-rgamma(50, 3, scale=2)
teste3g<-chi2.test(amostra,  .05)


#APLICAÇÕES
amostra<-log(mtcars$hp)
aplicacao<-norm.test(amostra, .05)

#DISTRIBUIÇÃO NORMAL
norm.test5<-function(amostra, alpha){
                lik <- function(par) {
```

```
        (length(amostra)/2)*log(2*pi)+(length(amostra)/2)*log(par
[2])+
                (1/2)*sum((amostra-par[1])^2)/par[2]
                }

## função pelo estimador de máxima verossimilhança
emv <- nlminb(c(1,1),lik)
amp <- 0.001
pontos <- seq(2,7,amp)
f.hat <- dnorm(pontos,emv$par[1],sqrt(emv$par[2]))

#Estimador não paramétrico
f.til<-density(amostra, n=length(pontos))$y


#Divergência da amostra com seu kernel
dkl <- -amp*sum(f.hat*(log(f.hat)-log(f.til)))

#Gráfico para comparação:
plot(pontos,f.hat, col=2, main="Comparação Amostra, EMV e
Kernel - Normal",
ylim=c(0,1.01), xlim=c(2,8), ylab="Densidade", xlab="")
hist(amostra, freq=F,add=T)
lines(density(amostra), col=3)
```

```
legend('topright', col=c(1,2,3), c('Amostra','EMV','Kernel'), lty=1,
bty="n")


#Bootstrap

i<-j<-0

m<-10000

while(i<m){

        sim<-rnorm(length(amostra),5,1)

                lik2 <- function(par) {

                (length(sim)/2)*log(2*pi)+(length(sim)/2)*log(par[2])+

                (1/2)*sum((sim-par[1])^2)/par[2]

}

        emv2 <- nlminb(c(1,1),lik2)

        f.hat2 <- dnorm(pontos,emv2$par[1],sqrt(emv2$par[2]))


        dklsim <- -amp*sum(f.hat2*(log(f.hat2)-log(f.til)))

        if(!is.na(dklsim)){

        if(dklsim>dkl){j<-j+1}}

        i<-i+1}

pvalor<-j/m

if(pvalor>alpha){conclusao<-"Não significativo"}else{conclusao<-
"Significativo!"}


resultado<-data.frame(cbind(dkl, pvalor, conclusao, emv$par[1],
emv$par[2]))
```

```
names(resultado)<-c("DKL","p-valor","Conclusão", "Média",
"Variância")


return(resultado)

}


aplicacao2<-norm.test5(amostra, 0.5)


#APLICAÇÃO AO BANCO DE DADOS CARS

data(cars)


norm.test.geral<-function(amostra, alpha, media, variancia){

                lik <- function(par) {

       (length(amostra)/2)*log(2*pi)+(length(amostra)/2)*log(par
[2])+

                (1/2)*sum((amostra-par[1])^2)/par[2]

                }


## função pelo estimador de máxima verossimilhança

emv <- nlminb(c(1,1),lik)

amp <- 0.001

pontos <- seq(0,27,amp)

f.hat <- dnorm(pontos,emv$par[1],sqrt(emv$par[2]))
```

```
#Estimador não paramétrico
f.til<-density(amostra, n=length(pontos))$y


#Divergência da amostra com seu kernel
dkl <- -amp*sum(f.hat*(log(f.hat)-log(f.til)))

#Gráfico para comparação:
plot(pontos,f.hat, col=2, main="Velocidade",
ylim=c(0,.5), xlim=c(0,27), ylab="Densidade", xlab="")
hist(amostra, freq=F,add=T)
lines(density(amostra), col=3)
legend('topright', col=c(1,2,3), c('Amostra','EMV','Kernel'), lty=1,
bty="n")

#Bootstrap
i<-j<-0
m<-10000
while(i<m){
        sim<-rnorm(length(amostra), media, variancia)
                lik2 <- function(par) {
                (length(sim)/2)*log(2*pi)+(length(sim)/2)*log(par[2])+
                (1/2)*sum((sim-par[1])^2)/par[2]
}
```

```
        emv2 <- nlminb(c(1,1),lik2)
        f.hat2 <- dnorm(pontos,emv2$par[1],sqrt(emv2$par[2]))

        dklsim <- -amp*sum(f.hat2*(log(f.hat2)-log(f.til)))
        if(!is.na(dklsim)){
        if(dklsim>dkl){j<-j+1}}
        i<-i+1}
pvalor<-j/m
if(pvalor>alpha){conclusao<-"Não significativo"}else{conclusao<-
"Significativo!"}

resultado<-data.frame(cbind(dkl, pvalor, conclusao, emv$par[1],
emv$par[2]))
names(resultado)<-c("DKL","p-valor","Conclusão", "Média",
"Variância")

return(resultado)
}

aplicacao3<-norm.test.geral(cars$speed, .05, 0, 1)
aplicacao4<-norm.test.geral(cars$speed, .05, 7, 1)
aplicacao5<-norm.test.geral(cars$speed, .05, 15, 1)
aplicacao6<-norm.test.geral(cars$speed, .05, 15, 30)
```

```
# DADOS COM CENSURA À DIREITA
require(survival)

as.wei<-function(tempo,status,alpha){
n<-length(tempo)
km <- survfit(Surv(tempo,status)~rep(1,n))
#names(km)
#km$surv
#km$time

amp <- 0.001
pontos <- seq(0.001,10.001,amp)

# calculo do e.m.v. da Weibull
# par[1] = alfa (forma), par[2] = beta (escala)
# f(x) = (a/b) (x/b)^(a-1) exp(- (x/b)^a)
# "lik" é a função de verossimilhança multiplicada por -1

lik <- function(par) {
  -sum(-km$n.censor*(km$time/par[2])^par[1]
      +km$n.event*(log(par[1]/(par[2]^par[1]))
                +(par[1]-1)*log(km$time)
                -(km$time/par[2])^par[1]))
}
```

```
emv <- nlminb(c(1,1),lik)


## "f.hat" é o vetor com o valor da estimativa de m.v. da densidade

f.hat <- dweibull(pontos,emv$par[1],emv$par[2])


## "f.til" é o vetor com a estimativa n.p. suavizada via kernel da densidade

h <- 1


mat.aux <- matrix(rep(pontos,n),ncol=n)-matrix(rep(km$time,rep(length(pontos),n)),ncol=n)

dS <- -diff(c(1,km$surv))

f.til <- (dnorm(mat.aux/h,0,1))%*%dS/h


dkl <- amp*sum(f.hat*(log(f.hat)-log(f.til)))


#Bootstrap

i<-j<-0

m<-10000

while(i < m){

        tt <- rweibull(n,5,5)

        cc <- rweibull(n,5,5)

        tempo2 <- tt[tt<=cc]
```

```
        status2 <- rep(1,length(tempo2))
        status2 <- c(status2,rep(0,(n-length(tempo2))))
        tempo2 <- c(tempo2,cc[cc<tt])
        km2 <- survfit(Surv(tempo2,status2)~rep(1,n))

        lik2 <- function(par) {
          -sum(-km2$n.censor*(km2$time/par[2])^par[1]
      +km2$n.event*(log(par[1]/(par[2]^par[1]))
                +(par[1]-1)*log(km2$time)
                -(km2$time/par[2])^par[1]))
}

        emv2 <- nlminb(c(1,1),lik2)
        f.hat2 <- dweibull(pontos,emv2$par[1],emv2$par[2])
        sim <- amp*sum(f.hat2*(log(f.hat2)-log(f.til)))
        if(!is.na(sim)){if(sim > dkl){j<-j+1}}
        i<-i+1}
pvalor<-j/m

if(pvalor>alpha){conclusao<-"Não significativo"}else{conclusao<-
"Significativo!"}

resultado<-data.frame(cbind(dkl, pvalor, conclusao, emv$par[1],
emv$par[2]))
```

```
names(resultado)<-c("DKL","p-valor","Conclusão", "Forma",
"Escala")


return(resultado)

}


#Simulações

n <- 10

set.seed(2)

tt <- rweibull(n,5,5)

set.seed(2)

cc <- rweibull(n,5,5)

temp <- tt[tt<=cc]

cens <- rep(1,length(temp))

cens <- c(cens,rep(0,(n-length(temp))))

temp <- c(temp,cc[cc<tt])


(sim1<-as.wei(temp,cens, .05))


n <- 20


set.seed(3)

tt <- rweibull(n,5,5)

set.seed(3)
```

```
cc <- rweibull(n,5,5)
temp <- tt[tt<=cc]
cens <- rep(1,length(temp))
cens <- c(cens,rep(0,(n-length(temp))))
temp <- c(temp,cc[cc<tt])

(sim2<-as.wei(temp,cens, .05))

n <- 50
set.seed(3)
tt <- rweibull(n,5,5)
set.seed(3)
cc <- rweibull(n,5,5)
temp <- tt[tt<=cc]
cens <- rep(1,length(temp))
cens <- c(cens,rep(0,(n-length(temp))))
temp <- c(temp,cc[cc<tt])

(sim3<-as.wei(temp,cens, .05))
```